AN APPLICATION OF THE FINITE ELEMENT METHOD TO THREE DIMENSIONAL HEAT CONDUCTION PROBLEMS

by

Şükrü Erden ERUÇ

B.S. in Boğaziçi University, 1983

Submitted to the Institute for Graduate Studies in Science and Engineering in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

Boğaziçi University

1986

APPROVED BY

Yard. Doç. Dr. Vahan KALENDEROĞLU [illegible]
(Thesis Supervisor)

Doç. Dr. İlsen ÖNSAN [illegible]

Prof. Dr. Akın TEZEL [illegible]

DATE OF APPROVAL March 11th, 1986

ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis instructor, Yrd. Doç. Dr. Vahan Kalenderoğlu for his advice and helpful criticisms. His patience throughout this long lasting work is worth mentioning.

I appreciate the ready availability of Dr. Tamer Şıkoğlu at times when I was no longer able to cope with the continual breakdowns of the computer system.

My sincere appreciations go to my colleagues Ersin Karabudak and Engin Atik for their generosity and willingness with the computer work. I am grateful to my friends Ali İhsan Okyay and Aydan Eksin for their cooperation during the final touches and typing process; and finally, I must thank my friends İsmail Karahan and Bürçin Mat for their devotion in listening to my complaints and their continuous encouragements.

Erden Eruç

ABSTRACT

A finite element solution scheme for three-dimensional transient conduction heat transfer problems is devised and its validity is substantiated through application to representative problems.

The governing matrix equation is first derived by applying two different methods, namely the variational approach and the Galerkin approach, and its interpretations are stated for the cases of steady state and transient conduction heat transfer. Using this model, the solution is reached when different types of finite elements are used where derivation of the element matrices and element load vectors are introduced. For the transient case, a specific cube problem is discussed for which the exact series solution and the finite element solutions are compared.

On the basis of the comparisons and results obtained above, it is concluded that, given certain conditions, the finite element method is safely applicable to three-dimenisonal heat conduction problems.

KISA ÖZET

Üç boyutlu zamana bağımlı ısı transferi problemleri için bir sonlu elemanlarla çözüm yöntemi oluşturulmuş ve bu yöntemin geçerliliği örnek problemlere uygulamalarla gösterilmiştir.

Varyasyonel ve Galerkin yaklaşımları ile ana matris denklemi çıkartılmış, durağan ve zamana bağımlı haller için yorumlanmıştır. Bu modelin uygulanması sayesinde değişik elemanlar kullanılarak sonuca ulaşılmıştır. Her bir eleman için eleman matrisleri ve vektörlerinin elde ediliş yolu gösterilmiştir. Zamana bağımlı halde sonlu eleman metodunun verdiği sonuçlarla Fourier seri çözümünün karşılaştırıldığı bir küp problemi tartışılmıştır.

Karşılaştırmalara ve elde edilen sonuçlara dayanarak, belli koşullar sağlandığı taktirde, sonlu elemanlar metodunun üç boyutlu ısı transferi problemlerine uygulanabileceği sonucuna ulaşılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF NOMENCLATURE

A	Area
A_{ijk}	Area of face ijk
C	specific heat
$[D]$	thermal conductivity matrix
h	convection heat transfer coefficient
I	the functional to be minimized
I^e	contribution of element e to the functional I
$[J]$	Jacobian matrix
k_x, k_y, k_z	thermal conductivities along x,y,z axes
$[K]$	conductivity matrix of complete body after incorporation of boundary conditions
$[K]^e = [K_{ij}]$	conductivity matrix of element e.
L_1, L_2, L_3, L_4	natural coordinates of a tetrahedron element
l_x, l_y, l_z	direction cosines
N_i	interpolation function associated with the i'th nodal degree of freedom
$[N]$	matrix of nodal interpolation functions
$\vec{P}^e$	load vector of element e in the local coordinate system
$\vec{P}$	global load vector
p	number of nodes per element
q	rate of heat flow
$\dot{q}$	rate of heat generation per unit volume

r, s	natural coordinates of a quadrilateral element
r, s, t	natural coordinates of a hexahedron element
S_1, S_2, S_3	part of a surface of a body
S^e	surface of element e
S_1^e, S_2^e, S_3^e	part of surface of element e
t	time
T	temperature
T_i	temperature at node i
T_o	initial temperature
T_∞	surrounding temperature
T_s	specified temperature
$\vec{T}^e$	vector of nodal temperature of element e
V^e	volume of element e
w	wieghts of collocation points
x	x-coordinate
(x_i, y_i, z_i)	(x,y,z) coordinates of node i
α_i	coefficients
$\lambda_1, \lambda_2, \lambda_3$	length of sides of tetrahedron face
ρ	density of a solid
$\emptyset$	field variable
$\emptyset_i$	value of field variable at node i
$\vec{\emptyset}^e$	vector of nodal values of the field variable of element e
$\vec{\emptyset}$	vector of nodal values of the field variables of complete body
$[\,]^T$	transpose of
$\dot{T}$	(dot over) derivative with respect to time
Δt	increment of time

INTRODUCTION

Modern technology has forced mankind to use tools and machine parts which are complex in nature, thus, having highly irregular shapes. This complexity results in difficulties when trying to solve mechanical or thermal problems over the body if analytical methods are applied.

In the past much effort has been directed towards the solution of such problems which are generally of three dimensional nature. The solutions have been based on either experimental or numerical approach when the solution is impossible analytically. Numerical approaches have almost exclusively used a direct finite difference method solution of the governing differential equations. An alternative approach based on a variational formulation and known as the finite element method is worked out, and the discretization achieved by this process has led to many advantages in a computer solution.

The FEM was originally developed for stress analysis, and its superiority for such problems is now widely recognized. As a consequence, the FEM has gained popularity in the solution of steady state and transient heat conduction problems.

A critical comparison of the two methods should be based on the criteria of flexibility and ease of application, accuracy and efficiency. The FEM is clearly superior to the FDM on the basis of flexibility and ease of application [8] Composite bodies containing several different anisotropic materials are easily handled. The isoparametric element [7] can be used to fit irregularly shaped boundaries. Grading of the net in regions having high temperature gradients is easily accomplished. Any type of thermal boundary condition may be readily applied.

The two remaining criteria, accuracy and efficiency, are interrelated. Excellent accuracy can be obtained with either method if a sufficient number of nodal points is used. However, increasing the number of nodal points increases the computer time required for the solution, and the memory requirements are highly augmented.

General finite element formulation leads to the replacement of distributed properties of a subregion or an element by a linear matrix relationship connecting generalized potentials associated with a number of discrete nodal points. The shapes of elements and their degree of interconnection, number of nodes, are a feature by which the accuracy of representation can be improved.

The transient field problem of the type encountered in heat conduction problems is formulated in terms of the finite element process using the variational or Galerkin approach. The extension of the three dimensional analysis was made by Zienkiewicz, Arlett and Bahrani first in 1968.

The transient problem was first suggested by Visser in 1965. The solution of heat conduction problems and step-by-step general processes are outlined in many references. In addition to the simplest tetrahedral elements the work has been extended to more precise forms such as parabolic and cubic elements with curved sides. An additional work has been reproduced using hexahedral elements similarly including the higher order forms.

In the general finite element method, the actual continuum is represented as an assembly of subdivisions called finite elements. These elements are considered to be interconnected at specified joints called nodes. The nodes usually lie on the element boundaries where adjacent elements are considered to be connected. Since the actual variation of the field variable inside the continuum is not known, it is assumed that the variation of the field variable inside a finite element can be approximated by single function. These approximating functions, also called interpolation models, are defined in terms of the values of the field variables at the nodes. Once the field equations for the whole continuum are written, the new unknowns become the nodal values of the field variable. By solving the field equations, which are generally in the form of the matrix equations, the nodal values of the field variable will be known. When these values are known the interpolation functions approximate the field variable throughout the body.

The solution of a general continuum problem by the finite element method always follows an orderly step-by-step process.

i. Discretization of the structure: The first step in the FEM is to divide the structure into subdivisions. Hence, the body that is being analyzed has to be modeled with suitable finite elements. The number, type, size and arrangement of the elements have to be decided.

ii. Selection of a proper interpolation model: Since the field variable under any specified load conditions cannot be predicted exactly, some suitable solution within an element to approximate the unknown distribution is assumed. The assumed solution must be simple from computational point of view, but it should satisfy certain convergence requirements.

iii. Derivation of the element matrices and load vectors: From the assumed interpolation model, the element conductivity matrix $[K_1]^e$, convection matrix $[K_2]^e$ and the load vector $\vec{p}^e$ of element (e) are to be derived by using either equilibrium conditions or a suitable variational principle.

iv. Assembly of element matrices to obtain global equilibrium equations: Since the body is composed of several finite elements, the individual element matrices and load vectors are to be assembled in a suitable manner, and the overall equilibrium equations can be formulated as

$$[K] \cdot \vec{T} = \vec{P} \qquad (1)$$

where $[K]$ is the effective conductivity matrix, $\vec{T}$ is the temperature solution vector and $\vec{P}$ is the effective load vector.

v. Modification: The overall epuilibrium equations (1) must be modified to account for the boundary conditions of the problem.

vi. Computation of field values at the nodes: The solution of equation (1) follows in order to obtain the nodal field values. At this stage of the FE formulation, solution algorithms such as the Gaussian elimination method are utilized.

The work presented here also used a similar process in all the programs that were developed.

Some advantages of the FEM can be summarized as follows:

- its ability to use elements of various types, sizes and shapes, and to model a body of arbitrary geometry,
- its ability to accomodate arbitrary boundary conditions and arbitrary thermal loading,
- its ability to model composite structures involving different body components such as combinations of plates, bars, solids, etc.
- the existing possibility for the derivation of new, more suitable finite elements which will produce better results with correspondingly shorter execution times,
- the finite element structure closely resembles the actual structure instead of being a quite different abstraction that is hard to visualize.

Other approximate methods - for example, the finite difference method - lack these attributes or accomodate them less readily. In the meantime, some disadvantages of the method can be stated as:

- a specific numerical result is obtained for a specific problem. A general closed form solution, which would permit one to examine system response to changes in various parameters, is not produced,

- experience and judgement (common sense) are needed in order to construct a good FEM,

- a large computer and a reliable computer program are essential since the requirement for a large computer core storage is immense. Therefore, the possibility exists that FEM could not be used on a smaller computer.

- input and output data may be large and tedious to prepare and interpret.

These drawbacks, however, are not unique to the FEM.

During the work done in the thesis studies, a CDC 170/815 model computer system installed at the Computer Center of Boğaziçi University and the high level computer language FORTRAN V is utilized.

PART I

DERIVATION OF THE FINITE ELEMENT EQUATION FOR THE THREE DIMENSIONAL HEAT CONDUCTION

In the heat transfer problems, the basic unknown is the temperature variation within the body, and the finite element equations can be derived either by using a suitable variational principle or from the governing differential equation using the Galerkin method. It is seen in the end that the same equilibrium equation is obtained independent of the choice in derivation methods.

The governing differential equation for heat conduction in three dimensional bodies can be obtained when the energy balance is considered for an infinitesimal cube in space. The procedure is given in detail by Hollman [1]. The resulting equation becomes

$$\frac{\partial}{\partial x}\left(k_x\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z\frac{\partial T}{\partial z}\right) + \dot{q} = \rho c\frac{\partial T}{t} \tag{2}$$

which is given in a form that would enable the study of anisotropic three dimensional solid body.

1.1. Variational Approach

1.1.1. Variational Statement of the Problem

The three dimensional heat conduction problem can be stated in an equivalent variational form as follows, find the temperature distribution T(x,y,z,t) inside the solid body which minimizes the integral

$$I = \frac{1}{2}\int_V \left[k_x\left(\frac{\partial T}{\partial x}\right) + k_y\left(\frac{\partial T}{\partial y}\right) + k_z\left(\frac{\partial T}{\partial z}\right) + 2\left(\dot{q} + \rho c\frac{\partial T}{\partial t}\right)T\right] dV$$

$$+ \int_{S_2} qT dS + \frac{1}{2}\int_{S_3} h(T-T_\infty)^2 \, dS \qquad (3)$$

and satisfies the boundary conditions

$$T(x,y,z,t) = T_s \qquad \text{for} \qquad t > 0 \quad \text{on } S_1$$

$$k_x\frac{\partial T}{\partial x}.l_x + k_y\frac{\partial T}{\partial y}.l_y + k_z\frac{\partial T}{\partial z}.l_z + q = 0 \qquad \text{for} \quad t>0 \text{ on } S_2$$

$$k_x\frac{\partial T}{\partial x}.l_x + k_y\frac{\partial T}{\partial y}.l_y + k_z\frac{\partial T}{\partial z}.l_z + h(T-T_\infty) = 0 \quad \text{for } t>0 \text{ on } S_3$$

and the initial condition

$$T(x,y,z,t)\Big|_{t=0} = T_o(x,y,z) \quad \text{in} \quad V$$

where l_x , l_y and l_z are the direction cosines. The equation (2) is first order in time and requires only one initial condition.

S_1 is the boundary on which the value of temperature is specified as $T_s(t)$, (Drichtlet condition); S_2 is the boundary on which the heat flux q is specified; S_3 is the boundary on which the connective heat flux $h(T-T_\infty)$ is specified (Newmann condition).

1.1.2. Derivation of the Finite Element Equation Using the Variational Approach

If a suitable form for variation of temperature in each finite element is assumed and expressed as

$$T^e(x,y,z,t) = \left[N(x,y,z)\right] \vec{T}^e \tag{4}$$

where the domain V is divided into E finite elements each having p nodes. In this case N_i (i=1 top) are the interpolation functions corresponding to the nodes numbered one to p in local numbering within the element. Similarly,

$$\vec{T} = \left[T_1(t), T_2(t), \ldots, T_p(t)\right]^{eT} \tag{5}$$

Keeping in mind the term $\frac{\partial T}{\partial t}$ as fixed, the variations can be taken in the volume.

As an initial step, the functional I in equation (3) has to be expressed as a sum of E elemental quantities I^e

$$I = \sum_{e=1}^{E} I^e \tag{6}$$

where

$$I^e = \frac{1}{2} \int_{V^e} \left[k_x\left(\frac{\partial T}{\partial x}\right) + k_y\left(\frac{\partial T}{\partial y}\right) + k_z\left(\frac{\partial T}{\partial z}\right) - 2\left(\dot{q} - \rho c \frac{\partial T^e}{\partial t}\right) T \right] dV$$

$$+\int_{S_2^e} q \cdot T^e \, dS + \frac{1}{2}\int_{S_3^e} h\,(T - T_\infty)^2 \, dS$$

for the minimization of the functional I, use

$$\frac{\partial I}{\partial T_i} = \sum_{e=1}^{E} \frac{\partial I^e}{\partial T_i} = 0 \quad , \quad i = 1,\ldots,NNP \qquad (7)$$

where NNP stands for the number of nodal points. From equation (6) the equation below follows

$$\frac{\partial I^e}{\partial T_i} = \int_{V^e} \left[k_x \frac{\partial T^e}{\partial x} \frac{\partial}{\partial T_i}\left(\frac{\partial T^e}{\partial x}\right) + k_y \frac{\partial T^e}{\partial y} \frac{\partial}{\partial T_i}\left(\frac{\partial T^e}{\partial y}\right) + k_z \frac{\partial T^e}{\partial z} \frac{\partial}{\partial T_i}\left(\frac{\partial T^e}{\partial z}\right) \right.$$

$$\left. - \left(\dot{q} - \rho c \frac{\partial T^e}{\partial t}\right) \frac{\partial T^e}{\partial T_i} \right] dV + \int_{S_2^e} q \frac{\partial T^e}{\partial T_i} \, dS + \int_{S_3^e} h\,(T^e - T_\infty) \frac{\partial T^e}{\partial T_i} \, dS \qquad (8)$$

in which the surface integrals do not exist if the node i is not on the surface, Equation (4) gives

$$\frac{\partial T^e}{\partial x} = \left[\frac{\partial N_1}{\partial x}, \frac{\partial N_2}{\partial x}, \ldots, \frac{\partial N_r}{\partial x} \right] \vec{T}^e$$

$$\frac{\partial}{\partial T_i}\left(\frac{\partial T^e}{\partial x}\right) = \frac{\partial N_i}{\partial x}$$

$$\frac{\partial T^e}{\partial T_i} = N_i$$

$$\frac{\partial T}{\partial t} = [N]\, \dot{\vec{T}}^e \quad , \text{ where } \quad \dot{\vec{T}}^e = \left[\frac{\partial T_1^e}{\partial t}, \ldots, \frac{T_r^e}{t} \right]^T$$

if the equations above are substituted into equation (8) and if the similar terms are put together the variation becomes

$$\left(\frac{\partial I^e}{\partial T_i}\right)_{jk} = \int_{V^e}\left[k_x \frac{\partial N_j}{\partial x}\frac{\partial N_k}{\partial x} + k_y \frac{\partial N_j}{\partial y}\frac{\partial N_k}{\partial y} + k_z \frac{\partial N_j}{\partial z}\frac{\partial N_k}{\partial z}\right]\vec{T}^e \, dV$$

$$+\int_{S_3^e} hN_j N_k \vec{T}^e dS + \int_{V^e} \rho c N_j N_k \dot{\vec{T}}^e dV + \int_{V^e} \dot{q} N_k \, dV$$

$$-\int_{S_2^e} qN_k \, dS + \int_{S_3^e} hT_\infty N_k \, dS$$

if the first integral is named as $K_{1,jk}$ and the rest as $K_{2,jk}$, $K_{3,jk}$, $P_{1,k}$, $P_{2,k}$ and $P_{3,k}$ respectively, the equation (9) can be rewritten as a matrix equation

$$\frac{\partial I}{\partial \vec{T}} = \sum_{e=1}^{E} \frac{\partial I^e}{\partial \vec{T}^e} = \sum_{e=1}^{E} \left[[K_1]+[K_2]\right]\vec{T}^e + [K_3]\,\dot{\vec{T}}^e - \vec{P} = 0$$

this can be restated in the overall matrix equation as

$$[K_3]\,\dot{\vec{T}} + [K_1 + K_2]\,\vec{T} = \vec{P} \qquad (10)$$

In equation (10), the matrices are global and are obtained after the assembly process applied to each element matrix.

1.2. Galerkin Approach

If equation (4) is reassumed over each element, the Galerkin method can be used readily. In the Galerkin method the integral of the weighted residue over the domain of the element is set equal to zero by taking the weights same as the interpolation functions N_i. The criteria to be satisfied at any instant of time is

$$\int_{V^e} N_i \left[\frac{\partial}{\partial x}(k_x \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(k_y \frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(k_z \frac{\partial T}{\partial z}) + \dot{q} - \rho c \frac{\partial T}{\partial t} \right] dV = 0 \quad (11)$$

$$i = 1 \text{ top}$$

Note that the Green-Gauss theorem can be applied to the above equation to give

$$-\int_V \left[k_x \frac{\partial N_i}{\partial x} \frac{\partial T^e}{\partial x} + k_y \frac{\partial N_i}{\partial y} \frac{\partial T^e}{\partial y} + k_z \frac{\partial N_i}{\partial z} \frac{\partial T^e}{\partial z} \right] dV$$

$$+ \int_S N_i \left[k_x \frac{\partial T^e}{\partial x} \cdot l_x + k_y \frac{\partial T^e}{\partial y} \cdot l_y + k_z \frac{\partial T}{\partial z} \cdot l_z \right] dS$$

$$+ \int_V N_i \left[\dot{q} - \rho c \frac{\partial T}{\partial t} \right] dV = 0 \quad \text{for } i = 1 \text{ top} \quad (12)$$

the boundary conditions specified by equation (3) forces the terms in the surface integral to be modified. Since the surface S is composed of three types of boundaries, the integral can also be restated as the sum of three integrals. The term concerning S_1 will be automatically zero, for the derivatives will be zero for constant temperatures. The terms concerning S_2 and S_3 can be summed as

$$\int_{S_2+S_3} N_i \left[k_x \frac{\partial T^e}{\partial x} \cdot l_x + k_y \frac{\partial T}{\partial y} \cdot l_y + k_z \frac{\partial T}{\partial z} \cdot l_z \right] dS$$

$$= -\int_{S_2} N_i \, q \, dS - \int_{S_3} N_i \, h \, (T^e - T_\infty) \, dS \quad (13)$$

when equation (4) is assumed, and when a similar procedure to that done in variational solution is applied to equations (12) and (13), the general matrix equation applicable to three dimensional conduction heat transfer problems is obtained once again. The form and notation of the equation has not the slightest difference from those of equation (10).

1.3. Interpretation of the Finite Element Equation

The index notation used in the derivation of equation (10) can be given in matrix notation which is easier to understand and more clear. Define

$$\begin{aligned}
[K_1]^e &= \int_{V^e} [B]^T [D] [B] \, dV \qquad (14)\\
[K_2]^e &= \int_{S_3^e} h \, [N]^T [N] \, dS\\
[K_3]^e &= \int_{V^e} \rho c \, [N]^T [N] \, dV\\
\vec{P}^e &= \vec{P}_1^e - \vec{P}_2^e + \vec{P}_3^e \quad , \text{ where}\\
\vec{P}_1^e &= \int_{V^e} \dot{q} \, [N]^T \, dV\\
\vec{P}_2^e &= \int_{S_2^e} q \, [N]^T \, dS\\
\vec{P}_3^e &= \int_{S_3^e} h \, T_\infty \, [N]^T \, dS
\end{aligned}$$

$$[D] = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix}$$

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \cdots & \frac{\partial N_p}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \cdots & \frac{\partial N_p}{\partial y} \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{z} & \cdots & \frac{\partial N_p}{\partial z} \end{bmatrix}$$

The above restatements are used throughout the programs and they form the basics in the algorithms. When suitable interpolation functions are used in place of N above, any kind of element can be used in the FEM formulations.

1.3.1. Steady State Application

Steady state is when the problem becomes independent of time, i.e. when the equation (2) becomes

$$\frac{\partial}{\partial x}(k_x\frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(k_y\frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(k_z\frac{\partial T}{\partial z}) + \dot{q} = 0 \qquad (15)$$

which states the general partial differential equation for three dimensional steady state conduction heat transfer problem in anisotropic materials. Similarly, the equation (10) is modified by setting $T = 0$ as

$$\left[K_1 + K_2\right] \cdot \vec{T} = \vec{P} \qquad (16)$$

which is in a form that can be treated easily once the matrices and the load vector are obtained thru the FEM analysis. The temperatures at the nodes $i = 1$ to NNP defined in the vector T are sought.

1.3.2. Transient Application

The equation (10) obtained after the FEM analysis is a set of first order linear differential equations. It can be seen that the term $\left[K_3\right]\vec{T}$ is the additional term that appears because of the unsteady state, where $\left[K_3\right]$ is called element heat capacity matrix.

In order to be able to solve the equation (10) numerically, a finite difference solution is applied in the time domain. This is a two point solution scheme and is based on approximating the first derivative of the time dependent quantity T as

$$\left.\frac{d\vec{T}}{dt}\right|_t = \frac{\vec{T}_1 - \vec{T}_0}{\Delta t} \qquad (17)$$

where

$$\vec{T}_1 = \vec{T}\left(t + \frac{\Delta t}{2}\right) \qquad \text{and} \qquad \vec{T}_0 = \vec{T}\left(t - \frac{\Delta t}{2}\right)$$

and Δt is the time step. The equation (17) approximates the derivative at the midpoint of the time interval t, so the terms $\vec{T}$ and $\vec{P}$ involved in equation (10) also have to be evaluated at the same point. Define

$$\left.\vec{T}\right|_t = \frac{\vec{T}_1 + \vec{T}_0}{2} \qquad (18)$$

$$\left.\vec{P}\right|_t = \frac{\vec{P}_1 + \vec{P}_0}{2}$$

The substitution of equations (17) and (18) in equation (10) gives

$$\frac{1}{2}\left[K_1 + K_2\right](\vec{T}_1 + \vec{T}_0) + \frac{1}{\Delta t}\left[K_3\right](\vec{T}_1 - \vec{T}_0) = \left.\vec{P}\right|_t$$

rearranging,

$$\left\{\left[K_1 + K_2\right] + \frac{2}{\Delta t}\left[K_3\right]\right\} \vec{T}\Big|_t = \frac{2}{\Delta t}\left[K_3\right] \vec{T}_o - \vec{P}\Big|_t \qquad (19)$$
$$\vec{T}_1 = 2\,\vec{T}\Big|_t - \vec{T}_o$$

The set of equations (19) shows that the nodal temperatures $\vec{T}$ at the time $(t + \Delta t/2)$ can be computed once the nodal temperatures at time $(t - \Delta t/2)$ are known since $\vec{P}\big|_t$ can be evaluated before solving equations (19). Thus, the known initial conditions on nodal temperatures can be used to find the solution at subsequent time steps.

PART II

DERIVATION OF ELEMENT MATRICES USED IN FINITE ELEMENT ANALYSIS

The previous derivation of matrix differential equation (10), in general terms, finds a straight forward application in the broad three dimensional heat transfer subject. No matter which kind of problem is solved, the first and very important step to be taken in order to get a start on the subject is to define which type of element is to be used. The procedure differs only in the algebra used, while the derivations and the evaluation of integrals remain the same when different elements are applied.

In the study accomplished, mainly two types of elements are used, the latter being an isoparametric element [3] according to the definition which requires the field variable and the Cartesian coordinates at a given point within the element, both, to be defined in terms of the nodal field variable values, for each, using the same interpolation functions. This can be restated as

$$x = [N]\begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_r \end{Bmatrix} \quad \text{and} \quad \phi = [N]\begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_r \end{Bmatrix} \tag{20}$$

These two types of elements happen to be the tetrahedron element and the hexahedron element for which the derivations follow.

2.1. Tetrahedron Element

The three dimensional tetrahedron element is a flat faced tetrahedron with four nodes, one at each corner. Let the nodes be labeled as i,j,k and l in local numeration. Note that the nodes will sometimes be recalled as 1,2,3 and 4 respectively.

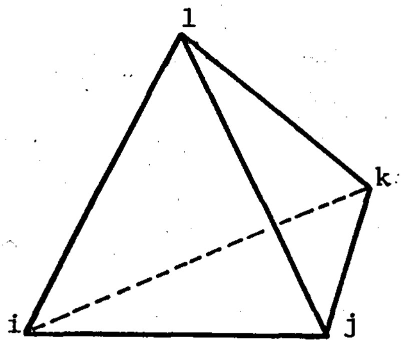

Figure 2.1

Let the values ϕ_i , ϕ_j , ϕ_k and ϕ_ℓ be the nodal values of the field variable and assign the Cartesian coordinates

$$(x_i, y_i, z_i) \;,\; (x_j, y_j, z_j) \;,\; (x_k, y_k, z_k) \;,\; (x_\ell, y_\ell, z_\ell)$$

for each node.

If a linear variation of the field variable is assumed as

$$\emptyset(x,y,z) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z \qquad (21)$$

which implies the nodal conditions

$$\begin{aligned}
\emptyset_i &= \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z \\
\emptyset_j &= \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z \\
\emptyset_k &= \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z \\
\emptyset_\ell &= \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z
\end{aligned} \qquad (22)$$

The set of equations (22) can be solved to give

$$\begin{aligned}
\alpha_1 &= (a_i \emptyset_i + a_j \emptyset_j + a_k \emptyset_k + a_\ell \emptyset_\ell) \,/\, 6V \\
\alpha_2 &= (b_i \emptyset_i + b_j \emptyset_j + b_k \emptyset_k + b_\ell \emptyset_\ell) \,/\, 6V \\
\alpha_3 &= (c_i \emptyset_i + c_j \emptyset_j + c_k \emptyset_k + c_\ell \emptyset_\ell) \,/\, 6V \\
\alpha_4 &= (d_i \emptyset_i + d_j \emptyset_j + d_k \emptyset_k + d_\ell \emptyset_\ell) \,/\, 6V
\end{aligned} \qquad (23)$$

where V is the volume of the tetrahedron defined as

$$V = \frac{1}{6} \begin{vmatrix} 1 & x_i & y_i & z_i \\ 1 & x_j & y_j & z_j \\ 1 & x_k & y_k & z_k \\ 1 & x_\ell & y_\ell & z_\ell \end{vmatrix} \qquad (24)$$

and where the coefficient a , b , c and d are given as

$$a_i = \begin{vmatrix} x_j & y_j & z_j \\ x_k & y_k & z_k \\ x_\ell & y_\ell & z_\ell \end{vmatrix}$$

$$b_i = -\begin{vmatrix} 1 & y_j & z_j \\ 1 & y_k & z_k \\ 1 & y_\ell & z_\ell \end{vmatrix}$$

$$c_i = -\begin{vmatrix} x_j & 1 & z_j \\ x_k & 1 & z_k \\ x_\ell & 1 & z_\ell \end{vmatrix} \qquad (24)$$

$$d_i = -\begin{vmatrix} x_j & y_j & 1 \\ x_k & y_k & 1 \\ x_\ell & y_\ell & 1 \end{vmatrix}$$

The rest of the coefficients are obtained by using the same determinants as given in equations (25) then the cyclic interchange of the subscripts in the order i, j, k, l. When these determinants are substituted in equation (21), the below equation is obtained in closed form

$$\emptyset\,(x,y,z) = N_i\,(x,y,z)\,\emptyset_i + N_j\,(x,y,z)\,\emptyset_j + N_k\,(x,y,z)\,\emptyset_k + N_\ell\,(x,y,z)\,\emptyset_\ell$$

$$\emptyset = \left[N(x,y,z)\right]\vec{\emptyset}^{e} \qquad (25)$$

where

$$\vec{\emptyset}^{e} = \begin{Bmatrix} \emptyset_1 \\ \emptyset_2 \\ \emptyset_3 \\ \emptyset_4 \end{Bmatrix}$$

and

$$N_i(x,y,z) = (a_i + b_i x + c_i y + d_i z) \ / \ 6V$$
$$N_j(x,y,z) = (a_j + b_j x + c_j y + d_j z) \ / \ 6V$$
$$N_k(x,y,z) = (a_k + b_k x + c_k y + d_k z) \ / \ 6V$$
$$N_\ell(x,y,z) = (a_\ell + b_\ell x + c_\ell y + d_\ell z) \ / \ 6V$$

In order to ease the calculations and to obtain consistency, a set of local coordinates must be defined within the element. These natural coordinates consist of four variables each corresponding to one corner respectively

$$L_1 = \frac{V_1}{V} \ , \ L_2 = \frac{V_2}{V} \ , \ L_3 = \frac{V_3}{V} \ , \ L_4 = \frac{V_4}{V} \qquad (26)$$

where V is the volume of the tetrahedron formed by the points P and the vertices other than i. The coordinates of point P within the element is given as (x,y,z) in global Cartesian coordinates and as (L_1, L_2, L_3, L_4) in local coordinate system.

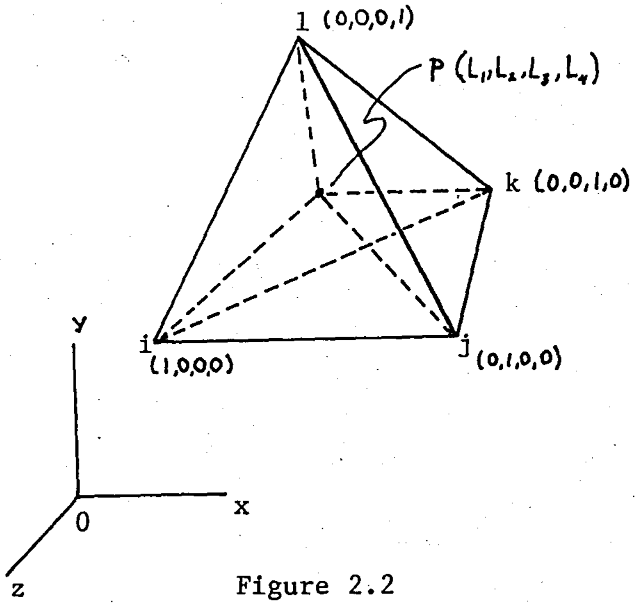

Figure 2.2

The definition of the local coordinates imply that node i has the local coordinates (1,0,0,0), node j has them as (0,1,0,0), etc. Another deduction can be made as

$$L_1 + L_2 + L_3 + L_4 = 1 \tag{27}$$

The cartesian and local coordinates are related by

$$\begin{aligned} x &= L_1 x_1 + L_2 x_2 + L_3 x_3 + L_4 x_4 \\ y &= L_1 y_1 + L_2 y_2 + L_3 y_3 + L_4 y_4 \\ z &= L_1 z_1 + L_2 z_2 + L_3 z_3 + L_4 z_4 \end{aligned} \tag{28}$$

which is implied by the geometry given in Figure 2 and by equations (20). The assembly of equations (27) and (28) give

$$\begin{Bmatrix} 1 \\ x \\ y \\ z \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{Bmatrix}$$

or

$$\begin{Bmatrix} L \\ L \\ L \\ L \end{Bmatrix} = \frac{1}{6V} \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix} \begin{Bmatrix} 1 \\ x \\ y \\ z \end{Bmatrix} \tag{29}$$

where the indices a_i , b_i , c_i and d_i are defined previously by equations (24).

In order to use the natural coordinates in the integrals, they have to be differentiable and integrable. If a function is given in local coordinates, it can be differentiated with respect to cartesian coordinates as

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial L_1} \cdot \frac{\partial L_1}{\partial x} + \frac{\partial f}{\partial L_2} \cdot \frac{\partial L_2}{\partial x} + \frac{\partial f}{\partial L_3} \cdot \frac{\partial L_3}{\partial x} + \frac{\partial f}{\partial L_4} \cdot \frac{\partial L_4}{\partial x} \tag{30}$$

$$\frac{\partial f}{\partial y} = \sum_{i=1}^{4} \frac{\partial f}{\partial L_i} \cdot \frac{\partial L_i}{\partial y}$$

$$\frac{\partial f}{\partial z} = \sum_{i=1}^{4} \frac{\partial f}{\partial L_i} \cdot \frac{\partial L_i}{\partial z}$$

where

$$\frac{\partial L_i}{\partial x} = \frac{b_i}{6V} \quad , \quad \frac{\partial L_i}{\partial y} = \frac{c_i}{6V} \quad , \quad \frac{\partial L_i}{\partial z} = \frac{d_i}{6V}$$

which can be obtained directly if equation (29) is differentiated.

The integration of the polynomial terms given in natural coordinates can be performed by using the relation

$$\int_V L_1^{\alpha} \cdot L_2^{\beta} \cdot L_3^{\gamma} \cdot L_4^{\delta} \cdot dV = \frac{\alpha!\ \beta!\ \gamma!\ \delta}{(\alpha + \beta + \gamma + \delta + 3)!} 6V \tag{31}$$

given over a volume.

When surface integrals are required, the interpolation functions remain the same, only, the functions corresponding to nodes not on the related surface are set equal to zero. If the necessary terms in the surface integrals are set equal to zero the remaining terms can be integrated using

$$\int_S L_1^{\alpha} \cdot L_2^{\beta} \cdot L_3^{\gamma} \cdot dS = \frac{\alpha!\ \beta!\ \gamma!}{(\alpha + \beta + \gamma + 2)!} 2A \qquad (32)$$

where

$$A = \left[S(S - \lambda_1)(S - \lambda_2)(S - \lambda_3)\right]^{1/2} \qquad (33)$$

in which S are defined as demonstrated in Figure 3.

$$\lambda_1 = \left[(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2\right]^{1/2}$$
$$\lambda_2 = \left[(x_j - x_k)^2 + (y_j - y_k)^2 + (z_j - z_k)^2\right]^{1/2}$$
$$\lambda_3 = \left[(x_i - x_k)^2 + (y_i - y_k)^2 + (z_i - z_k)^2\right]^{1/2}$$
$$S = (\lambda_1 + \lambda_2 + \lambda_3) / 2$$

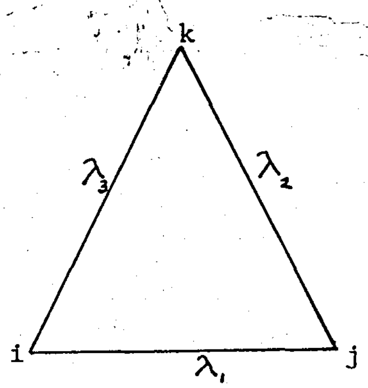

Figure 2.3

The surface integrals and their application require the equa- tion (33) to be used frequently. The derivatives above are valid for all tetrahedron elements.

2.1.1. Simplex Tetrahedron Element

Given that the body is divided into E tetrahedron elements, the temperature distribution over element e can be stated as

$$T = [N]\,\vec{T}^{e}$$

where

$$[N] = [N_i \; N_j \; N_k \; N_\ell] = [L_1\,,\,L_2\,,\,L_3\,,\,L_4]$$

$$\vec{T}^{e} = \begin{Bmatrix} T_i \\ T_j \\ T_k \\ T_\ell \end{Bmatrix}$$

A typical simplex tetrahedron element is illustrated in Figure 2.1. Given equations (14), the element matrices and thermal load vectors are derived as

$$[D] = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix}$$

$$[B] = \begin{bmatrix} \frac{\partial N_i}{\partial x} & \frac{\partial N_j}{\partial x} & \cdots\cdots \\ \frac{\partial N_i}{\partial y} & \cdots\cdots\cdots\cdots & \\ \frac{\partial N_i}{\partial z} & \cdots\cdots & \frac{\partial N_\ell}{\partial z} \end{bmatrix} = \frac{1}{6V^{e}} \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix}$$

giving

$$[K_1]^{e} = \int_{V^{e}} [B]^{T}[D][B]\,dV$$

$$= \frac{k_x}{36V} \begin{bmatrix} b_1^2 & & & \text{SYM} \\ b_1\,b_2 & b_2^2 & & \\ b_1\,b_3 & b_2\,b_3 & b_3^2 & \\ b_1\,b_4 & b_2\,b_4 & b_3\,b_4 & b_4^2 \end{bmatrix}$$

$$+ \frac{k_y}{36V} \begin{bmatrix} c_1^2 & & \text{SYM} & \\ c_1 c_2 & c_2^2 & & \\ c_1 c_3 & c_2 c_3 & c_3^2 & \\ c_1 c_4 & c_2 c_4 & c_3 c_4 & c_4^2 \end{bmatrix}$$

$$+ \frac{k_z}{36V} \begin{bmatrix} d_1^2 & & \text{SYM} & \\ d_1 d_2 & d_2^2 & & \\ d_1 d_3 & d_2 d_3 & d_3^2 & \\ d_1 d_4 & d_2 d_4 & d_3 d_4 & d_4^2 \end{bmatrix}$$

$$\left[K_2\right]^e_{ijk} = \int_{S^e_{ijk}} h \, [N]^T [N] \, dS \quad , \text{ setting } \; L_4 = 0$$

$$= \frac{hA_{ijk}}{12} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$\left[K_2\right]^e$ matrix can be obtained readily for the other surfaces in a similar form.

$$\left[K_3\right]^e = \int_{V^e} \rho c \, [N]^T [N] \, dV$$

$$= \frac{(\rho c)^e V^e}{20} \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix}$$

$$\left[P_1\right]^e = \int_{V^e} \dot{q} \, [N]^T \, dV = \frac{(\dot{q} \, V)^e}{4} \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix}$$

$$\left[P_2\right]^e_{ijk} = \int_{S^e_{ijk}} q \, [N]^T \, dS = \frac{(q A_{ijk})^e}{3} \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix}$$

$$\left[P_3\right]^e_{ijk} = \int_{S^e_{ijk}} h\,T_\infty \left[N\right]^T dS = \frac{(h\,T_\infty A_{ijk})^e}{3} \begin{Bmatrix}1\\1\\1\\1\end{Bmatrix}$$

2.1.2. Parabolic Tetrahedron Element

A typical parabolic element is demonstrated in Figure 2.4. Keeping the same assumptions as those considered for the simplex element, the interpolation functions can be obtained readily using the Lagrange interpolation formula.

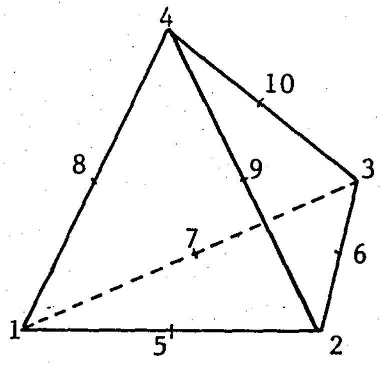

Figure 2.4

$$N_i = f^i(L_1)\cdot f^i(L_2)\cdot f^i(L_3)\cdot f^i(L_4) \qquad (34)$$

where

$$f^i(L_j) = \begin{cases} \prod_{k=1}^{r} \frac{1}{k}(m\,L_j - k + 1) & \text{for } r \geqslant 1 \\ 1 & \text{for } r = 0 \end{cases}$$

in which

m = degree of approximation; one for simplex, two for quadratic, etc.

$i = 1$ to p

$r = m\, L_j^i$

L_j^i = value of the natural coordinate L_j at node i.

With some effort, the general equation (34) can be obtained from the approximation polynomials and the derivations that follow.

In the parabolic tetrahedron element, the approximation polynomial is chosen to be quadratic as

$$\emptyset(x,y,z) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z + \alpha_5 x^2 + \alpha_6 y^2 + \alpha_7 z^2 + \alpha_8 xy + \alpha_9 yz + \alpha_{10} xz$$

The calculations with ten unknown coefficients require ten nodes, and ten equations are obtained from which the coefficients can be extracted. This is a cumbersome process, rather, the Lagrange interpolation formula should be applied.

The shape functions for the parabolic element are

$$N_i = L_i(2L_i - 1) \quad \text{for} \quad i = 1,2,3,4$$
$$N_5 = 4\,L_1 L_2$$
$$N_6 = 4\,L_2 L_3$$
$$N_7 = 4\,L_1 L_3$$
$$N_8 = 4\,L_1 L_4$$
$$N_9 = 4\,L_2 L_4$$
$$N_{10} = 4\,L_3 L_4$$

and if the field vector is defined as

$$\vec{\phi}^e = \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{10} \end{Bmatrix}$$

then

$$\phi(x,y,z) = \left[N_1 , N_2 , \ \ldots , N_{10} \right] \vec{\phi}^e$$

The natural coordinates of the nodes can be given as

$$\vec{L}_1 = \begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1/2 \\ 0 \\ 1/2 \\ 1/2 \\ 0 \\ 0 \end{Bmatrix} , \quad \vec{L}_2 = \begin{Bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1/2 \\ 1/2 \\ 0 \\ 0 \\ 1/2 \\ 0 \end{Bmatrix} , \quad \vec{L}_3 = \begin{Bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1/2 \\ 1/2 \\ 0 \\ 0 \\ 1/2 \end{Bmatrix} , \quad \vec{L}_4 = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1/2 \\ 1/2 \\ 1/2 \end{Bmatrix}$$

Having defined the interpolation functions, the other terms appearing in the integrals are set as

$$[D] = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix}$$

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \cdots & \frac{\partial N_{10}}{\partial x} \\ \frac{\partial N_1}{\partial y} & \cdots & \cdots & \frac{\partial N_{10}}{\partial y} \\ \frac{\partial N_1}{\partial z} & \cdots & \cdots & \frac{\partial N_{10}}{\partial z} \end{bmatrix}$$

where

$$\frac{\partial N_i}{\partial x} = (4L_i - 1)\, b_i/6V \qquad \text{for} \quad i = 1,2,3,4$$

$$\frac{\partial N_5}{\partial x} = 4(L_2 b_1 + L_1 b_2)/6V \quad , \qquad \frac{\partial N_8}{\partial x} = 4(L_4 b_1 + L_1 b_4)/6V$$

$$\frac{\partial N_6}{\partial x} = 4(L_3 b_2 + L_2 b_3)/6V \qquad \frac{\partial N_9}{\partial x} = 4(L_4 b_2 + L_2 b_4)/6V$$

$$\frac{\partial N_7}{\partial x} = 4(L_1 b_3 + L_3 b_1)/6V \qquad \frac{\partial N_{10}}{\partial x} = 4(L_4 b_3 + L_3 b_4)/6V$$

The derivatives with respect to y are obtained similarly if b_i in the above equations are replaced with c_i , and those with respect to z are rewritten with d_i's instead.

Once the above terms are put in the equations (14), the element matrices are obtained after a lengthy algebraic work. Should it be necessary, the open forms of the element matrices can be deduced from the program listing in Appendix E.

2.1.3. Cubic Tetrahedron Element

A typical cubic element is given in Figure 2.5. As was the case in parabolic element, the shape functions can be readily obtained thru the Lagrange interpolation formula as given in equation (34).

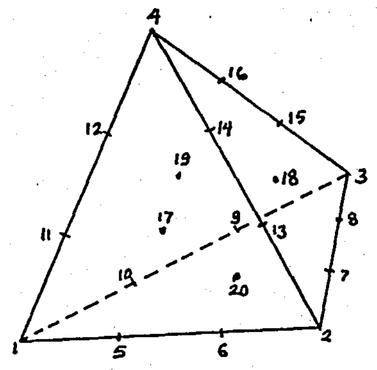

Figure 2.5.

The cubic approximation polynomial includes the terms produced by the Pascal tetrahedron as

$$\begin{aligned}\emptyset(x,y,z) = {} & \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z + \alpha_5 x^2 + \alpha_6 y^2 + \alpha_7 z^2 \\ & + \alpha_8 xy + \alpha_9 yz + \alpha_{10} xz + \alpha_{11} x^3 + \alpha_{12} y^3 + \alpha_{13} z^3 \\ & + \alpha_{14} x^2 y + \alpha_{15} x^2 z + \alpha_{16} y^2 z + \alpha_{17} xy^2 + \alpha_{18} xz^2 + \alpha_{19} yz^2 \\ & + \alpha_{20} xyz\end{aligned}$$

The twenty unknown coefficients require twenty nodes as illustrated in Figure 2.5, thus a system of equations consisting of twenty equilibrium conditions. The field variable is defined as

$$\emptyset(x,y,z) = \lceil N_1 , N_2 , \ldots , N_{20} \rceil \, \vec{\emptyset}^e = \lceil N(x,y,z) \rceil \, \vec{\emptyset}^e$$
$$\vec{\emptyset}^e = \left[\emptyset_1 , \emptyset_2 , \ldots , \emptyset_{20} \right]^T$$

The shape functions can be stated as

$$N_i = L_i (3 L_i - 1)(3 L_i - 2)/2 \quad , \quad \text{for } i = 1,2,3,4$$

$$N_5 = 9 L_1 L_2 (3 L_1 - 1)/2 \quad , \quad \text{for one-third points of edges.}$$

$$N_6 = 9 L_1 L_2 (3 L_2 - 1)/2$$

$$N_7 = 9 L_2 L_3 (3 L_2 - 1)/2$$

$$N_8 = 9 L_2 L_3 (3 L_3 - 1)/2 \quad , \quad \text{etc.}$$

$$N_{17} = 27 L_1 L_2 L_4 \quad , \text{ for mid-face nodes}$$

$$N_{18} = 27 L_2 L_3 L_4$$

$$N_{19} = 27 L_1 L_3 L_4$$

$$N_{20} = 27 L_1 L_2 L_3$$

The natural coordinates of the nodes can be given as follows

$$\vec{L}_1 = \begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 2/3 \\ 1/3 \\ 0 \\ 0 \\ 1/3 \\ 2/3 \\ 2/3 \\ 1/3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/3 \\ 0 \\ 1/3 \\ 1/3 \end{Bmatrix} , \quad \vec{L}_2 = \begin{Bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1/3 \\ 2/3 \\ 2/3 \\ 1/3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2/3 \\ 1/3 \\ 0 \\ 0 \\ 1/3 \\ 1/3 \\ 0 \\ 1/3 \end{Bmatrix} , \quad \vec{L}_3 = \begin{Bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1/3 \\ 2/3 \\ 2/3 \\ 1/3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2/3 \\ 1/3 \\ 0 \\ 1/3 \\ 1/3 \\ 1/3 \end{Bmatrix} , \quad \vec{L}_4 = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/3 \\ 2/3 \\ 1/3 \\ 2/3 \\ 1/3 \\ 2/3 \\ 1/3 \\ 1/3 \\ 1/3 \\ 0 \end{Bmatrix}$$

In order to obtain the matrix $[B]$, the differentials of N_i with respect to x, y and z are required.

$$\frac{\partial N_i}{\partial x} = (27 L_i^2 - 18 L_i + 2) b_i / 12V \qquad i = 1,2,3,4$$

$$\frac{\partial N_5}{\partial x} = \frac{9}{12V} \left[(6L_1 L_2 - L_2) b_1 + (3L_1^2 - L_1) b_2 \right]$$

$$\frac{\partial N_6}{\partial x} = \frac{9}{12V} \left[(3L_2^2 - L_2) b_1 + (6L_1 L_2 - L_1) b_2 \right]$$

$$\frac{\partial N_7}{\partial x} = \frac{9}{12V}\left[(6L_2L_3 - L_2)b_2 + (3L_2^2 - L_2)b_3\right]$$

$$\frac{\partial N_8}{\partial x} = \frac{9}{12V}\left[(3L_3^2 - L_3)b_2 + (6L_2L_3 - L_2)b_3\right]$$

$$\frac{\partial N_9}{\partial x} = \frac{9}{12V}\left[(3L_3^2 - L_3)b_1 + (6L_1L_3 - L_1)b_3\right]$$

$$\frac{\partial N_{10}}{\partial x} = \frac{9}{12V}\left[(6L_1L_3 - L_1)b_1 + (3L_1^2 - L_1)b_3\right]$$

$$\frac{\partial N_{11}}{\partial x} = \frac{9}{12V}\left[(6L_1L_4 - L_4)b_1 + (3L_1^2 - L_1)b_4\right]$$

$$\frac{\partial N_{12}}{\partial x} = \frac{9}{12V}\left[(3L_4^2 - L_4)b_1 + (6L_1L_4 - L_1)b_4\right]$$

$$\frac{\partial N_{13}}{\partial x} = \frac{9}{12V}\left[(6L_2L_4 - L_4)b_2 + (3L_2^2 - L_2)b_4\right]$$

$$\frac{\partial N_{14}}{\partial x} = \frac{9}{12V}\left[(3L_4^2 - L_4)b_2 + (6L_2L_4 - L_2)b_4\right]$$

$$\frac{\partial N_{15}}{\partial x} = \frac{9}{12V}\left[(6L_3L_4 - L_4)b_3 + (3L_3^2 - L_3)b_4\right]$$

$$\frac{\partial N_{16}}{\partial x} = \frac{9}{12V}\left[(3L_4^2 - L_4)b_3 + (6L_3L_4 - L_3)b_4\right]$$

$$\frac{\partial N_{17}}{\partial x} = \frac{27}{6V}(L_2L_4b_1 + L_1L_4b_2 + L_1L_2b_4)$$

$$\frac{\partial N_{18}}{\partial x} = \frac{27}{6V}(L_3L_4b_2 + L_2L_4b_3 + L_2L_3b_4)$$

$$\frac{\partial N_{19}}{\partial x} = \frac{27}{6V}(L_3L_4b_1 + L_1L_4b_3 + L_1L_3b_4)$$

$$\frac{\partial N_{20}}{\partial x} = \frac{27}{6V}(L_2L_3b_1 + L_1L_3b_2 + L_1L_2b_3)$$

Just as the parabolic element, the differentials with respect to y and z are obtained by changing b_i with c_i and d_i , respectively.

The exact integration that has to be followed in order to obtain the element matrices is very lenghty and cumbersome, which in the end gives complex polynomial expressions thus increasing the computer time needed for the execution. Therefore, the exact integration is omitted. Instead, a numeric integration algorithm, namely, a five point Gauss Quadrature is applied over the volume.

$$\int_{V^e} f(L_1, L_2, L_3, L_4)\, dV = \sum_{i=1}^{5} w_i\, f(L_1, L_2, L_3, L_4)\Big|_i \qquad (35)$$

where collocation points and the weights are taken as

$$w_1 = -\frac{4}{5}, \quad L_1 = L_2 = L_3 = L_4 = \frac{1}{4}$$

$$w_2 = \frac{9}{20}, \quad L_1 = \frac{1}{3}, \quad L_2 = L_3 = L_4 = \frac{1}{6}$$

$$w_3 = \frac{9}{20}, \quad L_2 = \frac{1}{3}, \quad L_1 = L_3 = L_4 = \frac{1}{6}$$

$$w_4 = \frac{9}{20}, \quad L_3 = \frac{1}{3}, \quad L_1 = L_2 = L_4 = \frac{1}{6}$$

$$w_5 = \frac{9}{20}, \quad L_4 = \frac{1}{3}, \quad L_1 = L_2 = L_3 = \frac{1}{6}$$

For the surface integrals, the modified shape functions are used, and the integration is carried on similarly,

$$\int_S f(L_1, L_2, L_3)\, dS = \sum_{i=1}^{7} w_i\, f(L_1, L_2, L_3)\Big|_i \qquad (36)$$

where

$$w_1 = \frac{27}{60} \quad , \quad L_1 = L_2 = L_3 = \frac{1}{3}$$

$$w_2 = \frac{8}{60} \quad , \quad L_1 = L_2 = \frac{1}{2} \quad , \quad L_3 = 0$$

$$w_3 = \frac{8}{60} \quad , \quad L_2 = L_3 = \frac{1}{2} \quad , \quad L_1 = 0$$

$$w_4 = \frac{8}{60} \quad , \quad L_1 = L_3 = \frac{1}{2} \quad , \quad L_2 = 0$$

$$w_5 = \frac{3}{60} \quad , \quad L_1 = 1 \quad , \quad L_2 = L_3 = 0$$

$$w_6 = \frac{3}{60} \quad , \quad L_2 = 1 \quad , \quad L_1 = L_3 = 0$$

$$w_7 = \frac{3}{60} \quad , \quad L_3 = 1 \quad , \quad L_1 = L_3 = 0$$

The individual matrices within each integral are evaluated at the collocation points and then are multiplied to give the element matrix.

2.2. Hexahedron Element

2.2.1. Simplex Hexahedron Element

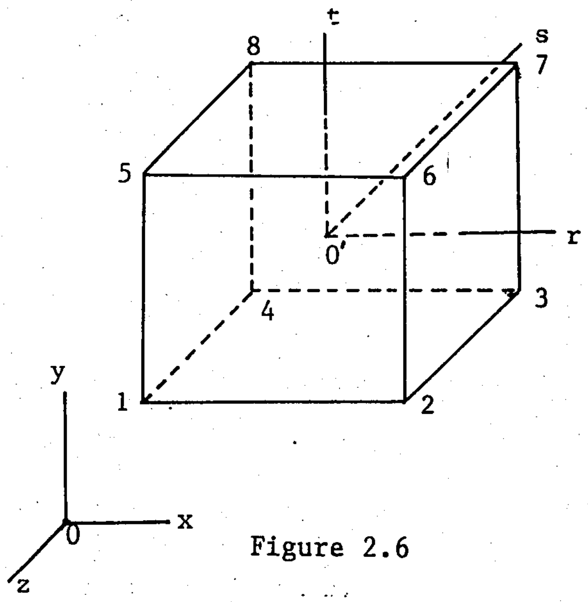

Figure 2.6

The hexahedron element is, by definition, an isoparametric element. For this reason, equations (20) can be written as follows in the case of simplex hexahedron element.

$$\phi(x,y,z) = \left[N_1, N_2, \ \ldots \ , N_8\right] \vec{\phi}^e = \left[N\right] \vec{\phi}^e$$
$$\vec{\phi}^e = \left[\phi_1, \phi_2, \ \ldots \ , \phi_8\right]^T \tag{37}$$

$$x = \left[N\right]\begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_8 \end{Bmatrix}, \quad y = \left[N\right]\begin{Bmatrix} y_1 \\ y_2 \\ \vdots \\ y_8 \end{Bmatrix}; \quad z = \left[N\right]\begin{Bmatrix} z_1 \\ z_2 \\ \vdots \\ z_8 \end{Bmatrix}$$

Once the hexahedron element is chosen to be the basic parent element in the FEM analysis of a body, a local coordinate system has to be developed. The new system is a unit right handed one with its origin at the center of the hexahedron.

When a coordinate transformation is concerned, a Jacobian has to be used. If a field variable ϕ is assumed, then

$$\begin{Bmatrix} \dfrac{\partial \phi}{\partial r} \\[2ex] \dfrac{\partial \phi}{\partial s} \\[2ex] \dfrac{\partial \phi}{\partial t} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial r} & \dfrac{\partial y}{\partial r} & \dfrac{\partial z}{\partial r} \\[2ex] \dfrac{\partial x}{\partial s} & \dfrac{\partial y}{\partial s} & \dfrac{\partial z}{\partial s} \\[2ex] \dfrac{\partial x}{\partial t} & \dfrac{\partial y}{\partial t} & \dfrac{\partial z}{\partial t} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial \phi}{\partial x} \\[2ex] \dfrac{\partial \phi}{\partial y} \\[2ex] \dfrac{\partial \phi}{\partial z} \end{Bmatrix} = \left[J\right] \begin{Bmatrix} \dfrac{\partial \phi}{\partial x} \\[2ex] \dfrac{\partial \phi}{\partial y} \\[2ex] \dfrac{\partial \phi}{\partial z} \end{Bmatrix}$$

which can be rewritten as a transformation function

$$\begin{Bmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \\ \frac{\partial \phi}{\partial z} \end{Bmatrix} = \left[J \right]^{-1} \begin{Bmatrix} \frac{\partial \phi}{\partial r} \\ \frac{\partial \phi}{\partial s} \\ \frac{\partial \phi}{\partial t} \end{Bmatrix} \tag{38}$$

Equations (37) imply that

$$[J] = \begin{bmatrix} \sum_{i=1}^{8} \frac{\partial N_i}{\partial r} x_i & \sum_{i=1}^{8} \frac{\partial N_i}{\partial r} y_i & \sum_{i=1}^{8} \frac{\partial N_i}{\partial r} z_i \\ \sum_{i=1}^{8} \frac{\partial N_i}{\partial s} x_i & \sum_{i=1}^{8} \frac{\partial N_i}{\partial s} y_i & \sum_{i=1}^{8} \frac{\partial N_i}{\partial s} z_i \\ \sum_{i=1}^{8} \frac{\partial N_i}{\partial t} x_i & \sum_{i=1}^{8} \frac{\partial N_i}{\partial t} y_i & \sum_{i=1}^{8} \frac{\partial N_i}{\partial t} z_i \end{bmatrix} \tag{39}$$

where

$$N_i = \frac{1}{8} (1 + rr_i)(1 + ss_i)(1 + tt_i) \quad , \quad i = 1 \text{ to } 8$$

$$\frac{\partial N}{\partial r} = \frac{r_i}{8} (1 + ss_i)(1 + tt_i)$$

$$\frac{\partial N}{\partial s} = \frac{s_i}{8} (1 + rr_i)(1 + tt_i)$$

$$\frac{\partial N}{\partial t} = \frac{t_i}{8} (1 + rr_i)(1 + ss_i)$$

in which r_i , s_i and t_i are the coordinates of the nodes in the local coordinate system.

$$\vec{r} = \begin{Bmatrix} -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \end{Bmatrix} , \quad \vec{s} = \begin{Bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \end{Bmatrix} , \quad \vec{t} = \begin{Bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix}$$

The integration of the functions of r, s and t have to be performed numerically keeping (38) in mind and with

$$dV = dx \cdot dy \cdot dz = \det\left[J\right] dr \cdot ds \cdot dt$$

A three point Gauss Quadrature [2,8] will give exact results for this case. For three dimensions

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(r,s,t)\, dr \cdot ds \cdot dt = \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3} w_i w_j w_k f(r_i, s_j, t_k) \tag{40}$$

and for two dimensions,

$$\int_{-1}^{1}\int_{-1}^{1} f(r,s)\, dr \cdot ds = \sum_{i=1}^{3}\sum_{j=1}^{3} w_i w_j f(r_i, s_j) \tag{41}$$

where

$$w_1 = w_3 = 0.\overline{555} \quad , \quad w_2 = 0.\overline{888}$$

$$-r_1 = r_3 = .77459\,6669241483 \quad , \quad r_2 = 0$$

The integrals in equations (40) and (41) are evaluated at 27 and nine collocation points, respectively.

The surface integrals require the planar interpolation functions to be known.

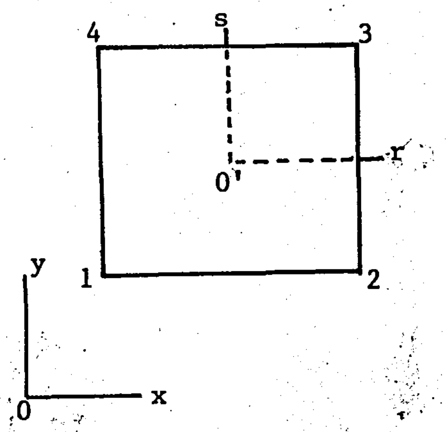

Figure 2.7

The quadrilateral in Figure 2.7 is also isoparametric, so that

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} N_1 & N_2 & N_3 & N_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & N_1 & N_2 & N_3 & N_4 \end{bmatrix} \begin{Bmatrix} x_1 \\ \vdots \\ x_4 \\ y_1 \\ \vdots \\ y_4 \end{Bmatrix}$$

$$\emptyset = [N_1\ N_2\ N_3\ N_4]\ \vec{\emptyset}^e$$
$$\vec{\emptyset}^e = [\emptyset_1\ \emptyset_2\ \emptyset_3\ \emptyset_4]^T$$

$$\begin{Bmatrix} \frac{\partial \emptyset}{\partial r} \\ \frac{\partial \emptyset}{\partial s} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} \begin{Bmatrix} \frac{\partial \emptyset}{\partial x} \\ \frac{\partial \emptyset}{\partial y} \end{Bmatrix} \quad \text{or} \quad \begin{Bmatrix} \frac{\partial \emptyset}{\partial x} \\ \frac{\partial \emptyset}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial \emptyset}{\partial r} \\ \frac{\partial \emptyset}{\partial s} \end{Bmatrix} \qquad (42)$$

where the Jacobian in the transformation function (42) can be written in open form as

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{4} \frac{\partial N_i}{\partial r} x_i & \sum_{i=1}^{4} \frac{\partial N_i}{\partial r} y_i \\ \sum_{i=1}^{4} \frac{\partial N_i}{\partial s} x_i & \sum_{i=1}^{4} \frac{\partial N_i}{\partial s} y_i \end{bmatrix}$$

and the shape function as

$$N_i = \frac{1}{4}(1 + rr_i)(1 + ss_i) \quad , \quad i = 1,2,3,4$$

$$\frac{\partial N_i}{\partial r} = \frac{r_i}{4}(1 + ss_i)$$

$$\frac{\partial N}{\partial s} = \frac{s_i}{4}(1 + rr_i)$$

$$\vec{r} = \begin{Bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{Bmatrix} \quad , \quad \vec{s} = \begin{Bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{Bmatrix}$$

The integrations (41) follow with

$$dA = dx\ dy = \det[J]\,dr\ ds$$

where the limits of integration are -1 to 1

2.2.2. Parabolic Hexahedron Element

A typical parabolic element is demonstrated in Figure 2.8 for which the same nomenclature is used as that for simplex hexahedron.

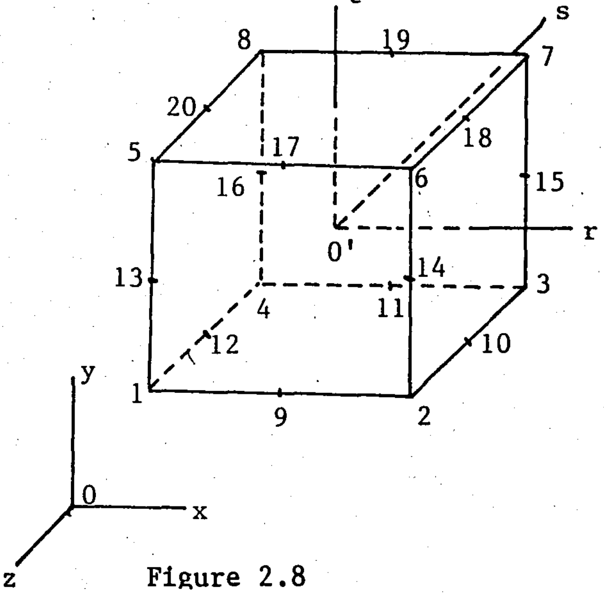

Figure 2.8

This element is isoparametric and the equations (20) can be rewritten for this case as

$$\phi(x,y,z) = [N_1, N_2, \ldots, N_{20}]\,\vec{\phi}^e$$

$$\vec{\phi}^e = [\phi_1\, \phi_2 \phi_3 \ldots\ \phi_{20}]^T$$

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{bmatrix} N_1\ N_2 \ldots N_{20} & 0\ 0 \ldots 0 & 0\ 0 \ldots 0 \\ 0\ 0 \ldots 0 & N_1\ N_2 \ldots N_{20} & 0\ 0 \ldots 0 \\ 0\ 0 \ldots 0 & 0\ 0 \ldots 0 & N_1\ N_2 \ldots N_{20} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{20} \\ y_1 \\ y_2 \\ \vdots \\ y_{20} \\ z_1 \\ z_2 \\ \vdots \\ z_{20} \end{Bmatrix}$$

The transformation function is again valid.

$$\begin{Bmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \\ \frac{\partial \phi}{\partial z} \end{Bmatrix} = \begin{bmatrix} \sum_{i=1}^{20} \frac{\partial N_i}{\partial r} x_i & \sum_{i=1}^{20} \frac{\partial N_i}{\partial r} y_i & \sum_{i=1}^{20} \frac{\partial N_i}{\partial r} z_i \\ \sum_{i=1}^{20} \frac{\partial N_i}{\partial s} x_i & \sum_{i=1}^{20} \frac{\partial N_i}{\partial s} y_i & \sum_{i=1}^{20} \frac{\partial N_i}{\partial s} z_i \\ \sum_{i=1}^{20} \frac{\partial N_i}{\partial t} x_i & \sum_{i=1}^{20} \frac{\partial N_i}{\partial t} y_i & \sum_{i=1}^{20} \frac{\partial N_i}{\partial t} z_i \end{bmatrix} \begin{Bmatrix} \frac{\partial \phi}{\partial r} \\ \frac{\partial \phi}{\partial s} \\ \frac{\partial \phi}{\partial t} \end{Bmatrix}$$

where

$$N_i = \frac{1}{8}(1+rr_i)(1+ss_i)(1+tt_i)(rr_i+ss_i+tt_i-2) \quad , \quad i=1 \text{ to } 8$$

$$N_i = \frac{1}{4}(1-r^2)(1+ss_i)(1+tt_i) \quad , \quad i=9,11,17,19$$

$$N = \frac{1}{4}(1-s^2)(1+rr_i)(1+tt_i) \quad , \quad i=10,12,18,20$$

$$N = \frac{1}{4}(1-t^2)(1+rr_i)(1+ss_i) \quad , \quad i=13,14,15,16$$

in which the local coordinates of the nodes can be given with the vectors

$$\vec{r} = \begin{Bmatrix} -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ 0 \\ 1 \\ 0 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ 0 \\ 1 \\ 0 \\ -1 \end{Bmatrix}, \quad \vec{s} = \begin{Bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ 0 \\ 1 \\ 0 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ 0 \\ 1 \\ 0 \end{Bmatrix}, \quad \vec{t} = \begin{Bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix}$$

For the surfaces of the hexahedron, the following has to be defined:

$$\emptyset = \left[N_1 \; N_2 \ldots \; N_8 \right] \vec{\emptyset}^e$$

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} N_1 \, N_2 \ldots & N_8 & 0 \; 0 & \ldots & 0 \\ 0 \; 0 \; \ldots & 0 & N_1 \, N_2 & \ldots & N_8 \end{bmatrix} \begin{Bmatrix} x_1 \\ \vdots \\ x_8 \\ y_1 \\ \vdots \\ y_8 \end{Bmatrix}$$

$$[J] = \begin{bmatrix} \dfrac{\partial x}{\partial r} & \dfrac{\partial y}{\partial r} \\ \dfrac{\partial x}{\partial s} & \dfrac{\partial y}{\partial s} \end{bmatrix} = \begin{bmatrix} \sum\limits_{i=1}^{8} \dfrac{\partial N_i}{\partial r} x_i & \sum\limits_{i=1}^{8} \dfrac{\partial N_i}{\partial r} y_i \\ \sum\limits_{i=1}^{8} \dfrac{\partial N_i}{\partial s} x_i & \sum\limits_{i=1}^{8} \dfrac{\partial N_i}{\partial s} y_i \end{bmatrix}$$

where

$$N_i = \frac{1}{4}(1+rr_i)(1+ss_i)(rr_i+ss_i-1) \quad \text{for } i=1,2,3,4$$

$$N_5 = \frac{1}{2}(1-r^2)(1+ss_5)$$

$$N_6 = \frac{1}{2}(1+rr_6)(1-s^2)$$

$$N_7 = \frac{1}{2}(1-r^2)(1+ss_7)$$

$$N_8 = \frac{1}{2}(1+rr_8)(1-s^2)$$

and

$$\vec{r} = \begin{Bmatrix} -1 \\ 1 \\ 1 \\ -1 \\ 0 \\ 1 \\ 0 \\ -1 \end{Bmatrix} , \quad \vec{s} = \begin{Bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ 0 \\ 1 \\ 0 \end{Bmatrix}$$

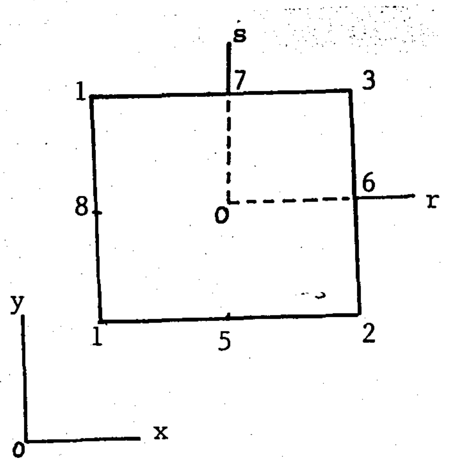

Figure 2.9

From the definitions given above for the interpolation functions, the differentials can be derived with some effort before substituting them in the matrices where necessary within the integrals (14). The integration process is carried on as in simplex hexahedron by using definitions (40) and (41).

2.2.3. Cubic Hexahedron Element

The typical cubic hexahedron element is illustrated in Figure 2.10. Like the previous hexahedron elements, this one is also an isoparametric element where the field variable is given by

$$\emptyset(x,y,z) = [N_1 \; N_2 \ldots N_{32}]\vec{\emptyset}^e$$
$$\vec{\emptyset}^e = [\emptyset_1 \; \emptyset_2 \ldots \emptyset_{32}]^T$$

and the coordinates by

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{bmatrix} N_1 N_2 \ldots N_{32} & 0\;0 \ldots 0 & 0\;0 \ldots 0 \\ 0\;0 \ldots 0 & N_1 N_2 \ldots N_{32} & 0\;0 \ldots 0 \\ 0\;0 \ldots 0 & 0\;0 \ldots 0 & N_1 N_2 \ldots N_{32} \end{bmatrix} \begin{Bmatrix} x_1 \\ \vdots \\ x_{32} \\ y_1 \\ \vdots \\ y_{32} \\ z_1 \\ \vdots \\ z_{32} \end{Bmatrix}$$

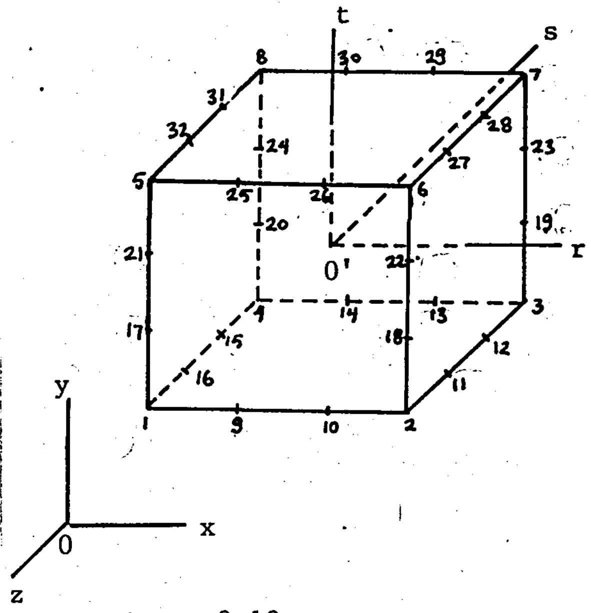

Figure 2.10

The Jacobian, also, is defined similarly for the element volume,

$$[J] = \begin{bmatrix} \dfrac{\partial x}{\partial r} & \dfrac{\partial y}{\partial r} & \dfrac{\partial z}{\partial r} \\ \dfrac{\partial x}{\partial s} & \dfrac{\partial y}{\partial s} & \dfrac{\partial z}{\partial s} \\ \dfrac{\partial x}{\partial t} & \dfrac{\partial y}{\partial t} & \dfrac{\partial z}{\partial t} \end{bmatrix} = \begin{bmatrix} \sum\limits_{i=1}^{32} \dfrac{\partial N_i}{\partial r} x_i & \sum\limits_{i=1}^{32} \dfrac{\partial N_i}{\partial r} y_i & \sum\limits_{i=1}^{32} \dfrac{\partial N_i}{\partial r} z_i \\ \sum\limits_{i=1}^{32} \dfrac{\partial N_i}{\partial s} x_i & \sum\limits_{i=1}^{32} \dfrac{\partial N_i}{\partial s} y_i & \sum\limits_{i=1}^{32} \dfrac{\partial N_i}{\partial s} z_i \\ \sum\limits_{i=1}^{32} \dfrac{\partial N_i}{\partial t} x_i & \sum\limits_{i=1}^{32} \dfrac{\partial N_i}{\partial t} y_i & \sum\limits_{i=1}^{32} \dfrac{\partial N_i}{\partial t} z_i \end{bmatrix}$$

where

$$N_i = \frac{1}{64}(1+rr_i)(1+ss_i)(1+tt_i)(9r^2+9s^2+9t^2-19), \text{ for } i=1 \text{ to } 8$$

$$N_i = 9(1-r^2)(1+9rr_i)(1+ss_i)(1+tt_i) \quad , \text{ for } i=9,10,13,14,25,26,29,30$$

$$N_i = 9(1-s^2)(1+9ss_i)(1+rr_i)(1+tt_i) \quad , \text{ for } i=11,12,15,16,27,28,31,32$$

$$N_i = 9(1-t^2)(1+9tt_i)(1+rr_i)(1+ss_i) \quad , \text{ for } i=17,18,19,20,21,22,23,24$$

and the local coordinates of the nodes are given by the vectors

$$\vec{r} = \left\{ \begin{matrix} -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1/3 \\ 1/3 \\ 1 \\ 1 \\ 1/3 \\ -1/3 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1/3 \\ 1/3 \\ 1 \\ 1 \\ 1/3 \\ -1/3 \\ -1 \\ -1 \end{matrix} \right\} \qquad \vec{s} = \left\{ \begin{matrix} -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1/3 \\ 1/3 \\ 1 \\ 1 \\ 1/3 \\ -1/3 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1/3 \\ 1/3 \\ 1 \\ 1 \\ 1/3 \\ -1/3 \end{matrix} \right\} \qquad \vec{t} = \left\{ \begin{matrix} -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1/3 \\ -1/3 \\ -1/3 \\ -1/3 \\ 1/3 \\ 1/3 \\ 1/3 \\ 1/3 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} \right\}$$

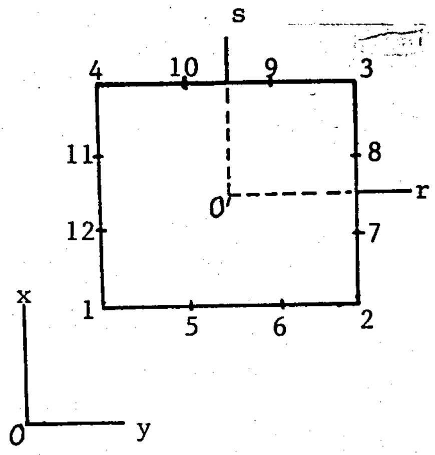

Figure 2.11

For the surfaces of the hexahedron the following are to be defined,

$$\emptyset(x,y) = [N_1\ N_2 \ldots N_{12}]\ \vec{\emptyset}^e = [N]\ \vec{\emptyset}^e$$

$$\vec{\emptyset}^e = [\emptyset_1\ \emptyset_2 \ldots \emptyset_{12}]^T$$

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} N_1 & N_2 & \ldots & N_{12} & 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & N_1 & N_2 & \ldots & N_{12} \end{bmatrix} \begin{Bmatrix} x_1 \\ \vdots \\ x_{12} \\ y_1 \\ \vdots \\ y_{12} \end{Bmatrix}$$

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{12} \frac{\partial N_i}{\partial r} x_i & \sum_{i=1}^{12} \frac{\partial N_i}{\partial r} y_i \\ \sum_{i=1}^{12} \frac{\partial N_i}{\partial s} x_i & \sum_{i=1}^{12} \frac{\partial N}{\partial s} y_i \end{bmatrix}$$

where

$$N_i = \frac{1}{32}(1+rr_i)(1+ss_i)(9r+9s-10) \quad \text{for} \quad i = 1,2,3,4$$

$$N_i = \frac{9}{32}(1+rr_i)(1-s^2)(1+9ss_i) \quad \text{for} \quad i = 7,8,11,12$$

$$N_i = \frac{9}{32}(1+ss_i)(1-r^2)(1+9rr_i) \quad \text{for} \quad i = 5,6,9,10$$

and

$$\vec{r} = \begin{Bmatrix} -1 \\ 1 \\ 1 \\ -1 \\ -1/3 \\ 1/3 \\ 1 \\ 1 \\ 1/3 \\ -1/3 \\ -1 \\ -1 \end{Bmatrix} , \quad \vec{s} = \begin{Bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1/3 \\ 1/3 \\ 1 \\ 1 \\ 1/3 \\ -1/3 \end{Bmatrix}$$

From the definitions given above for the interpolation functions, the necessary operations must be carried on. The integration is to be followed with the application of three point Gauss Quadrature defined by equations (40) and (41).

PART THREE

APPLICATIONS OF THE PROGRAMS DEVELOPED

Before proceeding to the problems concerning the transient conduction heat transfer, the validity of the tetrahedral elements are given by Rao [3]. These are tested by applying them to certain steady state problems for which the solutions are known. By doing so, an experience and insight to the element formation and element orientation in space was gained. It is highly probable to make mistakes in the data preparation for three dimensional problems because of the difficulty in visualizing the element configuration. This fact necessiated an automatic mesh generation program in order to solve conduction heat transfer problems in a cube with various boundary conditions. The generation program is devised such that it generates the data so as to include the information for different boundary conditions, while requiring the minimum information as input, and the minimum effort in updating. In the generation program, the basic configuration that is shown in Figure 3.1 is used as the repeating element which consists of six tetrahedrons. When the repetition in

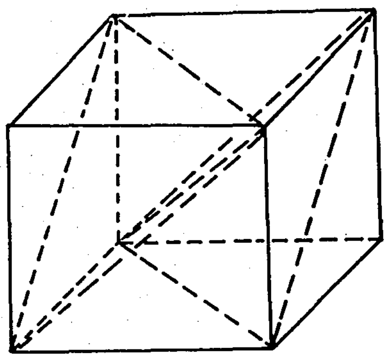

Figure 3.1

space is too uniform, as in Figure 2(a), it is seen that same symmetry problems arise which can be avoided by randomizing the orientation of the basic cube, shown in Figure 3.1, by rotations, as shown in 3.2(b). The latter is preferred in order to obtain the following results that are valid for both simplex and parabolic elements.

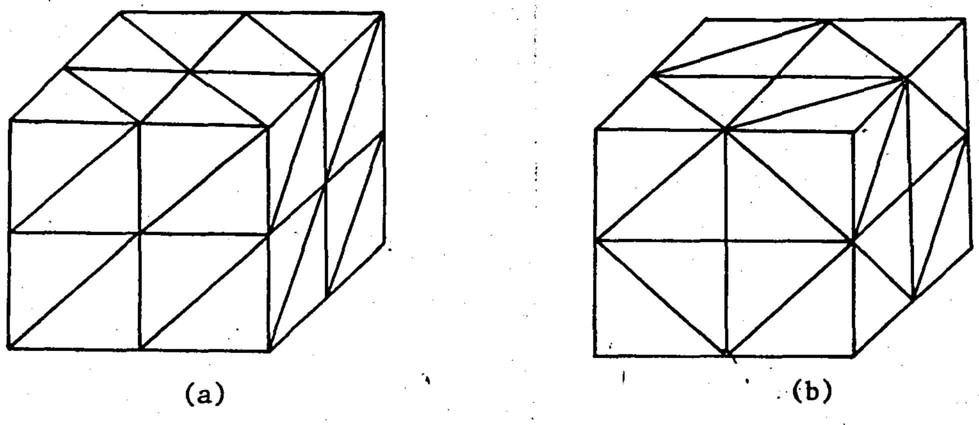

Figure 3.2

a. The cube with two opposing faces subject to heat flux, and other faces insulated, as shown in Figure 3.3, was tried to be solved yielding a singular matrix, therefore it has no steady state solution.

b. The cube with two opposing faces, one being specified at 100° and the other at 0° , and others insulated resulted with a perfect linear temperature distribution, ranging from 100° down to 0° as the solution.

c. The fin problem in Figure 3.4 with one face specified at 100° , and h=0. behaved like an insulated cube giving a uniform temperature at 100°.

d. The fin problem as in case (c) with h very large was expected to give the same result with a cube where the convection faces in Figure 15 are replaced with surfaces specified at T_s . It is seen that when all of the nodes on the base of the fin, including the edge nodes, are specified to be at $T_s = 100°$, then, a singularity condition arises which can be taken care of by no longer specifying the edge nodes. In other words, at the edges where convection and temperature specified boundaries meet, the advantage must be given to the convection boundaries.

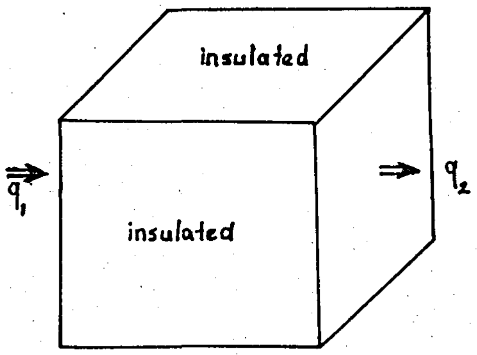

Figure 3.3

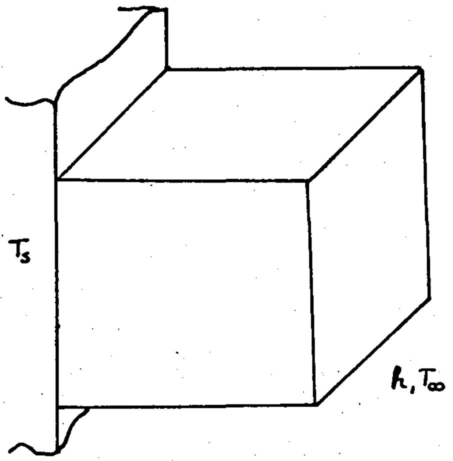

Figure 3.4

3.1. Statement of the Problem

The transient problem that has been worked on was chosen to be one which has an exact series solution and one which has an easy-to-generate geometry at the same time. The simplest case is decided to be a cube as demonstrated in Figure 3.5 where the zero faces are insulated and the other three faces are specified at $T = 100°$.

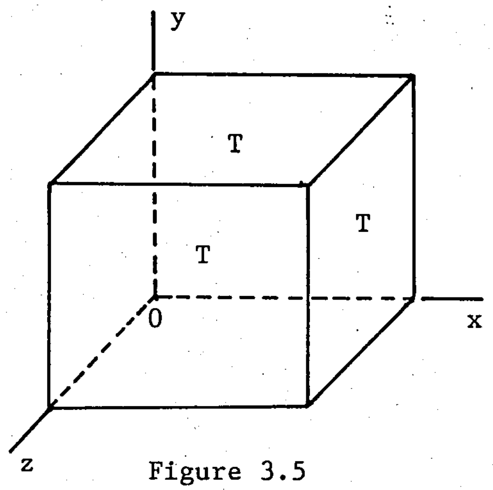

Figure 3.5

The one-dimensional solution to the problem [1] is obtained thru the Fourier method

$$\theta(x) = \frac{T(x) - T_s}{T_o - T_s} = \frac{2}{\mu_n}(-1)^{n+1}\cos\left(\mu_n\frac{x}{L}\right)\exp(-\mu_n^2 F_o) \quad (43)$$

where

$$\mu_n = (2n-1)\frac{\pi}{2} \qquad \text{for} \quad n = 1,2,3,\ldots$$

$F_o = \alpha t/L^2$, being the Fourier modulus.

In the problem solved, the cube is taken to be a unit cube by setting L equal to one.   is also equal to one, thus making the Fourier Modules behave as the time variable.

The three-dimensional cube solution is the superposition of the equation (43) in three directions giving

$$\theta(x,y,z) = \frac{T(x)-T_s}{T_0 - T_s} \cdot \frac{T(y)-T_s}{T_0 - T_s} \cdot \frac{T(z)-T_s}{T_0 - T_s} \qquad (44)$$

with initial temperature $T = 0^\circ$.

Another version of this problem can be produced by letting the specified temperature surfaces to be defined as correction surfaces for which a series approximation exists for small time values, not applicable to the time range given in the tables 3.9 and 3.10.

It should be noted that the so called "exact solution" is only another approximation to the problem, providing an approximate solution obtained by using a method other than FEM.

3.2. Discussion of the Results

The cube problem described in section 3.1 is first solved with a program using equations (43) and (44) in order to get an exact result, and to treat it as a reference. Then, the next step is to apply the finite element method to the problem. The finite element programs that are devised using different types of elements are each applied to the problem above and the results obtained are compared with the exact solution as follows.

The exact solution and FE solutions are first given in tabular form which is followed by an illustration in graphical form. A comparison of the results obtained is made based on refinement or element type. For the latter one, the comparison is made using the same number of nodes, and for either case, the node of concern is the corner at the origin in Figure 3.5.

The graphs show little oscillation at the beginning which gradually approaches the exact solution as time increases. The steady-state solution, $T=100^\circ$ in this case, is reached after $t=1.40$ seconds. The oscillation characteristics vanish as the mesh is refined and higher order elements are used.

The refinement of the mesh and the increase in the order of the interpolation polynomial used improves the results considerably. In the following tables and figures tha comparisons and plots are made by using coarse meshes which makes it easier to detect the deviations from the exact solution.

In figures 3.6, 3.7, 3.8 and 3.9, it can be observed, as a general trend, that the FE solution first drops below the exact solution but then crosses it and later converges tothe exact solution. This trend is tried to be avoided in two ways which proved to be unsuccessful.

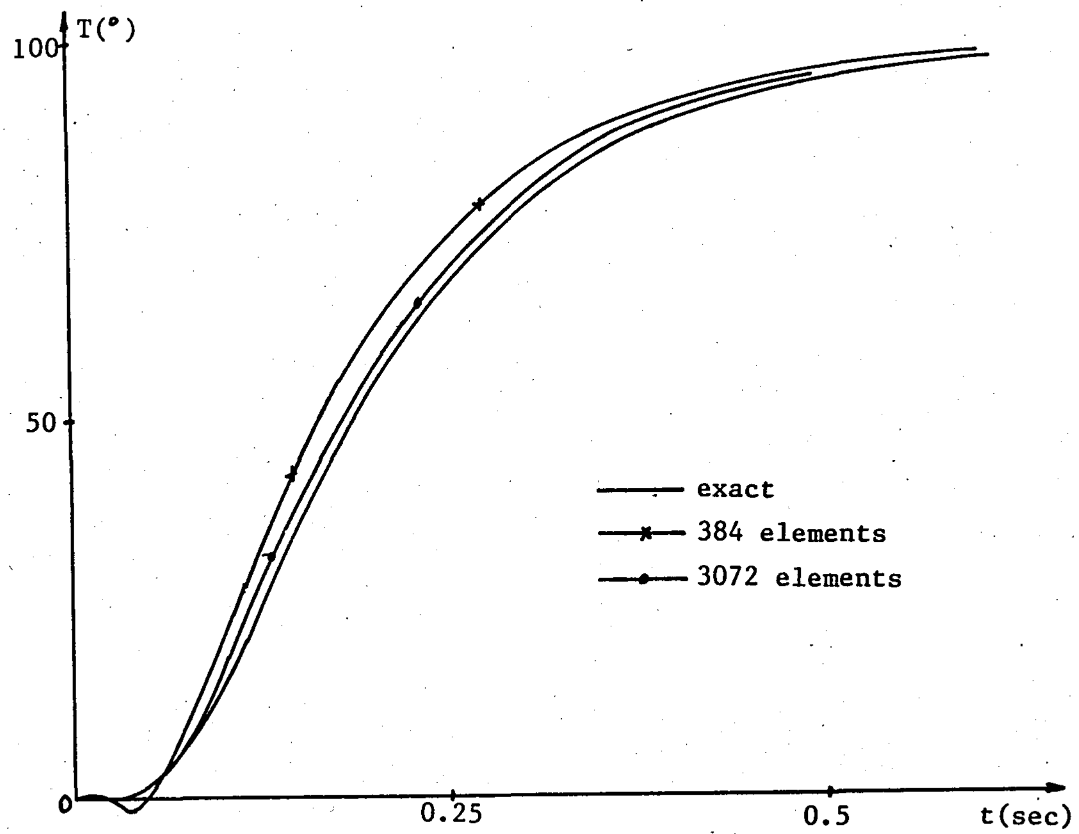

Figure 3.6 Plot of Table 3.1

TIME	NUMBER OF ELEMENTS			EXACT
	384	1296	3072	
.02	.16	0.00	0.00	0.00
.04	-.80	.20	.28	.24
.06	2.52	2.39	2.38	2.32
.08	10.27	8.31	7.80	7.27
.10	19.92	16.58	15.57	14.45
.20	61.62	57.48	55.96	53.93
.30	82.65	80.04	79.04	77.66
.40	92.19	90.71	90.13	89.32
.50	96.49	95.68	95.36	94.90
.60	98.42	97.99	97.82	97.57

Table 3.1. Comparison of Temperature Variation at the Corner Node Based on Refinement Using Simplex Tetrahedron (Specifed Surfaces)

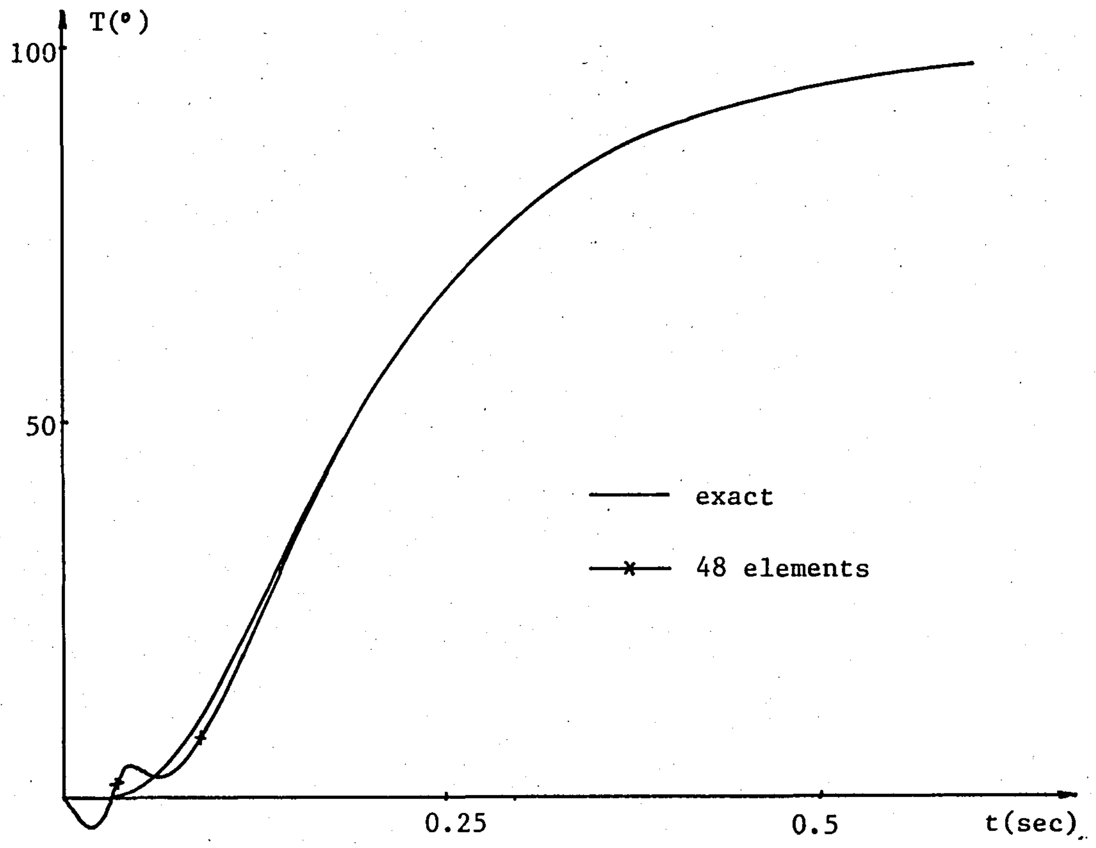

Figure 3.7 Plot of Table 3.2

TIME	NUMBER OF ELEMENTS			EXACT
	384	1296	3072	
.01	-6.01	0.00	0.00	0.00
.02	3.43	0.86	0.06	0.00
.04	4.60	1.39	0.58	0.24
.06	1.89	2.74	2.61	2.32
.08	4.57	6.95	7.24	7.27
.20	52.79	53.62	53.82	53.93
.30	77.35	77.57	77.64	77.66
.40	89.25	89.30	89.32	89.32
.50	94.90	94.90	94.91	94.90
.60	97.58	97.57	97.57	97.57

Table 3.2. Comparison of Temperature Variation at the Corner Node Based on Refinement Using Parabolic Tetrahedron (Specified Surfaces)

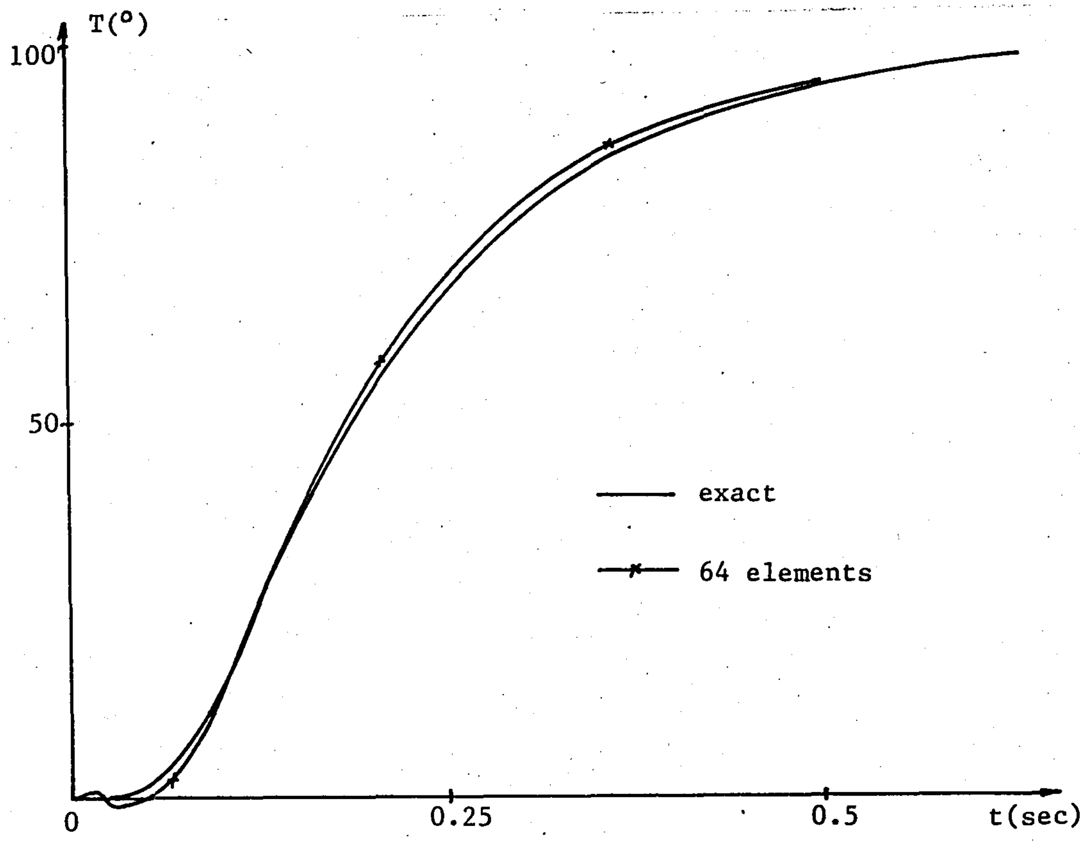

Figure 3.8 Plot of Table 3.3.

TIME	NUMBER OF ELEMENTS		EXACT
	64	216	
.02	0.74	0.00	0.00
.04	-1.42	0.01	0.24
.06	0.21	1.49	2.32
.08	6.25	6.69	7.27
.10	14.72	14.39	14.45
.20	56.14	54.90	53.93
.30	79.10	78.32	77.66
.40	90.12	89.11	89.32
.50	95.33	95.10	94.90
.60	97.80	97.67	97.57

Table 3.3. Comparison of Temperature Variation at the Corner Node Based on Refinement Using Simplex Hexahedron (Specified Surfaces)

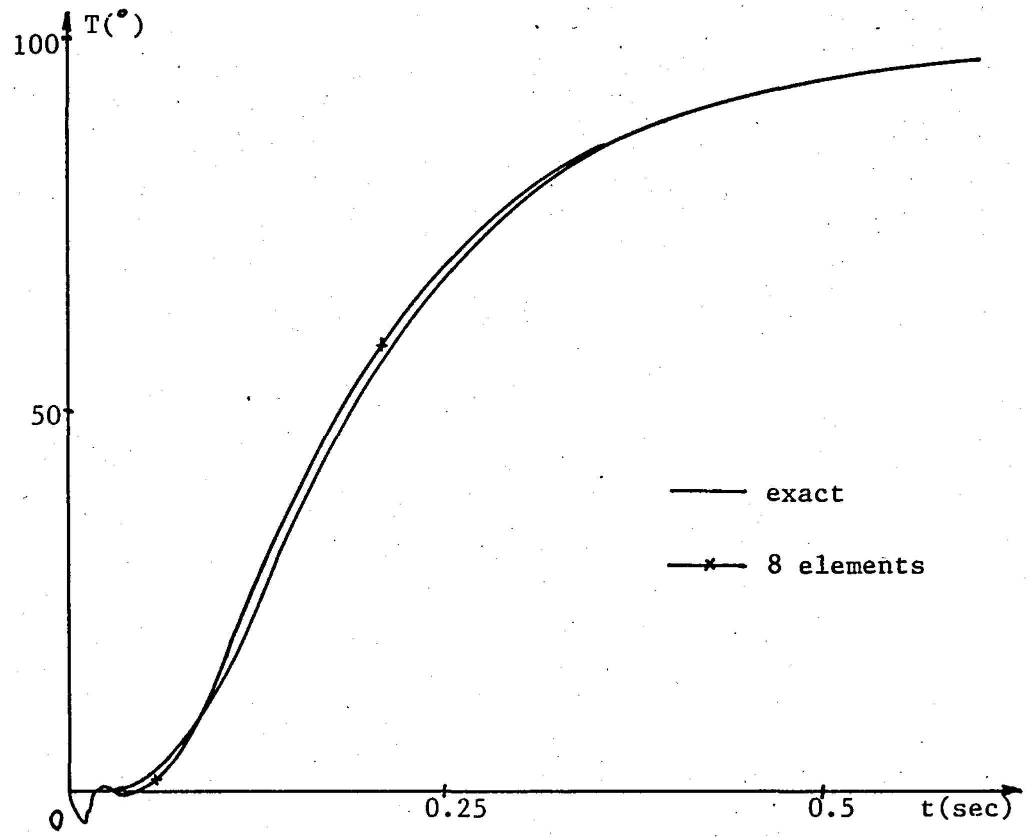

Figure 3.9 Plot of Table 3.4

TIME	NUMBER OF ELEMENTS		EXACT
	8	27	
.01	-4.95	-1.27	0.00
.02	0.27	0.31	0.00
.04	-0.03	0.81	0.24
.06	1.45	2.31	2.32
.08	7.01	7.20	7.27
.10	15.00	14.57	14.45
.20	55.43	54.41	53.93
.30	78.51	77.90	77.66
.40	89.74	89.42	89.32
.50	95.11	94.94	94.90

Table 3.4. Comparison of Temperature Variation at the Corner Node Based on Refinemet Using Parabolic Hexahedron (Specified Surfaces)

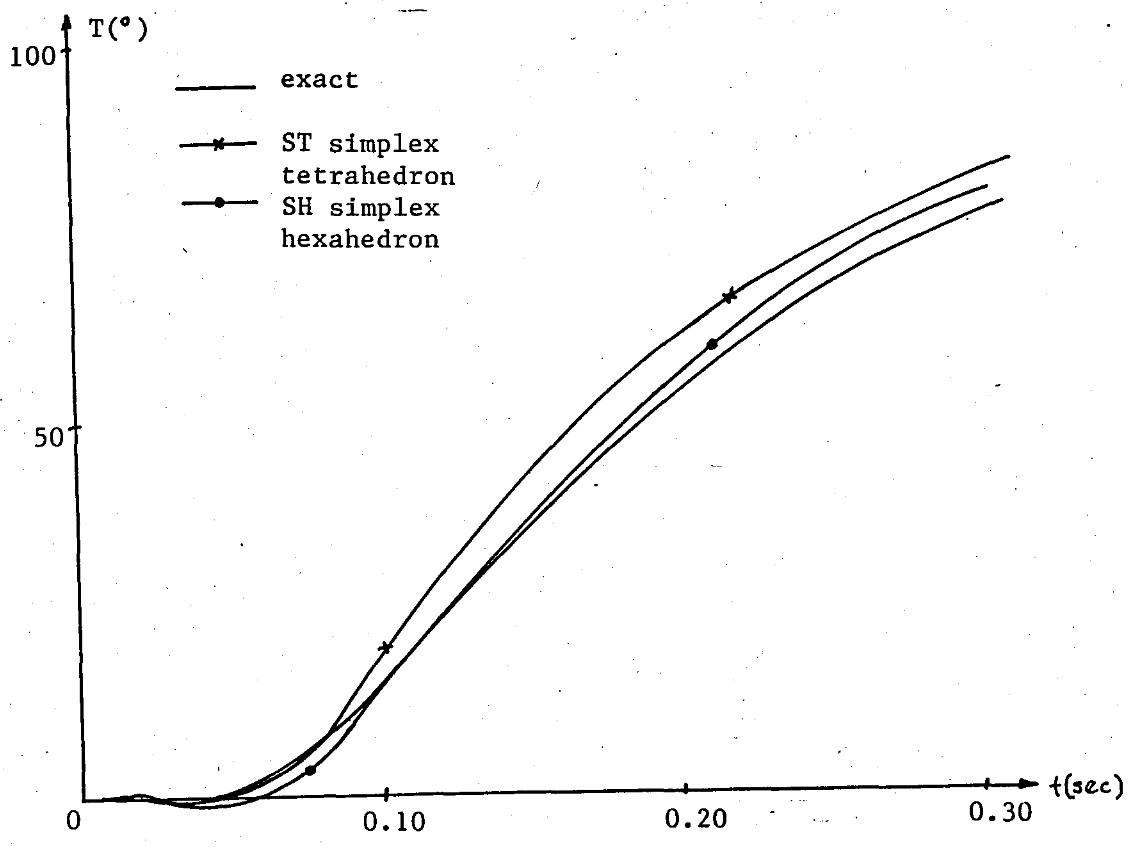

Figure 3.10. Plot of Table 3.5

TIME	TYPE OF ELEMENTS		EXACT
	ST	SH	
.01	.04	-.14	0.00
.02	.16	.74	0.00
.04	-.80	-1.74	.24
.06	2.52	.21	2.32
.08	10.27	6.25	7.27
.10	19.92	14.72	14.45
.20	61.62	56.14	53.93
.30	82.65	79.10	77.66
.40	92.19	90.12	89.32
.50	96.49	95.33	94.90

Table 3.5. Comparison of Temperature Variation at the Corner Node Based on Type of Element, Each Using 125 Nodes (Specified Surfaces)

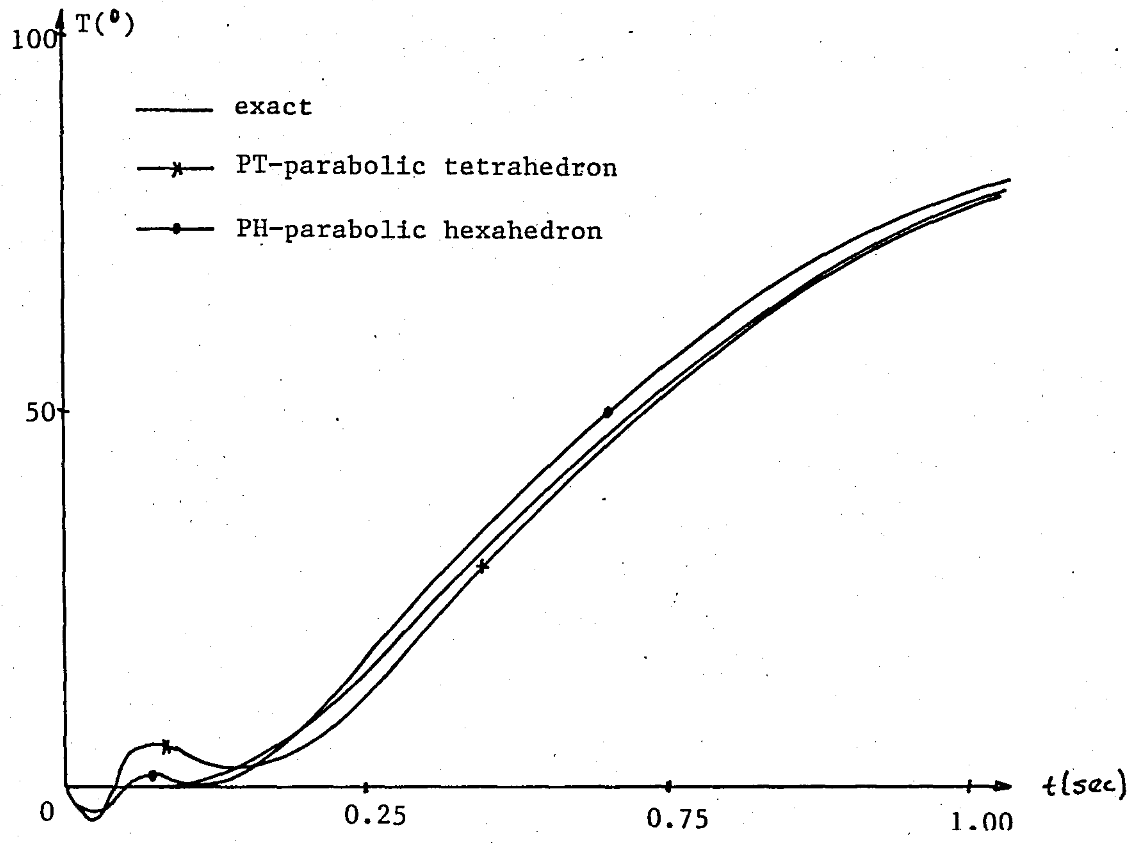

Figure 3.11 Plot of Table 3.6

TIME	TYPE OF ELEMENT		EXACT
	PT	PH	
.01	-6.01	-4.95	0.00
.02	3.43	.27	0.00
.04	4.60	-.03	.24
.06	1.89	1.45	2.32
.08	4.57	7.01	7.27
.10	11.28	15.00	14.45
.20	52.79	55.43	53.93
.30	77.35	78.51	77.66
.40	89.25	89.74	89.32
.50	94.90	95.11	94.90

Table 3.6 Comparison of Temperature Variation at the Corner Node Based on Type of Element (Specified Surfaces) PT: 125 nodes PH: 81 nodes

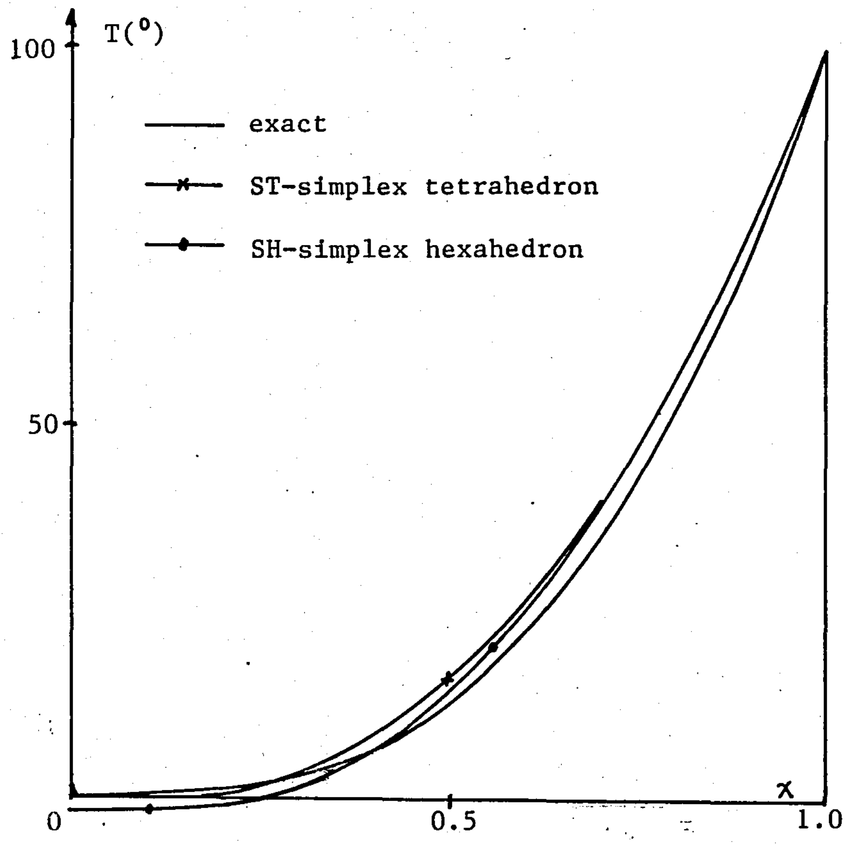

Figure 3.12. Plot of Table 3.7

X COORDINATE	TYPE OF ELEMENTS ST	SH	EXACT
0.0	.27	-1.25	.94
.25	1.29	.87	2.20
.50	15.97	13.39	11.94
.75	46.50	46.98	43.41
1.00	100.00	100.00	100.00

Table 3.7. Temperature Variation Along X-Axis at t=.05, Comparison Based On Type of Element Each Using 125 Nodes (Specified Surfaces)

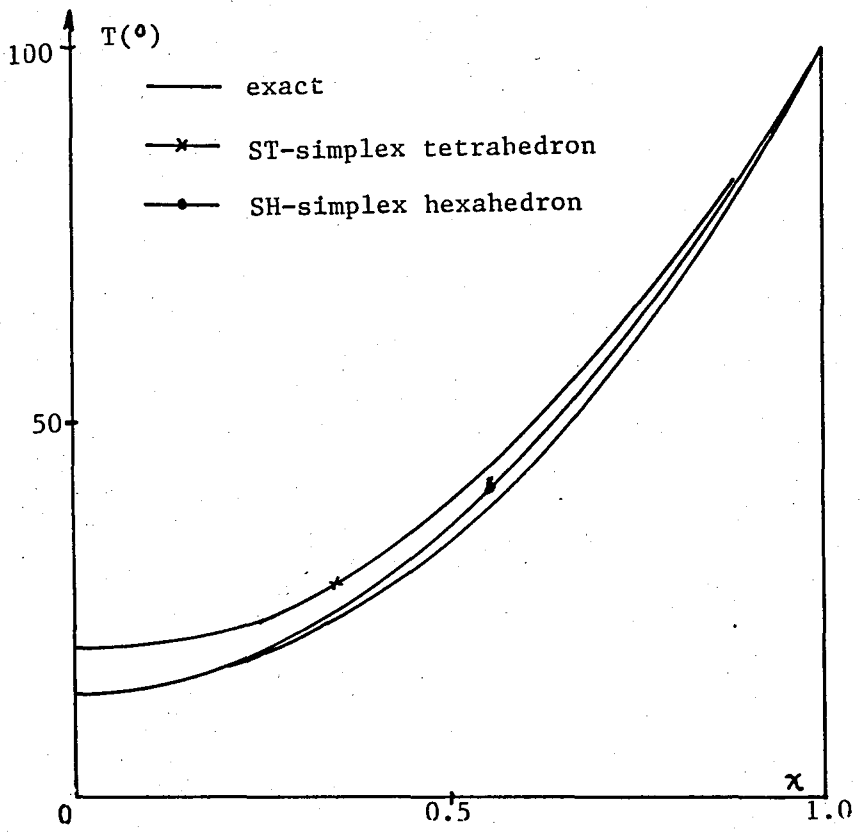

Figure 3.13. Plot of Table 3.8

X COODINATE	TYPE OF ELEMENTS ST	SH	EXACT
0.0	19.92	14.72	14.45
.25	22.14	19.73	18.78
.50	40.00	35.81	33.70
.75	64.73	63.77	61.81
1.00	100.00	100.00	100.00

Table 3.8. Temperature Variation Along X-Axis at t=.10, Comparison Based On Type of Element Each Using 125 Nodes (Specified Surfaces)

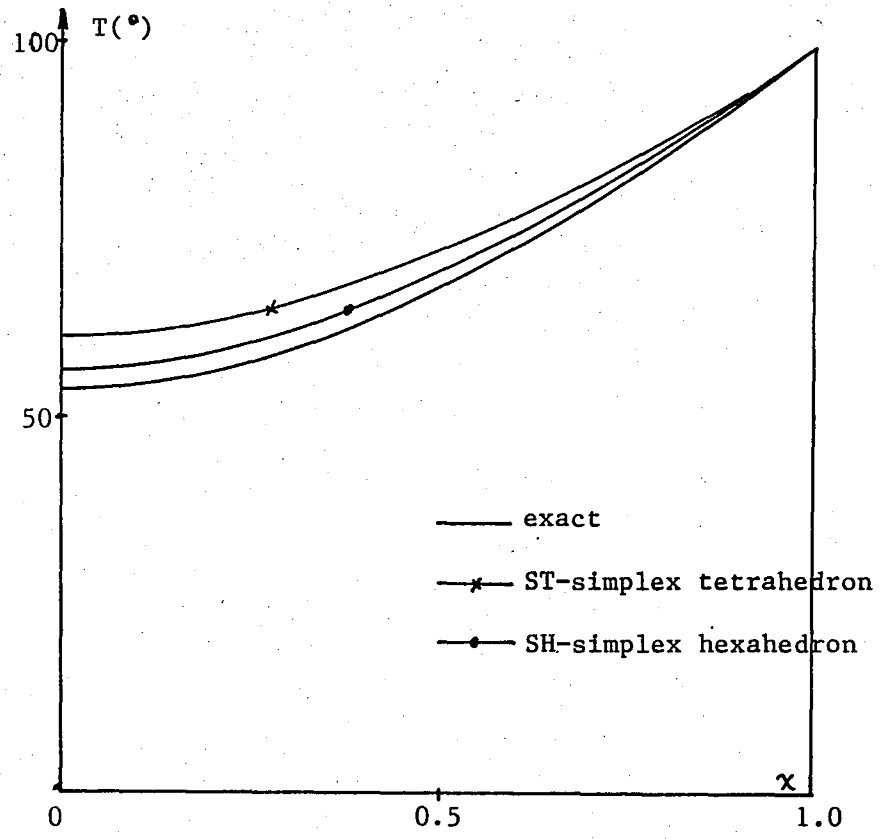

Figure 3.14 Plot of Table 3.9

X COORDINATE	TYPE OF ELEMENTS ST	SH	EXACT
0.0	61.62	56.14	53.93
.25	62.92	59.40	57.28
.50	72.44	68.79	67.00
.75	84.33	83.03	81.98
1.00	100.00	100.00	100.00

Figure 3.9. Temperature Variation Along X-Axis at t=.20, Comparison Based On Type of Element Each Using 125 Nodes (Specified Surfaces)

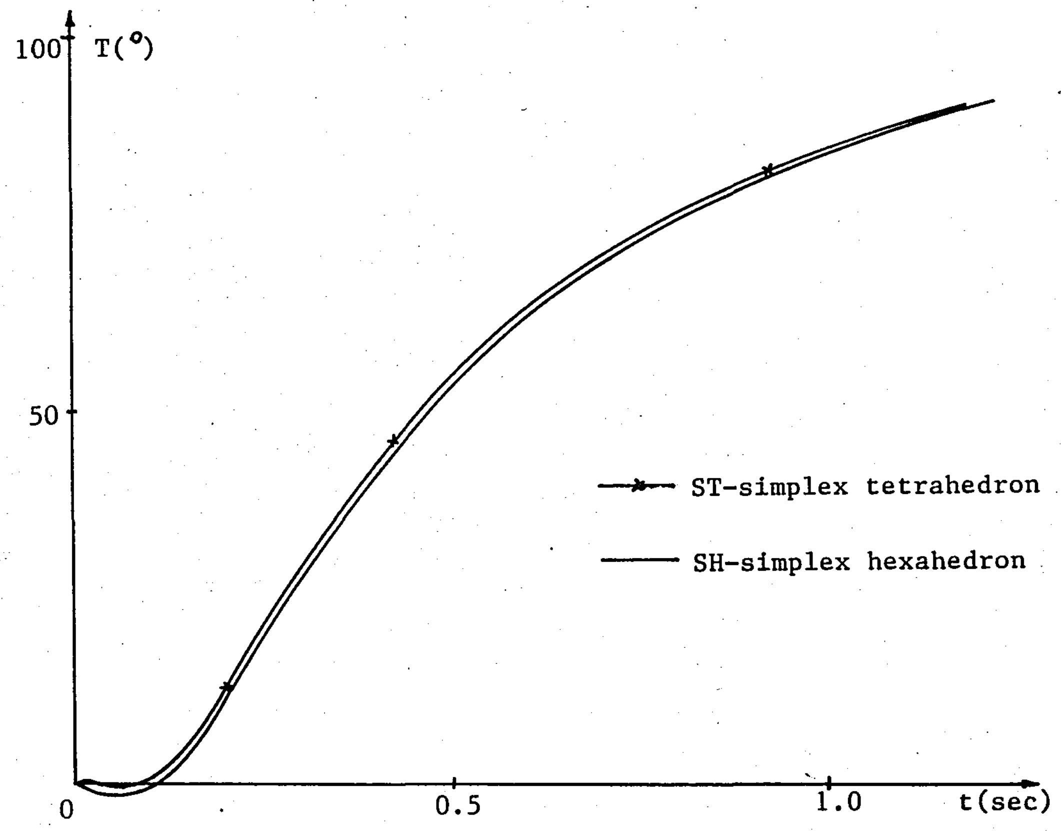

Figure 3.15 Plot of Table 3.10

TIME	TYPE OF ELEMENTS	
	ST	SH
0.1	1.55	0.88
0.2	14.10	12.88
0.3	29.44	28.27
0.4	43.08	42.09
0.5	54.35	53.54
0.6	63.44	62.79
0.7	70.74	70.22
0.8	76.59	76.17
0.9	81.27	80.93
1.0	85.01	84.74

Table 3.10 Comparison of Temperature Variation at the Corner Node Based on Type of Element, Each Using 125 Nodes (Correction Boundaries)

TIME	TETRAHEDRON		HEXAHEDRON	
	5x5x5	7x7x7	5x5x5	7x7x7
0.1	1.55	1.84	0.88	1.54
0.2	14.10	14.11	12.83	13.55
0.3	29.44	29.26	28.27	28.71
0.4	43.08	42.86	42.09	42.39
0.5	54.35	54.12	53.54	53.74
0.6	63.44	63.24	62.79	62.93
0.7	70.74	70.56	70.22	70.31
0.8	76.59	76.43	76.17	76.23
0.9	81.27	81.13	80.93	80.97
1.0	85.01	84.90	84.74	85.77

Table 3.11 Comparison of Temperature Variation at the Corner Node Based on Refinement with 125 and 343 Nodes for Each Element (Convection Boundaries)

a. Using a variable time step: The observation that a small time increment meant a small deviation from the exact solution led to the use of a small time increment at the beginning of time domain which later would be followed by larger increments.

b. Using a variable specified temperature: The solution obtained for the cube problem has a linear relationship with the specified temperature which means that a smaller T_s gives smaller deviations. Using this property, it was thought that using a gradual increase in specified temperature instead of a step change could give better results. This also diverged.

Tables 3.10 3.11 and plot 3.15 show that the simplex tetrahedron and the simplex hexahedron elements give a good correlation when compared to each other for the problem where surfaces are subject to convection.

The programs occupy a larger memory space and require longer execution times as the mesh is refined. Depending on the time increment chosen and the final time at which a solution is sought, the cost of the execution is augmented considerably.

Two-dimensional abstracts of the problems could first be analysed before applying the three-dimensional programs. Hence, the execution time and the size of the matrices would be decreased, while obtaining acceptable results.

CONCLUSIONS

Fundamental concepts pertaining to the finite element analysis of problems of three dimensional conduction heat transfer have been treated in detail. Different types of elements have been discussed and, for each element, the element matrices and load vectors have been derived. Programs have been developed which used these elements and they have been applied to a selected set of problems involving various boundary conditions on a hexahedron.

On the basis of the theoretical and numerical results presented in the preceding sections, the following conclusions are reached concerning the applicability of the finite-element approximation technique to problems of three-dimensional conduction heat transfer.

The finite element method provides an efficient and reliable method of obtaining highly accurate solutions to problems involving irregularly shaped bodies for which a solution using conventional methods could be impossible. The convergence to the exact solution has been shown to improve as the mesh is refined. For a given level of accuracy this may allow solution to problems on a coarser mesh than is predicted, which in turn, would save computer time and money.

When using the tetrahedron element, it was seen that the results obtained were highly dependent on the orientation of the elements in space which was not the case with the hexahedron element. This inefficiency lost its effect as the mesh was refined or higher order elements were used. Given the same number of nodes, the hexahedron element has been shown to be more accurate and more flexible when compared to the tetrahedron elements of the same order of approximation.

It is seen that the higher order approximations require more nodes given the same number of elements. The main disadvantage of the application of the finite element approximation technique to the three-dimensional conduction heat transfer problems is the necessity of storing the system matrices and vectors, and followingly, of solving the corresponding set of equations. The system limitations as far as speed and memory capacities are concerned have to be overcome in order to refine the mesh. The matrices are reduced in size by using some properties inherent to the finite element method such as the banded form of the matrices and the symmetry they have. For cases when even the banded solution scheme lacks the capability to create enough space in the core memory, a frontal subroutine has been adapted to the programs which uses the disc files in order to keep the major part of the information.

The programs should be tested throughly to see the behaviour along the lines where different boundaries meet, thus, to see the singularity conditions. Zienkiewicz stated that the curvature of the surfaces also effect convergence which gives another objective for the tests.

The programs are developed in modular form each consisting of subroutines that can easily be replaced which gives the ability to use different elements in the same program with minimum effort for adaptation.

The programs can be improved to enable the solution of time dependent heat generation. Less readily can they be modified to solve variable boundary conditions. The radiation boundary condition can be included in the programs when an iteration process is adapted [3]. The oscillation characteristics of the results may be taken care of when a three point integration scheme in the time domain is used rather than the two point finite difference approximation that the programs already make use of.

In summary then, the finite element approximation technique as applied to three-dimensional transient conduction heat transfer problems can be considered to be both reliable and efficient. There is no reason why the programs should not be extended to more complex forms and applied to more irregularly shaped bodies than a simple hexahedron.

APPENDIX A

ABOUT THE PROGRAMS

The programs are designed to apply the three-dimensional finite element method for the solution of linear, transient, three-dimensional, anisotropic heat conduction problems in volumes with random geometry with time independent heat generation and time independent boundary conditions. Although the programs as they are presented can only solve time independent heat generation problems, with not much of an effort, they can be put into a form which would enable the solution of heat generation problems with step changes in time.

The programs are structured in banded form and, thus, use much less space than that would otherwise be used. For cases that require even more memory space, the frontal method is applied at the cost of increased execution time.

$$\begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 \\ a & a_{22} & a_{23} & 0 & 0 & 0 \\ 0 & a & a_{33} & a_{34} & 0 & 0 \\ 0 & 0 & a & a_{44} & a_{45} & 0 \\ 0 & 0 & 0 & a & a_{55} & a_{56} \\ 0 & 0 & 0 & 0 & a & a_{66} \end{bmatrix} 0 = \begin{bmatrix} a_{11} & a_{12} \\ a_{22} & a_{23} \\ a_{33} & a_{34} \\ a_{44} & a_{45} \\ a_{55} & a_{56} \\ a_{66} & 0 \end{bmatrix}$$

Mapping of $[a_{ik}] \Rightarrow$ $(j = k-i+1$, for $k > i) \Rightarrow (a_{ij})$

Figure A-1

The square matrix in Figure A-1 is shown being mapped onto a banded matrix.

The programs start with the data input and initiation of the matrices. The data is given in the following order,

a. first card gives number of nodes and number of elements

b. nodal data is given, coordinates and specified temperatures

c. time step and total number of steps is given

d. the conductivities, density and the specific heat are given

e. the element data follows.

After the element matrices ECM and EHCM are calculated, they are placed in the global matrices CM and HCM by using the subroutine MASS in which the bandwidth of the banded global matrices is calculated. The subroutine BC calculates the element convection matrix and the surface dependent vectors where they are added onto the global matrix CM and load vector Q, respectively.

Next step is to form the effective conductivity matrix which actually is on the left-hand-side of equation (19). The modification and reduction phases follow in order to include fixed temperature data and start the Gaussian elimination.

The effective load vector is, only then, calculated which is the right-hand-side of equation (19). This is the point where step changes in heat generation vector can be made.

Then comes the solution and back substitution phases of the Gaussian elimination method in a revised form for banded matrices.

Last step is the printing of the temperatures at each node. The node number and the temperature at that point are printed.

The data have to be prepared carefully, where a generation program can be devised if the geometry of the body permits, since the risk of making mistakes is highly probable due to the difficulty in visualizing the element configuration in space.

APPENDIX B

ABOUT THE FRONTAL SUBROUTINE

The FRONTAL method can be considered as a particular technique for first assembling finite element conductivities and nodal field values into a global conductivity matrix and load vector, then solving for the unknown displacements by means of a Gaussian Elimination and back substitution process. Although it is designed to minimize the core storage requirement, it results with an increase in the execution time.

The subroutine is modular, in that it is a self-contained frontal solver which can be employed in any finite element program. It is assumed that the element conductivity matrices and the load vector have been generated elsewhere and are available from disc file. It is further assumed that each nodal point has the same degree of freedom; however, this number can be optional. Also, the matrices are assumed to be symmetric. Although the quadratic tetrahedron element is exclusively used in the FE program, the frontal solver presented will be compatible with any type of element, subject to above conditions.

The main idea of the Frontal solution is to assemble the equations and to eliminate the variables at the same time. As soon as the coefficients of an equation are completely assembled, from the contributions of all relevant elements, the corresponding variable can be eliminated. Therefore, the complete global matrices are never formed as such, since after elimination the reduced equation is immediately transferred to back-up disc-storage.

The core contains, at any given instant, the upper triangular part of a square matrix containing the equations which are being formed at that particular time. These equations, their corresponding nodes and degrees of freedom are termed the "front". The number of unknowns in the front is the "frontwidth"; this length generally changes continually during the assembly-reduction process that takes place in the subroutine. The maximum size of problem which can be solved is governed by the "maximum frontwidth". The equations, nodes and degrees of freedom belonging to the front are termed "active"; those which are yet to be considered "inactive"; those which have passed through the front and have been eliminated are said to be "deactivated".

During the assembly/elimination process, the elements are considered each in turn according to a prescribed order which is defined by the generation program prior to the application of the FE program containing the subroutine FRONT. Whenever a new element is called in, its element matrix and element load vector are read from the disc file and summed either into existing equations, if the nodes are already active, or into new equations which have to be included in the front if the nodes are being activated for the first time. If some nodes are appearing for the last time, the corresponding equations can be eliminated and stored away on a disc file and are thus deactivated. In so doing, they free space in the front which can be employed during assembly of the next element.

Following this part is the back-substitution phase which can be taken as a frontal process in reverse.

The fixed temperatures at certain nodes are taken care of within the subroutine FRONT.

The explanation in detail of the subroutine FRONT and subroutine listing can be found in [ ] pp 192-204.

APPENDIX C

LISTING OF FE PROGRAM

USING A PARABOLIC TETRAHEDRON

WITH FRONTAL SUBROUTINE

```
      PROGRAM TRPAFR(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C
C
C =====================================================================
C PARABOLIC III-DIMENSIONAL TETRAHEDRAL FINITE ELEMENT PROGRAM FOR
C THE SOLUTION OF LINEAR,TRANSIENT,III-DIMENSIONAL,ISOTROPIC HEAT
C CONDUCTION PROBLEMS IN CUBIC VOLUMES WITH TIME-INDEPENDENT HEAT
C GENERATION AND TIME-INDEPENDENT BOUNDARY CONDITIONS
C =====================================================================
C
C
C
C     TERMINOLOGY
C
C
C
C      AIJK,AIKL,AILJ         AREAS ON FACES IJK, IKL, & LJ RESPECTIVELY
C      ATIJK,ATIKL,AILJ       AMBIENT TEMPERATURES ON FACES IJK, IKL &
C                                        ILJ RESPECTIVELY
C      CX,CY,CZ               CONDUCTION PARAMETERS IN THE DIRECTIONS
C                               X, Y & Z RESPECTIVELY
C      DT                     TIME INCREMENT
C      ECODE                  ELEMENT CODE
C                               0 IF ELEMENT IS AN INTERNAL ELEMENT
C                               1 IF ELEMENT IS A SIDE ELEMENT
C                               2 IF ELEMENT IS AN EDGE ELEMENT
C                               3 IF ELEMENT IS A CORNER ELEMENT
C      HIJK,HIKL,HILJ         CONVECTIVE HEAT TRANSFER COEFFICIENTS
C                               ON EDGES IJK, IKL & ILJ RESPECTIVELY
C      ICOUNT                 COUNTING INDEX FOR PRINTING THE
C                               COMPUTED TEMPERATURE DISTRIBUTION
C      INT                    PRINTING INDEX FOR THE COMPUTED
C                               TEMPERATURE DISTRIBUTION
C      MBAND                  BANDWIDTH
C      MBMAX                  MAXIMUM ALLOWABLE BANDWIDTH
C      NDT                    NUMBER OF TIME INCREMENTS
C      NE                     NUMBER OF ELEMENTS
C      NEMAX                  MAXIMUM NUMBER OF ELEMENTS
C      NNP                    NUMBER OF NODAL POINTS
C      NNPMAX                 MAXIMUM NUMBER OF NODAL POINTS
C      QIJK,QIKL,QILJ         SPECIFIED INCOMING HEAT FLUX COMPONENTS
C                               ALONG THE NORMALS OF FACES IJK, IKL &
C                               ILJ RESPECTIVELY
C      DENS                   DENSITY OF THE MATERIAL
C      SIJ,SIK,SIL,SKL,
C      SJK,SJL                LENGTHS OF SIDES IJ, IK, IL, KL, JK &
C                                JL ON RELATED FACES, RESPECTIVELY
C      SPH                    SPECIFIC HEAT OF THE MATERIAL
C      TIME                   TIME LEVEL
C      VOLUME                  ELEMENT VOLUME
C
C
C      NPCODE(I),I=1,2,...,NNP  BOUNDARY CONDITION CODE OF THE ITH
C                                  NODE
C                                0 IF TEMPERATURE IS NOT SPECIFIED AT
C                                  THE ITH NODE
C                                1 IF TEMPERATURE IS SPECIFIED AT THE
C                                  ITH NODE
C
C      QGEN(I),I=1,2,...,NE     RATE OF HEAT GENERATION PER UNIT
C                                 VOLUME WITHIN THE ITH ELEMENT AT
C                                 ALL TIMES
C      X1(I),I=1,2,...,NNP     X-COORDINATE OF THE ITH NODE
C      X2(I),I=1,2,...,NNP     Y-COORDINATE OF THE ITH NODE
C      X3(I),I=1,2,...,NNP     Z-COORDINATE OF THE ITH NODE
C
```

```
C       Q                           THERMAL LOAD VECTOR,1*NNP
C       T                           TEMPERATURE VECTOR,1*NNP
C       FLV                         EFFECTIVE LOAD VECTOR,1*NNP
C
C
C
C       IX(I,J),I=1,2,...,NE        NODAL POINT LABEL OF THE JTH NODE
C               J=1,2,3,4             OF THE ITH ELEMENT
C       CM                          EFFECTIVE CONDUCTIVITY MATRIX,NNP*MBAND
C       ECM                         ELEMENT CONDUCTIVITY MATRIX,4*4
C       EHCM                        ELEMENT HEAT CAPACITY MATRIX,4*4
C       HCM                         HEAT CAPACITY MATRIX,NNP*MBAND
C
C
C
C      DECLERATIONS
C
C       * THE PROGRAM IS DESIGNED TO TREAT THE MATRICES, FORMED
C       DURING EXECUTION, IN BANDED FORM.
C       * THE DATA OBTAINED FROM PROGRAM GNRATE SHOULD BE UPDATED
C       IN ORDER FOR IT TO INCLUDE THE VALUES FOR DT, NDT, SPH & DENS.
C       * THE DATA SUPPLIES ONLY THE CORNER NODES OF THE TETRAHEDRON
C       ELEMENTS AND THE REMAINING SIX NODES ARE GENERATED WITHIN THE
C       PROGRAM ASSUMING STRTAIGHT SIDED TETRAHEDRONS.
C       * THE PROGRAM USES THE FRONTAL SOLUTION SCHEME IN ORDER TO
C       DECREASE THE COMPUTER MEMORY USED DURING EXECUTION  AT ANY TIME.
C
       INTEGER E

       DIMENSION X1(729),X2(729),X3(729),
      &Q(10),EHCM(10,10),QGEN(384),ECODE(384),
      &QIJK(384),HIJK(384),ATIJK(384),QILJ(384),HILJ(384),
      &ATILJ(384),QIKL(384),HIKL(384),ATIKL(384)
       COMMON NE,NNP,NMAT,MBAND,I,J,K,L,VOLUME
       COMMON/SUBBC/ECM(10,10)
       COMMON/SUBHGV/ELOAD(10)
       COMMON/FRON1/T(729),ZASDIS(729),NOFIX(729)
       COMMON/AAA/IFPRE(729,1)
       COMMON/LGDATA/PRESC(729,1),IX(384,10)
       COMMON/NPC/NPCODE(729)
C
C       1. READ THE DATA
C
       READ(5,*)NNP,NE
       READ(5,*)(K,X1(I),X2(I),X3(I),NPCODE(I),T(I),I=1,NNP)
       READ(5,*)DT,NDT
       READ(5,*)CX,CY,CZ,DENS,SPH
       READ(5,*)(K,(IX(I,J),J=1,4),ECODE(I),QGEN(I),QIJK(I),
      &HIJK(I),ATIJK(I),QILJ(I),HILJ(I),ATILJ(I),QIKL(I),HIKL(I),
      &ATIKL(I),I=1,NE)
       CALL GOFIX(NVFIX)
C
C ===============================================================
C       FORM ELEMENT CONDUCTIVITY AND HEAT CAPACITY MATRICES
C ===============================================================
C
       DO 1009 E=1,NE
C
C       2. INITIALIZE ELEMENT CONDUCTIVITY AND HEAT
C          CAPACITY MATRICES
C
       DO 1008 M=1,10
       ELOAD(M)=0.
       DO 1008 N=1,10
       ECM(M,N)=0.
       EHCM(M,N)=0.
```

```
1008  CONTINUE
C
C       3.FORM ELEMENT CONCUCTIVITY AND HEAT CAPACITY MATRICES
C
C          3.1.DEFINE THE VARIABLE NAMES I,J,K
C
      I=IX(E,1)
      J=IX(E,2)
      K=IX(E,3)
      L=IX(E,4)
      XI=X1(I)
      XJ=X1(J)
      XK=X1(K)
      XL=X1(L)
      YI=X2(I)
      YJ=X2(J)
      YK=X2(K)
      YL=X2(L)
      ZI=X3(I)
      ZJ=X3(J)
      ZK=X3(K)
      ZL=X3(L)
C
C       3.2. DEFINE THE OTHER NODES ASSUMING STRAIGHT SIDEC
C            TETRAHEDRONS
C
  500 X5=(XI+XJ)/2.
      Y5=(YI+YJ)/2.
      Z5=(ZI+ZJ)/2.
      DO 501 IA=1,NNP
      IF(ABS(X1(IA)-X5).LE.1.E-5.AND.ABS(X2(IA)-Y5).LE.1.E-5.AND.
     &ABS(X3(IA)-Z5).LE.1.E-5)GO TO 502
  501 CONTINUE
  502 IX(E,5)=IA

      X6=(XJ+XK)/2.
      Y6=(YJ+YK)/2.
      Z6=(ZJ+ZK)/2.
      DO 503 IA=1,NNP
      IF(ABS(X1(IA)-X6).LE.1.E-5.AND.ABS(X2(IA)-Y6).LE.1.E-5.AND.
     &ABS(X3(IA)-Z6).LE.1.E-5)GO TO 504
  503 CONTINUE
  504 IX(E,6)=IA

      X7=(XK+XI)/2.
      Y7=(YK+YI)/2.
      Z7=(ZK+ZI)/2.
      DO 505 IA=1,NNP
      IF(ABS(X1(IA)-X7).LE.1.E-5.AND.ABS(X2(IA)-Y7).LE.1.E-5.AND.
     &ABS(X3(IA)-Z7).LE.1.E-5)GO TO 506
  505 CONTINUE
  506 IX(E,7)=IA

      X8=(XL+XI)/2.
      Y8=(YL+YI)/2.
      Z8=(ZL+ZI)/2.
      DO 507 IA=1,NNP
      IF(ABS(X1(IA)-X8).LE.1.E-5.AND.ABS(X2(IA)-Y8).LE.1.E-5.AND.
     &ABS(X3(IA)-Z8).LE.1.E-5)GO TO 508
  507 CONTINUE
  508 IX(E,8)=IA

      X9=(XL+XJ)/2.
      Y9=(YL+YJ)/2.
      Z9=(ZL+ZJ)/2.
      DO 509 IA=1,NNP
      IF(ABS(X1(IA)-X9).LE.1.E-5.AND.ABS(X2(IA)-Y9).LE.1.E-5.AND.
     &ABS(X3(IA)-Z9).LE.1.E-5)GO TO 510
  509 CONTINUE
  510 IX(E,9)=IA

      X10=(XL+XK)/2.
      Y10=(YL+YK)/2.
      Z10=(ZL+ZK)/2.
      DO 511 IA=1,NNP
      IF(ABS(X1(IA)-X10).LE.1.E-5.AND.ABS(X2(IA)-Y10).LE.1.E-5.AND.
     &ABS(X3(IA)-Z10).LE.1.E-5)GO TO 512
  511 CONTINUE
  512 IX(E,10)=IA
C
C          3.3.CALCULATE GEOMETRICAL PARAMETERS OF THE ELEMENT
C
      CALL EG00(X1,X2,X3)
C
C          3.4.FORM THE ELEMENT CONDUCTIVITY MATRIX
C
      CALL ECM00(X1,X2,X3,CX,CY,CZ)
C
C          3.5.FORM THE ELEMENT HEAT CAPACITY MATRIX
C
      CALL EHCM00(DENS,SPH,EHCM)

      WRITE(7)EHCM
C
C ===================================================================
C       BOUNDARY CONDITIONS
C ===================================================================
C
      IF(ECODE(E).EQ.0) GO TO 42
      CALL BC00(X1,X2,X3,QIJK,HIJK,ATIJK,QILJ,HILJ,
     &ATILJ,QIKL,HIKL,ATIKL,Q,ECODE,E)
42    CONTINUE
      WRITE(8)ELOAD
C
C ===================================================================
C       FORM THE EFFECTIVE CONDUCTIVITY MATRIX
C ===================================================================
C
      DO 43 I=1,10
      DO 43 J=1,10
      ECM(I,J)=ECM(I,J)+2.*EHCM(I,J)/DT
43    CONTINUE
C
C ===================================================================
C       WRITE THE EFFECTIVE ELEMENT CONDUCTIVITY MATRIX ON DISC
C ===================================================================
C
      WRITE(1)ECM
1009  CONTINUE
C
C ===================================================================
C       CALCULATE AND PRINT THE TEMPERATURE DISTRIBUTION
C ===================================================================
C
9     TIME=0.
      ICOUNT=0.
      ICASE=0
      INT=0
C
C        1.PRINT THE INITIAL TEMPERATURE DISTRIBUTION
C
      PRINT 133,TIME,(I,T(I),I=1,NNP)
      DO 64 ITIME=1,NDT
C
C       2.FORM THE EFFECTIVE ELEMENT LOAD VECTOR
C
      REWIND 8
      REWIND 3
      REWIND 7
      DO 67 E=1,NE
      READ(7)EHCM
      READ(8)ELOAD
      DO 62 I=1,10
      DO 62 J=1,10
   62 ELOAD(I)=ELOAD(I)+2.*EHCM(I,J)*T(IX(E,J))/DT
C
C       3.FORM THE HEAT GENERATION VECTOR AND ADD IT INTO THE
C         THE EFFECTIVE LOAD VECTOR
C
      I=IX(E,1)
      J=IX(E,2)
      K=IX(E,3)
      L=IX(E,4)
      CALL EGOO(X1,X2,X3)
      CALL HGV(E,DENS,NPCODE,QGEN)
      WRITE(3)ELOAD
67    CONTINUE
C
C       4.SOLVE FOR NODAL TEMPERATURES
C
      CALL FRONT(NVFIX,ICASE)
C
C
      DO 290 N=1,NNP
290   T(N)=2.*ZASDIS(N)-T(N)
C
C       5.UPDATE THE TIME LEVEL AND PRINT NODAL
C         TEMPERATURES
C
      TIME=TIME+DT
      ICOUNT=ICOUNT+1
      IF(ICOUNT-INT) 64,65,65
65    PRINT 136,TIME,(I,T(I),I=1,NNP)
      ICOUNT=0.
64    CONTINUE
C
C ============================================================
C       FORMATS.......
C ============================================================
C
133   FORMAT(1H1//15X,60(1H=)/26X,'INPUT TABLE-TEMPERATURE DISTRI'
     &,'BUTION'/15X,60(1H=)/37X,'TIME=',F12.5/15X,60(1H-)/40X,'NODAL PO'
     &,'INT'/27X,'NUMBER',26X,'TEMPERATURES'/15X,60(1H-)/(28X,I5,24X,E15
     &.6))
136   FORMAT(1H1//15X,60(1H=)/26X,'OUTPUT TABLE-TEMPERATURE DISTRI'
     &,'BUTION'/15X,60(1H=)/37X,'TIME=',F12.5/15X,60(1H-)/40X,'NODAL PO'
     &,'INT'/27X,'NUMBER',26X,'TEMPERATURES'/15X,60(1H-)/(28X,I5,24X,E15
     &.6))
138   FORMAT(1H1,35(/),50X,'BANDWIDTH.......',I3)
C
      STOP
      END
C
      SUBROUTINE BCOO(X1,X2,X3,QIJK,HIJK,ATIJK,QILJ,HILJ,
     &ATILJ,QIKL,HIKL,ATIKL,Q,ECODE,E)
      INTEGER E
      DIMENSION X1(729),X2(729),X3(729),
      &Q(10),QIJK(384),HIJK(384),ATIJK(384),QILJ(384),
      &HILJ(384),ATILJ(384),QIKL(384),HIKL(384),ATIKL(384),
      &ECODE(384),GK2(10,10),P2(10),P3(10)
       COMMON NE,NNP,NMAT,MBAND,I,J,K,L,VOLUME
       COMMON/SUBBC/ECM(10,10)
       COMMON/SUBHGV/ELOAD(10)
       COMMON/LGDATA/TS(729),IX(384,10)
       COMMON/NPC/NPCODE(729)

       I=IX(E,1)
       J=IX(E,2)
       K=IX(E,3)
       L=IX(E,4)

       XI=X1(I)
       XJ=X1(J)
       XK=X1(K)
       XL=X1(L)
       YI=X2(I)
       YJ=X2(J)
       YK=X2(K)
       YL=X2(L)
       ZI=X3(I)
       ZJ=X3(J)
       ZK=X3(K)
       ZL=X3(L)

       QIJKD=QIJK(E)
       HIJKD=HIJK(E)
       ATIJKD=ATIJK(E)
       QILJD=QILJ(E)
       HILJD=HILJ(E)
       ATILJD=ATILJ(E)
       QIKLD=QIKL(E)
       HIKLD=HIKL(E)
       ATIKLD=ATIKL(E)
       ECODED=ECODE(E)

       DO 21 LL=1,10
       P2(LL)=0.
       P3(LL)=0.
       DO 21 LLL=1,10
    21 GK2(LL,LLL)=0.

       IF(ECODED.EQ.2) GO TO 200
       IF(ECODED.EQ.1) GO TO 100

C ***  FACE IKL

       SIK=SQRT((XI-XK)**2.+(YI-YK)**2.+(ZI-ZK)**2.)
       SKL=SQRT((XK-XL)**2.+(YK-YL)**2.+(ZK-ZL)**2.)
       SIL=SQRT((XL-XI)**2.+(YL-YI)**2.+(ZL-ZI)**2.)
       SS=(SIK+SKL+SIL)/2.
       AIKL=SQRT(SS*(SS-SIK)*(SS-SKL)*(SS-SIL))
       CON1=HIKLD*AIKL/180.
       CON2=QIKLD*AIKL/3.
       CON3=HIKLD*AIKL*ATIKLD/3.
       P2(7)=P2(7)+CON2
       P2(8)=P2(8)+CON2
       P2(10)=P2(10)+CON2
       P3(7)=P3(7)+CON3
       P3(8)=P3(8)+CON3
       P3(10)=P3(10)+CON3
       GK2(1,1)=GK2(1,1)+6.*CON1
       GK2(1,3)=GK2(1,3)-CON1
       GK2(1,4)=GK2(1,4)-CON1
      GK2(1,10)=GK2(1,10)-4.*CON1
      GK2(3,1)=GK2(3,1)-CON1
      GK2(3,3)=GK2(3,3)+6.*CON1
      GK2(3,4)=GK2(3,4)-CON1
      GK2(3,8)=GK2(3,8)-4.*CON1
      GK2(4,1)=GK2(4,1)-CON1
      GK2(4,3)=GK2(4,3)-CON1
      GK2(4,4)=GK2(4,4)+6.*CON1
      GK2(4,7)=GK2(4,7)-4.*CON1
      GK2(7,4)=GK2(7,4)-4.*CON1
      GK2(7,7)=GK2(7,7)+32.*CON1
      GK2(7,8)=GK2(7,8)+16.*CON1
      GK2(7,10)=GK2(7,10)+16.*CON1
      GK2(8,3)=GK2(8,3)-4.*CON1
      GK2(8,7)=GK2(8,7)+16*CON1
      GK2(8,8)=GK2(8,8)+32.*CON1
      GK2(8,10)=GK2(8,10)+16.*CON1
      GK2(10,1)=GK2(10,1)-4.*CON1
      GK2(10,7)=GK2(10,7)+16.*CON1
      GK2(10,8)=GK2(10,8)+16.*CON1
      GK2(10,10)=GK2(10,10)+32.*CON1
  200 CONTINUE
C ***  FACE ILJ
      SIJ=SQRT((XI-XJ)**2.+(YI-YJ)**2.+(ZI-ZJ)**2.)
      SJL=SQRT((XJ-XL)**2.+(YJ-YL)**2.+(ZJ-ZL)**2.)
      SIL=SQRT((XL-XI)**2.+(YL-YI)**2.+(ZL-ZI)**2.)
      SS=(SIJ+SJL+SIL)/2.
      AILJ=SQRT(SS*(SS-SIJ)*(SS-SJL)*(SS-SIL))
      CON1=HILJD*AILJ/180.
      CON2=QILJD*AILJ/3.
      CON3=HILJD*AILJ*ATILJD/3.
      P2(5)=P2(5)+CON2
      P2(8)=P2(8)+CON2
      P2(9)=P2(9)+CON2
      P3(5)=P3(5)+CON3
      P3(8)=P3(8)+CON3
      P3(9)=P3(9)+CON3
      GK2(1,1)=GK2(1,1)+6.*CON1
      GK2(1,2)=GK2(1,2)-CON1
      GK2(1,4)=GK2(1,4)-CON1
      GK2(1,9)=GK2(1,9)-4.*CON1
      GK2(2,1)=GK2(2,1)-CON1
      GK2(2,2)=GK2(2,2)+6.*CON1
      GK2(2,4)=GK2(2,4)-CON1
      GK2(2,8)=GK2(2,8)-4.*CON1
      GK2(4,1)=GK2(4,1)-CON1
      GK2(4,2)=GK2(4,2)-CON1
      GK2(4,4)=GK2(4,4)+6.*CON1
      GK2(4,5)=GK2(4,5)-4.*CON1
      GK2(5,4)=GK2(5,4)-4.*CON1
      GK2(5,5)=GK2(5,5)+32.*CON1
      GK2(5,8)=GK2(5,8)+16.*CON1
      GK2(5,9)=GK2(5,9)+16.*CON1
      GK2(8,2)=GK2(8,2)-4.*CON1
      GK2(8,5)=GK2(8,5)+16.*CON1
      GK2(8,8)=GK2(8,8)+32.*CON1
      GK2(8,9)=GK2(8,9)+16.*CON1
      GK2(9,1)=GK2(9,1)-4.*CON1
      GK2(9,5)=GK2(9,5)+16.*CON1
      GK2(9,8)=GK2(9,8)+16.*CON1
      GK2(9,9)=GK2(9,9)+32.*CON1
  100 CONTINUE
C ***  FACE IJK
```

```
      SIJ=SQRT((XI-XJ)**2.+(YI-YJ)**2.+(ZI-ZJ)**2.)
      SJK=SQRT((XJ-XK)**2.+(YJ-YK)**2.+(ZJ-ZK)**2.)
      SIK=SQRT((XK-XI)**2.+(YK-YI)**2.+(ZK-ZI)**2.)
      SS=(SIJ+SJK+SIK)/2.
      AIJK=SQRT(SS*(SS-SIJ)*(SS-SJK)*(SS-SIK))
      CON1=HIJKD*AIJK/180.
      CON2=QIJKD*AIJK/3.
      CON3=HIJKD*AIJK*ATIJKD/3.
      P2(5)=P2(5)+CON2
      P2(6)=P2(6)+CON2
      P2(7)=P2(7)+CON2
      P3(5)=P3(5)+CON3
      P3(6)=P3(6)+CON3
      P3(7)=P3(7)+CON3
      GK2(1,1)=GK2(1,1)+6.*CON1
      GK2(1,2)=GK2(1,2)-CON1
      GK2(1,3)=GK2(1,3)-CON1
      GK2(1,6)=GK2(1,6)-4.*CON1
      GK2(2,1)=GK2(2,1)-CON1
      GK2(2,2)=GK2(2,2)+6.*CON1
      GK2(2,3)=GK2(2,3)-CON1
      GK2(2,7)=GK2(2,7)-4.*CON1
      GK2(3,1)=GK2(3,1)-CON1
      GK2(3,2)=GK2(3,2)-CON1
      GK2(3,3)=GK2(3,3)+6.*CON1
      GK2(3,5)=GK2(3,5)-4.*CON1
      GK2(5,3)=GK2(5,3)-4.*CON1
      GK2(5,5)=GK2(5,5)+32.*CON1
      GK2(5,6)=GK2(5,6)+16.*CON1
      GK2(5,7)=GK2(5,7)+16.*CON1
      GK2(6,1)=GK2(6,1)-4.*CON1
      GK2(6,5)=GK2(6,5)+16.*CON1
      GK2(6,6)=GK2(6,6)+32.*CON1
      GK2(6,7)=GK2(6,7)+16.*CON1
      GK2(7,2)=GK2(7,2)-4.*CON1
      GK2(7,5)=GK2(7,5)+16.*CON1
      GK2(7,6)=GK2(7,6)+16.*CON1
      GK2(7,7)=GK2(7,7)+32.*CON1

      DO 4 LL=1,10
      ELOAD(LL)=ELOAD(LL)-P2(LL)+P3(LL)
      DO 4 JJ=1,10
    4 ECM(LL,JJ)=ECM(LL,JJ)+GK2(LL,JJ)
      RETURN
      END
C
      SUBROUTINE EGOO(X1,X2,X3)
      DIMENSION X1(729),X2(729),X3(729)
      COMMON NE,NNP,NMAT,MBAND,I,J,K,L,VOLUME

      XI=X1(I)
      XJ=X1(J)
      XK=X1(K)
      XL=X1(L)
      YI=X2(I)
      YJ=X2(J)
      YK=X2(K)
      YL=X2(L)
      ZI=X3(I)
      ZJ=X3(J)
      ZK=X3(K)
      ZL=X3(L)
      VOLUME=(XJ*YK*ZL+XK*YL*ZJ+XL*YJ*ZK-XL*YK*ZJ-XK*YJ*ZL-XJ*YL*ZK
     &      -XI*YK*ZL-XK*YL*ZI-XL*YI*ZK+XL*YK*ZI+XK*YI*ZL+XI*YL*ZK
     &      +XI*YJ*ZL+XJ*YL*ZI+XL*YI*ZJ-XL*YJ*ZI-XJ*YI*ZL-XI*YL*ZJ
     &        -XI*YJ*ZK-XJ*YK*ZI-XK*YI*ZJ+XK*YJ*ZI+XJ*YI*ZK+XI*YK*ZJ)/6.
      VOLUME=ABS(VOLUME)

      RETURN
      END
C
      SUBROUTINE ECHOO(X1,X2,X3,CX,CY,CZ)
      DIMENSION X1(729),X2(729),X3(729)
      COMMON NE,NNP,NMAT,MBAND,I,J,K,L,VOLUME
      COMMON/SUBBC/ECM(10,10)

      XI=X1(I)
      XJ=X1(J)
      XK=X1(K)
      XL=X1(L)
      YI=X2(I)
      YJ=X2(J)
      YK=X2(K)
      YL=X2(L)
      ZI=X3(I)
      ZJ=X3(J)
      ZK=X3(K)
      ZL=X3(L)

      B1=YK*ZJ+YJ*ZL+YL*ZK-YK*ZL-YJ*ZK-YL*ZJ
      B2=YK*ZL+YI*ZK+YL*ZI-YK*ZI-YL*ZK-YI*ZL
      B3=YJ*ZI+YL*ZJ+YI*ZL-YJ*ZL-YI*ZJ-YL*ZI
      B4=YJ*ZK+YI*ZJ+YK*ZI-YJ*ZI-YK*ZJ-YI*ZK

      C1=XK*ZL+XJ*ZK+XL*ZJ-XK*ZJ-XJ*ZL-XL*ZK
      C2=XK*ZI+XI*ZL+XL*ZK-XK*ZL-XI*ZK-XL*ZI
      C3=XJ*ZL+XI*ZJ+XL*ZI-XJ*ZI-XI*ZL-XL*ZJ
      C4=XJ*ZI+XK*ZJ+XI*ZK-XJ*ZK-XI*ZJ-XK*ZI

      D1=XK*YJ+XL*YK+XJ*YL-XK*YL-XJ*YK-XL*YJ
      D2=XK*YL+XL*YI+XI*YK-XK*YI-XL*YK-XI*YL
      D3=XJ*YI+XL*YJ+XI*YL-XJ*YL-XI*YJ-XL*YI
      D4=XJ*YK+XI*YJ+XK*YI-XJ*YI-XK*YJ-XI*YK

      ECH(1,1)=3.*(CX*B1**2.+CY*C1**2.+CZ*D1**2.)
      ECM(1,2)=-CX*B1*B2-CY*C1*C2-CZ*D1*D2
      ECM(1,3)=-CX*B1*B3-CY*C1*C3-CZ*D1*D3
      ECM(1,4)=-CX*B1*B4-CY*C1*C4-CZ*D1*D4
      ECM(1,5)=CX*B1*(-B1+3.*B2)+CY*C1*(-C1+3.*C2)+CZ*D1*(-D1+3.*D2)
      ECM(1,6)=-CX*B1*(B2+B3)-CY*C1*(C2+C3)-CZ*D1*(D2+D3)
      ECM(1,7)=CX*B1*(-B1+3.*B3)+CY*C1*(-C1+3.*C3)+CZ*D1*(-D1+3.*D3)
      ECM(1,8)=CX*B1*(-B1+3.*B4)+CY*C1*(-C1+3.*C4)+CZ*D1*(-D1+3.*D4)
      ECM(1,9)=-CX*B1*(B2+B4)-CY*C1*(C2+C4)-CZ*D1*(D2+D4)
      ECM(1,10)=-CX*B1*(B3+B4)-CY*C1*(C3+C4)-CZ*D1*(D3+D4)
      ECM(2,1)=ECM(1,2)
      ECM(2,2)=3.*(CX*B2**2.+CY*C2**2.+CZ*D2**2.)
      ECM(2,3)=-CX*B2*B3-CY*C2*C3-CZ*D2*D3
      ECM(2,4)=-CX*B2*B4-CY*C2*C4-CZ*D2*D4
      ECM(2,5)=CX*B2*(-B2+3.*B1)+CY*C2*(-C2+3.*C1)+CZ*D2*(-D2+3.*D1)
      ECM(2,6)=CX*B2*(-B2+3.*B3)+CY*C2*(-C2+3.*C3)+CZ*D2*(-D2+3.*D3)
      ECM(2,7)=-CX*B2*(B1+B3)-CY*C2*(C1+C3)-CZ*D2*(D1+D3)
      ECM(2,8)=-CX*B2*(B1+B4)-CY*C2*(C1+C4)-CZ*D2*(D1+D4)
      ECM(2,9)=CX*B2*(-B2+3.*B4)+CY*C2*(-C2+3.*C4)+CZ*D2*(-D2+3.*D4)
      ECM(2,10)=-CX*B2*(B3+B4)-CY*C2*(C3+C4)-CZ*D2*(D3+D4)
      ECM(3,1)=ECM(1,3)
      ECM(3,2)=ECM(2,3)
      ECM(3,3)=3.*(CX*B3**2.+CY*C3**2.+CZ*D3**2.)
      ECM(3,4)=-CX*B3*B4-CY*C3*C4-CZ*D3*D4
      ECM(3,5)=-CX*B3*(B1+B2)-CY*C3*(C1+C2)-CZ*D3*(D1+D2)
      ECM(3,6)=CX*B3*(-B3+3.*B2)+CY*C3*(-C3+3.*C2)+CZ*D3*(-D3+3.*D2)
      ECM(3,7)=CX*B3*(-B3+3.*B1)+CY*C3*(-C3+3.*C1)+CZ*D3*(-D3+3.*D1)
      ECM(3,8)=-CX*B3*(B1+B4)-CY*C3*(C1+C4)-CZ*D3*(D1+D4)
      ECM(3,9)=-CX*B3*(B2+B4)-CY*C3*(C2+C4)-CZ*D3*(D2+D4)
      ECM(3,10)=CX*B3*(-B3+3.*B4)+CY*C3*(-C3+3.*C4)+CZ*D3*(-D3+3.*D4)
      ECM(4,1)=ECM(1,4)
      ECM(4,2)=ECM(2,4)
      ECM(4,3)=ECM(3,4)
      ECM(4,4)=3.*(CX*B4**2.+CY*C4**2.+CZ*D4**2.)
      ECM(4,5)=-CX*B4*(B1+B2)-CY*C4*(C1+C2)-CZ*D4*(D1+D2)
      ECM(4,6)=-CX*B4*(B2+B3)-CY*C4*(C2+C3)-CZ*D4*(D2+D3)
      ECM(4,7)=-CX*B4*(B1+B3)-CY*C4*(C1+C3)-CZ*D4*(D1+D3)
      ECM(4,8)=CX*B4*(-B4+3.*B1)+CY*C4*(-C4+3.*C1)+CZ*D4*(-D4+3.*D1)
      ECM(4,9)=CX*B4*(-B4+3.*B2)+CY*C4*(-C4+3.*C2)+CZ*D4*(-D4+3.*D2)
      ECM(4,10)=CX*B4*(-B4+3.*B3)+CY*C4*(-C4+3.*C3)+CZ*D4*(-D4+3.*D3)
      ECM(5,1)=ECM(1,5)
      ECM(5,2)=ECM(2,5)
      ECM(5,3)=ECM(3,5)
      ECM(5,4)=ECM(4,5)
      ECM(5,5)=8.*(CX*(B1**2.+B2**2.+B1*B2)
     &              +CY*(C1**2.+C2**2.+C1*C2)
     &              +CZ*(D1**2.+D2**2.+D1*D2))
      ECM(5,6)=4.*(CX*(B2*B3+B1*B2+2.*B1*B3+B2**2.)
     &              +CY*(C2*C3+C1*C2+2.*C1*C3+C2**2.)
     &              +CZ*(D2*D3+D1*D2+2.*D1*D3+D2**2.))
      ECM(5,7)=4.*(CX*(B1*B3+B1*B2+2.*B2*B3+B1**2.)
     &              +CY*(C1*C3+C1*C2+2.*C2*C3+C1**2.)
     &              +CZ*(D1*D3+D1*D2+2.*D2*D3+D1**2.))
      ECM(5,8)=4.*(CX*(B1*B2+B1*B4+2.*B2*B4+B1**2.)
     &              +CY*(C1*C2+C1*C4+2.*C2*C4+C1**2.)
     &              +CZ*(D1*D2+D1*D4+2.*D2*D4+D1**2.))
      ECM(5,9)=4.*(CX*(B1*B2+B2*B4+2.*B1*B4+B2**2.)
     &              +CY*(C1*C2+C2*C4+2.*C1*C4+C2**2.)
     &              +CZ*(D1*D2+D2*D4+2.*D1*D4+D2**2.))
      ECM(5,10)=4.*(CX*(B1*B3+B2*B3+B1*B4+B2*B4)
     &               +CY*(C1*C3+C2*C3+C1*C4+C2*C4)
     &               +CZ*(D1*D3+D2*D3+D1*D4+D2*D4))
      ECM(6,1)=ECM(1,6)
      ECM(6,2)=ECM(2,6)
      ECM(6,3)=ECM(3,6)
      ECM(6,4)=ECM(4,6)
      ECM(6,5)=ECM(5,6)
      ECM(6,6)=8.*(CX*(B2**2.+B3**2.+B2*B3)
     &              +CY*(C2**2.+C3**2.+C2*C3)
     &              +CZ*(D2**2.+D3**2.+D2*D3))
      ECM(6,7)=4.*(CX*(B2*B3+B1*B3+2.*B1*B2+B3**2.)
     &              +CY*(C2*C3+C1*C3+2.*C1*C2+C3**2.)
     &              +CZ*(D2*D3+D1*D3+2.*D1*D2+D3**2.))
      ECM(6,8)=4.*(CX*(B1*B2+B2*B4+B1*B3+B3*B4)
     &              +CY*(C1*C2+C2*C4+C1*C3+C3*C4)
     &              +CZ*(D1*D2+D2*D4+D1*D3+D3*D4))
      ECM(6,9)=4.*(CX*(B2*B4+B2*B3+2.*B3*B4+B2**2.)
     &              +CY*(C2*C4+C2*C3+2.*C3*C4+C2**2.)
     &              +CZ*(D2*D4+D2*D3+2.*D3*D4+D2**2.))
      ECM(6,10)=4.*(CX*(B2*B3+B3*B4+2.*B2*B4+B3**2.)
     &               +CY*(C2*C3+C3*C4+2.*C2*C4+C3**2.)
     &               +CZ*(D2*D3+D3*D4+2.*D2*D4+D3**2.))
      ECM(7,1)=ECM(1,7)
      ECM(7,2)=ECM(2,7)
      ECM(7,3)=ECM(3,7)
      ECM(7,4)=ECM(4,7)
      ECM(7,5)=ECM(5,7)
      ECM(7,6)=ECM(6,7)
      ECM(7,7)=8.*(CX*(B1**2.+B3**2.+B1*B3)
     &              +CY*(C1**2.+C3**2.+C1*C3)
     &              +CZ*(D1**2.+D3**2.+D1*D3))
      ECM(7,8)=4.*(CX*(B1*B3+B1*B4+2.*B3*B4+B1**2.)
     &              +CY*(C1*C3+C1*C4+2.*C3*C4+C1**2.)
     &                  +CZ*(D1*D3+D1*D4+2.*D3*D4+D1**2.))
      ECH(7,9)=4.*(CX*(B2*B3+B1*B4+B1*B2+B3*B4)
     &                  +CY*(C2*C3+C1*C4+C1*C2+C3*C4)
     &                  +CZ*(D2*D3+D1*D4+D1*D2+D3*D4))
      ECH(7,10)=4.*(CX*(B3*B4+B1*B3+2.*B1*B4+B3**2.)
     &                   +CY*(C3*C4+C1*C3+2.*C1*C4+C3**2.)
     &                   +CZ*(D3*D4+D1*D3+2.*D1*D4+D3**2.))
      ECH(8,1)=ECH(1,8)
      ECH(8,2)=ECH(2,8)
      ECH(8,3)=ECH(3,8)
      ECH(8,4)=ECH(4,8)
      ECH(8,5)=ECH(5,8)
      ECH(8,6)=ECH(6,8)
      ECH(8,7)=ECH(7,8)
      ECH(8,8)=8.*(CX*(B1**2.+B4**2.+B1*B4)
     &                  +CY*(C1**2.+C4**2.+C1*C4)
     &                  +CZ*(D1**2.+D4**2.+D1*D4))
      ECH(8,9)=4.*(CX*(B1*B4+B2*B4+2.*B1*B2+B4**2.)
     &                  +CY*(C1*C4+C2*C4+2.*C1*C2+C4**2.)
     &                  +CZ*(D1*D4+D2*D4+2.*D1*D2+D4**2.))
      ECH(8,10)=4.*(CX*(B1*B4+B3*B4+2.*B1*B3+B4**2.)
     &                   +CY*(C1*C4+C3*C4+2.*C1*C3+C4**2.)
     &                   +CZ*(D1*D4+D3*D4+2.*D1*D3+D4**2.))
      ECH(9,1)=ECH(1,9)
      ECH(9,2)=ECH(2,9)
      ECH(9,3)=ECH(3,9)
      ECH(9,4)=ECH(4,9)
      ECH(9,5)=ECH(5,9)
      ECH(9,6)=ECH(6,9)
      ECH(9,7)=ECH(7,9)
      ECH(9,8)=ECH(8,9)
      ECH(9,9)=8.*(CX*(B2**2.+B4**2.+B2*B4)
     &                  +CY*(C2**2.+C4**2.+C2*C4)
     &                  +CZ*(D2**2.+D4**2.+D2*D4))
      ECH(9,10)=4.*(CX*(B3*B4+B2*B4+2.*B2*B3+B4**2.)
     &                   +CY*(C3*C4+C2*C4+2.*C2*C3+C4**2.)
     &                   +CZ*(D3*D4+D2*D4+2.*D2*D3+D4**2.))
      ECH(10,1)=ECH(1,10)
      ECH(10,2)=ECH(2,10)
      ECH(10,3)=ECH(3,10)
      ECH(10,4)=ECH(4,10)
      ECH(10,5)=ECH(5,10)
      ECH(10,6)=ECH(6,10)
      ECH(10,7)=ECH(7,10)
      ECH(10,8)=ECH(8,10)
      ECH(10,9)=ECH(9,10)
      ECH(10,10)=8.*(CX*(B3**2.+B4**2.+B3*B4)
     &                    +CY*(C3**2.+C4**2.+C3*C4)
     &                    +CZ*(D3**2.+D4**2.+D3*D4))

      DO 20 M=1,10
      DO 20 N=1,10
   20 ECH(M,N)=ECH(M,N)/(180.*VOLUME)

      RETURN
      END
C
      SUBROUTINE EHCM00(DENS,SPH,EHCM)
      DIMENSION EHCM(10,10)
      COMMON NE,NNP,NMAT,MBAND,I,J,K,L,VOLUME
      CON=DENS*SPH*VOLUME

      EHCM(1,1)=CON/70.
      EHCM(1,2)=CON/420.
      EHCM(1,3)=CON/420.
      EHCM(1,4)=CON/420.
EHCM(1,5)=-CON/105.
EHCM(1,6)=-CON/70.
EHCM(1,7)=-CON/105.
EHCM(1,8)=-CON/105.
EHCM(1,9)=-CON/70.
EHCM(1,10)=-CON/70.
EHCM(2,1)=EHCM(1,2)
EHCM(2,2)=CON/70.
EHCM(2,3)=CON/420.
EHCM(2,4)=CON/420.
EHCM(2,5)=-CON/105.
EHCM(2,6)=-CON/105.
EHCM(2,7)=-CON/70.
EHCM(2,8)=-CON/70.
EHCM(2,9)=-CON/105.
EHCM(2,10)=-CON/70.
EHCM(3,1)=EHCM(1,3)
EHCM(3,2)=EHCM(2,3)
EHCM(3,3)=CON/70.
EHCM(3,4)=CON/420.
EHCM(3,5)=-CON/70.
EHCM(3,6)=-CON/105.
EHCM(3,7)=-CON/105.
EHCM(3,8)=-CON/70.
EHCM(3,9)=-CON/70.
EHCM(3,10)=-CON/105.
EHCM(4,1)=EHCM(1,4)
EHCM(4,2)=EHCM(2,4)
EHCM(4,3)=EHCM(3,4)
EHCM(4,4)=CON/70.
EHCM(4,5)=-CON/70.
EHCM(4,6)=-CON/70.
EHCM(4,7)=-CON/70.
EHCM(4,8)=-CON/105.
EHCM(4,9)=-CON/105.
EHCM(4,10)=-CON/105.
EHCM(5,1)=EHCM(1,5)
EHCM(5,2)=EHCM(2,5)
EHCM(5,3)=EHCM(3,5)
EHCM(5,4)=EHCM(4,5)
EHCM(5,5)=8.*CON/105.
EHCM(5,6)=4.*CON/105.
EHCM(5,7)=4.*CON/105.
EHCM(5,8)=4.*CON/105.
EHCM(5,9)=4.*CON/105.
EHCM(5,10)=2.*CON/105.
EHCM(6,1)=EHCM(1,6)
EHCM(6,2)=EHCM(2,6)
EHCM(6,3)=EHCM(3,6)
EHCM(6,4)=EHCM(4,6)
EHCM(6,5)=EHCM(5,6)
EHCM(6,6)=8.*CON/105.
EHCM(6,7)=4.*CON/105.
EHCM(6,8)=2.*CON/105.
EHCM(6,9)=4.*CON/105.
EHCM(6,10)=4.*CON/105.
EHCM(7,1)=EHCM(1,7)
EHCM(7,2)=EHCM(2,7)
EHCM(7,3)=EHCM(3,7)
EHCM(7,4)=EHCM(4,7)
EHCM(7,5)=EHCM(5,7)
EHCM(7,6)=EHCM(6,7)
EHCM(7,7)=8.*CON/105.
EHCM(7,8)=4.*CON/105.
EHCM(7,9)=2.*CON/105.
EHCM(7,10)=4.*CON/105.
```

```
      EHCM(8,1)=EHCM(1,8)
      EHCM(8,2)=EHCM(2,8)
      EHCM(8,3)=EHCM(3,8)
      EHCM(8,4)=EHCM(4,8)
      EHCM(8,5)=EHCM(5,8)
      EHCM(8,6)=EHCM(6,8)
      EHCM(8,7)=EHCM(7,8)
      EHCM(8,8)=8.*CON/105.
      EHCM(8,9)=4.*CON/105.
      EHCM(8,10)=4.*CON/105.
      EHCM(9,1)=EHCM(1,9)
      EHCM(9,2)=EHCM(2,9)
      EHCM(9,3)=EHCM(3,9)
      EHCM(9,4)=EHCM(4,9)
      EHCM(9,5)=EHCM(5,9)
      EHCM(9,6)=EHCM(6,9)
      EHCM(9,7)=EHCM(7,9)
      EHCM(9,8)=EHCM(8,9)
      EHCM(9,9)=8.*CON/105.
      EHCM(9,10)=4.*CON/105.
      EHCM(10,1)=EHCM(1,10)
      EHCM(10,2)=EHCM(2,10)
      EHCM(10,3)=EHCM(3,10)
      EHCM(10,4)=EHCM(4,10)
      EHCM(10,5)=EHCM(5,10)
      EHCM(10,6)=EHCM(6,10)
      EHCM(10,7)=EHCM(7,10)
      EHCM(10,8)=EHCM(8,10)
      EHCM(10,9)=EHCM(9,10)
      EHCM(10,10)=8.*CON/105.

      RETURN
      END
C
      SUBROUTINE HGV(N,DENS,NPCODE,QGEN)
      DIMENSION QGEN(384),NPCODE(729)
      COMMON NE,NNP,NMAT,MBAND,I,J,K,L,VOLUME
      COMMON/SUBHGV/ELOAD(10)
      COMMON/LGDATA/PRESC(729,1),IX(384,10)

      CON=-QGEN(N)*VOLUME/20.
      DO 20 M=1,4
      I=IX(N,M)
      IF(NPCODE(I).EQ.1)GO TO 20
      ELOAD(I)=ELOAD(I)+CON
   20 CONTINUE
      CON=QGEN(N)*VOLUME/5.
      DO 21 M=5,10
      I=IX(N,M)
      IF(NPCODE(I).EQ.1)GO TO 21
      ELOAD(I)=ELOAD(I)+CON
   21 CONTINUE

      RETURN
      END
C
      SUBROUTINE FRONT(NVFIX,ICASE)
      DIMENSION FIXED(729),EQUAT(120),VECRV(120),GLOAD(120),GSTIF(7260)
     &,ESTIF(10,10),IFFIX(729),NACVA(120),LOCEL(16),NDEST(16),ASDIS(729)
      COMMON/LGDATA/PRESC(729,1),LNODS(384,10)
      COMMON NELEM,NPOIN,NMAT,MBAND,I,J,K,L,VOLUME
      COMMON/AAA/IFPRE(729,1)
      COMMON/FRON1/T(729),ZASDIS(729),NOFIX(729)
      COMMON/SUBHGV/ELOAD(10)
      NFUNC(I,J)=(J*J-J)/2+I
      ICASE=ICASE+1
      NEVAB=10
      NDIME=3
      MFRON=120
      MSTIF=7260


C
C              INTERPRET FIXITY DATA IN VECTOR FORM
C
      NTOTV=NPOIN*1
      DO 100 ITOTV=1,NTOTV
      IFFIX(ITOTV)=0
  100 FIXED(ITOTV)=0.0
      DO 110 IVFIX=1,NVFIX
      NLOCA=(NOFIX(IVFIX)-1)*1
      DO 110 IDOFN=1,1
      NGASH=NLOCA+IDOFN
      IFFIX(NGASH)=IFPRE(IVFIX,IDOFN)
  110 FIXED(NGASH)=PRESC(IVFIX,IDOFN)



C              CHANGE THE SIGN OF THE LAST APPEARANCE OF EACH NODE
C
      DO 140 IPOIN=1,NPOIN
      KLAST=0
      DO 130 IELEM=1,NELEM
      DO 120 INODE=1,10
      IF(LNODS(IELEM,INODE).NE.IPOIN)GO TO 120
      KLAST=IELEM
      NLAST=INODE
  120 CONTINUE
  130 CONTINUE
      IF(KLAST.NE.0)LNODS(KLAST,NLAST)=-IPOIN
  140 CONTINUE
C
C              START BY INITIALIZING EVERYTHING THAT MATTERS TO ZERO
C
      DO 150 ISTIF=1,MSTIF
  150 GSTIF(ISTIF)=0.0
      DO 160 IFRON=1,MFRON
      GLOAD(IFRON)=0.0
      EQUAT(IFRON)=0.0
      VECRV(IFRON)=0.0
  160 NACVA(IFRON)=0
C
C              AND PREPARE FOR DISK READING AND WRITING OPERATIONS
C
C
      REWIND 1
      REWIND 2
      REWIND 3
      REWIND 4
C
C              ENTER MAIN ELEMENT ASSEMBLY-REDUCTION LOOP
C
      NFRON=0
      KELVA=0
      DO 380 IELEM=1,NELEM
      READ(3)ELOAD
      KEVAB=0
      READ(1) ESTIF
      DO 170 INODE=1,10
      DO 170 IDOFN=1,1
      NPOSI=(INODE-1)*1+IDOFN
      LOCNO=LNODS(IELEM,INODE)
      IF(LOCNO.GT.0)LOCEL(NPOSI)=(LOCNO-1)*1+IDOFN
      IF(LOCNO.LT.0)LOCEL(NPOSI)=(LOCNO+1)*1-IDOFN
  170 CONTINUE
C
C           START BY LOOKING FOR EXISTING DESTINATIONS
C
      DO 210 IEVAB=1,NEVAB
      NIKNO=IABS(LOCEL(IEVAB))
      KEXIS=0
      DO 180 IFRON=1,NFRON
      IF(NIKNO.NE.NACVA(IFRON))GO TO 180
      KEVAB=KEVAB+1
      KEXIS=1
      NDEST(KEVAB)=IFRON
  180 CONTINUE
      IF(KEXIS.NE.0)GO TO 210
C
C           WE NOW SEEK NEW EMPTY PLACES FOR DESTINATION VECTOR
C
C
      DO 190 IFRON=1,MFRON
      IF(NACVA(IFRON).NE.0)GO TO 190
      NACVA(IFRON)=NIKNO
      KEVAB=KEVAB+1
      NDEST(KEVAB)=IFRON
      GO TO 200
  190 CONTINUE
C
C           THE NEW PLACES MAY DEMAND AN INCREASE
C           IN CURRENT FRONTWIDTH
C
C
  200 IF(NDEST(KEVAB).GT.NFRON)NFRON=NDEST(KEVAB)
  210 CONTINUE
C
C           ASSEMBLE ELEMENT LOADS
C
      DO 240 IEVAB=1,NEVAB
      IDEST=NDEST(IEVAB)
      GLOAD(IDEST)=GLOAD(IDEST)+ELOAD(IEVAB)
C
C           ASSEMBLE THE ELEMENT STIFFNESSES-BUT NOT IN RESOLUTION
C
      IF(ICASE.GT.1)GO TO 230
      DO 220 JEVAB=1,IEVAB
      JDEST=NDEST(JEVAB)
      NGASH=NFUNC(IDEST,JDEST)
      NGISH=NFUNC(JDEST,IDEST)
      IF(JDEST.GE.IDEST)GSTIF(NGASH)=GSTIF(NGASH)+ESTIF(IEVAB,JEVAB)
      IF(JDEST.LT.IDEST)GSTIF(NGISH)=GSTIF(NGISH)+ESTIF(IEVAB,JEVAB)
  220 CONTINUE
  230 CONTINUE
  240 CONTINUE
C
C           RE-EXAMINE EACH ELEMENT NODE ,TO
C           ENQUIRE WHICH CAN BE ELIMINATED
C
      DO 370 IEVAB=1,NEVAB
      NIKNO=-LOCEL(IEVAB)
      IF(NIKNO.LE.0)GO TO 370
C
C           FIND POSITONS OF VARIABLES READY FOR ELIMINATON
```

```
C
      DO 350 IFRON=1,NFRON
      IF(NACVA(IFRON).NE.NIKNO)GO TO 350
C
C             EXTRACT THE COEFFICIENTS OF THE NEW EQUATION
C             FOR ELIMINATION
C
      IF(ICASE.GT.1)GO TO 260
      DO 250 JFRON=1,MFRON
      IF(IFRON.LT.JFRON)NLOCA=NFUNC(IFRON,JFRON)
      IF(IFRON.GE.JFRON)NLOCA=NFUNC(JFRON,IFRON)
      EQUAT(JFRON)=GSTIF(NLOCA)
250   GSTIF(NLOCA)=0.
260   CONTINUE
C
C             AND EXTRACT THE CORRESPONDING RIGHT HAND SIDES
C
      EQRHS=GLOAD(IFRON)
      GLOAD(IFRON)=0.
      KELVA=KELVA+1
C
C             WRITE EQUATIONS TO DISK OR TO TAPE
C
      IF(ICASE.GT.1)GO TO 270
      WRITE(2)EQUAT,EQRHS,IFRON,NIKNO
      GO TO 280
270   WRITE(4)EQRHS
      READ(2)EQUAT,DUMMY,IDUMM,NIKNO
280   CONTINUE
C
C             DEAL WITH PIVOT
C
      PIVOT=EQUAT(IFRON)
      EQUAT(IFRON)=0.0
C
C             ENQUIRE WHETHER THE PRESENT VARIABLE IS FREE OR PRESCRIBE
C
      IF(IFFIX(NIKNO).EQ.0)GO TO 300
C
C             DEAL WITH A PRESCRIBED DEFLECTION
C
      DO 290 JFRON=1,NFRON
290   GLOAD(JFRON)=GLOAD(JFRON)-FIXED(NIKNO)*EQUAT(JFRON)
      GO TO 340
C
C             ELIMINATE A FREE VARIABLE - DEAL WITH THE RIGHT HAND SIDE
C             FIRST
C
300   DO 330 JFRON=1,NFRON
      GLOAD(JFRON)=GLOAD(JFRON)-EQUAT(JFRON)*EQRHS/PIVOT
C
C             NOW DEAL WIHT THE COEFFICIENTS IN CORE
C
      IF(ICASE.GT.1)GO TO 320
      IF(EQUAT(JFRON).EQ.0.0)GO TO 330
      NLOCA=NFUNC(0,JFRON)
      DO 310 LFRON=1,JFRON
      NGASH=LFRON+NLOCA
310   GSTIF(NGASH)=GSTIF(NGASH)-EQUAT(JFRON)*EQUAT(LFRON)/PIVOT
320   CONTINUE
330   CONTINUE
340   EQUAT(IFRON)=PIVOT
C
C             RECORD THE NEW VACANT SPACE ,AND REDUCE FRONTWIDTH IF
C             POSSIBLE
      NACVA(IFRON)=0
      GO TO 360
C
C           COMPLETE THE ELEMENT LOOP IN THE FORWARD ELIMINATION
C
350   CONTINUE
360   IF(NACVA(NFRON).NE.0)GO TO 370
      NFRON=NFRON-1
      IF(NFRON.GT.0)GO TO 360
370   CONTINUE
380   CONTINUE
C
C           ENTER BACK-SUBSTITUTION PHASE,LOOP BACKWARDS
C           THROUGH VARIABLES
C
      DO 410 IELVA=1,KELVA
C
C           READ A NEW EQUATION
C
      BACKSPACE 2
      READ(2)EQUAT,EQRHS,IFRON,NIKNO
      BACKSPACE 2
      IF(ICASE.EQ.1)GO TO 390
      BACKSPACE 4
      READ(4)EQRHS
      BACKSPACE 4
390   CONTINUE
C
C           PREPARE TO BACK-SUSTITUTE FROM THE CURRENT EQUATION
C
      PIVOT=EQUAT(IFRON)
      IF(IFFIX(NIKNO).EQ.1)VECRV(IFRON)=FIXED(NIKNO)
      IF(IFFIX(NIKNO).EQ.0)EQUAT(IFRON)=0.
C
C           BACK-SUBSTITUTE IN THE CURRENT EQUATION
C
      DO 400 JFRON=1,MFRON
400   EQRHS=EQRHS-VECRV(JFRON)*EQUAT(JFRON)
C
C           PUT THE FINAL VALUES WHERE THEY BELONG
C
      IF(IFFIX(NIKNO).EQ.0)VECRV(IFRON)=EQRHS/PIVOT
      IF(IFFIX(NIKNO).EQ.1)FIXED(NIKNO)=-EQRHS
      ASDIS(NIKNO)=VECRV(IFRON)
410   CONTINUE
      DO 450 IPOIN=1,NPOIN
      NGASH=IPOIN*1
      NGISH=NGASH-1+1
      ZASDIS(IPOIN)=ASDIS(NGISH)
450   CONTINUE
C
C           POST FRONT - RESET ALL ELEMENT CONNECTION NUMBERS
C           TO POSITIVE VALUES FOR SUBSEQUENT USE IN STRESS
C           CALCULATION
C
      DO 520 IELEM=1,NELEM
      DO 520 INODE=1,10
520   LNODS(IELEM,INODE)=IABS(LNODS(IELEM,INODE))
      RETURN
      END
C
      SUBROUTINE GOFIX(K)
      COMMON NE,NNP
      COMMON/FRON1/T(729),ZASDIS(729),
     &NOFIX(729)
      COMMON/LGDATA/PRESC(729,1),IX(384,10)
      COMMON/NPC/NPCODE(729)
      COMMON/AAA/IFPRE(729,1)
      K=0
      DO 1 I=1,NNP
      IF(NPCODE(I).EQ.1)THEN
      K=K+1
      NOFIX(K)=I
      IFPRE(K,1)=1
      PRESC(K,1)=T(I)
      ENDIF
    1 CONTINUE
      RETURN
      END
14.48.45.UCLP, BU, P03     ,        1.171KLNS.
```

APPENDIX D

LISTING OF A FE PROGRAM USING SIMPLEX TETRAHEDRON

```
      PROGRAM TRSIMP(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C
C
C ==================================================================
C LINEAR TII-DIMENSIONAL TETRAHEDRAL FINITE ELEMENT PROGRAM FOR THE
C SOLUTION OF LINEAR,TRANSIENT,TII-DIMENSIONAL,ISOTROPIC HEAT
C CONDUCTION PROBLEMS IN CUBIC VOLUMES WITH TIME-INDEPENDENT HEAT
C GENERATION AND TIME-INDEPENDENT BOUNDARY CONDITIONS
C ==================================================================
C
C
C
C     TERMINOLOGY
C
C
C
C       AIJK,AIKL,AILJ          AREAS ON FACES IJK, IKL, & LJ RESPECTIVELY
C       ATIJK,ATIKL,AILJ        AMBIENT TEMPERATURES ON FACES IJK, IKL &
C                                 ILJ ESPECTIVELY
C       CX,CY,CZ                CONDUCTION PARAMETERS IN THE DIRECTIONS
C                                 X, Y & Z RESPECTIVELY
C       DT                      TIME INCREMENT
C       ECODE                   ELEMENT CODE
C                                 0 IF ELEMENT IS AN INTERNAL ELEMENT
C                                 1 IF ELEMENT IS A SIDE ELEMENT
C                                 2 IF ELEMENT IS AN EDGE ELEMENT
C                                 3 IF ELEMENT IS A CORNER ELEMENT
C       HIJK,HIKL,HILJ          CONVECTIVE HEAT TRANSFER COEFFICIENTS
C                                 ON EDGES IJK, IKL & ILJ RESPECTIVELY
C       ICOUNT                  COUNTING INDEX FOR PRINTING THE
C                                 COMPUTED TEMPERATURE DISTRIBUTION
C       INT                     PRINTING INDEX FOR THE COMPUTED
C                                 TEMPERATURE DISTRIBUTION
C       MBAND                   BANDWIDTH
C       MBMAX                   MAXIMUM ALLOWABLE BANDWIDTH
C       NDT                     NUMBER OF TIME INCREMENTS
C       NE                      NUMBER OF ELEMENTS
C       NEMAX                   MAXIMUM NUMBER OF ELEMENTS
C       NNP                     NUMBER OF NODAL POINTS
C       NNPMAX                  MAXIMUM NUMBER OF NODAL POINTS
C       QIJK,QIKL,QILJ          SPECIFIED INCOMING HEAT FLUX COMPONENTS
C                                 ALONG THE NORMALS OF FACES IJK, IKL &
C                                 ILJ RESPECTIVELY
C       DENS                           DENSITY OF THE MATERIAL
C       SIJ,SIK,SIL,SKL,
C       SJK,SJL                 LENGTHS OF SIDES IJ, IK, IL, KL, JK &
C                                 JL ON RELATED FACES RESPECTIVELY
C       SPH                     SPECIFIC HEAT OF THE MATERIAL
C       TIME                    TIME LEVEL
C       VOLUME                  ELEMENT VOLUME
C
C
C       NPCODE(I),I=1,2,...,NNP BOUNDARY CONDITION CODE OF THE ITH
C                                   NODE
C                                 0 IF TEMPERATURE IS NOT SPECIFIED AT
C                                   THE ITH NODE
C                                 1 IF TEMPERATURE IS SPECIFIED AT THE
C                                   ITH NODE
C       QGEN(I),I=1,2,...,NE    RATE OF HEAT GENERATION PER UNIT
C                                 VOLUME WITHIN THE ITH ELEMENT AT
C                                 ALL TIMES
C       X1(I),I=1,2,...,NNP     X-COORDINATE OF THE ITH NODE
C       X2(I),I=1,2,...,NNP     Y-COORDINATE OF THE ITH NODE
C       X3(I),I=1,2,...,NNP     Z-COORDINATE OF THE ITH NODE
C
```

```
C       Q                           THERMAL LOAD VECTOR,1*NNP
C       T                           TEMPERATURE VECTOR,1*NNP
C       ELV                         EFFECTIVE LOAD VECTOR,1*NNP
C
C
C
C       IX(I,J),I=1,2,...,NE        NODAL POINT LABEL OF THE JTH NODE
C               J=1,2,3,4             OF THE ITH ELEMENT, NE*4
C       CM                          EFFECTIVE CONDUCTIVITY MATRIX,NNP*MBAND
C       ECM                         ELEMENT CONDUCTIVITY MATRIX,4*4
C       EHCM                        ELEMENT HEAT CAPACITY MATRIX,4*4
C       HCM                         HEAT CAPACITY MATRIX,NMP*MBAND
C
C
C
C      DECLERATIONS
C
C       * THE PROGRAM IS DESIGNED TO TREAT THE MATRICES, FORMED
C       DURING EXECUTION, IN BANDED FORM.
C       * THE DATA OBTAINED FROM PROGRAM GNRATE SHOULD BE UPDATED
C       IN ORDER FOR IT TO INCLUDE THE VALUES FOR DT, NDT, SPH & DENS.
C

      PARAMETER (NEMAX=1296,NNPMAX=343,NMMAX=1,MBMAX=70)
      INTEGER E

      DIMENSION IX(NEMAX,4),X1(NNPMAX),X2(NNPMAX),X3(NNPMAX),
     &Q(NNPMAX),ELV(NNPMAX),ECM(4,4),EHCM(4,4),T(NNPMAX),QGEN(NEMAX),
     &CM(NNPMAX,MBMAX),HCM(NNPMAX,MBMAX),NPCODE(NNPMAX),ECODE(NEMAX),
     &QIJK(NEMAX),HIJK(NEMAX),ATIJK(NEMAX),QILJ(NEMAX),HILJ(NEMAX),
     &ATILJ(NEMAX),QIKL(NEMAX),HIKL(NEMAX),ATIKL(NEMAX)
      COMMON NE,NNP,NMAT,MBAND,I,J,K,L,VOLUME

      READ(5,*)NNP,NE
      READ(5,*)(K,X1(I),X2(I),X3(I),NPCODE(I),T(I),I=1,NNP)
      READ(5,*)DT,NDT
      READ(5,*)CX,CY,CZ,DENS,SPH
      READ(5,*)(K,(IX(I,J),J=1,4),ECODE(I),QGEN(I),QIJK(I),
     &HIJK(I),ATIJK(I),QILJ(I),HILJ(I),ATILJ(I),QIKL(I),HIKL(I),
     &ATIKL(I),I=1,NE)

C
C ================================================================
C        FORM SYSTEM CONDUCTIVITY AND HEAT CAPACITY MATRICES
C ================================================================
C
C        1.INITIALIZE THE BANDWIDTH,SYSTEM CONDUCTIVITY AND
C          HEAT CAPACITY MATRICES,AND LOAD VECTORS
C
      MBAND=0
      DO 1004 I=1,NNP
      Q(I)=0.
      ELV(I)=0.
      DO 1004 J=1,MBMAX
      CM(I,J)=0.
      HCM(I,J)=0.
 1004 CONTINUE
      DO 1009 E=1,NE
C
C        2.INITIALIZE ELEMENT CONDUCTIVITY AND HEAT
C          CAPACITY MATRICES
C
      DO 1008 M=1,4
      DO 1008 N=1,4
      ECM(M,N)=0.
      EHCM(M,N)=0.
```

```
1008  CONTINUE
C
C       3. FORM ELEMENT CONDUCTIVITY AND HEAT CAPACITY MATRICES
C
C           3.1.DEFINE THE VARIABLE NAMES I,J,K
C
      I=IX(E,1)
      J=IX(E,2)
      K=IX(E,3)
      L=IX(E,4)
C
C           3.2.CALCULATE GEOMETRICAL PARAMETERS OF THE ELEMENT
C
      CALL EGOO(NNPMAX,X1,X2,X3)
C
C           3.3.FORM THE ELEMENT CONDUCTIVITY MATRIX
C
      CALL ECMOO(NEMAX,NNPMAX,NMMAX,X1,X2,X3,CX,CY,CZ,ECM)
C
C           3.4.FORM THE ELEMENT HEAT CAPACITY MATRIX
C
      CALL EHCMOO(NEMAX,NMMAX,DENS,SPH,EHCM)
C
C           3.5.ADD ELEMENT MATRICES TO GET SYSTEM MATRICES
C
      CALL MASSOO(NEMAX,NNPMAX,MBMAX,E,IX,ECM,EHCM,CM,HCM)
1009  CONTINUE
C
C           3.6.PRINT THE BANDWIDTH
C
      PRINT 138,MBAND
C
C ==================================================================
C       BOUNDARY CONDITIONS
C ==================================================================
C
      DO 42 E=1,NE
      IF(ECODE(E).EQ.0) GO TO 42
      CALL BCOO(NNPMAX,MBMAX,X1,X2,X3,QIJK,HIJK,ATIJK,QILJ,HILJ,
     &ATILJ,QIKL,HIKL,ATIKL,Q,CM,ECODE,E,IX,NEMAX)
42    CONTINUE
C
C ==================================================================
C       FORM THE EFFECTIVE CONDUCTIVITY MATRIX
C ==================================================================
C
      DO 43 I=1,NNP
      DO 43 J=1,MBAND
      CM(I,J)=CM(I,J)+2.*HCM(I,J)/DT
43    CONTINUE
C
C ==================================================================
C       MODIFY THE EFFECTIVE CONDUCTIVITY MATRIX AND THE THERMAL
C       LOAD VECTOR
C ==================================================================
C
      CALL MDFOO(NNPMAX,MBMAX,NPCODE,T,Q,CM)
C
C ==================================================================
C       REDUCE THE EFFECTIVE CONDUCTIVITY MATRIX
C ==================================================================
C
      DO 280 N=1,NNP
      DO 260 L=2,MBAND
      C=CM(N,L)/CK(N,1)
      I=N+L-1
      IF(NNP-I)260,240,240
240   J=0
      DO 250 K=L,MBAND
      J=J+1
250   CM(I,J)=CM(I,J)-C*CM(N,K)
260   CM(N,L)=C
280   CONTINUE
C
C ==================================================================
C       CALCULATE AND PRINT THE TEMPERATURE DISTRIBUTION
C ==================================================================
C
9     TIME=0.
      ICOUNT=0.
      INT=10
C
C      1.PRINT THE INITIAL TEMPERATURE DISTRIBUTION
C
      PRINT 133,TIME,(I,T(I),I=1,NNP)
      DO 64 ITIME=1,NDT
C
C      2.FORM THE EFFECTIVE LOAD VECTOR
C
      DO 62 I=1,NNP
      ELV(I)=ELV(I)+Q(I)
      IF(NPCODE(I).EQ.1) GO TO 62
      DO 1062 J=1,MBAND
      K=I+J-1
      IF((K-NNP).GT.0) GO TO 1062
      ELV(I)=ELV(I)+2.*HCM(I,J)*T(K)/DT
      IF((J-1).NE.0) GO TO 1062
      DO 2062 L=2,MBAND
      M=I-L+1
      IF(M.LE.0) GO TO 2062
      ELV(I)=ELV(I)+2.*HCM(M,L)*T(M)/DT
2062  CONTINUE
1062  CONTINUE
62    CONTINUE
C
C      3.FORM THE HEAT GENERATION VECTOR AND ADD IT INTO THE
C        THE EFFECTIVE LOAD VECTOR
C
      DO 58 E=1,NE
      I=IX(E,1)
      J=IX(E,2)
      K=IX(E,3)
      L=IX(E,4)
      CALL EG00(NNPMAX,X1,X2,X3)
      CALL HGV(NEMAX,NNPMAX,NMMAX,E,DENS,NPCODE,ELV,QGEN,IX)
58    CONTINUE
C
C      4.SOLVE FOR NODAL TEMPERATURES
C
      DO 290 N=1,NNP
      DO 485 L=2,MBAND
      I=N+L-1
      IF(NNP-I)270,285,285
285   ELV(I)=ELV(I)-CM(N,L)*ELV(N)
485   CONTINUE
270   ELV(N)=ELV(N)/CM(N,1)
290   CONTINUE
      N=NNP
300   N=N-1
      IF(N)350,500,350
350   DO 400 K=2,MBAND
      L=N+K-1
      IF(NNP-L)400,370,370
370   ELV(N)=ELV(N)-CM(N,K)*ELV(L)
400   CONTINUE
      GO TO 300
500   CONTINUE
      DO 63 I=1,NNP
      T(I)=2.*ELV(I)-T(I)
      ELV(I)=0.
63    CONTINUE
C
C       5.UPDATE THE TIME LEVEL AND PRINT NODAL
C         TEMPERATURES
C
      TIME=TIME+DT
      ICOUNT=ICOUNT+1
      IF(ICOUNT-INT) 64,65,65
65    PRINT 136,TIME,(I,T(I),I=1,NNP)
      ICOUNT=0.
64    CONTINUE
C
C ====================================================================
C       FORMATS......
C ====================================================================
C
133   FORMAT(1H1//15X,60(1H=)/26X,'INPUT TABLE-TEMPERATURE DISTRI'
     &,'BUTION'/15X,60(1H=)/37X,'TIME=',F12.5/15X,60(1H-)/40X,'NODAL PO'
     &,'INT'/27X,'NUMBER',26X,'TEMPERATURES'/15X,60(1H-)/(28X,I5,24X,E15
     &.6))
136   FORMAT(1H1//15X,60(1H=)/26X,'OUTPUT TABLE-TEMPERATURE DISTRI'
     &,'BUTION'/15X,60(1H=)/37X,'TIME=',F12.5/15X,60(1H-)/40X,'NODAL PO'
     &,'INT'/27X,'NUMBER',26X,'TEMPERATURES'/15X,60(1H-)/(28X,I5,24X,E15
     &.6))
138   FORMAT(1H1,35(/),50X,'BANDWIDTH.......',I3)
C
      STOP
      END

      SUBROUTINE BCOO(NNPMAX,MBMAX,X1,X2,X3,QIJK,HIJK,ATIJK,QILJ,HILJ,
     &ATILJ,QIKL,HIKL,ATIKL,Q,CM,ECODE,E,IX,NEMAX)
      INTEGER E
      DIMENSION X1(NNPMAX),X2(NNPMAX),X3(NNPMAX),CM(NNPMAX,MBMAX),
     &Q(NNPMAX),QIJK(NEMAX),HIJK(NEMAX),ATIJK(NEMAX),QILJ(NEMAX),
     &HILJ(NEMAX),ATILJ(NEMAX),QIKL(NEMAX),HIKL(NEMAX),ATIKL(NEMAX),
     &ECODE(NEMAX),IX(NEMAX,4),GK2(4,4),P2(4),P3(4)
      COMMON NE,NNP,NMAT,MBAND,I,J,K,L,VOLUME

      I=IX(E,1)
      J=IX(E,2)
      K=IX(E,3)
      L=IX(E,4)

      XI=X1(I)
      XJ=X1(J)
      XK=X1(K)
      XL=X1(L)
      YI=X2(I)
      YJ=X2(J)
      YK=X2(K)
      YL=X2(L)
      ZI=X3(I)
      ZJ=X3(J)
      ZK=X3(K)
      ZL=X3(L)

      QIJKD=QIJK(E)
      HIJKD=HIJK(E)
      ATIJKD=ATIJK(E)
      QILJD=QILJ(E)
      HILJD=HILJ(E)
      ATILJD=ATILJ(E)
      QIKLD=QIKL(E)
      HIKLD=HIKL(E)
      ATIKLD=ATIKL(E)
      ECODED=ECODE(E)

      DO 21 LL=1,4
      P2(LL)=0.
      P3(LL)=0.
      DO 21 LLL=1,4
   21 GK2(LL,LLL)=0.

      IF(ECODED.EQ.2) GO TO 200
      IF(ECODED.EQ.1) GO TO 100

C ***  FACE IKL

      SIK=SQRT((XI-XK)**2.+(YI-YK)**2.+(ZI-ZK)**2.)
      SKL=SQRT((XK-XL)**2.+(YK-YL)**2.+(ZK-ZL)**2.)
      SIL=SQRT((XL-XI)**2.+(YL-YI)**2.+(ZL-ZI)**2.)
      SS=(SIK+SKL+SIL)/2.
      AIKL=SQRT(SS*(SS-SIK)*(SS-SKL)*(SS-SIL))
      CON1=HIKLD*AIKL/12.
      CON2=QIKLD*AIKL/3.
      CON3=HIKLD*AIKL*ATIKLD/3.
      DO 101 K=1,4
      P2(K)=P2(K)+CON2
      P3(K)=P3(K)+CON3
      DO 101 L=1,4
      GK2(K,L)=GK2(K,L)+CON1
  101 IF(K.EQ.L) GK2(K,L)=GK2(K,L)+CON1
      P2(2)=P2(2)-CON2
      P3(2)=P3(2)-CON3
      DO 102 K=1,4
      GK2(K,2)=GK2(K,2)-CON1
  102 GK2(2,K)=GK2(2,K)-CON1
  200 CONTINUE

C ***  FACE ILJ

      SIJ=SQRT((XI-XJ)**2.+(YI-YJ)**2.+(ZI-ZJ)**2.)
      SJL=SQRT((XJ-XL)**2.+(YJ-YL)**2.+(ZJ-ZL)**2.)
      SIL=SQRT((XL-XI)**2.+(YL-YI)**2.+(ZL-ZI)**2.)
      SS=(SIJ+SJL+SIL)/2.
      AILJ=SQRT(SS*(SS-SIJ)*(SS-SJL)*(SS-SIL))
      CON1=HILJD*AILJ/12.
      CON2=QILJD*AILJ/3.
      CON3=HILJD*AILJ*ATILJD/3.
      DO 103 K=1,4
      P2(K)=P2(K)+CON2
      P3(K)=P3(K)+CON3
      DO 103 L=1,4
      GK2(K,L)=GK2(K,L)+CON1
  103 IF(K.EQ.L) GK2(K,L)=GK2(K,L)+CON1
      P2(3)=P2(3)-CON2
      P3(3)=P3(3)-CON3
      DO 104 K=1,4
      GK2(K,3)=GK2(K,3)-CON1
  104 GK2(3,K)=GK2(3,K)-CON1
  100 CONTINUE

C ***  FACE IJK
```

```
      STJ=SQRT((XI-XJ)**2.+(YI-YJ)**2.+(ZI-ZJ)**2.)
      SJK=SQRT((XJ-XK)**2.+(YJ-YK)**2.+(ZJ-ZK)**2.)
      SIK=SQRT((XK-XI)**2.+(YK-YI)**2.+(ZK-ZI)**2.)
      SS=(SIJ+SJK+SIK)/2.
      AIJK=SQRT(SS*(SS-SIJ)*(SS-SJK)*(SS-SIK))
      CON1=HIJKD*AIJK/12.
      CON2=QIJKD*AIJK/3.
      CON3=HIJKD*AIJK*ATIJKD/3.
      DO 105 K=1,4
      P2(K)=P2(K)+CON2
      P3(K)=P3(K)+CON3
      DO 105 L=1,4
      GK2(K,L)=GK2(K,L)+CON1
  105 IF(K.EQ.L) GK2(K,L)=GK2(K,L)+CON1
      P2(4)=P2(4)-CON2
      P3(4)=P3(4)-CON3
      DO 106 K=1,4
      GK2(K,4)=GK2(K,4)-CON1
  106 GK2(4,K)=GK2(4,K)-CON1

      DO 4 LL=1,4
      I=IX(E,LL)
      Q(I)=Q(I)-P2(LL)+P3(LL)
      DO 4 M=1,4
      K=IX(E,M)
      J=K-I+1
      IF(MBAND.GE.J) GO TO 2
      MBAND=J
 2    IF(J.LE.0) GO TO 4
      CM(I,J)=CM(I,J)+GK2(LL,M)
 4    CONTINUE

      RETURN
      END


      SUBROUTINE ECOO(NNPMAX,X1,X2,X3)
      DIMENSION X1(NNPMAX),X2(NNPMAX),X3(NNPMAX)
      COMMON NE,NNP,NMAT,MBAND,I,J,K,L,VOLUME

      XI=X1(I)
      XJ=X1(J)
      XK=X1(K)
      XL=X1(L)
      YI=X2(I)
      YJ=X2(J)
      YK=X2(K)
      YL=X2(L)
      ZI=X3(I)
      ZJ=X3(J)
      ZK=X3(K)
      ZL=X3(L)
      VOLUME=(XJ*YK*ZL+XK*YL*ZJ+XL*YJ*ZK-XL*YK*ZJ-XK*YJ*ZL-XJ*YL*ZK
     &         -XI*YK*ZL-XK*YL*ZI-XL*YI*ZK+XL*YK*ZI+XK*YI*ZL+XI*YL*ZK
     &         +XI*YJ*ZL+XJ*YL*ZI+XL*YI*ZJ-XL*YJ*ZI-XJ*YI*ZL-XI*YL*ZJ
     &         -XI*YJ*ZK-XJ*YK*ZI-XK*YI*ZJ+XK*YJ*ZI+XJ*YI*ZK+XI*YK*ZJ)/6.
      VOLUME=ABS(VOLUME)

      RETURN
      END


      SUBROUTINE ECHOO(NEMAX,NNPMAX,NMMAX,X1,X2,X3,CX,CY,CZ,ECM)
      DIMENSION X1(NNPMAX),X2(NNPMAX),X3(NNPMAX),ECM(4,4)
      COMMON NE,NNP,NMAT,MBAND,I,J,K,L,VOLUME
      XI=X1(I)
      XJ=X1(J)
      XK=X1(K)
      XL=X1(L)
      YI=X2(I)
      YJ=X2(J)
      YK=X2(K)
      YL=X2(L)
      ZI=X3(I)
      ZJ=X3(J)
      ZK=X3(K)
      ZL=X3(L)

      B1=YK*ZJ+YJ*ZL+YL*ZK-YK*ZL-YJ*ZK-YL*ZJ
      B2=YK*ZL+YI*ZK+YL*ZI-YK*ZI-YL*ZK-YI*ZL
      B3=YJ*ZI+YL*ZJ+YI*ZL-YJ*ZL-YI*ZJ-YL*ZI
      B4=YJ*ZK+YI*ZJ+YK*ZI-YJ*ZI-YK*ZJ-YI*ZK

      C1=XK*ZL+XJ*ZK+XL*ZJ-XK*ZJ-XJ*ZL-XL*ZK
      C2=XK*ZI+XI*ZL+XL*ZK-XK*ZL-XI*ZK-XL*ZI
      C3=XJ*ZL+XI*ZJ+XL*ZI-XJ*ZI-XI*ZL-XL*ZJ
      C4=XJ*ZI+XK*ZJ+XI*ZK-XJ*ZK-XI*ZJ-XK*ZI

      D1=XK*YJ+XL*YK+XJ*YL-XK*YL-XJ*YK-XL*YJ
      D2=XK*YL+XL*YI+XI*YK-XK*YI-XL*YK-XI*YL
      D3=XJ*YI+XL*YJ+XI*YL-XJ*YL-XI*YJ-XL*YI
      D4=XJ*YK+XI*YJ+XK*YI-XJ*YI-XK*YJ-XI*YK

      ECM(1,1)=CX*B1**2.+CY*C1**2.+CZ*D1**2.
      ECM(1,2)=CX*B1*B2+CY*C1*C2+CZ*D1*D2
      ECM(1,3)=CX*B1*B3+CY*C1*C3+CZ*D1*D3
      ECM(1,4)=CX*B1*B4+CY*C1*C4+CZ*D1*D4
      ECM(2,1)=ECM(1,2)
      ECM(2,2)=CX*B2**2.+CY*C2**2.+CZ*D2**2.
      ECM(2,3)=CX*B2*B3+CY*C2*C3+CZ*D2*D3
      ECM(2,4)=CX*B2*B4+CY*C2*C4+CZ*D2*D4
      ECM(3,1)=ECM(1,3)
      ECM(3,2)=ECM(2,3)
      ECM(3,3)=CX*B3**2.+CY*C3**2.+CZ*D3**2.
      ECM(3,4)=CX*B3*B4+CY*C3*C4+CZ*D3*D4
      ECM(4,1)=ECM(1,4)
      ECM(4,2)=ECM(2,4)
      ECM(4,3)=ECM(3,4)
      ECM(4,4)=CX*B4**2.+CY*C4**2.+CZ*D4**2.

      DO 20 M=1,4
      DO 20 N=1,4
   20 ECM(M,N)=ECM(M,N)/(36.*VOLUME)

      RETURN
      END


      SUBROUTINE EHCMOD(NEMAX,NMMAX,DENS,SPH,EHCM)
      DIMENSION EHCM(4,4)
      COMMON NE,NNP,NMAT,MBAND,I,J,K,L,VOLUME

      CON=DENS*SPH*VOLUME/20.
      DO 20 M=1,4
      DO 20 N=1,4
      EHCM(M,N)=CON
   20 IF(M.EQ.N) EHCM(M,N)=EHCM(M,N)+CON

      RETURN
      END
      SUBROUTINE MASS00(NEMAX,NNPMAX,MBMAX,N,IX,ECM,EHCM,CM,HCM)
      DIMENSION IX(NEMAX,4),ECM(4,4),EHCM(4,4),CM(NNPMAX,MBMAX),
     &HCM(NNPMAX,MBMAX)
      COMMON NE,NNP,NMAT,MBAND,I,J,K,L,VOLUME

      DO 4 LL=1,4
      I=IX(N,LL)
      DO 4 M=1,4
      K=IX(N,M)
      J=K-I+1
      IF(MBAND.GE.J) GO TO 2
      MBAND=J
2     IF(J.LE.0) GO TO 4
      CM(I,J)=CM(I,J)+ECM(LL,M)
      HCM(I,J)=HCM(I,J)+EHCM(LL,M)
4     CONTINUE

      RETURN
      END

      SUBROUTINE MDF00(NNPMAX,MBMAX,NPCODE,T,Q,CM)
      DIMENSION NPCODE(NNPMAX),T(NNPMAX),Q(NNPMAX),
     &CM(NNPMAX,MBMAX)
      COMMON NE,NNP,NMAT,MBAND,I,J,K,L,VOLUME

      DO 4900 I=1,NNP
      IF(NPCODE(I).EQ.0)GO TO 4900
44    DO 49 J=2,MBAND
      K=I-J+1
      IF(K)46,46,45
45    Q(K)=Q(K)-CM(K,J)*T(I)
      CM(K,J)=0.
46    L=I+J-1
      IF(NNP-L)48,47,47
47    Q(L)=Q(L)-CM(I,J)*T(I)
48    CM(I,J)=0.
49    CONTINUE
      CM(I,1)=1.0
      Q(I)=T(I)
4900  CONTINUE

      RETURN
      END

      SUBROUTINE HGV(NEMAX,NNPMAX,NMMAX,N,DENS,NPCODE,ELV,QGEN,IX)
      DIMENSION QGEN(NEMAX),NPCODE(NNPMAX),ELV(NNPMAX),IX(NEMAX,4)
      COMMON NE,NNP,NMAT,MBAND,I,J,K,L,VOLUME

      CON=QGEN(N)*VOLUME/4.
      DO 20 M=1,4
      I=IX(N,M)
      IF(NPCODE(I).EQ.1)GO TO 20
      ELV(I)=ELV(I)+CON
   20 CONTINUE

      RETURN
      END
17.45.22.UCLP, BU, P03     ,        0.625KLNS.
```

APPENDIX E

LISTING OF A FE PROGRAM

USING SIMPLEX HEXAHEDRON

```
      PROGRAM TRCUNUH(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C
C
C =================================================================
C LINEAR III-DIMENSIONAL HEXAHEDRAL FINITE ELEMENT PROGRAM FOR THE
C SOLUTION OF LINEAR,TRANSIENT,III-DIMENSIONAL,ISOTROPIC HEAT
C CONDUCTION PROBLEMS IN CUBIC VOLUMES WITH TIME-INDEPENDENT HEAT
C GENERATION AND TIME-INDEPENDENT BOUNDARY CONDITIONS
C =================================================================
C
C
C
C     TERMINOLOGY
C
C
C
C      ATA,ATB,ATC        AMBIENT TEMPERATURES ON FACES A, B &
C                           C  RESPECTIVELY
C      B                  B MATRIX IN  DEFINITION OF CONDUCTIVITY MATRIX
C      CX,CY,CZ           CONDUCTION PARAMETERS IN THE DIRECTIONS
C                           X, Y & Z RESPECTIVELY
C      D                  D MATRIX IN  DEFINITION OF CONDUCTIVITY MATRIX
C      DENS               DENSITY OF THE MATERIAL
C      DETJ               DETERMINANT OF THE JACOBIAN
C      DT                 TIME INCREMENT
C      KA,KB,KC           ELEMENT CODES SPECIFYING THE FACES ON THE
C                           BODY SURFACE
C      HA,HB,HC           CONVECTIVE HEAT TRANSFER COEFFICIENTS
C                           ON EDGES A, B & C RESPECTIVELY
C      ICOUNT             COUNTING INDEX FOR PRINTING THE
C                           COMPUTED TEMPERATURE DISTRIBUTION
C      INT                PRINTING INDEX FOR THE COMPUTED
C                           TEMPERATURE DISTRIBUTION
C      JAC                JACOBIAN MATRIX
C      JACI                 ADJOINT OF THE JACOBIAN MATRIX
C      MBAND              BANDWIDTH
C      MBMAX              MAXIMUM ALLOWABLE BANDWIDTH
C      NDT                NUMBER OF TIME INCREMENTS
C      NE                 NUMBER OF ELEMENTS
C      NEMAX              MAXIMUM NUMBER OF ELEMENTS
C      NNP                NUMBER OF NODAL POINTS
C      NNPMAX             MAXIMUM NUMBER OF NODAL POINTS
C      QA,QB,QC           SPECIFIED INCOMING HEAT FLUX COMPONENTS
C                           ALONG THE NORMALS OF FACES A, B &
C                           C RESPECTIVELY
C      SHAPE              NODAL SHAPE FUNCTIONS
C      SPH                SPECIFIC HEAT OF THE MATERIAL
C      TIME               TIME LEVEL
C      VOLUME             ELEMENT VOLUME
C      W                  WEIGHTS IN GAUSS QUADRATURE
C
C      NPCODE(I),
C         I=1,2,....,NNP  BOUNDARY CONDITION CODE OF THE ITH NODE
C                             0 IF TEMPERATURE IS NOT SPECIFIED AT
C                               THE ITH NODE
C                             1 IF TEMPERATURE IS SPECIFIED AT THE
C                               ITH NODE
C      QGEN(I),
C         I=1,2,....,NE   RATE OF HEAT GENERATION PER UNIT
C                             VOLUME WITHIN THE ITH ELEMENT AT
C                               ALL TIMES
C      X1(I),I=1,2,....,NNP    X-COORDINATE OF THE ITH NODE
C      X2(I),I=1,2,....,NNP    Y-COORDINATE OF THE ITH NODE
C      X3(I),I=1,2,....,NNP    Z-COORDINATE OF THE ITH NODE
C      R, S, T2                LOCAL COORDINATES OF THE NODES
```

```
C       RR, SS, TT                LOCAL COORDINATES OF COLLOCATION POINTS
C                                   USED IN GAUSS QUADRATURE
C        Q                        THERMAL LOAD VECTOR,1*NNP
C        T                        TEMPERATURE VECTOR,1*NNP
C        ELV                      EFFECTIVE LOAD VECTOR,1*NNP
C
C
C
C        IX(I,J),I=1,2,...,NE     NODAL POINT LABEL OF THE JTH NODE
C                J=1,2,3,4          OF THE ITH ELEMENT
C        CM                       CONDUCTIVITY MATRIX,NNP*MBAND
C        ECM                      ELEMENT CONDUCTIVITY MATRIX,4*4
C        EHCM                     ELEMENT HEAT CAPACITY MATRIX,4*4
C        HCM                      HEAT CAPACITY MATRIX,NMP*MBAND
C
C
C
C       DECLERATIONS
C
C
C        * THE PROGRAM IS DESIGNED TO TREAT THE MATRICES, FORMED
C        DURING EXECUTION, IN BANDED FORM.
C        * THE DATA OBTAINED FROM PROGRAM GNCUBE SHOULD BE UPDATED
C        IN ORDER FOR IT TO INCLUDE THE VALUES FOR DT, NDT, SPH & DENS.
C
C
       PARAMETER (NEMAX=400,NNPMAX=343,NMMAX=1,MBMAX=75)
       INTEGER E
       REAL JAC(3,3),JACI(3,3)
       DIMENSION IX(NEMAX,8),X1(NNPMAX),X2(NNPMAX),X3(NNPMAX),KC(NEMAX),
      &Q(NNPMAX),ELV(NNPMAX),ECM(8,8),EHCM(8,8),T(NNPMAX),QGEN(NEMAX),
      &CM(NNPMAX,MBMAX),HCM(NNPMAX,MBMAX),NPCODE(NNPMAX),KA(NEMAX),
      &QA(NEMAX),HA(NEMAX),ATA(NEMAX),QB(NEMAX),HB(NEMAX),KB(NEMAX),
      &ATB(NEMAX),QC(NEMAX),HC(NEMAX),ATC(NEMAX),B(3,8),D(3,3),HGV(8)
       COMMON/BC/W(3),S(8),R(8),T2(8),SHAPE(8),PROD(8,8),PROD1(8,8),
      &RR(3),SS(3),TT(3)
       COMMON NE,NNP,MBAND,I1,I2,I3,I4,I5,I6,I7,I8

       READ(5,*)NNP,NE
       READ(5,*)(K,X1(I),X2(I),X3(I),NPCODE(I),T(I),I=1,NNP)
       READ(5,*)DT,NDT
       READ(5,*)CX,CY,CZ,DENS,SPH
       READ(5,*)(K,(IX(I,J),J=1,8),KA(I),KB(I),KC(I),QGEN(I),QA(I),
      &HA(I),ATA(I),QB(I),HB(I),ATB(I),QC(I),HC(I),ATC(I),I=1,NE)

C
C ============================================================
C        FORM SYSTEM CONDUCTIVITY AND HEAT CAPACITY MATRICES
C ============================================================
C
C        1.INITIALIZE THE BANDWIDTH,SYSTEM CONDUCTIVITY AND
C          HEAT CAPACITY MATRICES,AND LOAD VECTORS
C
       MBAND=0
       DO 1004 I=1,NNP
       Q(I)=0.
       ELV(I)=0.
       DO 1004 J=1,MBMAX
       CM(I,J)=0.
       HCM(I,J)=0.
1004   CONTINUE
       W(1)=.5555555555556
       W(2)=.8888888888889
       W(3)=.5555555555556
       RR(1)=-.77459666924148
       RR(2)=0.
      RR(3)= .77459666924148
      SS(1)=-.77459666924148
      SS(2)= 0.
      SS(3)= .77459666924148
      TT(1)=-.77459666924148
      TT(2)= 0.
      TT(3)= .77459666924148
      D(1,1)=CX
      D(2,2)=CY
      D(3,3)=CZ
      D(1,2)=D(1,3)=D(2,1)=D(2,3)=D(3,1)=D(3,2)=0.
      R(1)=-1.
      R(2)= 1.
      R(3)= 1.
      R(4)=-1.
      R(5)=-1.
      R(6)= 1.
      R(7)= 1.
      R(8)=-1.
      S(1)=-1.
      S(2)=-1.
      S(3)= 1.
      S(4)= 1.
      S(5)=-1.
      S(6)=-1.
      S(7)= 1.
      S(8)= 1.
      T2(1)=-1.
      T2(2)=-1.
      T2(3)=-1.
      T2(4)=-1.
      T2(5)= 1.
      T2(6)= 1.
      T2(7)= 1.
      T2(8)= 1.

      DO 1009 E=1,NE
C
C       2.INITIALIZE ELEMENT CONDUCTIVITY AND HEAT
C         CAPACITY MATRICES
C
      DO 1008 M=1,8
      DO 1008 N=1,8
      ECM(M,N)=0.
      EHCM(M,N)=0.
1008  CONTINUE
C
C       3.FORM ELEMENT CONDUCTIVITY AND HEAT CAPACITY MATRICES
C
      DO 1100 I=1,3
      DO 1100 J=1,3
      DO 1100 K=1,3

      DO 1101 II=1,8
      B(1,II)=R(II)*(1+SS(J)*S(II))*(1+TT(K)*T2(II))/8.
      B(2,II)=S(II)*(1+RR(I)*R(II))*(1+TT(K)*T2(II))/8.
1101  B(3,II)=T2(II)*(1+RR(I)*R(II))*(1+SS(J)*S(II))/8.

      DO 1102 II=1,3
      DO 1102 JJ=1,3
1102  JAC(II,JJ)=0.

      DO 1103 II=1,8
      CON1=R(II)*(1+SS(J)*S(II))*(1+TT(K)*T2(II))/8.
      CON2=S(II)*(1+RR(I)*R(II))*(1+TT(K)*T2(II))/8.
      CON3=T2(II)*(1+RR(I)*R(II))*(1+SS(J)*S(II))/8.
      JAC(1,1)=CON1*X1(IX(E,II))+JAC(1,1)
      JAC(1,2)=CON1*X2(IX(E,II))+JAC(1,2)
      JAC(1,3)=CON1*X3(IX(E,II))+JAC(1,3)
      JAC(2,1)=CON2*X1(IX(E,II))+JAC(2,1)
      JAC(2,2)=CON2*X2(IX(E,II))+JAC(2,2)
      JAC(2,3)=CON2*X3(IX(E,II))+JAC(2,3)
      JAC(3,1)=CON3*X1(IX(E,II))+JAC(3,1)
      JAC(3,2)=CON3*X2(IX(E,II))+JAC(3,2)
 1103 JAC(3,3)=CON3*X3(IX(E,II))+JAC(3,3)

      JACI(1,1)=JAC(2,2)*JAC(3,3)-JAC(3,2)*JAC(2,3)
      JACI(1,2)=JAC(3,2)*JAC(1,3)-JAC(1,2)*JAC(3,3)
      JACI(1,3)=JAC(1,2)*JAC(2,3)-JAC(2,2)*JAC(1,3)
      JACI(2,1)=JAC(3,1)*JAC(2,3)-JAC(2,1)*JAC(3,3)
      JACI(2,2)=JAC(1,1)*JAC(3,3)-JAC(3,1)*JAC(1,3)
      JACI(2,3)=JAC(2,1)*JAC(1,3)-JAC(1,1)*JAC(2,3)
      JACI(3,1)=JAC(2,1)*JAC(3,2)-JAC(3,1)*JAC(2,2)
      JACI(3,2)=JAC(3,1)*JAC(1,2)-JAC(1,1)*JAC(3,2)
      JACI(3,3)=JAC(1,1)*JAC(2,2)-JAC(2,1)*JAC(1,2)

      DETJ= JAC(1,1)*JAC(2,2)*JAC(3,3)
     &      +JAC(2,1)*JAC(3,2)*JAC(1,3)
     &      +JAC(1,2)*JAC(2,3)*JAC(3,1)
     &      -JAC(3,1)*JAC(2,2)*JAC(1,3)
     &      -JAC(2,1)*JAC(1,2)*JAC(3,3)
     &      -JAC(1,1)*JAC(3,2)*JAC(2,3)

      DO 1104 II=1,3
      DO 1104 JJ=1,8
      PROD(II,JJ)=0.
      DO 1104 KK=1,3
 1104 PROD(II,JJ)=PROD(II,JJ)+JACI(II,KK)*B(KK,JJ)

      DO 1105 II=1,3
      DO 1105 JJ=1,8
      PROD1(II,JJ)=0.
      DO 1105 KK=1,3
 1105 PROD1(II,JJ)=PROD1(II,JJ)+D(II,KK)*PROD(KK,JJ)

      DO 1106 II=1,3
      DO 1106 JJ=1,8
      PROD(II,JJ)=0.
      DO 1106 KK=1,3
 1106 PROD(II,JJ)=PROD(II,JJ)+JACI(KK,II)*PROD1(KK,JJ)

      DO 1107 II=1,8
      DO 1107 JJ=1,8
      PROD1(II,JJ)=0.
      DO 1108 KK=1,3
 1108 PROD1(II,JJ)=PROD1(II,JJ)+B(KK,II)*PROD(KK,JJ)
 1107 PROD1(II,JJ)=PROD1(II,JJ)/DETJ

      DO 1109 II=1,8
      DO 1109 JJ=1,8
 1109 ECM(II,JJ)=ECM(II,JJ)+W(I)*W(J)*W(K)*PROD1(II,JJ)

C
C          3.4.FORM THE ELEMENT HEAT CAPACITY MATRIX
C

      DO 1110 II=1,8
 1110 SHAPE(II)=(1+RR(I)*R(II))*(1+SS(J)*S(II))*(1+TT(K)*T2(II))/8.

      DO 1111 II=1,8
      DO 1111 JJ=1,8
 1111 PROD(II,JJ)=SHAPE(II)*SHAPE(JJ)*DETJ*DENS*SPH
```

```
      DO 1112 II=1,8
      DO 1112 JJ=1,8
1112  EHCM(II,JJ)=EHCM(II,JJ)+W(I)*W(J)*W(K)*PROD(II,JJ)

1100  CONTINUE
C
C          3.5.ADD ELEMENT MATRICES TO GET SYSTEM MATRICES
C
      CALL MASS00(NEMAX,NNPMAX,MBMAX,E,IX,ECM,EHCM,CM,HCM)
1009  CONTINUE
C
C          3.6.PRINT THE BANDWIDTH
C
      PRINT 138,MBAND
C
C ====================================================================
C       BOUNDARY CONDITIONS
C ====================================================================
C
      DO 42 E=1,NE
      IF(KA(E).EQ.0) GO TO 42
      CALL BC00(NNPMAX,MBMAX,X1,X2,X3,QA,HA,ATA,QB,HB,
     &ATB,QC,HC,ATC,Q,CM,KA,KB,KC,E,IX,NEMAX)
42    CONTINUE
C
C ====================================================================
C       FORM THE EFFECTIVE CONDUCTIVITY MATRIX
C ====================================================================
C
      DO 43 I=1,NNP
      DO 43 J=1,MBAND
      CM(I,J)=CM(I,J)+2.*HCM(I,J)/DT
43    CONTINUE
C
C ====================================================================
C       MODIFY THE EFFECTIVE CONDUCTIVITY MATRIX AND THE THERMAL
C       LOAD VECTOR
C ====================================================================
C
      CALL MDF00(NNPMAX,MBMAX,NPCODE,T,Q,CM)
C
C ====================================================================
C       REDUCE THE EFFECTIVE CONDUCTIVITY MATRIX
C ====================================================================
C
      DO 280 N=1,NNP
      DO 260 L=2,MBAND
      C=CM(N,L)/CM(N,1)
      I=N+L-1
      IF(NNP-I)260,240,240
240   J=0
      DO 250 K=L,MBAND
      J=J+1
250   CM(I,J)=CM(I,J)-C*CM(N,K)
260   CM(N,L)=C
280   CONTINUE
C
C ====================================================================
C       CALCULATE AND PRINT THE TEMPERATURE DISTRIBUTION
C ====================================================================
C
9     TIME=0.
      ICOUNT=0.
      INT=10
C
```

```
C        1.PRINT THE INITIAL TEMPERATURE DISTRIBUTION
C
      PRINT 133,TIME,(I,T(I),I=1,NNP)
      DO 64 ITIME=1,NDT
C
C        2.FORM THE EFFECTIVE LOAD VECTOR
C
      DO 62 I=1,NNP
      ELV(I)=ELV(I)+Q(I)
      IF(NPCODE(I).EQ.1) GO TO 62
      DO 1062 J=1,MBAND
      K=I+J-1
      IF((K-NNP).GT.0) GO TO 1062
      ELV(I)=ELV(I)+2.*HCM(I,J)*T(K)/DT
      IF((J-1).NE.0) GO TO 1062
      DO 2062 L=2,MBAND
      M=I-L+1
      IF(M.LE.0) GO TO 2062
      ELV(I)=ELV(I)+2.*HCM(M,L)*T(M)/DT
2062  CONTINUE
1062  CONTINUE
62    CONTINUE
C
C        3.FORM THE HEAT GENERATION VECTOR AND ADD IT INTO THE
C          THE EFFECTIVE LOAD VECTOR
C
      DO 58 E=1,NE
      DO 1199 I=1,8
1199  HGV(I)=0.
      DO 1200 I=1,3
      DO 1200 J=1,3
      DO 1200 K=1,3

      DO 1201 II=1,3
      DO 1201 JJ=1,3
1201  JAC(II,JJ)=0.

      DO 1202 II=1,8
      CON1=R(II)*(1+SS(J)*S(II))*(1+TT(K)*T2(II))/8.
      CON2=S(II)*(1+RR(I)*R(II))*(1+TT(K)*T2(II))/8.
      CON3=T2(II)*(1+RR(I)*R(II))*(1+SS(J)*S(II))/8.
      JAC(1,1)=CON1*X1(IX(E,II))+JAC(1,1)
      JAC(1,2)=CON1*X2(IX(E,II))+JAC(1,2)
      JAC(1,3)=CON1*X3(IX(E,II))+JAC(1,3)
      JAC(2,1)=CON2*X1(IX(E,II))+JAC(2,1)
      JAC(2,2)=CON2*X2(IX(E,II))+JAC(2,2)
      JAC(2,3)=CON2*X3(IX(E,II))+JAC(2,3)
      JAC(3,1)=CON3*X1(IX(E,II))+JAC(3,1)
      JAC(3,2)=CON3*X2(IX(E,II))+JAC(3,2)
1202  JAC(3,3)=CON3*X3(IX(E,II))+JAC(3,3)

      DETJ= JAC(1,1)*JAC(2,2)*JAC(3,3)
     &     +JAC(2,1)*JAC(3,2)*JAC(1,3)
     &     +JAC(1,2)*JAC(2,3)*JAC(3,1)
     &     -JAC(3,1)*JAC(2,2)*JAC(1,3)
     &     -JAC(2,1)*JAC(1,2)*JAC(3,3)
     &     -JAC(1,1)*JAC(3,2)*JAC(2,3)

      DO 1203 II=1,8
1203  SHAPE(II)=(1+RR(I)*R(II))*(1+SS(J)*S(II))*(1+TT(K)*T2(II))*DETJ*
     &QGEN(E)/8.

      DO 1204 II=1,8
1204  HGV(II)=HGV(II)+W(I)*W(J)*W(K)*SHAPE(II)
1200  CONTINUE
```

```
      DO 1205 M=1,8
      I=IX(E,M)
      IF(NPCODE(I).EQ.1)GO TO 1205
      ELV(I)=ELV(I)+HGV(M)
1205  CONTINUE
58    CONTINUE
C
C       4.SOLVE FOR NODAL TEMPERATURES
C
      DO 290 N=1,NNP
      DO 485 L=2,MBAND
      I=N+L-1
      IF(NNP-I)270,285,285
285   ELV(I)=ELV(I)-CM(N,L)*ELV(N)
485   CONTINUE
270   ELV(N)=ELV(N)/CM(N,1)
290   CONTINUE
      N=NNP
300   N=N-1
      IF(N)350,500,350
350   DO 400 K=2,MBAND
      L=N+K-1
      IF(NNP-L)400,370,370
370   ELV(N)=ELV(N)-CM(N,K)*ELV(L)
400   CONTINUE
      GO TO 300
500   CONTINUE
      DO 63 I=1,NNP
      T(I)=2.*ELV(I)-T(I)
      ELV(I)=0.
63    CONTINUE
C
C       5.UPDATE THE TIME LEVEL AND PRINT NODAL
C         TEMPERATURES
C
      TIME=TIME+DT
      ICOUNT=ICOUNT+1
      IF(ICOUNT-INT) 64,65,65
65    PRINT 136,TIME,(I,T(I),I=1,NNP)
      ICOUNT=0.
64    CONTINUE
C
C ====================================================================
C       FORMATS......
C ====================================================================
C
133    FORMAT(1H1//15X,60(1H=)/26X,'INPUT TABLE-TEMPERATURE DISTRI'
     &,'BUTION'/15X,60(1H=)/37X,'TIME=',F12.5/15X,60(1H-)/40X,'NODAL PO'
     &,'INT'/27X,'NUMBER',26X,'TEMPERATURES'/15X,60(1H-)/(28X,I5,24X,E15
     &.6))
136    FORMAT(1H1//15X,60(1H=)/26X,'OUTPUT TABLE-TEMPERATURE DISTRI'
     &,'BUTION'/15X,60(1H=)/37X,'TIME=',F12.5/15X,60(1H-)/40X,'NODAL PO'
     &,'INT'/27X,'NUMBER',26X,'TEMPERATURES'/15X,60(1H-)/(28X,I5,24X,E15
     &.6))
138    FORMAT(1H1,35(/),50X,'BANDWIDTH.......',I3)
C
       STOP
       END

       SUBROUTINE BCOO(NNPMAX,MBMAX,X1,X2,X3,QA,HA,ATA,QB,HB,
     &ATB,QC,HC,ATC,Q,CM,KA,KB,KC,E,IX,NEMAX)
       INTEGER E
       REAL JAC(2,2)
       DIMENSION X1(NNPMAX),X2(NNPMAX),X3(NNPMAX),CM(NNPMAX,MBMAX),
     &Q(NNPMAX),QA(NEMAX),HA(NEMAX),ATA(NEMAX),QB(NEMAX),KA(NEMAX),
     &HB(NEMAX),ATB(NEMAX),QC(NEMAX),HC(NEMAX),ATC(NEMAX),KB(NEMAX),
     &KC(NEMAX),IX(NEMAX,8),GK2(8,8),P2(8),P3(8),GK2D(8,8),P2D(8),P3D(8)
     &,VEC(8)
      COMMON/BC/W(3),S(8),R(8),T2(8),SHAPE(8),PROD(8,8),PROD1(8,8),
     &RR(3),SS(3),TT(3)
      COMMON NE,NNP,MBAND,I1,I2,I3,I4,I5,I6,I7,I8

      QAD=QA(E)
      HAD=HA(E)
      ATAD=ATA(E)
      QBD=QB(E)
      HBD=HB(E)
      ATBD=ATB(E)
      QCD=QC(E)
      HCD=HC(E)
      ATCD=ATC(E)
      KAD=KA(E)
      KBD=KB(E)
      KCD=KC(E)

      DO 21 LL=1,8
      P2(LL)=0.
      P3(LL)=0.
      DO 21 LLL=1,8
   21 GK2(LL,LLL)=0.
      I85=0
   22 IF(I85.NE.0) GO TO 23
       QZ=QAD
       H=HAD
       AT=ATAD
       IF(KAD.EQ.1) GO TO 100
       IF(KAD.EQ.2) GO TO 200
       IF(KAD.EQ.3) GO TO 300
       IF(KAD.EQ.4) GO TO 400
       IF(KAD.EQ.5) GO TO 500
       IF(KAD.EQ.6) GO TO 600
   23 IF(I85.NE.1) GO TO 24
       QZ=QBD
       H=HBD
       AT=ATBD
       IF(KBD.EQ.3) GO TO 300
       IF(KBD.EQ.4) GO TO 400
       IF(KBD.EQ.5) GO TO 500
       IF(KBD.EQ.6) GO TO 600
       IF(KBD.EQ.7) GO TO 700
   24 IF(I85.NE.2) GO TO 25
       QZ=QCD
       H=HCD
       AT=ATCD
       IF(KCD.EQ.5) GO TO 500
       IF(KCD.EQ.6) GO TO 600
       IF(KCD.EQ.7) GO TO 700
   25 IF(I85.EQ.3) GO TO 700

C***** FACE I

100   DO 900 I=1,8
      SHAPE(I)=0.
      P2D(I)=0.
      P3D(I)=0.
      DO 900 J=1,8
900   GK2D(I,J)=0.
      DO 1100 I=1,3
      DO 1100 J=1,3

      JAC(1,1)=(-S(1)*(1+TT(I)*T2(1))*X2(IX(E,1))-
     &            S(4)*(1+TT(I)*T2(4))*X2(IX(E,4))-
     &          S(5)*(1+TT(I)*T2(5))*X2(IX(E,5))-
     &          S(8)*(1+TT(I)*T2(8))*X2(IX(E,8)))/4.
      JAC(1,2)=(-S(1)*(1+TT(I)*T2(1))*X3(IX(E,1))-
     &          S(4)*(1+TT(I)*T2(4))*X3(IX(E,4))-
     &          S(5)*(1+TT(I)*T2(5))*X3(IX(E,5))-
     &          S(8)*(1+TT(I)*T2(8))*X3(IX(E,8)))/4.
      JAC(2,1)=(T2(1)*(1-SS(J)*S(1))*X2(IX(E,1))+
     &          T2(4)*(1-SS(J)*S(4))*X2(IX(E,4))+
     &          T2(5)*(1-SS(J)*S(5))*X2(IX(E,5))+
     &          T2(8)*(1-SS(J)*S(8))*X2(IX(E,8)))/4.
      JAC(2,2)=(T2(1)*(1-SS(J)*S(1))*X3(IX(E,1))+
     &          T2(4)*(1-SS(J)*S(4))*X3(IX(E,4))+
     &          T2(5)*(1-SS(J)*S(5))*X3(IX(E,5))+
     &          T2(8)*(1-SS(J)*S(8))*X3(IX(E,8)))/4.

      DETJ= JAC(1,1)*JAC(2,2)-JAC(2,1)*JAC(1,2)

      SHAPE(1)=(1+TT(I)*T2(1))*(1-SS(J)*S(1))/4.
      SHAPE(4)=(1+TT(I)*T2(4))*(1-SS(J)*S(4))/4.
      SHAPE(5)=(1+TT(I)*T2(5))*(1-SS(J)*S(5))/4.
      SHAPE(8)=(1+TT(I)*T2(8))*(1-SS(J)*S(8))/4.

      DO 1101 II=1,8
      VEC(II)=SHAPE(II)*DETJ
      DO 1101 JJ=1,8
1101  PROD(II,JJ)=SHAPE(II)*SHAPE(JJ)*H*DETJ
      DO 1102 II=1,8
      P2D(II)=P2D(II)+W(I)*W(J)*VEC(II)*QZ
      P3D(II)=P3D(II)+W(I)*W(J)*VEC(II)*H*AT
      DO 1102 JJ=1,8
1102  GK2D(II,JJ)=GK2D(II,JJ)+W(I)*W(J)*PROD(II,JJ)
1100  CONTINUE

      DO 1103 I=1,8
      P2(I)=P2(I)+P2D(I)
      P3(I)=P3(I)+P3D(I)
      DO 1103 J=1,8
1103  GK2(I,J)=GK2(I,J)+GK2D(I,J)

      I85=I85+1
      GO TO 22

C ***  FACE II

200   DO 901 I=1,8
      SHAPE(I)=0.
      P2D(I)=0.
      P3D(I)=0.
      DO 901 J=1,8
901   GK2D(I,J)=0.
      DO 1200 I=1,3
      DO 1200 J=1,3

      JAC(1,1)=(S(2)*(1+TT(I)*T2(2))*X2(IX(E,2))+
     &          S(3)*(1+TT(I)*T2(3))*X2(IX(E,3))+
     &          S(6)*(1+TT(I)*T2(6))*X2(IX(E,6))+
     &          S(7)*(1+TT(I)*T2(7))*X2(IX(E,7)))/4.
      JAC(1,2)=(S(2)*(1+TT(I)*T2(2))*X3(IX(E,2))+
     &          S(3)*(1+TT(I)*T2(3))*X3(IX(E,3))+
     &          S(6)*(1+TT(I)*T2(6))*X3(IX(E,6))+
     &          S(7)*(1+TT(I)*T2(7))*X3(IX(E,7)))/4.
      JAC(2,1)=(T2(2)*(1+SS(J)*S(2))*X2(IX(E,2))+
     &          T2(3)*(1+SS(J)*S(3))*X2(IX(E,3))+
     &          T2(6)*(1+SS(J)*S(6))*X2(IX(E,6))+
     &          T2(7)*(1+SS(J)*S(7))*X2(IX(E,7)))/4.
```

```
       JAC(2,2)=(T2(2)*(1+SS(J)*S(2))*X3(IX(E,2))+
     &          T2(3)*(1+SS(J)*S(3))*X3(IX(E,3))+
     &          T2(6)*(1+SS(J)*S(6))*X3(IX(E,6))+
     &          T2(7)*(1+SS(J)*S(7))*X3(IX(E,7)))/4.

      DETJ= JAC(1,1)*JAC(2,2)-JAC(2,1)*JAC(1,2)

      SHAPE(2)=(1+TT(I)*T2(2))*(1+SS(J)*S(2))/4.
      SHAPE(3)=(1+TT(I)*T2(3))*(1+SS(J)*S(3))/4.
      SHAPE(6)=(1+TT(I)*T2(6))*(1+SS(J)*S(6))/4.
      SHAPE(7)=(1+TT(I)*T2(7))*(1+SS(J)*S(7))/4.

      DO 1201 II=1,8
      VEC(II)=SHAPE(II)*DETJ
      DO 1201 JJ=1,8
 1201 PROD(II,JJ)=SHAPE(II)*SHAPE(JJ)*H*DETJ
      DO 1202 II=1,8
      P2D(II)=P2D(II)+W(I)*W(J)*VEC(II)*QZ
      P3D(II)=P3D(II)+W(I)*W(J)*VEC(II)*H*AT
      DO 1202 JJ=1,8
 1202 GK2D(II,JJ)=GK2D(II,JJ)+W(I)*W(J)*PROD(II,JJ)
 1200 CONTINUE

      DO 1203 I=1,8
      P2(I)=P2(I)+P2D(I)
      P3(I)=P3(I)+P3D(I)
      DO 1203 J=1,8
 1203 GK2(I,J)=GK2(I,J)+GK2D(I,J)

      I85=I85+1
      GO TO 22

C ***  FACE III

  300 DO 902 I=1,8
      SHAPE(I)=0.
      P2D(I)=0.
      P3D(I)=0.
      DO 902 J=1,8
  902 GK2D(I,J)=0.
      DO 1300 I=1,3
      DO 1300 J=1,3

       JAC(1,1)=(-R(1)*(1+SS(J)*S(1))*X3(IX(E,1))-
     &          R(2)*(1+SS(J)*S(2))*X3(IX(E,2))-
     &          R(3)*(1+SS(J)*S(3))*X3(IX(E,3))-
     &          R(4)*(1+SS(J)*S(4))*X3(IX(E,4)))/4.
       JAC(1,2)=(-R(1)*(1+SS(J)*S(1))*X1(IX(E,1))-
     &          R(2)*(1+SS(J)*S(2))*X1(IX(E,2))-
     &          R(3)*(1+SS(J)*S(3))*X1(IX(E,3))-
     &          R(4)*(1+SS(J)*S(4))*X1(IX(E,4)))/4.
       JAC(2,1)=(S(1)*(1-RR(I)*R(1))*X3(IX(E,1))+
     &          S(2)*(1-RR(I)*R(2))*X3(IX(E,2))+
     &          S(3)*(1-RR(I)*R(3))*X3(IX(E,3))+
     &          S(4)*(1-RR(I)*R(4))*X3(IX(E,4)))/4.
       JAC(2,2)=(S(1)*(1-RR(I)*R(1))*X1(IX(E,1))+
     &          S(2)*(1-RR(I)*R(2))*X1(IX(E,2))+
     &          S(3)*(1-RR(I)*R(3))*X1(IX(E,3))+
     &          S(4)*(1-RR(I)*R(4))*X1(IX(E,4)))/4.

      DETJ= JAC(1,1)*JAC(2,2)-JAC(2,1)*JAC(1,2)

      SHAPE(1)=(1-RR(I)*R(1))*(1+SS(J)*S(1))/4.
      SHAPE(2)=(1-RR(I)*R(2))*(1+SS(J)*S(2))/4.
      SHAPE(3)=(1-RR(I)*R(3))*(1+SS(J)*S(3))/4.
      SHAPE(4)=(1-RR(I)*R(4))*(1+SS(J)*S(4))/4.

      DO 1301 II=1,8
      VEC(II)=SHAPE(II)*DETJ
      DO 1301 JJ=1,8
 1301 PROD(II,JJ)=SHAPE(II)*SHAPE(JJ)*H*DETJ
      DO 1302 II=1,8
      P2D(II)=P2D(II)+W(I)*W(J)*VEC(II)*QZ
      P3D(II)=P3D(II)+W(I)*W(J)*VEC(II)*H*AT
      DO 1302 JJ=1,8
 1302 GK2D(II,JJ)=GK2D(II,JJ)+W(I)*W(J)*PROD(II,JJ)
 1300 CONTINUE

      DO 1303 I=1,8
      P2(I)=P2(I)+P2D(I)
      P3(I)=P3(I)+P3D(I)
      DO 1303 J=1,8
 1303 GK2(I,J)=GK2(I,J)+GK2D(I,J)

      I85=I85+1
      GO TO 22

C***   FACE IV

 400  DO 903 I=1,8
      SHAPE(I)=0.
      P2D(I)=0.
      P3D(1)=0.
      DO 903 J=1,8
 903  GK2D(I,J)=0.
      DO 1400 I=1,3
      DO 1400 J=1,3

      JAC(1,1)=(R(5)*(1+SS(J)*S(5))*X3(IX(E,5))+
     &          R(6)*(1+SS(J)*S(6))*X3(IX(E,6))+
     &          R(7)*(1+SS(J)*S(7))*X3(IX(E,7))+
     &          R(8)*(1+SS(J)*S(8))*X3(IX(E,8)))/4.
      JAC(1,2)=(R(5)*(1+SS(J)*S(5))*X1(IX(E,5))+
     &          R(6)*(1+SS(J)*S(6))*X1(IX(E,6))+
     &          R(7)*(1+SS(J)*S(7))*X1(IX(E,7))+
     &          R(8)*(1+SS(J)*S(8))*X1(IX(E,8)))/4.
      JAC(2,1)=(S(5)*(1+RR(I)*R(5))*X3(IX(E,5))+
     &          S(6)*(1+RR(I)*R(6))*X3(IX(E,6))+
     &          S(7)*(1+RR(I)*R(7))*X3(IX(E,7))+
     &          S(8)*(1+RR(I)*R(8))*X3(IX(E,8)))/4.
      JAC(2,2)=(S(5)*(1+RR(I)*R(5))*X1(IX(E,5))+
     &          S(6)*(1+RR(I)*R(6))*X1(IX(E,6))+
     &          S(7)*(1+RR(I)*R(7))*X1(IX(E,7))+
     &          S(8)*(1+RR(I)*R(8))*X1(IX(E,8)))/4.

      DETJ= JAC(1,1)*JAC(2,2)-JAC(2,1)*JAC(1,2)

      SHAPE(5)=(1+RR(I)*R(5))*(1+SS(J)*S(5))/4.
      SHAPE(6)=(1+RR(I)*R(6))*(1+SS(J)*S(6))/4.
      SHAPE(7)=(1+RR(I)*R(7))*(1+SS(J)*S(7))/4.
      SHAPE(8)=(1+RR(I)*R(8))*(1+SS(J)*S(8))/4.

      DO 1401 II=1,8
      VEC(II)=SHAPE(II)*DETJ
      DO 1401 JJ=1,8
 1401 PROD(II,JJ)=SHAPE(II)*SHAPE(JJ)*H*DETJ
      DO 1402 II=1,8
      P2D(II)=P2D(II)+W(I)*W(J)*VEC(II)*QZ
      P3D(II)=P3D(II)+W(I)*W(J)*VEC(II)*H*AT
      DO 1402 JJ=1,8
1402  GK2D(II,JJ)=GK2D(II,JJ)+W(I)*W(J)*PROD(II,JJ)
1400  CONTINUE

      DO 1403 I=1,8
      P2(I)=P2(I)+P2D(I)
      P3(I)=P3(I)+P3D(I)
      DO 1403 J=1,8
1403  GK2(I,J)=GK2(I,J)+GK2D(I,J)

      I85=I85+1
      GO TO 22

C ***  FACE V

500   DO 904 I=1,8
      SHAPE(I)=0.
      P2D(I)=0.
      P3D(I)=0.
      DO 904 J=1,8
904   GK2D(I,J)=0.
      DO 1500 I=1,3
      DO 1500 J=1,3

      JAC(1,1)=(T2(3)*(1+RR(I)*R(3))*X1(IX(E,3))+
     &          T2(4)*(1+RR(I)*R(4))*X1(IX(E,4))+
     &          T2(7)*(1+RR(I)*R(7))*X1(IX(E,7))+
     &          T2(8)*(1+RR(I)*R(8))*X1(IX(E,8)))/4.
      JAC(1,2)=(T2(3)*(1+RR(I)*R(3))*X2(IX(E,3))+
     &          T2(4)*(1+RR(I)*R(4))*X2(IX(E,4))+
     &          T2(7)*(1+RR(I)*R(7))*X2(IX(E,7))+
     &          T2(8)*(1+RR(I)*R(8))*X2(IX(E,8)))/4.
      JAC(2,1)=(R(3)*(1+TT(J)*T2(3))*X1(IX(E,3))+
     &          R(4)*(1+TT(J)*T2(4))*X1(IX(E,4))+
     &          R(7)*(1+TT(J)*T2(7))*X1(IX(E,7))+
     &          R(8)*(1+TT(J)*T2(8))*X1(IX(E,8)))/4.
      JAC(2,2)=(R(3)*(1+TT(J)*T2(3))*X2(IX(E,3))+
     &          R(4)*(1+TT(J)*T2(4))*X2(IX(E,4))+
     &          R(7)*(1+TT(J)*T2(7))*X2(IX(E,7))+
     &          R(8)*(1+TT(J)*T2(8))*X2(IX(E,8)))/4.

      DETJ= JAC(1,1)*JAC(2,2)-JAC(2,1)*JAC(1,2)

      SHAPE(3)=(1+RR(I)*R(3))*(1+TT(J)*T2(3))/4.
      SHAPE(4)=(1+RR(I)*R(4))*(1+TT(J)*T2(4))/4.
      SHAPE(7)=(1+RR(I)*R(7))*(1+TT(J)*T2(7))/4.
      SHAPE(8)=(1+RR(I)*R(8))*(1+TT(J)*T2(8))/4.

      DO 1501 II=1,8
      VEC(II)=SHAPE(II)*DETJ
      DO 1501 JJ=1,8
1501  PROD(II,JJ)=SHAPE(II)*SHAPE(JJ)*H*DETJ
      DO 1502 II=1,8
      P2D(II)=P2D(II)+W(I)*W(J)*VEC(II)*QZ
      P3D(II)=P3D(II)+W(I)*W(J)*VEC(II)*H*AT
      DO 1502 JJ=1,8
1502  GK2D(II,JJ)=GK2D(II,JJ)+W(I)*W(J)*PROD(II,JJ)
1500  CONTINUE

      DO 1503 I=1,8
      P2(I)=P2(I)+P2D(I)
      P3(I)=P3(I)+P3D(I)
      DO 1503 J=1,8
1503  GK2(I,J)=GK2(I,J)+GK2D(I,J)
```

```
      I85=I85+1
      GO TO 22

C ***  FACE VI

600   DO 905 I=1,8
      SHAPE(I)=0.
      P2D(I)=0.
      P3D(I)=0.
      DO 905 J=1,8
905   GK2D(I,J)=0.
      DO 1600 I=1,3
      DO 1600 J=1,3

      JAC(1,1)=(T2(1)*(1-RR(I)*R(1))*X1(IX(E,1))+
     &             T2(2)*(1-RR(I)*R(2))*X1(IX(E,2))+
     &             T2(5)*(1-RR(I)*R(5))*X1(IX(E,5))+
     &             T2(6)*(1-RR(I)*R(6))*X1(IX(E,6)))/4.
      JAC(1,2)=(T2(1)*(1-RR(I)*R(1))*X2(IX(E,1))+
     &             T2(2)*(1-RR(I)*R(2))*X2(IX(E,2))+
     &             T2(5)*(1-RR(I)*R(5))*X2(IX(E,5))+
     &             T2(6)*(1-RR(I)*R(6))*X2(IX(E,6)))/4.
      JAC(2,1)=(-R(1)*(1+TT(J)*T2(1))*X1(IX(E,1))-
     &             R(2)*(1+TT(J)*T2(2))*X1(IX(E,2))-
     &             R(5)*(1+TT(J)*T2(5))*X1(IX(E,5))-
     &             R(6)*(1+TT(J)*T2(6))*X1(IX(E,6)))/4.
      JAC(2,2)=(-R(1)*(1+TT(J)*T2(1))*X2(IX(E,1))-
     &             R(2)*(1+TT(J)*T2(2))*X2(IX(E,2))-
     &             R(5)*(1+TT(J)*T2(5))*X2(IX(E,5))-
     &             R(6)*(1+TT(J)*T2(6))*X2(IX(E,6)))/4.


      DETJ=  JAC(1,1)*JAC(2,2)-JAC(2,1)*JAC(1,2)

      SHAPE(1)=(1-RR(I)*R(1))*(1+TT(J)*T2(1))/4.
      SHAPE(2)=(1-RR(I)*R(2))*(1+TT(J)*T2(2))/4.
      SHAPE(5)=(1-RR(I)*R(5))*(1+TT(J)*T2(5))/4.
      SHAPE(6)=(1-RR(I)*R(6))*(1+TT(J)*T2(6))/4.
      DO 1601 II=1,8
      VEC(II)=SHAPE(II)*DETJ
      DO 1601 JJ=1,8
1601  PROD(II,JJ)=SHAPE(II)*SHAPE(JJ)*H*DETJ
      DO 1602 II=1,8
      P2D(II)=P2D(II)+W(I)*W(J)*VEC(II)*QZ
      P3D(II)=P3D(II)+W(I)*W(J)*VEC(II)*H*AT
      DO 1602 JJ=1,8
1602  GK2D(II,JJ)=GK2D(II,JJ)+W(I)*W(J)*PROD(II,JJ)
1600  CONTINUE

      DO 1603 I=1,8
      P2(I)=P2(I)+P2D(I)
      P3(I)=P3(I)+P3D(I)
      DO 1603 J=1,8
1603  GK2(I,J)=GK2(I,J)+GK2D(I,J)

      I85=I85+1
      GO TO 22

700   DO 4 LL=1,8
      I=IX(E,LL)
      Q(I)=Q(I)-P2(LL)+P3(LL)
      DO 4 M=1,8
      K=IX(E,M)
      J=K-I+1
      IF(MBAND.GE.J) GO TO 2
      MBAND=J
2     IF(J.LE.0) GO TO 4
      CM(I,J)=CM(I,J)+GK2(LL,M)
4     CONTINUE

      RETURN
      END


      SUBROUTINE MASS00(NEMAX,NNPMAX,MBMAX,N,IX,ECM,EHCM,CM,HCM)
      DIMENSION IX(NEMAX,8),ECM(8,8),EHCM(8,8),CM(NNPMAX,MBMAX),
     &HCM(NNPMAX,MBMAX)
      COMMON NE,NNP,MBAND,I1,I2,I3,I4,I5,I6,I7,I8

      DO 4 LL=1,8
      I=IX(N,LL)
      DO 4 M=1,8
      K=IX(N,M)
      J=K-I+1
      IF(MBAND.GE.J) GO TO 2
      MBAND=J
2     IF(J.LE.0) GO TO 4
      CM(I,J)=CM(I,J)+ECM(LL,M)
      HCM(I,J)=HCM(I,J)+EHCM(LL,M)
4     CONTINUE

      RETURN
      END


      SUBROUTINE MDF00(NNPMAX,MBMAX,NPCODE,T,Q,CM)
      DIMENSION NPCODE(NNPMAX),T(NNPMAX),Q(NNPMAX),
     &CM(NNPMAX,MBMAX)
      COMMON NE,NNP,MBAND,I1,I2,I3,I4,I5,I6,I7,I8

      DO 4900 I=1,NNP
      IF(NPCODE(I).EQ.0)GO TO 4900
44    DO 49 J=2,MBAND
      K=I-J+1
      IF(K)46,46,45
45    Q(K)=Q(K)-CM(K,J)*T(I)
      CM(K,J)=0.
46    L=I+J-1
      IF(NNP-L)48,47,47
47    Q(L)=Q(L)-CM(I,J)*T(I)
48    CM(I,J)=0.
49    CONTINUE
      CM(I,1)=1.0
      Q(I)=T(I)
4900  CONTINUE

      RETURN
      END
14.49.56.UCLP, BU, P03    ,        0.948KLNS.
```

REFERENCES

1. Hollman, Fundamentals of Heat Transfer, McGraw Hill, 1979.
2. Cook, R. D., Concepts and Applications of Finite Element Analysis, John Wiley and Sons Inc., 1974
3. Rao, S. S., The Finite Element Method in Engineering, Pergamon Press, 1982.
4. Zienkiewicz, O. C., The Finite Element Method in Engineerin Science, McGraw Hill, 1971.
5. Kalenderoğlu, V., "Finite Element Analpsis of Radiative Transport in Gray Participating Media", Ph.D. Thesis, Boğaziçi Üniversitesi, 1980
6. Zienkiewicz, O.C. and Parekh, C.J., "Transient Field Problems: Two Dimensional and Three Dimensional Analysis by Isoparametric Finite Elements", Int. J. Num. Methods Engr., V.2, pp.61-71, 1970.
7. Gray, W.H. and Schnurr A., "A Comparison of the Finite Element and Finite Difference Methods for the Analysis of Steady Two Dimensional Heat Conduction Problems", Computer Methods in Applied Mechanics and Engineering, V.6, pp.243-245, 1975.
8. Ralston, A. and Rabinowitz, P., A First Course in Numerical Analysis, 2nd edition, McGraw Hill, 1978.