FINITE ELEMENT ANALYSIS OF HEAT CONDUCTION IN THE ABSORBER PLATE OF A SOLAR COLLECTOR

by

Faramarz Mossayebi

Submitted in Partial Fulfillment of the Requirments

for the Degree of

Master of Science in Engineering

in the

Mechanical Engineering

Adviser — 3/13/87 — Date

Sally M. Hotchkiss — March 27, 1987
Dean of the Graduate School — Date

YOUNGSTOWN STATE UNIVERSITY

March, 1987

ABSTRACT

FINITE ELEMENT ANALYSIS OF HEAT CONDUCTION IN THE ABSORBER PLATE OF A SOLAR COLLECTOR

Faramarz Mossayebi

Master of Science in Engineering

Youngstown State University, 1987

The Finite Element Method (FEM) is used to model the absorber plate of a flat plate solar collector. The Finite Element equations for steady-state temperature distribution are then derived by the Galerkin approach. Based on these formulations, a computer program is written in FORTRAN language to obtain the temperature field. The program contains all the necessary algorithms to handle two-dimensional Laplace and Poisson's equations. Only some basic input data is essential to run the program.

The Finite Element solution is compared to the analytical solution and/or the solution by the Finite Difference Method, where possible. The results show good agreement.

ACKNOWLEDGEMENTS

The author wishes to express his thanks to his adviser, Dr. Hyun M. Kim of the Mechanical Engineering Department for his continuous assistance throughout the preparation and completion of the thesis. Thanks are also due to Dr. Frank A. D'Isa and Dr. Thomas I. Elias for their cooperation and patience in reading this thesis.

TABLE OF CONTENTS

LIST OF SYMBOLS

SYMBOL	DEFINITION	UNIT
A	Area	m^2
a_1, a_2, a_3	Coefficients of shape functions	
H	Monthly average, daily total horizontal radiation	$KW.hr/m^2.day$
h_c	Average convection heat transfer coefficient	$W/m^2.K$
I_o	Solar constant	W/m^2
K	Thermal conductivity	W/m.K
L_1, L_2, L_3	Natural coordinate system for triangle	
l_x, l_y	Direction cosines of the outward drawn normal to the boundary	
N_i, N_j, N_k	Shape functions	
q	Heat generation	W/m^3
S	Net energy absorbed	W/m^2
T	Temperature	$^\circ C$
T_i	Inlet temperature	$^\circ C$
T_o	Outlet temperature	$^\circ C$
T_∞	Ambient temperature	$^\circ C$
t	Thickness	m
W_i	Weighting functions	
X_i, Y_i	Nodal coordinates	m
α	Absorptance	
ε	Residual	
Ω	Solution domain	
Γ	Surface that bounds the solution domain	
τ	Transmittance	

LIST OF FIGURES

LIST OF TABLES

TABLE PAGE

CHAPTER I

INTRODUCTION

Flat plate solar collectors are the most common unsophisticated device for harnessing solar energy at low cost. A typical collector consists of an absorber plate, piping for coolants, transparent cover glazing, thermal insulation and a casing. The heart of the system is the absorber plate. It is desired to determine the steady-state temperature distribution on the absorber plate. This type of problem belongs to one of the classical groups of problems in heat transfer analysis. Yet, an accurate prediction of temperature distribution on the absorber plate is quite difficult due to nonuniform boundary conditions and other uncertain conditions. The main objective of this thesis is focused on numerical analysis rather than mathematical modeling of an absorber plate. In other words, the methodology and procedure of finite element formulation, and its implementation to numerical computing, have been dealt with in detail.

Finite element method is an approximate method for solving differential equations of boundary and/or initial value problems. The name "Finite Element Method" first appeared in 1960, when it was used in a paper on a plane elasticity problem by Clough [1]. However, Turner, et al.[2]

were the pioneers with their paper on solution of plane stress problems by means of triangular elements, which was published in 1956. In 1965, Zienkviewicz and Cheung [3] reported that the method is applicable to all field problems which can be cast into variational form.

The range of applications for the finite element method was greatly enlarged when Szabo and Lee [4] and Zienkiewicz [5] showed that the finite element equations could be derived by using a weighted residual procedure.

The finite element method reduces a continuum problem, which theoretically has infinite number of unknowns, to one with finite number of unknowns by dividing the solution region to a finite number of subdomains called "finite elements". The field variable within each element is then expressed in terms of some assumed approximate functions. These approximate functions (also called interpolation functions) are defined in terms of the value of the field variables at "nodes". The nodes usually lie on the element boundaries where adjacent elements are connected. The finite element equations which govern all isolated elements are then derived. Finally these elements are assembled to form a global system of equations. After incorporation of the boundary conditions, the nodal value of the field variable is determined from the global system of equations.

In the process of solving the global system of equations, matrix technique combined with digital computer is generally employed. A complete compact computer program which could handle the general heat conduction problem with various boundary conditions was written and applied to the problem.

CHAPTER II

ABSORBER PLATE DESCRIPTION AND MATHEMATICAL MODEL

When a flat plate solar collector absorbs solar radiation the temperature of the absorber plate gradually rises until it is high enough above ambient such that the rate of heat loss from the plate to the ambient just balances the rate of heat gain from absorbtion of solar rays. Practically a hot metal sheet is not of any value by itself. In a solar collector the collected heat is carried off by movement of a fluid, either as air blown over the plate or a fluid flowing through tubes attached to the plate. A typical liquid-cooled flat plate collector is illustrated in Fig. 1. Assuming that the spacings of the tubes attached to the absorber plate are equal, only one

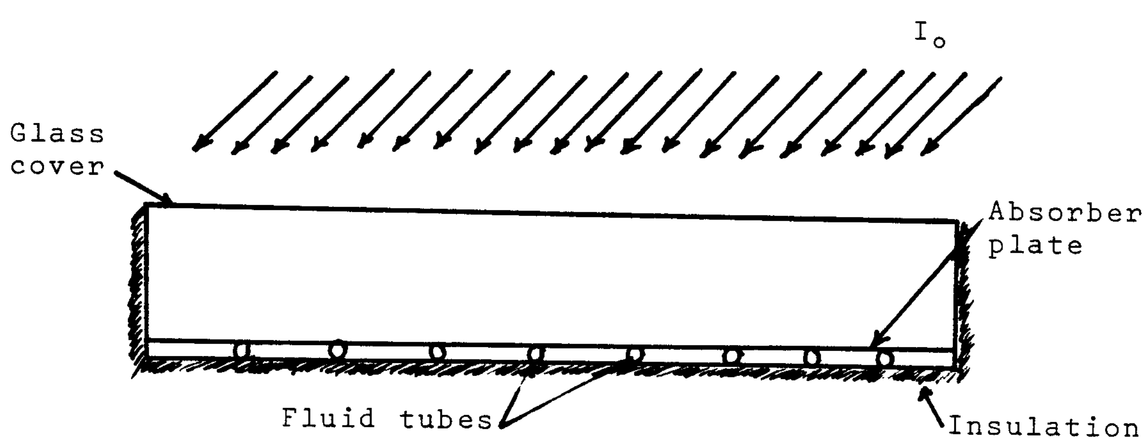


Fig. 1. Liquid-cooled flat plate collector

section of the absorber plate is considered for analysis as shown in Fig. 2. Other major assumptions made are as follows [6]:

1. The absorber plate is made of aluminum with constant properties, and receives constant solar flux.
2. There is no convective and conduction heat loss in the vertical direction.
3. The inlet (T_i) and outlet (T_o) temperatures of fluid are constant.
4. The temperature variation of the fluid from T_i to T_o is linear.

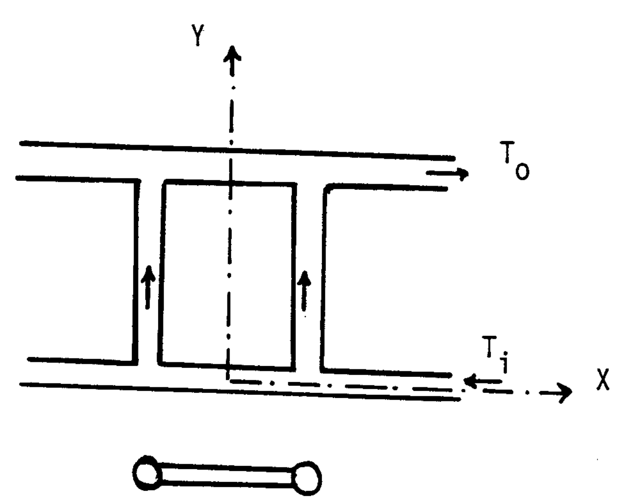


Fig. 2. A section of absorber plate

Considering the symmetry of the absorber plate with respect to y axis (Fig. 2) and the above assumptions, the mathematical model of the plate can be constructed as shown in Figure 3. From Fourier's law of heat conduction, the governing differential equation of the model can be written as

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = 0 \qquad (1)$$

Analysis of three dimensional problems by the finite element method (and, in general, any numerical method) requires extensive programming efforts and computational capabilities. However, due to the fact that the thickness of the absorber plate is small, temperature gradient in the z direction is assumed to be negligible. Hence, the lumping technique reduces the problem to a two dimensional problem (Fig. 3) as follows:

$$\int_0^t \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) dz = - \int_0^t \left(\frac{\partial^2 T}{\partial z^2} \right) dz \qquad (2)$$

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = - \frac{q}{K} \quad , \qquad (3)$$

where q is calculated in Appendix D. The boundary conditions for the above equation are as follows

$$\frac{\partial T}{\partial x} (0,y) = 0 \qquad 0 < y < 0.6 \text{ m}$$

$$- K \frac{\partial T}{\partial x} = h_c (T - T_\infty) \qquad \text{at } x = 0.15 \text{ m}$$

$$T (x,0) = T_i = 50^\circ c$$

$$T (x,0.6) = T_o = 53^\circ c$$

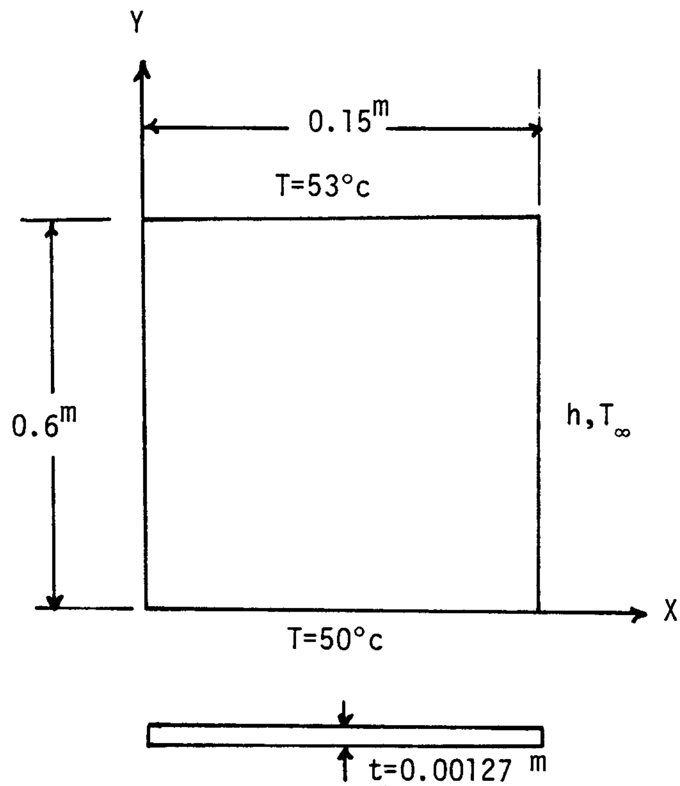


Fig. 3. Mathematical model of the absorber plate

CHAPTER III

FORMULATION OF FINITE ELEMENT EQUATIONS FOR HEAT CONDUCTION IN THE ABSORBER PLATE

In the finite element method, there are basically four different approaches in the formulation of element properties: direct approach, energy balance approach, variational approach, and weighted residual approach [7]. The most versatile approach for a continuum problem is the weighted residual approach which is adopted in this analysis. A brief discussion of the weighted residual approach is presented in Appendix C.

The Finite Element Formulation

Assuming that the solution domain is divided into n triangular elements, the overall finite element equations can be obtained by deriving the equations for each element and assembling them.

The element equations are derived by assuming a linear variation of T in each element, as it is discussed in Appendix A. Therefore

$$T'(x,y) = N(x,y)\ T^{e} \qquad (5)$$

where T^{e} and N(x,y) are given by equations (68) and (69) respectively as

$$N(x,y)=[N_i \;\; N_j \;\; N_k]=1/2A \begin{bmatrix} a_i+b_ix+c_iy \\ a_j+b_jx+c_jy \\ a_k+b_kx+c_ky \end{bmatrix}^T \tag{6}$$

$$T^e= \begin{Bmatrix} T_i \\ T_j \\ T_k \end{Bmatrix} \tag{7}$$

Substituting eq. (5) into the equation (3) yields

$$K \frac{\partial^2 T'}{\partial x^2} + K \frac{\partial^2 T'}{\partial y^2} + q = \varepsilon \neq 0 \quad . \tag{8}$$

Applying the weighted residual(Galerkin) principle,

$$\int_\Omega N_i \,\varepsilon \, d\,\Omega = \int_\Omega N_i \left(K \frac{\partial^2 T'}{\partial x^2} + K \frac{\partial^2 T'}{\partial y^2} + q \right) d\,\Omega = 0 \;, \tag{9}$$

where Ω is the domain. This equation can be transformed into a first degree equation by noting that

$$K \frac{\partial}{\partial x} \left(N_i \frac{\partial T'}{\partial x}\right) = K\, N_i \frac{\partial^2 T'}{\partial x^2} + K \frac{\partial N_i}{\partial x} \frac{\partial T'}{\partial x} \tag{10}$$

or

$$K\, N_i \frac{\partial^2 T'}{\partial x^2} = K \frac{\partial}{\partial x} \left(N_i \frac{\partial T'}{\partial x}\right) - K \frac{\partial N_i}{\partial x} \frac{\partial T'}{\partial x} \tag{11}$$

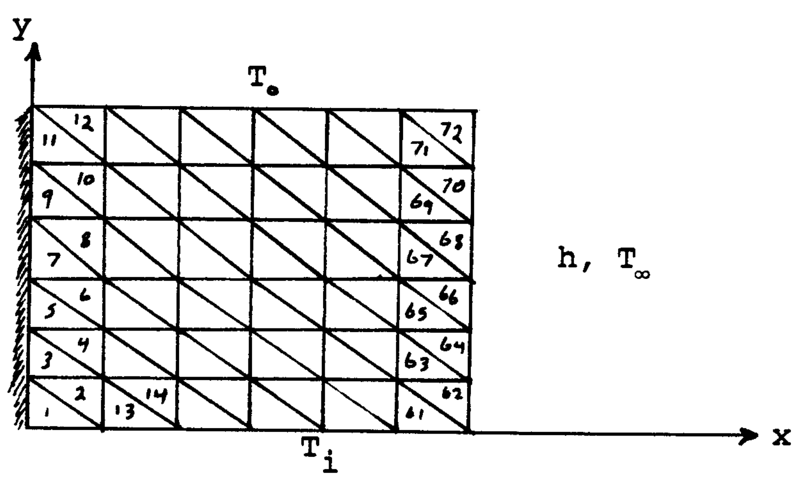


Fig. 4. Finite element modeling of the absorber plate

Similarly,

$$K N_i \frac{\partial^2 T'}{\partial y^2} = K \frac{\partial}{\partial y} (N_i \frac{\partial T'}{\partial y}) - K \frac{\partial N_i}{\partial y} \frac{\partial T'}{\partial y} \quad . \qquad (12)$$

Substituting equations (11) and (12) into equation (9) yields

$$-\int_\Omega N_i (\frac{\partial N_i}{\partial x} \frac{\partial T'}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial T'}{\partial y}) d\,\Omega + \int_\Omega K\{\frac{\partial}{\partial x} (N_i \frac{\partial T'}{\partial x}) + \frac{\partial}{\partial y} (N_i \frac{\partial T'}{\partial y})\} d\,\Omega + \int_\Omega N_i\, q\, d\Omega = 0 \quad . \qquad (13)$$

Applying Gauss theorem to the second integral of equation (13) yields

$$-\int_\Omega N_i (\frac{\partial N_i}{\partial x} \frac{\partial T'}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial T'}{\partial y}) d\,\Omega + \int_\Gamma K N_i (\frac{\partial T'}{\partial x} \ell_x + \frac{\partial T'}{\partial y} \ell_y) d\Gamma + \int_\Omega N_i\, q\, d\Omega = 0 \quad , \qquad (14)$$

where Γ is the surface which bounds the region Ω, and ℓ_x and ℓ_y are direction cosines of the outward drawn normal to the boundary. The surface can have a combination of two different kinds of boundaries, convection and prescribed temperature. For instance, the element number 72 in the mesh of Fig. 4 is subjected to convection and a prescribed temperature. This element is redrawn as shown in Fig. 5,

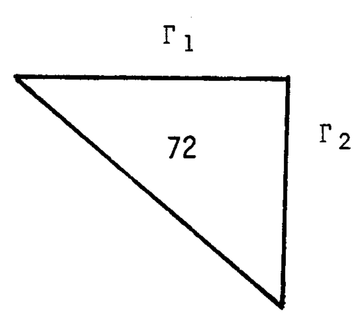


Fig. 5. Boundary conditions for a triangular element

where Γ_1 and Γ_2 denote prescribed temperature and convection boundaries, respectively. The surface integral of (14) can be written as

$$\int_{\Gamma} K\, N_i \left(\frac{\partial T'}{\partial x}\ell_x + \frac{\partial T'}{\partial y}\ell_y\right) d\Gamma = \int_{\Gamma_1} K\, N_i \left(\frac{\partial T'}{\partial x}\ell_x + \frac{\partial T'}{\partial y}\ell_y\right) d\Gamma_1 + \int_{\Gamma_2} K\, N_i \left(\frac{\partial T'}{\partial x}\ell_x + \frac{\partial T'}{\partial y}\ell_y\right) d\Gamma_2 \quad . \tag{15}$$

Because the temperature over Γ_1 is prescribed and constant, the surface integral over Γ_1 is zero. Since

$$K\left(\frac{\partial T'}{\partial x}\ell_x + \frac{\partial T'}{\partial y}\ell_y\right) = -h\,(T - T_\infty) \quad ,$$

it follows that

$$\int_{\Gamma_2} K\, N_i \left(\frac{\partial T'}{\partial y}\ell_x + \frac{\partial T'}{\partial y}\ell_y\right) d\Gamma_2 = -\int_{\Gamma_2} h\, N_i (T' - T_\infty)\, d\Gamma_2 . \tag{16}$$

Substituting equation (16) into equation (14) yields

$$-\int_{\Omega} K N_i \left(\frac{\partial N_i}{\partial x}\frac{\partial T'}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial T'}{\partial y}\right) d\Omega + \int_{\Gamma_2} h\, N_i (T_\infty - T)\, d\Gamma_2 + \int_{\Omega} N_i\, q\, d\Omega = 0 \quad . \tag{17}$$

Expanding the second integral and rearranging yields

$$-\int_{\Omega} K N_i \left(\frac{\partial N_i}{\partial x}\frac{\partial T'}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial T'}{\partial y}\right) d\Omega - \int_{\Gamma_2} h\, N_i\, T\, d\Gamma_2 + \int_{\Gamma_2} h\, N_i\, T_\infty\, d\Gamma_2 + \int_{\Omega} N_i\, q\, d\Omega = 0 \quad . \tag{18}$$

The partial derivatives of T' with respect to x and y are obtained from equation (5) as

$$\frac{\partial T'}{\partial x} = \frac{\partial}{\partial x}(N_i\, T^e) = \frac{\partial N_i}{\partial x} T^e$$
$$\frac{\partial T'}{\partial y} = \frac{\partial}{\partial y}(N_i\, T^e) = \frac{\partial N_i}{\partial y} T^e \quad . \tag{19}$$

Substituting equation (19) into equation (17) and writing it in terms of matrices yields

$$[\ K\ ]\ T'^{e} + [\ K_h\ ]\ T'^{e} - f_q - f_h = 0 \quad , \qquad (20)$$

where

$$[\ K\ ] = \int_{\Omega} K\ [B]^T\ [B]\ d\Omega\ ,$$

$$[\ K_h\ ] = \int_{\Gamma 2} h\ [N]^T\ [N]\ d\Gamma_2 ,$$

$$f_q = \int_{\Omega} q[N]^T\ d\ \Omega ,$$

$$f_h = \int_{\Gamma 2} h\ T\ \ [N]^T\ d\Gamma 2 ,$$

and

$$[B] = \begin{bmatrix} \dfrac{\partial N_1}{\partial x} & \dfrac{\partial N_2}{\partial x} & \dfrac{\partial N_3}{\partial x} \\ \dfrac{\partial N_1}{\partial y} & \dfrac{\partial N_2}{\partial y} & \dfrac{\partial N_3}{\partial y} \end{bmatrix} .$$

Construction of element characteristic matrices

The element conduction and force matrices are calculated from equation (20). Matrix [K] is calculated for each element while $[K_h]$ is calculated for those elements which are subjected to convection heat loss. The vectors f_q and f_h are calculated for the elements which are subjected to internal heat generation and convection, respectively.

Assuming that the solar absorber plate is discretized by 9 nodes and 8 elements as shown in Fig. 6, the matrix K must be calculated for all eight elements, while K_h is calculated for element numbers 6 and 8 which are subjected to convection heat loss. In order to determine matrix K, the matrix B must be evaluated. However, to calculate B, the partial derivatives of shape functions N_i

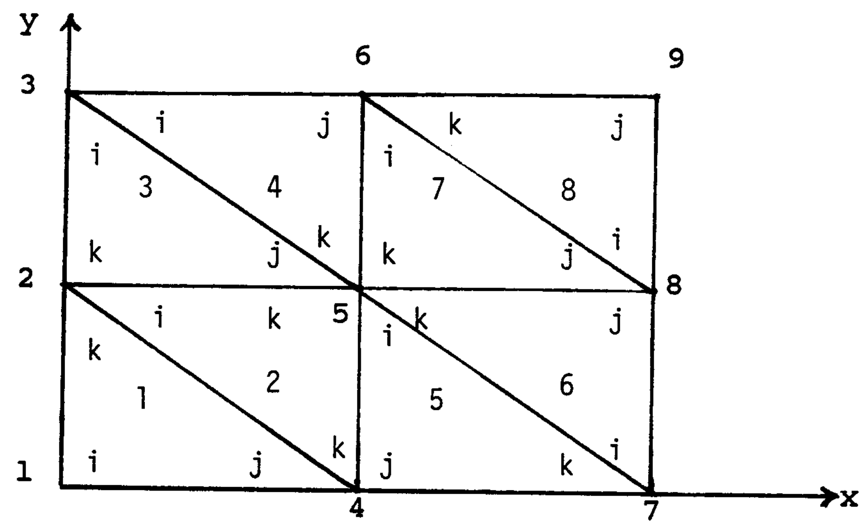


Fig. 6. Discretiziation of the absorber plate

with respect to x and y should be derived. From equation (69),

$N_e=[N_i\ \ N_j\ \ N_k] = 1/2A[\ a_i+b_ix+c_iy \quad a_j+b_jx+c_jy \quad a_k+b_kx+c_ky]$.

Therefore

$$\frac{\partial N_i}{\partial x} = \frac{b_i}{2A} \qquad \frac{\partial N_i}{\partial y} = \frac{c_i}{2A}$$
$$\frac{\partial N_j}{\partial x} = \frac{b_j}{2A} \qquad \frac{\partial N_j}{\partial y} = \frac{c_j}{2A}$$
$$\frac{\partial N_k}{\partial x} = \frac{b_k}{2A} \qquad \frac{\partial N_k}{\partial y} = \frac{c_k}{2A} \quad , \qquad (21)$$

where b_i, b_j, ..., and c_k are given by equation (63) and A is the area of the triangle. Substituting equation (21) into equation (20) yields

$$[B] = 1/2A \begin{bmatrix} b_i & b_j & b_k \\ c_i & c_j & c_k \end{bmatrix} \quad . \qquad (22)$$

Since B is constant and independent of x and y,

$$[K] = \int_A k\ [\ B\ ]^T\ [\ B\ ]\, dA = k[\ B\ ]^T[\ B\ ] \int_A dA \quad . \quad (23)$$

Preforming the integration gives

$$[K]=k/(4A)\begin{bmatrix} b_i + c_i & b_ib_j+c_ic_j & b_ib_k+c_ic_k \\ & b_j + c_j & b_jb_k+c_jc_k \\ \text{symmetric} & & b_k + c_k \end{bmatrix} . \qquad (24)$$

Considering element number 1,

$x_i=y_i=0.0$

$x_j=0.075$ $\quad y_i=0.0$

$x_k=0.0$ $\quad y_k=0.3$.

Substituting into equation (63) yields

$bi=-0.3$ $\quad c_i=-0.075$

$b_j= 0.3$ $\quad c_j= 0.0$

$b_k= 0.0$ $\quad c_k= 0.075$.

Substituting into equation (24) results

$$K_1 = k/4A_1\begin{bmatrix} 0.09562 & -0.09 & -0.00562 \\ & 0.09 & 0.0 \\ \text{symmetric} & & 0.00562 \end{bmatrix} . \qquad (25)$$

Since element number 1 is not subjected to convection heat loss,

$$[K_{h1}] = \{f_{h1}\} = 0 \quad .$$

If the Element Conduction Matrix (ECM) is defined as

$$[ECM]_e = [K]_e + [K_h]_e \quad , \qquad (26)$$

then

$$[ECM_1] = [K_1] \quad . \qquad (27)$$

Since all the elements have constant heat generation, from equation (20)

$$f_{q1} = \int_{\Omega^1} q\ [N]^T\ d\,\Omega^1 = q \int_{\Omega^1} \begin{Bmatrix} N_i \\ N_j \\ N_k \end{Bmatrix} d\,\Omega^1 \quad , \qquad (28)$$

where Ω^1 is the domain of element number 1. The evaluation of this integral is painless if the area coordinates are employed. The concept of area coordinates and its relating integral formulas are discussed in Appendix B. Assuming that L_1 is measured from the side opposite to node i,

$L_1 = N_i$,

$L_2 = N_j$, and

$L_3 = N_k$. (29)

Substituting into equation (28) and using equation (76) with the assumption that the thickness is unity, yields

$$f_{q1} = q \int_{A_1} \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} dA_1 = qA_1\ /3 \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix} \quad . \qquad (30)$$

If the Element Force vector (EF) is defined as

$$\{EF\} = f_q + f_h \quad , \qquad (31)$$

then

$$\{EF_1\} = f_{q1} \quad . \qquad (32)$$

The heat generated within the absorber plate (q) is approximated to be 139433 W/m^3 (Appendix D). Therefore

$$\{EF_1\} = A_1 \begin{Bmatrix} 46477.68 \\ 46477.68 \\ 46477.68 \end{Bmatrix} \quad . \qquad (33)$$

Since element numbers 2,3,4,5, and 7 are not subjected to convection heat loss, repeating the above procedure yields the following element property matrices for aforementioned elements:

$$[ECM_2] = k/(4A_2)\begin{bmatrix} 0.09 & 0.0 & -0.09 \\ & 0.00562 & -0.09562 \\ \text{symmetric} & & 0.00562 \end{bmatrix} \quad (34)$$

$$[ECM_3] = k/(4A_3)\begin{bmatrix} 0.00562 & -0.00562 & 0.0 \\ & 0.09562 & -0.09 \\ \text{symmetric} & & 0.09 \end{bmatrix} \quad (35)$$

$$[ECM_4] = k/(4A_4)\begin{bmatrix} 0.09 & 0.0 & -0.09 \\ & 0.00562 & -0.00562 \\ \text{symmetric} & & 0.09562 \end{bmatrix} \quad (36)$$

$$[ECM_5] = k/(4A_5)\begin{bmatrix} 0.00562 & -0.00562 & 0.0 \\ & 0.09562 & -0.09 \\ \text{symmetric} & & 0.09 \end{bmatrix} \quad (37)$$

$$[ECM_7] = k/(4A_7)\begin{bmatrix} 0.00562 & -0.00562 & 0.0 \\ & 0.09562 & -0.09 \\ \text{symmetric} & & 0.09 \end{bmatrix}. \quad (38)$$

Since the areas of all the elements and the heat generated within each element are equal,

$$\{EF_1\}=\{EF_2\}=\{EF_3\}=\{EF_4\}=\{EF_5\}=\{EF_7\}= A_1 \begin{Bmatrix} 46477.63 \\ 46477.63 \\ 46477.63 \end{Bmatrix}. \quad (39)$$

Since element numbers 6 and 8 are subjected to convection, $\{K_h\}$ and f_h matrices are not zero. Considering element number 6, $[K_6]$ and f_{q6} are calculated as

$$[K_6] = k/4A_6 \begin{bmatrix} 0.00562 & -0.00562 & 0.0 \\ & 0.09562 & -0.09 \\ \text{symmetric} & & 0.09 \end{bmatrix} , \qquad (40)$$

and

$$f_{q6} = A_6 \begin{Bmatrix} 46477.63 \\ 46477.63 \\ 46477.63 \end{Bmatrix} . \qquad (41)$$

However, $[K_h]$ is given by

$$[K_h]=\int_\Gamma h[N]^T[N]d\Gamma = h\int_\Gamma \begin{bmatrix} N_i^2 & N_iN_j & N_iN_k \\ & N_j^2 & N_jN_k \\ \text{symmetric} & & N_k^2 \end{bmatrix} d\Gamma , \qquad (42)$$

which must be evaluated over the surface from which the element is subjected to convection. For element number 6, side i-j is subjected to convection. Since N_k is zero along this side, equation (42) reduces to

$$[K_{h6}] = h\int_{\Gamma ij} \begin{bmatrix} N_i^2 & N_iN_j & 0.0 \\ & N_j^2 & 0.0 \\ \text{symmetric} & & 0.0 \end{bmatrix} d\Gamma . \qquad (43)$$

Employing the area coordinates and using related integral formulas gives

$$[K_{h6}] = h\Gamma ij/6 \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} , \qquad (44)$$

where Γij is the length of side i-j and is calculated by

$$\Gamma ij = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} = 0.3 . \qquad (45)$$

Substituting $\Gamma ij = 0.3$ and $h=304.0\ W/m^2K$ (Appendix D) into equation (43) and then forming the element conduction matrix gives

$$[ECM_6]=[K_6]+[K_{h6}]=\begin{bmatrix} 59.92 & -14.28 & 0.0 \\ & 531.92 & -471.99 \\ \text{symmetric} & & 471.99 \end{bmatrix} . \quad (46)$$

Furthermore, f_{h6} is obtained as follows:

$$f_{h6}=\int_{\Gamma_{ij}} h\ T_\infty\ [N]^T d\Gamma_{ij} = h\ T_\infty \int_{\Gamma_{ij}} \begin{Bmatrix} N_i \\ N_j \\ N_k \end{Bmatrix} d\Gamma_{ij} \ . \quad (47)$$

Using the area coordinates and integrating yields

$$f_{h6} = h\ T_\infty\ \Gamma_{ij}/2 \begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix} . \quad (48)$$

Substituting $T_\infty = 51°C$,

$$f_{h6} = \begin{Bmatrix} 2328.58 \\ 2328.58 \\ 0.0 \end{Bmatrix} . \quad (49)$$

The element force matrix is then obtained as

$$\{EF_6\} = f_{q6} + f_{h6} = \begin{Bmatrix} 2851.45 \\ 28251.45 \\ 522.87 \end{Bmatrix} . \quad (50)$$

Similarly, the element conduction and force matrices for element number 8 are calculated as

$$[ECM_8]=[K_8]+[K_{h8}]=\begin{bmatrix} 59.92 & -14.28 & 0.0 \\ & 531.92 & -471.99 \\ \text{symmetric} & & 471.99 \end{bmatrix} , \quad (51)$$

and

$$\{EF_8\} = f_{q8} + f_{h8} = \begin{Bmatrix} 2851.45 \\ 2851.45 \\ 522.87 \end{Bmatrix} \quad . \tag{52}$$

The FORTRAN coding for calculation of these element property matrices is shown in Table 1. The variables are defined as follows:

GK = element conduction matrix,

FQ = heat generation force vector,

FH = convection force vector,

GKH = element convection matrix,

CC = thermal conductivity,

H = coefficient of convection heat loss,

SEIJ = length of I-J side of the element, and

A = area of the element.

Table 1. Fortran coding for calculation of property matrices

```
      C.....
      C... obtain nodal coordinates
      C.....
       B1=Y2-Y3
       B2=Y3-Y1
       B3=Y1-Y2
       C1=X3-X2
       C2=X1-X3
       C3=X2-X1
      C... calculate area
       A(I)=(C3*B1-C1*B3)/2
       GK(1,1)=B1**2+C1**2
       GK(1,2)=B1*B2+C1*C2
       GK(1,3)=B1*B3+C1*C3
       GK(2,2)=B2**2+C2**2
       GK(2,3)=B2*B3+C2*C3
       GK(3,3)=B3**2+C3**2
       GK(2,1)=GK(1,2)
       GK(3,1)=GK(1,3)
       GK(3,2)=GK(2,3)
       DO 20 M=1,3
       FQ(M)=QD(I)*A(I)/3.0
       FH(M)=0.0
       DO 20 N=1,3
       GK(M,N)=GK(M,N)*CC/(4*A(I))
       GKH(M,N)=0.0
   20  CONTINUE
       IF(INBC(I).EQ.0) GO TO 30
   C...boundary on ij side of element
       SEIJ=SQRT((X2-X1)**2+(Y2-Y1)**2)
       CT=H(I)*SEIJ/6.0
       GKH(1,1)=2.0*CT
       GKH(1,2)=CT
       GKH(2,1)=GKH(1,2)
       GKH(2,2)=GKH(1,1)
       FH(1)=H(I)*SEIJ*AT(I)/2.0
       FH(2)=FH(1)
   30  CONTINUE
   C....
   C....
```

Assemblage of Element Equations

Once the element properties are determined, the next step is to construct the overall system equations which in a sense is equivalent to constructing the solution domain with the elements that comprise it. The assembly is based on the principle of compatibility; that is, at the nodes the value of the unknown field variable is the same for all elements joining at that node.

The element conduction matrix and force vector for element 1 are rewritten in Table 2 . The location of any coefficient EC_{ij} in the global conduction matrix, [GC], is identified by the global degrees of freedom corresponding to the local degrees of freedom. The location of the coefficients EC_{ij} in [GC] and EF_i in {GF} for element 1 is shown in Table 3 . The conduction matrix and the force vector of the second element are shown in Table 4 . These elements are placed in [GC] and {GF} at appropriate locations as shown in Table 5 .

The final global conduction matrix and force vector are obtained by adding the contributions of elements 3 through 8 to those shown in Table 5 . If there is no contribution from any elements to any coefficient of [GCM], then that coefficient will be taken as zero. The final global conduction matrix and force vector are shown in Table 6. The matrix [GCM] is written in a symmetric banded form.

Table 2. Coduction and force matrices of element 1

	local d.o.f	global d.o.f	i	j	k
			1	4	2
	i	1	501.49	-471.99	-29.49
$[EC_1]=$	j	4		471.99	0.0
	k	2	symmetric		29.49
	i	1	522.87		
$\{EF_1\}=$	j	4	522.87		
	k	2	522.87		

Table 3. Location of the coefficients of $[EC_1]$ and $\{EF_1\}$ in [GCM] and {GF}.

$[Ec_1]=$

global d.o.f	1	2	3	4	5	6	7	8	9
1	501.49	-29.49		-471.99					
2	-29.49	29.49		0.0					
3									
4	-471.99	0.0		471.99					
5									
6									
7									
8									
9									

$\{EF_1\}=$

global d.o.f	
1	522.87
2	522.87
3	
4	522.87
5	
6	
7	
8	
9	

Table 4. Conduction and force matrices of element 2

	local d.o.f	global d.o.f	i (2)	j (4)	k (5)
$[EC_2]=$	i	2	471.99	0.0	-471.99
	j	4		29.49	-29.49
	k	5	symmetric		501.49
$\{EF_2\}=$	i	2	522.87		
	j	4	522.87		
	k	5	522.87		

Table 5. Assembly of $[EC_1]$,$[EC_2]$,$\{EF_1\}$,and $\{EF_2\}$

	global d.o.f	1	2	3	4	5	6	7	8	9
	1	501.49	-29.49		-471.99					
	2	-29.49	29.49+ 471.99		0.0+ 0.0	-471.99				
	3									
$[EC_1]+$ $[EC_2]=$	4	-471.99	0.0+ 0.0		471.99 +29.49	-29.49				
	5		-471.99		-29.49	501.49				
	6									
	7									
	8									
	9									

Table 5. (Continued)

$\{EF_1\}+\{EF_2\}=$	1	522.87
	2	522.87+522.87
	3	
	4	522.87+522.87
	5	522.87
	6	
	7	
	8	
	9	

Table 6. Global conduction and force matrices

[GCM] =	501.49	-29.49	0.0	-471.99
	1002.99	-29.49	0.0	-943.99
	501.49	0.0	0.0	-471.99
	1002.99	-58.99	0.0	-471.99
	2005.99	-58.99	0.0	-943.99
	1002.99	0.0	0.0	-471.99
	531.93	-14.28	0.0	
	1063.87	-14,28		
	531.93			

Table 6. (Continued)

$$\{GF\} = \begin{Bmatrix} 522.87 \\ 1568.62 \\ 1045.74 \\ 1568.62 \\ 3660.11 \\ 1045.74 \\ 3374.32 \\ 5702.9 \\ 3374.32 \end{Bmatrix}$$

Table 7 shows the FORTRAN coding that has been used for the assembly process. Since the matrix [GCM] is banded and symmetric, only the upper triangular matrix is calculated and stored. The variables are defined as follows:

NEL = number of elements

NNODE = total number of nodes

NBW = number of band width

GCM = global conduction matrix

ECM = element conduction matrix

GF = global force vector

EF = element force vector

NENN(i,j) = global node number corresponding to the j^{th} corner of i^{th} element. (53)

Table 7. Fortran coding for assembly of element matrices

```
C....
C     initialize the matrices
C...
      DO 5 I=1,NNODE
      GF = 0.0
      DO 5 J=1,NBW
      GCM(I,J)=0.0
5     CONTINUE
C...
C...
      DO 10 I=1,NEL
      DO 15 M=1,3
      IM=NENN(I,M)
      GF(IM)=GF(IM)+ EF(M)
      DO 15 N=1,3
      IN=NENN(I,N)-IM+1
      IF(IN.LE.0) GO TO 15
      GCM(IM,IN)=GCM(IM,IN) + ECM(M,N)
15    CONTINUE
10    CONTINUE
```

Incorporation of Dirichlet Boundary Conditions

After all the element characteristic matrices have been assembled into the global conduction matrix and force vector, the system equations can be written as

$$[\ GCM\ ]\{\ T\ \} = \{\ GF\ \} \qquad . \tag{54}$$

This system of equations must be modified whenever some of the nodal temperatures are prescribed. If the ith coefficient of { T } is prescribed, then the modification proceeds as follows:

1. Subtract the product of the (j,i) coefficient of [GCM] times the known ith coefficient of { GF } from the jth coefficient of { GF }.

2. Replace the ith row and the ith column of [GCM] by zero.

3. Set the (i,i) coefficient of [GCM] to unity.
4. Make the ith coefficient of { GF } equal to the prescribed value [8].

The temperatures of nodes 1,3,4,6,7, and 9 of the mesh of Fig. 6 are prescribed and are

$$T_1=T_4=T_7=T_i= 50.0\ ^\circ C$$

$$T_3=T_6=T_9=T_o= 53.0\ ^\circ C$$

The system of equations is given by

$$\begin{bmatrix} a11 & a12 & a13 & a14 \ldots\ldots\ldots\ldots a19 \\ a21 & a22 & a23 & a24 \ldots\ldots\ldots\ldots a29 \\ \vdots & \vdots & \vdots & \vdots \\ a91 & a92 & a93 & a94 \ldots\ldots\ldots\ldots a99 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ \vdots \\ T_9 \end{Bmatrix} = \begin{Bmatrix} f1 \\ f2 \\ \vdots \\ f9 \end{Bmatrix} , \qquad (55)$$

where a11,a12,.....,and f9 are the same as those shown in Table 6. Considering node number 1 and implementing the above procedures yields

$$\begin{bmatrix} 1 & 0 & 0 & \ldots\ldots\ldots\ldots\ldots & 0 \\ 0 & a22 & a23 & \ldots\ldots\ldots\ldots\ldots & a29 \\ 0 & a32 & a33 & a34 \ldots\ldots\ldots\ldots & a39 \\ 0 & a42 & a43 & a44 \quad a45 \ldots\ldots & a49 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & a92 & a93 & a94 \quad a95 & a96..a99 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ \vdots \\ T_9 \end{Bmatrix} = \begin{Bmatrix} 50.0 \\ 3043.12 \\ f3 \\ 25168.12 \\ \vdots \\ f9 \end{Bmatrix} . \qquad (56)$$

Repeating the procedure for the remaining nodes will result in the following system of equations:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1002.99 & 0 & 0 & -943.99 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 2005.99 & 0 & 0 & -943.9 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & \text{this matrix} \\ 206.24 & 0 & & \text{is written in the} \\ 1 & & & \text{banded form} \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \\ T_9 \end{Bmatrix} = \begin{Bmatrix} 50.0 \\ 4607.11 \\ 53.0 \\ 50.0 \\ 9737.11 \\ 53.0 \\ 50.0 \\ 7173.79 \\ 53.0 \end{Bmatrix} \quad . \qquad (57)$$

To implement incorporation of the Dirichlet boundary conditions according to the aforementioned procedure, a subroutine is written which is shown in Table 8 . The program assumes that the matrix [GCM] is stored in band form. The subroutine is called for each prescribed nodal degree of freedom by the following FORTRAN statement

```
CALL DIRBC (GCM,GF,NNODE,NBW,ND,PT)   ,
```

where

ND = node subjected to prescribed temperature

ST = value of the prescribed temperature.

Solution of System Equations

The last step of the finite element method is the solution of system equations. Although there are many methods available for solving a system of linear equations, the Choleski's method is used. This method is briefly discussed in Appendix E.

Table 8. Fortran coding for incorporation of the Dirichlet boundary conditions.

```
      SUBROUTINE DIRBC (GCM,GF,NNODE,NBW,M,ST)
      DIMENSION GCM(NNODE,NBW),GF(NNODE)
      DO 5 K = 2,NBW
      I1=M-K+1
      I2=M+K-1
      IF(I1.GE.1) GF(I1)=GF(I1)-A(I1,K)*ST
 5    IF(I2.LE.NNODE) GF(I2)=GF(I2)-A(M,K)*ST
      GF(M) = ST
      DO 10 J = 1,NBW
      I1=M-J+1
      IF(I1.GE.1) A(I1,J)=0.0
 10   A(M,J) = 0.0
      A(M,1) = 1.0
      RETURN
      END
```

The Choleski's method is implemented by using two subroutines which are shown in Tables 9 and 10. The subroutine DECOM performs the decomposition of the global conduction matrix [GCM] into an upper triangular matrix, while the subroutine CHOLE solves the system equations and stores the results in array TEM. The subroutines are called by the following FORTRAN statements

CALL DECOM (NNODE,NBW,GCM)

CALL CHOLE (NNODE,NBW,GCM,TEM)

Using Choleski's method to solve the system of equations given by equation (57) yields

$$\begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \\ T_9 \end{Bmatrix} = \begin{Bmatrix} 50.0 \\ 73.32 \\ 53.0 \\ 50.0 \\ 73.02 \\ 53.0 \\ 50.0 \\ 71.53 \\ 53.0 \end{Bmatrix} \quad . \tag{58}$$

Table 9. Fortran coding for decomposition of global conduction matrix.

```
      SUBROUTINE DECOM (NNODE,NBW,A)
      DIMENSION A(NNODE,NBW)
      DOUBLE PRECISION D
      A(1,1) = SQRT(A(1,1))
      DO 10 I=2,NBW
 10   A(1,I)=A(1,I)/A(1,1)
      DO 20 I=2,NNODE
      I1=I+1
      I2=I-1
      D=A(I,1)
      DO 30 J=1,I2
      I3=I+1-J
      IF(I3.GT.NBW) GO TO 30
      D=D-A(J,I3)** 2
 30   CONTINUE
      A(I,1)=DSQRT(D)
      DO 40 IJ=2,NBW
      IF(I+IJ-1.GT.NNODE) GO TO 20
      D=A(I,IJ)
      DO 50 J=1,I2
      I3=I+1-J
      I4=I+IJ-1
      IF(I4.GT.NBW) GO TO 50
      IF(I3.GT.NBW) GO TO 50
      D=D-A(J,I3)*A(J,I4)
 50   CONTINUE
 40   A(I,IJ)=D/A(I,1)
 20   CONTINUE
      RETURN
      END
```

Table 10. Fortran coding for solution of system equations.

```
      SUBROUTINE CHOLE (NNODE,NBW,A,B)
      DIMENSION A(NNODE,NBW),B(NNODE)
      DOUBLE PRECISION D
      B(1)=B(1)/A(1,1)
      DO 10 I=2,NNODE
      D=B(I)
      DO 20 J=2,NBW
      I1=I+1-J
      IF(I1.LT.1) GO TO 20
      I2=I+1-I1
      IF(I2.GT.NBW) GO TO 20
      D=D-A(I1,I2)*B(I1)
 20   CONTINUE
      B(I)=D/A(I,1)
 10   CONTINUE
      B(NNODE)=B(NNODE)/A(NNODE,1)
C
C     compute the system unknowns
C
      DO 30 I=2,NNODE
      I3=NNODE+1-I
      D=B(I3)
      DO 40 J=2,NBW
      I4=I3-1+J
      IF(I4.GT.NNODE) GO TO 40
      D=D-A(I3,J)*B(I4)
 40   CONTINUE
      B(I3)=D/A(I3,1)
 30   CONTINUE
      RETURN
      END
```

CHAPTER IV

NUMERICAL ANALYSIS OF THE ABSORBER PLATE

The mesh of Fig. 6, which was used to illustrate the formulation of the problem by the finite elements method, does not produce an accurate solution to the problem. However, the solution is expected to converge as the size of the elements are reduced. On the other hand, there are two more possible mathematical models for representation of the absorber plate. These models, along with the initial model of Fig. 2, are shown in Fig. 7. For the model of Fig. 7-b,

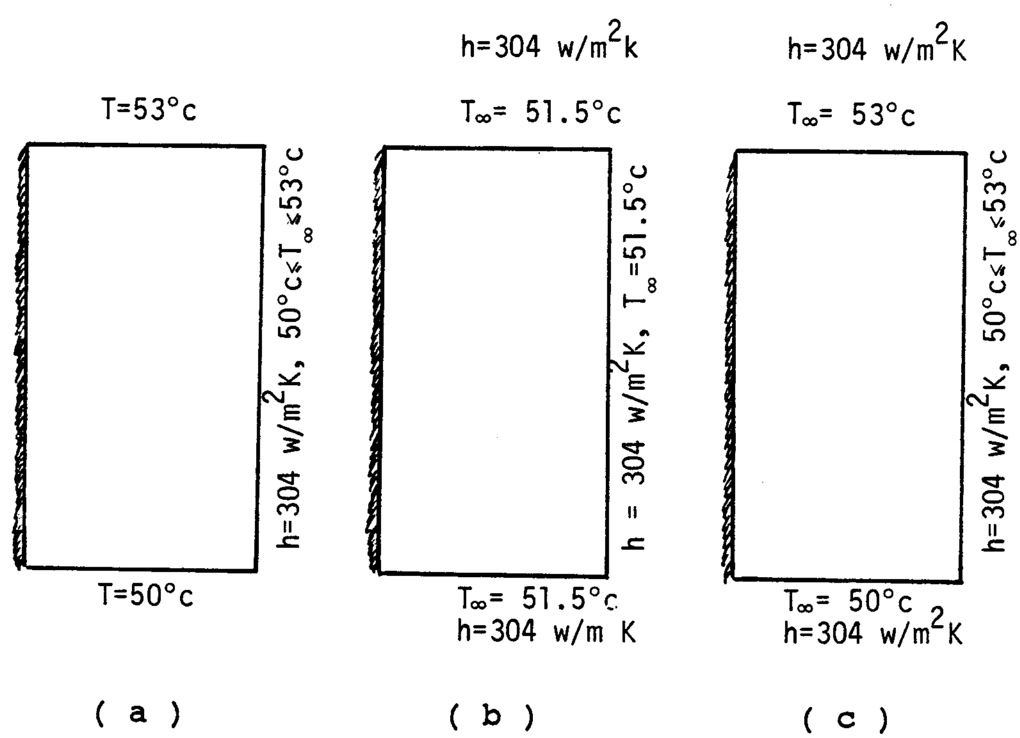


Fig. 7. Possible mathematical models for absorber plate

the mean temperature of the fluid flowing through the tubes is used as the ambient temperature (51.5°c). The model of Fig. 7-c assumes that the ambient temperatures around the top and bottom of the plate are equal to the outlet and inlet fluid temperatures, respectively. Again, the temperature variation of the fluid from T_i to T_o is assumed to be linear.

In order to achieve an accurate solution to the problem, a fine mesh with 451 nodes and 800 elements was constructed. The finite element solutions of the aforementioned problems are illustrated by means of the isothermal lines within the absorber plate, as shown in pages 36,37, and 38. The temperature distribution along the vertical and horizontal axes are also shown in page 39. Note that this is done only for the original mathematical model, since for this case the isothermal lines do not visualize the solution of the problem as good as the other two cases.

In order to investigate the accuracy of the finite element method, the well-known method of finite differences was used. The finite difference solution of the problems along with the finite element results are tabulated in Tables 11 and 12. The finite difference solution was obtained by dividing the solution region into 8 rectangles (9 nodes). The finite element solution was also obtained by using the same number of nodes (9 nodes and 16 elements) and the mesh with 451 nodes and 800 elements. The results are very close to each other, which implies that an excessive number of elements is not necessary for a reasonably accurate solution. It may be noted that for this particular problem, the finite difference solution converges faster than finite element because of the simple geometry of the problem.

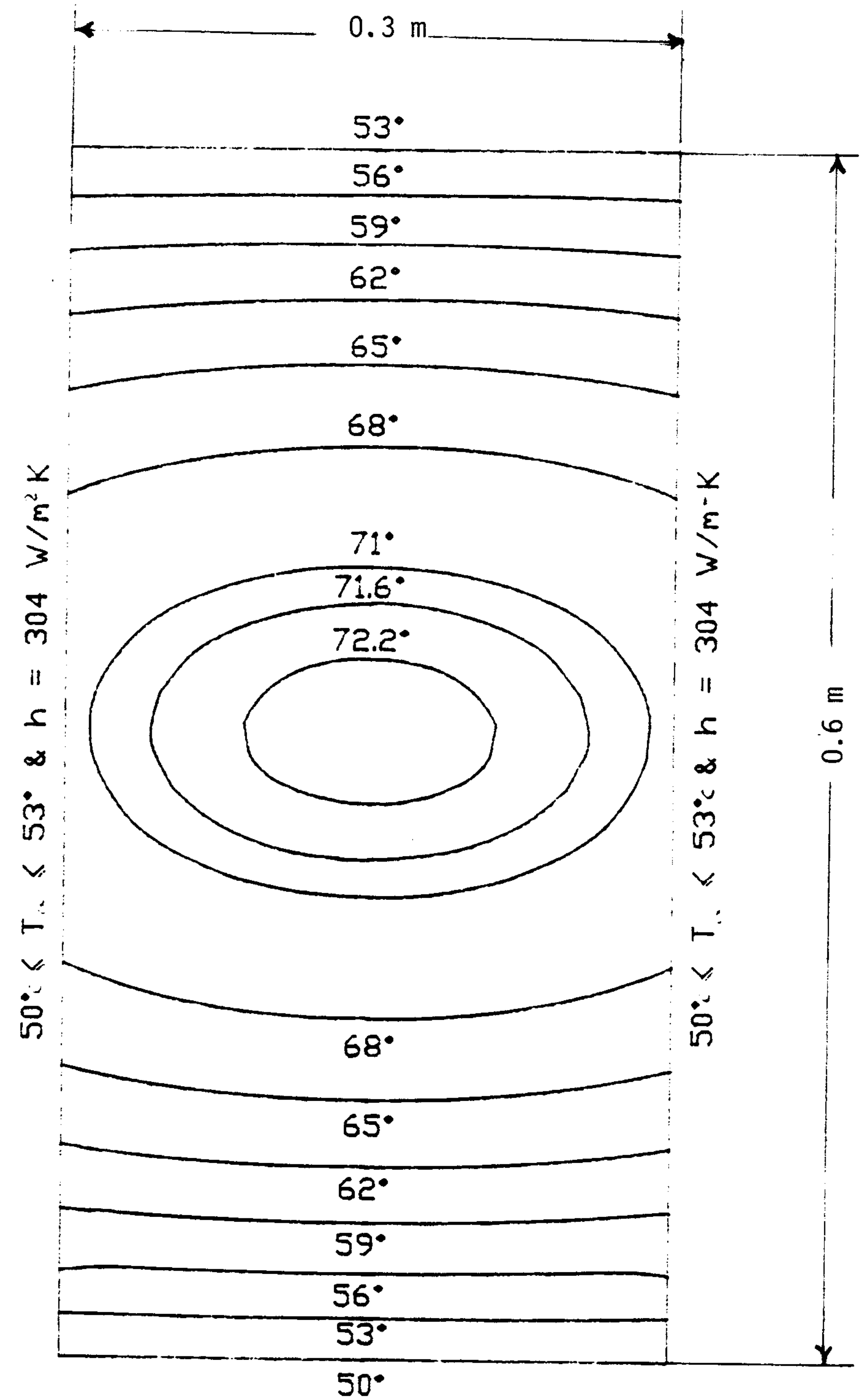


Fig. 8. Isothermal lines for model of Fig. 7-a

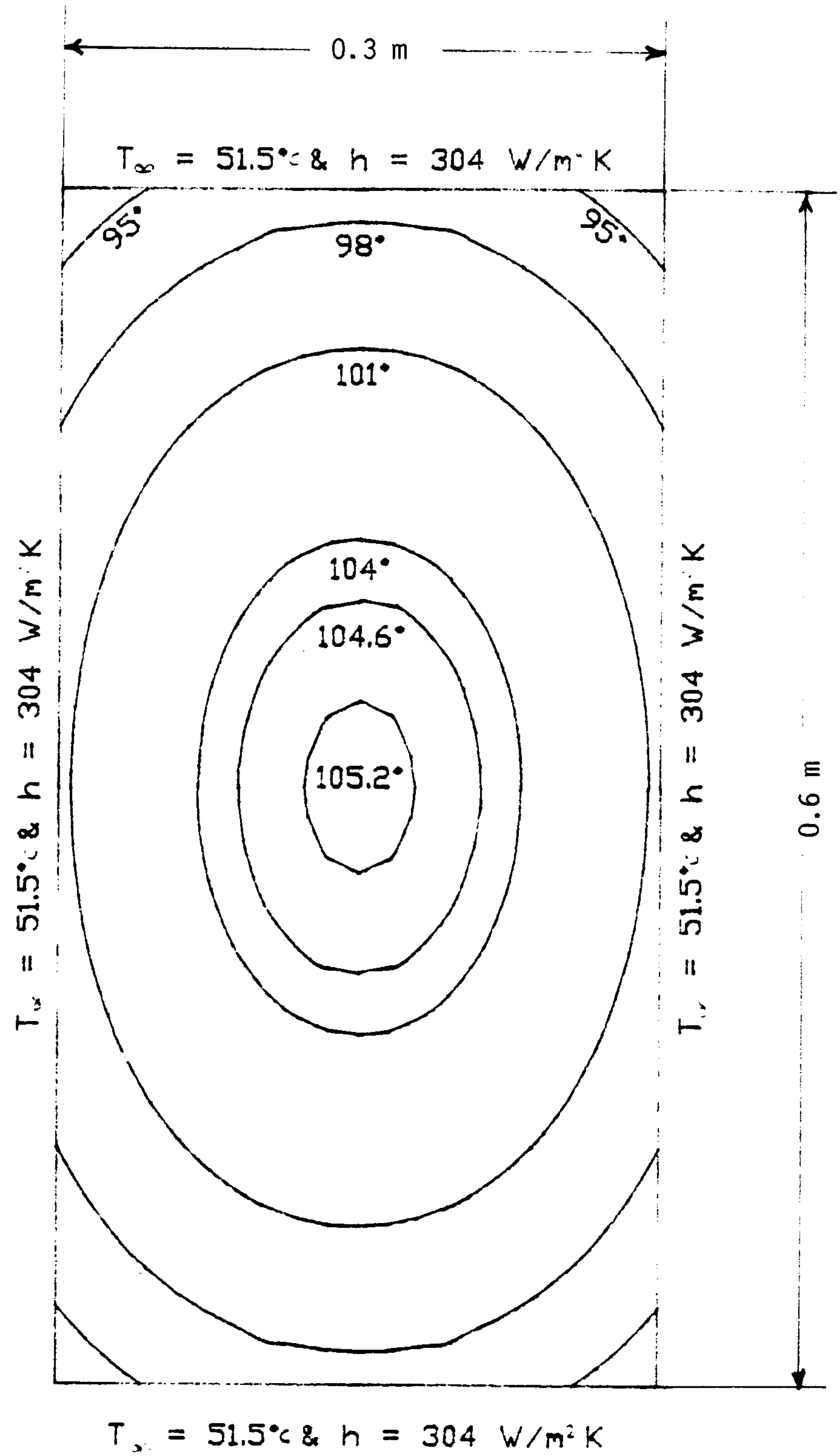


Fig. 9. Isothermal lines for model of Fig. 7-b

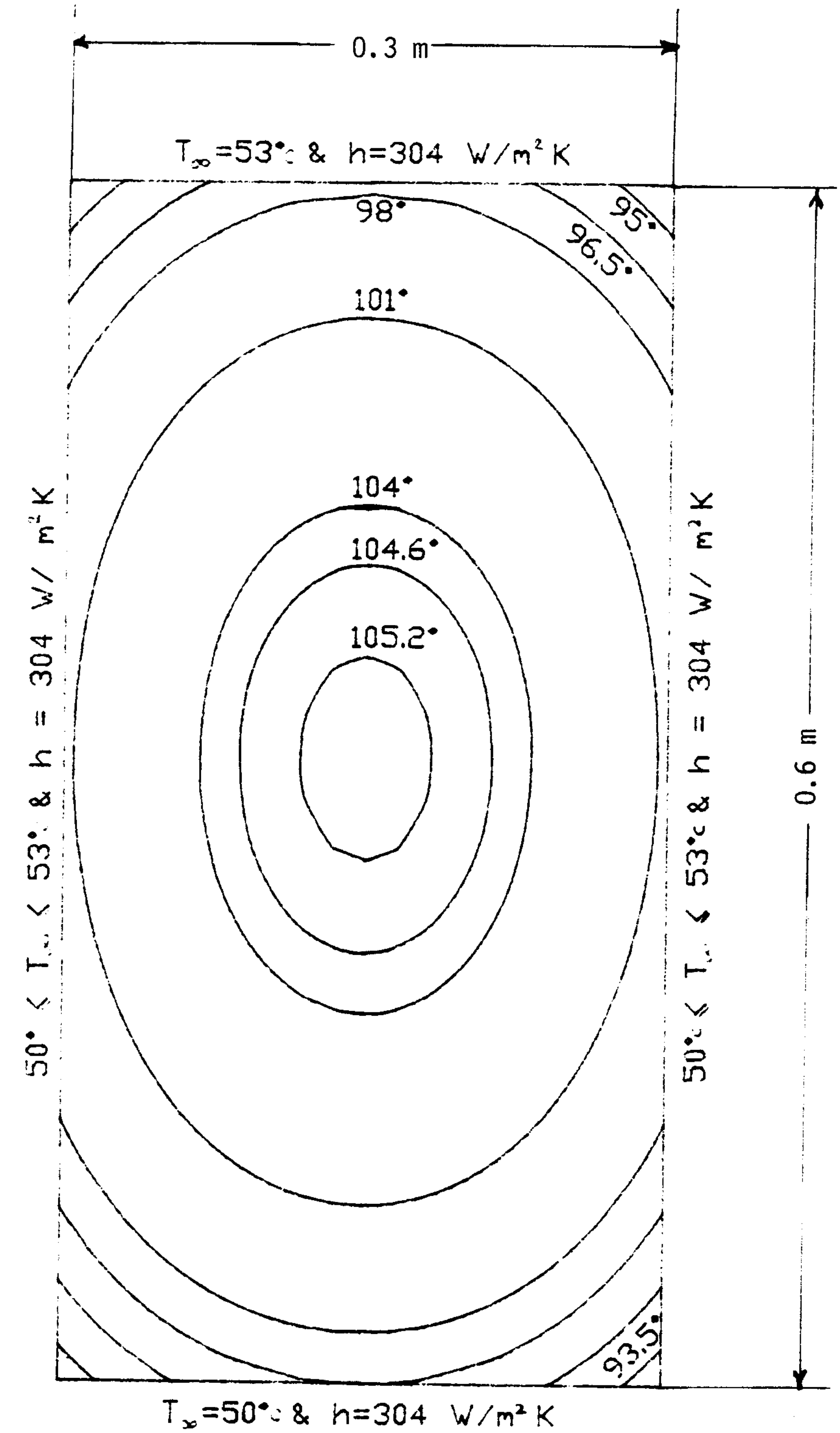
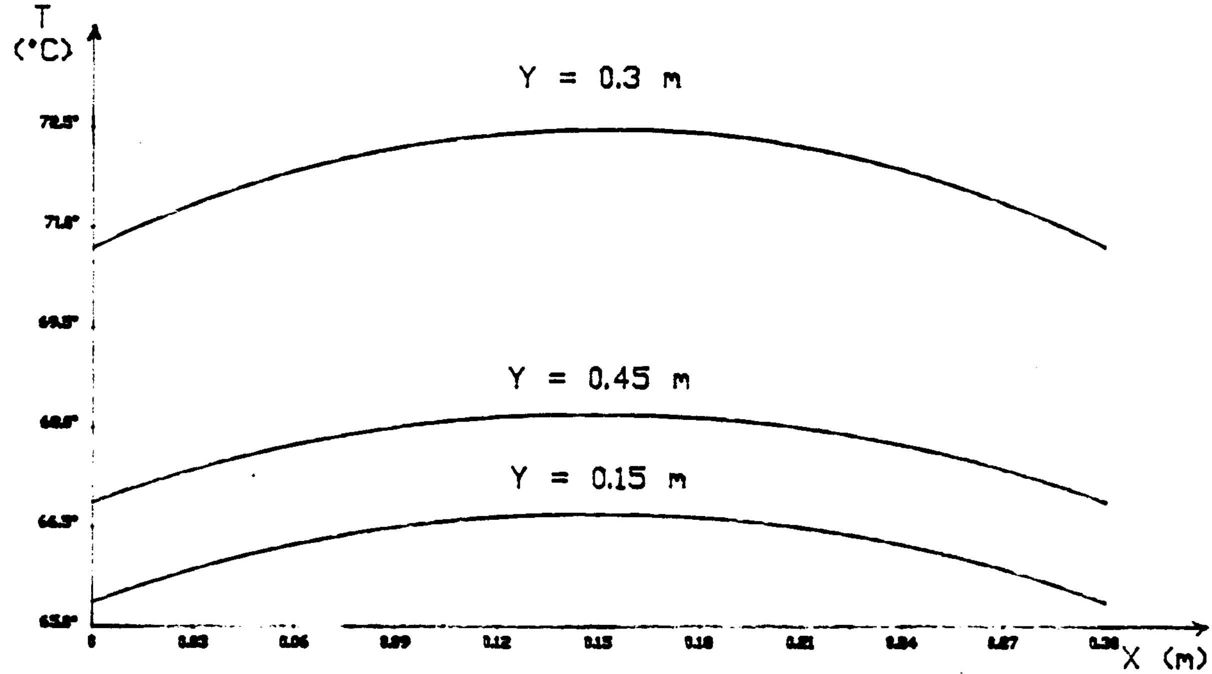
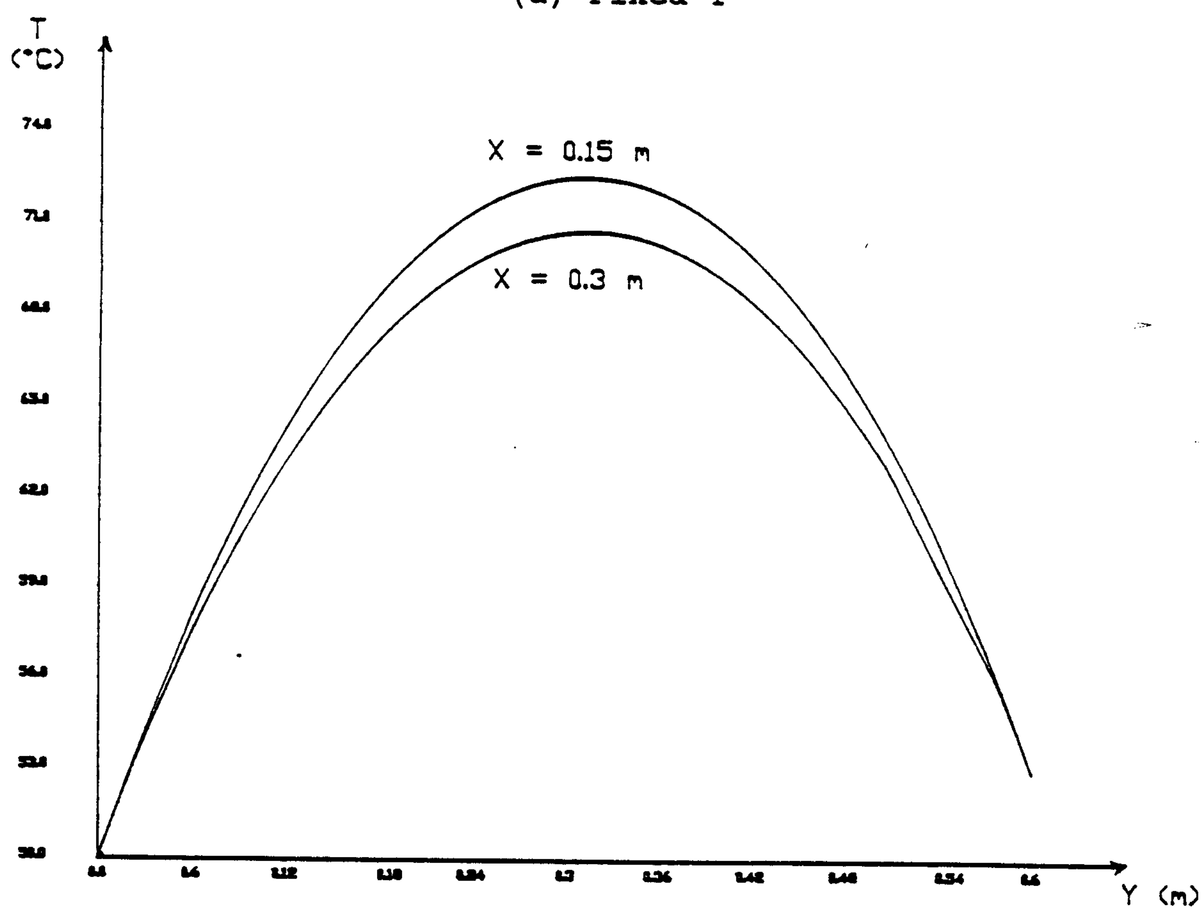


Fig. 10. Isothermal lines for model of Fig. 7-c



(a) Fixed Y



(b) Fixed X

Fig. 11. Temperature profiles for model of Fig. 7-a

Table 11. Comparison of finite element results for model of Fig. 7-b to finite difference and approximate analytical solutions.

FDM (9 nodes)	FEM (9 nodes)	FEM (451 nodes)	Analytical
105.551	105.6827	105.3435	105.3521
104.376	104.5124	104.1686	104.1778
100.829	100.9697	100.6223	100.6331
103.604	103.6134	103.3452	103.3655
102.473	102.5289	102.2146	102.2360
99.057	99.1745	98.8011	98.8236
97.412	96.9271	97.0306	97.0417
96.419	96.3323	95.9984	96.0526
93.417	93.6127	93.0064	93.0599

Table 12. Comparison of finite element and finite difference solutions for model of Fig. 7-c

F.E.M.(15 nodes)	F.E.M.(451 nodes)	F.D.M.(15 nodes)
97.1582	96.5292	96.845
95.6068	95.5512	95.836
92.8418	92.5343	92.779
103.5136	103.1205	103.284
102.3752	101.9811	102.144
98.5172	98.5371	98.699
105.6827	105.4169	105.551
104.5124	104.2413	104.376
100.9697	100.6928	100.829
103.9211	103.7367	103.924
102.8457	102.6155	102.803
99.5199	99.2302	99.415
97.4695	97.6632	97.978
96.8873	96.6643	97.003
94.2087	93.7057	94.055

CHAPTER V

CONCLUSION AND DISCUSSION

The finite element formulation for determination of temperature distribution in the absorber plate of a flat plate solar collector has been demonstrated and a computer program written based on these formulations.

This demonstration has shown that the finite element analysis is a valid and versatile method. The finite element solution of those problems which have analytical solutions, shows excellent agreement with the corresponding analytical solutions.

The finite element program used in this thesis is written based on the formulation of the equations by the Galerkin approach. It has several features which make it easy to use and economical. Storing the stiffness matrix in a symmetric banded form reduces the storage requirement of the program by more than half. For a numerical analysis the modification of the program,in general, may be a necessity when a new problem or a new mesh is constructed. The program is written to accommodate various problems with minimum modifications so that the human errors arising in the allocation of required memory storage for the array are greatly reduced.

When comparing the finite element and the finite difference solutions it seems that the latter method yields slightly more accurate values. This is to be anticipated because of the simplicity in the geometry of the problem and the triangular element used in this finite element formulation, which is based on the linear variation. The accuracy of the solution by the finite element method should increase if quadratic or isoparametric elements are used. Therefore, a general statement in favor of the finite difference method over the finite element method can not be justified. This is due to the fact that the versatility of the finite element method, especially the ability of the method to realistically model any geometric configuration, is far beyond that of the finite difference method.

Another versatility of this finite element program is the fact that it could be used to solve any field problem which is goverened by the Laplace or Poisson's equation and has the same type of boundary conditions.

For more general applications, the computer program can be improved by accommodating portions of programming or subroutines (1)to handle variable material properties, (2)to calculate the convection matrix for elements which have more than one side exposed to convective heat loss, (3)to renumber the node numbering in order to minimize the number of bandwith and therefore to minimize the size of the conduction matrix, and (4)to make automatic mesh generation.

APPENDIX A

TWO DIMENSIONAL SIMPLEX ELEMENT

Two dimensional simplex element is a triangle with three nodes, one at each corner, and straight sides [9]. As shown in Fig. 12, the nodes are labled counterclockwise from node i, which is specified arbitrarily. The global coordinates of nodes i, j, and k are $\{x_i, y_i\}$, $\{x_j, y_j\}$, and $\{x_k, y_k\}$. The nodal values of the scalar field variable are denoted as T_i, T_j , and T_k. The interpolation polynomial is

$$T(x,y)=a_1+a_2x+a_3y \quad , \tag{59}$$

with the nodal conditions

$$T=T_i \text{ at } x = x_i,\ y = y_i, \tag{60}$$

$$T=T_j \text{ at } x = x_j,\ y = y_j, \text{ and} \tag{61}$$

$$T=T_k \text{ at } x = x_k,\ y = y_k \quad . \tag{62}$$

Substituting the nodal conditions into equation (59) gives

$$\begin{aligned} T_i&=a_1+a_2x_i+a_3y_i \\ T_j&=a_1+a_2x_j+a_3y_j \\ T_k&=a_1+a_3x_k+a_3y_k \quad . \end{aligned} \tag{63}$$

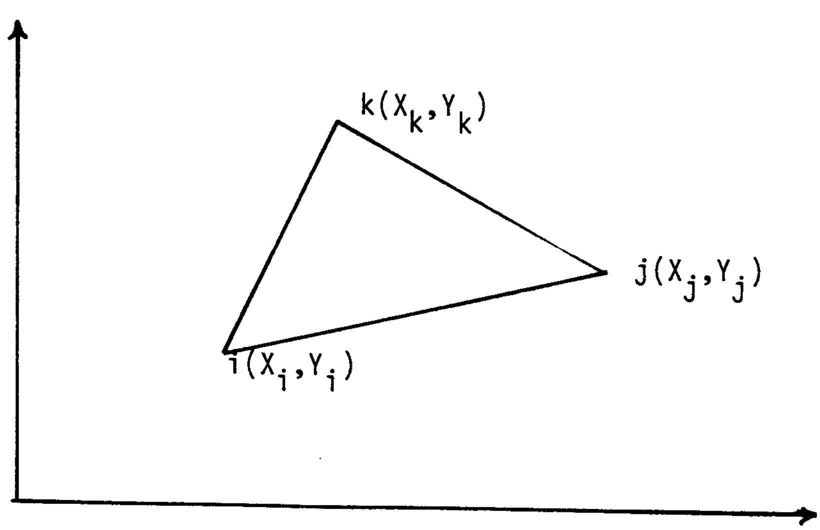


Fig.12 Two dimensional simplex element

Solving the system of equations for the polynomial coefficients yields

$$a_1=(a_iT_i+a_jT_j+a_kT_k)/(2A)$$
$$a_2=(b_iT_i+b_jT_j+b_kT_k)/(2A)$$
$$a_3=(c_iT_i+c_jT_j+c_kT_k)/(2A) \quad , \qquad (64)$$

where

$$a_i=x_iy_k-x_ky_j$$
$$a_j=x_ky_i-x_iy_k$$
$$a_k=x_iy_j-x_jy_i$$
$$b_i=y_j-y_k$$
$$b_j=y_k-y_i$$
$$b_k=y_i-y_j$$
$$c_i=x_k-x_j$$
$$c_j=x_i-x_k$$
$$c_k=x_j-x_i \quad , \qquad (65)$$

and A is area of the triangle. A is calculated by

$$A=1/2 \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{vmatrix} \quad . \qquad (66)$$

Substituting equation (64) into equation (59) yields

$$T(x,y)=\{(a_i+b_ix+c_jy)T_i+(a_j+b_jx+c_jy)T_j+(a_k+b_kx+c_ky)T_k\}/(2A), \qquad (67)$$

which can be written in matrix form as

$$T(x,y)=[\ N_i\ \ N_j\ \ N_k\ ] \begin{Bmatrix} T_i \\ T_j \\ T_k \end{Bmatrix} = [N(x,y)]T^e \quad , \qquad (68)$$

where T^e is the nodal unknown vector of element e, and the shape functions, $N(x,y)=[N_i\ N_j\ N_k]$ are

$$N_i=(a_i+b_i x+c_i y)/(2A)$$
$$N_j=(a_j+b_j x+c_j y)/(2A)$$
$$N_k=(a_k+b_k x+c_k y)/(2A) \quad . \tag{69}$$

APPENDIX B

INTERPOLATION FUNCTIONS IN TERMS OF LOCAL COORDINATES

The determination of the system equations involves the integration of the interpolation functions and/or their derivatives over the element. If the interpolation functions are written in terms of the local coordinate system, then the evaluation of these integrals will be easier. The local coordinate system is one located on or within the boundaries of the element. A special local coordinate system is a natural coordinate system whose coordinates range between zero and one.

For the triangular element the natural coordinate system is obtained by defining three coordinate ratios L_1, L_2, and L_3 as shown in Fig. 13. Each coordinate is the ratio of a perpendicular distance from one side, s, to the altitude, h, of that same side. These coordinates are also called area coordinates, because their value gives the area of subtriangles relative to the total area. Considering an arbitrary point B within the element, the total area is

$$A_t = bh/2 \quad , \tag{70}$$

while the area of the triangle formed by Bjk is

$$A_1 = bs/2 \quad . \tag{71}$$

Forming the ratio of these areas yields

$$A_1 \;/\; A_t = s/h = L_1 \quad . \tag{72}$$

Similarly

$$L_2 = A_2 \;/\; A_t \quad \text{and} \quad L_3 = A_3 \;/\; A_t \quad . \tag{73}$$

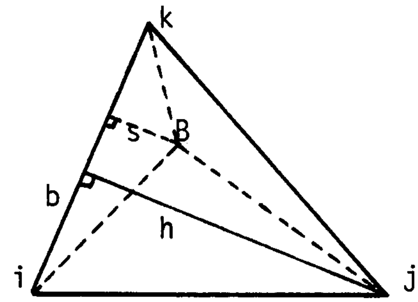


Fig. 13. Natural coordinate system for a triangle

Since

$$A_1 + A_2 + A_3 = A_t \quad , \tag{74}$$

$$L_1 + L_2 + L_3 = 1. \tag{75}$$

The natural coordinate in terms of the Cartesian Coordinates are given by [10]

$$L_1(x,y)=(a_i+b_ix+c_iy)/(2A)$$
$$L_2(x,y)=(a_j+b_jx+c_ky)/(2A)$$
$$L_3(x,y)=(a_k+b_kx+c_ky)/(2A) \quad , \tag{76}$$

where A is the area and the coefficients a_i, $a_j \ldots$, c_k are the same as those defined by (63). Since equations (76) and (69) are identical, the natural coordinates are precisely the interpolation functions for linear interpolation over a triangle. Thus

$$N_i = L_i \quad . \tag{77}$$

The advantage of using the area coordinate is the existence of integration equations which simplify the evaluation of length and area integrals. These equations are [11]

$$\int_\Gamma L_1^a L_2^b \, d\Gamma = (a!\ b!)(\Gamma)/(a+b+1)!$$

$$\int_A L_1^{a} L_2^{b} L_3^{c} \, dA = (a!\ b!\ c!)(2A)/(a+b+c+2)! \quad , \qquad (78)$$

where A = area of triangle

Γ = length along an edge of element.

APPENDIX C

WEIGHTED RESIDUAL METHOD

The method of weighted residual is an approximate technique for solution of partial differential equations. In this method, an approximate solution to the problem which satisfies the boundary conditions is assumed. Substitution of this approximate solution into the original differential equation results in some error or residual. This residual is then required to vanish in some average sense over the solution domain.

Suppose the governing equation for a problem is

$$L\,(T) - f = 0 \quad \text{in } \Omega \quad , \tag{79}$$

and its boundary conditions are

$$C_r = g_r \text{ in } \Gamma \quad . \tag{80}$$

The solution to equation (79) is then approximated by

$$T' = \quad N_i\, T_i \quad , \tag{81}$$

in which N_i are trial functions which satisfy the boundary conditions, and T_i are unknown parameters. Since T' is an approximate solution, substitution of T' into equation (79) results in

$$L\,(T') - f = \varepsilon = 0 \ . \tag{82}$$

The method of weighted residual requires that m unknown parameters T_i be determind by satisfying

$$\int_\Omega W_i\, \varepsilon\, d\Omega = \int_\Omega W_i\, (\, L(T') - f\,)\, d\Omega = 0 \quad , \tag{83}$$

where W_i are m linear independent weighting functions [12].

There are numerous means to choose the weighting function W_i, leading to Galerkin method, least-square method, method of moments, and collocation method. In the Galerkin method, the trial functions N_i are used as weighting functions. Thus

$$W_i = N_i \tag{84}$$

and

$$\int_\Omega W_i(L(T')-f)\, d\,\Omega = \int_\Omega N_i(L(T')-f)\, d\Omega \quad . \tag{85}$$

APPENDIX D

CALCULATION OF HEAT GENERATION AND CONVECTION HEAT TRANSFER COEFFICIENT

Because it was assumed that the problem is at steady state condition, the average solar heat flux reaching the solar collector can be assumed to be constant and equal to 10,000.0 KJ/m^2hr. Thus, the solar constant can be calculated as

$$I_o = 10{,}000{,}000.0 / 12(3600) = 231.48 \; w/m^2 \quad . \qquad (86)$$

If the collector has transmittance (τ) and absorptance (α) of 0.85 and 0.9 respectively, then the net energy absorbed by the plate is

$$S = I_o \; (\tau \; \alpha) = 231.48 \times 0.85 \times 0.9 = 177.1 \; w/m^2 \quad . \qquad (87)$$

Finally, the rate of heat generation per unit volume is

$$q = S/t = 177.1 / 0.00127 = 139{,}448.82 \; w/m^3 \quad . \qquad (88)$$

In order to calculate the convection heat transfer coefficient, one must determine whether the flow of fluid through the cooling tube is laminar or turbulent. If the coolant fluid is water, then the velocity of water through the tube can be determined from

$$V = 4 \; \dot{m} \; / \; \rho \; (\pi D^2) \quad , \qquad (89)$$

where $\dot{m}$ is the mass flow rate of water, ρ the density, and D the diameter of the tube. The mass flow rate is determined from following relationship :

$$Q = \dot{m} \; c_p \; (T_o - T_i) \quad , \qquad (90)$$

where Q is total heat generation in the plate, and c_p is specific heat of water. The total heat generated in the plate is

$$Q = S \; A = 177.48 \; (0.6 \times 0.15) = 15.97 \; W \quad . \qquad (91)$$

If the diameter of the tube is 1.0 cm, then

$$\dot{m} = Q/c_p(T_o-T_i)=15.97/4175(53-50)= 0.00127 \text{ kg/s} . \quad (92)$$

Thus, the velocity of water is

$$V = 4(0.00127)/(992.2)(0.01)^2(3.14) = 0.0163 \text{ m/s} . \quad (93)$$

Once the velocity of the water through the tube is determined, the Reynolds number can be calculated from

$$Re = VD/v \quad , \quad (94)$$

where v is the Kinematic Viscosity. Substituting the appropriate values into equation (94) yields

$$Re = (0.0163)(0.01)/(0.658 \times 10^{-6}) = 247.7 \quad . \quad (95)$$

Since the Reynolds number is less than 2300, the flow may be assumed to be laminar. The heat transfer coefficient for laminar flow can be evaluated from the emprical correlation of

$$Nu_D = 1.86(Re_D\ Pr)^{0.33} \left(\frac{D}{L}\right)^{0.33} \left(\frac{\mu_b}{\mu_s}\right)^{0.14} \quad (96)$$

if $(Re_D\ Pr\ D)/L$ is less than 10 [13]. In the above equation Pr is the Prandtl number, and μ_b and μ_s are the viscosity at the average bulk temperature and the wall temperature, respectively. If the empirical correction factor $\left(\frac{\mu_b}{\mu_s}\right)^{.14}$ which is to account for the effect of temperature variation, is assumed to be unity, then

$$Nu_D = 1.86\ (Re_D\ Pr\ D/L)^{0.33}$$

or

$$Nu_D = 1.86(247.7 \times 4.3 \times 0.01/0.6)^{0.33}$$

$$= 1.86(17.75)^{0.33} = 4.805 \quad . \quad (97)$$

Therefore

$$h_c = Nu_D\ K / D \quad , \quad (98)$$

where h_c is the average heat transfer coefficient and K the thermal conductivity of the fluid. Substituting equation (97) into equation (98) yields the average convection heat transfer coefficient as

$$h_c = (4.8)(0.633)/0.01 = 304 \ W/m^2 \ k \quad . \tag{99}$$

APPENDIX E

CHOLESKI METHOD

The Choleski method, also called the Banachiewicz method, uses the fact that a symmetric matrix can be expressed as the product of two triangular matrices, as [14]

$$A = S^T S \qquad (100)$$

or

$$\begin{bmatrix} a11 & a12 & \cdots & a1n \\ a21 & a22 & \cdots & a2n \\ a31 & a32 & \cdots & a3n \\ \vdots & \vdots & & \vdots \\ an1 & an2 & \cdots & ann \end{bmatrix} = \begin{bmatrix} s11 & & & \\ s12 & s22 & & \\ \vdots & \vdots & \ddots & \\ s1n & s2n & \cdots & snn \end{bmatrix} \begin{bmatrix} s11 & s12 & \cdots & s1n \\ & s22 & \cdots & s2n \\ & & \ddots & \vdots \\ & & & snn \end{bmatrix} .$$

Considering the rules of matrix multiplication,

$$a_{ij} = s_{1i}\, s_{1j} + s_{2i}\, s_{2j} + \ldots + s_{ii}\, s_{jj} \qquad i<j \qquad (101)$$

$$a_{ii} = s_{1i}^{\;2} + s_{2i}^{\;2} + \ldots\ldots + s_{ii}^{\;2} \qquad i=j \quad . \qquad (102)$$

Therefore the coefficients of the first row of S can be determined by

$$s_{11} = a_{11} \; ; \quad s_{1i} = a_{1j} / a_{11} \qquad (103)$$

and in general,

$$s_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} s_{ki}^{\;2}} \qquad (104)$$

$$s_{ij} = (a_{ij} - \sum_{k=1}^{i-1} s_{ki}\, s_{kj}) / s_{ii} \quad . \qquad (105)$$

Furthermore, the solution of the system

$$A X = F \qquad (106)$$

reduces to

$$S^T S X = F \quad , \qquad (107)$$

or

$$S^T C = F$$

$$S X = C \quad . \qquad (108)$$

The elements of C are determined from

$$c_1 = f_1 / s_{11} \tag{109}$$

and

$$c_i = \left(f_i - \sum_{k=1}^{i-1} s_{ki}\, c_k \right) / s_{ii} \qquad (i>1) \quad . \tag{110}$$

Once C is known, X can be found as

$$x_n = c_n / s_{nn} \tag{111}$$

and

$$x_i = \left(c_i - \sum_{k=i+1}^{n} s_{ik}\, x_k \right) / s_{ii} \qquad (i <n) \quad . \tag{112}$$

APPENDIX F

INPUT FORMAT AND LISTING OF THE "HTAFEM" PROGRAM

The purpose of this appendix is to define the input data which are needed in order to run the HTAFEM program. Moreover, other parameters which must be initially supplied to the program such that the input data are properly read by the program are defined.

The input is divided into three different sections. The number of the data card and the information which is provided to the program in each of these sections is :

I-TITLE CARD (format; 20A4)

Note	Columns	Variable	Entry
	1-80	TITLE	Enter the title for use in labeling the output.

II-NODAL POINT DATA CARDS (format; 3F10.5,I5)

Note	Columns	Variable	Entry
(1)	1-10	X-CORD	x-cordinates
	11-20	Y-CORD	y-coordinates
	21-30	PT	value of prescribed temperature
(2)	31-35	IDBC	Flag of Dirichilet boundary condition

NOTES :

(1) The total number of nodes (NNODE) controls the amount of data to be read in this section. This information must be supplied to the main program prior to the execution of the program (see page 66).

(2) The flag of Dirichilet boundary condition can only be assigned the following values :

IDBC=1; The node is subjected to prescribed temperature (PT),

IDBC=0; There is no prescribed temperature.

III-ELEMENTS DATA CARDS (format;6I5,4F10.5) (1)

Notes	Columns	Variable	Entry
(2)	1-5	NENN(I,1)	Node 1 of the element I.
	6-10	NENN(I,2)	Node 2 of the element I.
	11-15	NENN(I,3)	Node 3 of the element I.
(3)	16-20	INBC	Flag of Neumann boundary condition.
	21-25	IBCON(I)	Node I of the element which lies on the boundary.
	26-30	IBCON(J)	Node J of the element which lies on the boundary.
(4)	31-40	QD	Heat flux.
	41-50	Q	Heat generation within the element.

Notes	Columns	Variable	Entry
	51-60	H	Convection coefficient.
	61-70	TA	Ambient temperature.

NOTES :

(1) The total number of elements (NEL) controls the amount of data to be read in this section. This must be supplied to the main program prior to the execution of the program (see page 66).

(2) Numbering of the elements nodes must be counterclockwise.

(3) Side I-J of the element which is subjected to the Neumann boundary condition must be specified in a counterclockwise order. For example, if an element is numbered counterclockwise as 2,7, and 9, and if side 7-9 is subjected to boundary conditions, then IBCON(I)=7, and IBCON(J)=9. Moreover, only one side of an element can lie on the boundary surface.

(4) The heat flux into the body is negative.

A sample input data which corresponds to the mathematical model of Fig. 2, is shown in Fig. 13.

The HTAFEM program has been organized in a way that modifications to the program are localized. This is done by dividing the program into several subroutines. The organization of the program is illustrated in Fig. 14. When a problem is desired to be solved, the necessary parameters which control the memory allocation of the arrays and amount of input data to be read must be supplied to the main program. These parameters are : number of nodes (NNODE), number of elements (NEL), and number of the bandwidth (NBW). These parameters along with the thermal conductivity of the material (CC) are supplied to the main program by means of a DATA card, which has the following structure :

```
DATA NNODE,NEL,NBW,CC/ ---,---,---,---/  .
```

The arrays and their memory allocations which must be defined in the main program are summarized in Table 13.

The complete listing of the program, the flow charts, and a sample out-put which corresponds to the sample input illustrated in this appendix concludes this Appendix.

Table 13. Definition of the variables in the program

Name	definition
NNODE	number of nodes
NEL	number of elements
NBW	band width
CC	thermal conductivity
NENN(NEL,3)	element connectivity matrix
XCORD(NNODE)	x coordinates
YCORD(NNODE)	y coordinates
PT(NNODE)	prescribed temperature
IDBC(NNODE)	flag of Dirichilit b.c.'s
Q(NEL)	heat generation
QD(NEL)	heat flux
H(NEL)	convection heat-transfer coefficient
AT(NEL)	ambient temperature
IBCON(I,2)	location of Neumann b.c.'s for element I
INBC(NNODE)	flag of Neumann b.c.'s
A(NEL)	area of element
GCM(NNODE,NBW)	global conduction matrix
GF(NNODE)	global force vector
TEM(NNODE)	nodal temperature

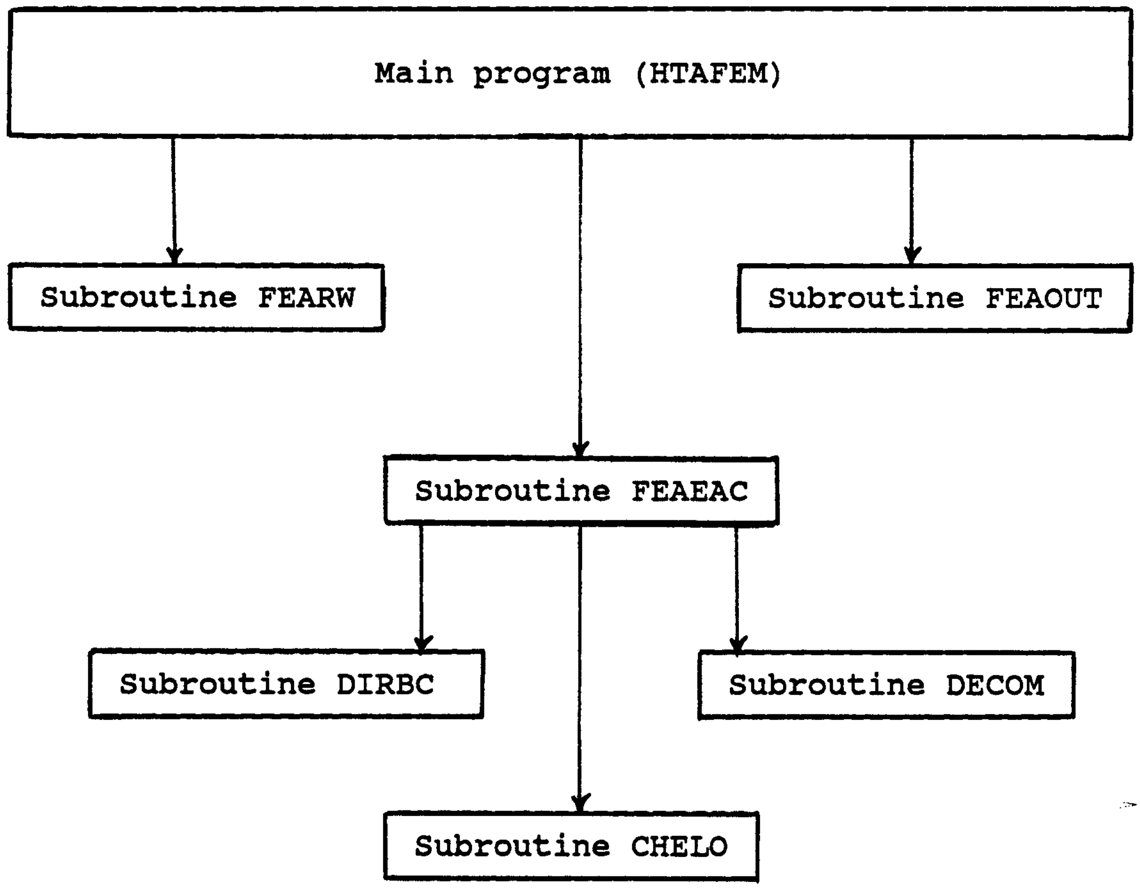


Fig. 14. Organization of the computer program

```
C*********************************************************************
C*    THIS PROGRAM IS FOR THE SOLUTION OF HEAT CONDUCTION IN A SOLID   *
C*    SUBJECTED TO CONVECTION,HEAT FLUX,HEAT GENERATION,AND PRESCRIBED *
C*    TEMPERATURE .                         FARAMARZ MOSSAYEBI         *
C*********************************************************************
C
C     THE PARAMETERS ARE :
C
C....NNODE=NUMBER OF NODES
C....NEL=NUMBER OF ELEMENT
C....NBW=BAND WIDTH
C....CC=THERMAL CONDUCTIVITY
C....NENN(NEL,3)=ELEMENT CONNECTIVITY MATRIX
C....XCORD,YCORD(NNODE)= NODAL COORDINATES
C....GCM(NNODE,NBW)=GLOBAL CONDUCTION MATRIX
C....GF(NNODE)=GLOBAL FORCE MATRIX
C....A(NEL)=AREA OF ELEMENT
C....Q(NEL)=HEAT GENERATION WITHIN AN ELEMENT
C....QD(NEL)=HEAT FLUX
C....H(NEL)=COEFFICIENT OF CONVECTION HEAT LOSS
C....AT(NEL)=AMBIENT TEMPERATURE
C....PT(NNODE)=VALUE OF PRESCRIBED TEMPERATURE
C....IDBC(NNODE)=BINARY FLAG OF DIRICHLET BOUNDARY CONDITIONS
C....INBC(NEL)=BINARY FLAG OF NUMEN BOUNDARY CONDITIONS
C....TEM(NNODE)=NODAL TEMPERATURE(SOLUTION).
C
C
C*********************************************************************
C*    THE DIMENSION OF THE ARRAYS AND THE "DATA CARD" MUST BE         *
C*    MODIFIED ACCORDING TO THE MODELING OF THE PROBLEM.  THIS        *
C*    MODIFICATION OCCURS ONLY IN THIS PROGRAM.                       *
C*********************************************************************
C
C
      DIMENSION NENN(8,3),XCORD(9),YCORD(9),GCM(9,4),GF(9),
     1A(8),INBC(8),IBCON(8,2),Q(8),PT(9),TEM(9),IDBC(9)
     2,QD(8),H(8),AT(8)
C
C...."DATA CARD"
C
      DATA NNODE,NEL,NBW,CC/9,8,4,236./
C
C....SUBROUTIN FEARW IS CALLED TO READ AND WRITE THE DATA
C
      CALL FEARW(NNODE,NEL,NBW,CC,NENN,XCORD,YCORD,GCM,GF,A,INBC,IBCON
     *,Q,PT,TEM,IDBC,QD,H,AT)
C
C.....SUBROUTINE FEAECA IS CALLED TO CALCULATE ELEMENT PROPERTY
C.....MATRICES. THIS SUBROUTINE ALSO ASSEMBLES THESE MATRICES AND
C.....CALLS THE PROPER SUBROUTINES TO MODIFY AND SOLVE THE SYSTEM
C.....EQUATIONS.
C
      CALL FEAECA(NNODE,NEL,NBW,NENN,XCORD,YCORD,CC,GCM,GF,A,INBC,
     *           IBCON,Q,TEM,PT,IDBC,QD,H,AT)
C
C.....SUBROUTINE FEAOUT IS CALLED TO WRITE THE SOLUTION.
      CALL FEAOUT(NNODE,TEM)
C
      STOP
      END
```

```
C***********************************************************************
C*    THIS SUBROUTINE READS AND WRITES THE INPUT DATA.                  *
C*                                                                      *
C***********************************************************************
C
C
      SUBROUTINE FEARW(NNODE,NEL,NBW,CC,NENN,XCORD,YCORD,GCM,GF,A,INBC,
     *                   IBCON,Q,PT,TEM,IDBC,QD,H,AT)
      DIMENSION NENN(NEL,3),XCORD(NNODE),YCORD(NNODE),GCM(NNODE,NBW),
     *GF(NNODE),A(NEL),INBC(NEL),IBCON(NEL,2),Q(NEL),PT(NNODE),QD(NEL),
     *H(NEL),AT(NEL),IDBC(NNODE),TEM(NNODE),TITLE(70)
C
       DO 10 I=1,NEL
       DO 10 J=1,3
10     NENN(I,J)=0
C
C
      READ(5,20)TITLE
20    FORMAT(70A1)
      WRITE(6,30)TITLE
30    FORMAT(' ****  ',70A1,///)
      WRITE(6,40)NNODE,NEL,NBW,CC
40    FORMAT('1TOTAL  NUMBER  OF  NODES =',5X,I5,/
     1       ' TOTAL NUMBER OF ELEMENTS =',5X,I5,/
     2       ' NUMBER   OF  BANDWIDTH   =',5X,I5,/
     3       ' THERMAL   CONDUCTIVETY   =',F15.8///)
C
      READ(5,50)(XCORD(I),YCORD(I),PT(I),IDBC(I),I=1,NNODE)
      WRITE(6,60)
      WRITE(6,70)(I,XCORD(I),YCORD(I),PT(I),IDBC(I),I=1,NNODE)
      READ(5,80)((NENN(I,J),J=1,3),INBC(I),(IBCON(I,K),K=1,2),Q(I),
     *QD(I),H(I),AT(I),I=1,NEL)
C
      WRITE(6,90)
      WRITE(6,100)(I,(NENN(I,J),J=1,3),INBC(I),(IBCON(I,JJ),JJ=1,2),
     *Q(I),QD(I),H(I),AT(I),I=1,NEL)
C
50    FORMAT(3F10.5,I5)
60    FORMAT('1',' NODE NUMBER',5X,'X_COORDINATE',9X,'Y_COORDINATE',9X
     * ,'PRES. TEMPERATURE ', 2X,'IDBC'/)
70    FORMAT(4X,I3,10X,F12.5,10X,F12.5,10X,F12.5,5X,I5)
80    FORMAT(6I5,4F10.5)
90    FORMAT(///33X,'E L E M E N T   C O N E C T I V I T Y   A N D   ',
     *'P R O P E R T I E S'////,'  ELMT   I    J    K',3X,
     *'INBC',2X,'IBCON',9X,'Q',14X,'QD',14X,'H',
     *15X,'AT',/29X,'I',4X,'J'///)
100   FORMAT(7I5,4F15.5)
C
C
      RETURN
      END
```

```
C************************************************************************
C*    THIS SUBROUTINE CALCULATES THE ELEMENT PROPERTY MATRICES AND      *
C*    ASSEMBLES THEM.                                                   *
C*                                                                      *
C************************************************************************
C
C
      SUBROUTINE FEAECA(NNODE,NEL,NBW,NENN,XCORD,YCORD,CC,GCM,GF,A,INBC,
     1                IBCON,Q,TEM,PT,IDBC,QD,H,AT)
      DIMENSION XCORD(NNODE),YCORD(NNODE),NENN(NEL,3),GCM(NNODE,NBW),
     1        GF(NNODE),A(NEL),INBC(NEL),IBCON(NEL,2),Q(NEL),PT(NNODE),
     2      GK(3,3),GKH(3,3),FQ(3),FQP(3),FH(3),TEM(NNODE),IDBC(NNODE)
     3        ,QD(NEL),H(NEL),AT(NEL)
C     DOUBLE PRECISION DIFF(1)
C
C.....INITIALIZE THE GLOBAL COEFFICIENT AND TEMPERATURE MATRICES
C
      DO 10 I=1,NNODE
      GF(I)=0.0
      DO 10 J=1,NBW
      GCM(I,J)=0.0
10    CONTINUE
C
C.....OBTAIN LOCAL X & Y CORDINATES
C
C
      DO 100 I=1,NEL
      N1=NENN(I,1)
      N2=NENN(I,2)
      N3=NENN(I,3)
      X1=XCORD(N1)
      X2=XCORD(N2)
      X3=XCORD(N3)
      Y1=YCORD(N1)
      Y2=YCORD(N2)
      Y3=YCORD(N3)
      B1=Y2-Y3
      B2=Y3-Y1
      B3=Y1-Y2
      C1=X3-X2
      C2=X1-X3
      C3=X2-X1
C
C.....COMPUTE THE AREA OF EACH ELEMENT
C
C
      A(I)=(C3*B1-C1*B3)/2.0
      A(I)=ABS(A(I))
C
C.....COMPUTE ELEMENT COEFFICIENT MATRIX
C
      GK(1,1)=B1**2+C1**2
      GK(1,2)=B1*B2+C1*C2
      GK(1,3)=B1*B3+C1*C3
      GK(2,2)=B2**2+C2**2
      GK(2,3)=B2*B3+C2*C3
      GK(3,3)=B3**2+C3**2
```

```
      GK(2,1)=GK(1,2)
      GK(3,1)=GK(1,3)
      GK(3,2)=GK(2,3)
C
C
C.....INITIALIZE THE MATRICES AND INTRODUCE HEAT GENERATION
C
      DO 20 M=1,3
      FQ(M)=QD(I)*A(I)/3.0
      FQP(M)=0.0
      FH(M)=0.0
      DO 20 N=1,3
      GK(M,N)=GK(M,N)*CC/(4.0*A(I))
      GKH(M,N)=0.0
20    CONTINUE
C
C.....INTRODUCE THE HEAT FLUX AND CONVECTION BOUNDARY CONDITIONS
C
      IF (INBC(I).EQ.0) GO TO 30
C
C.....DETERMINE WHICH SIDE OF ELEMENT IS SUBJECTED TO BOUNDARY CONDITION
C
      NNI=IBCON(I,1)
      IF (NNI.EQ.N1) GO TO 40
C
      IF (NNI.EQ.N3) GO TO 50
C
      SEJK=SQRT((X3-X2)**2+(Y3-Y2)**2)
      CT=H(I)*SEJK/6.0
      GKH(2,2)=2.0*CON
      GKH(2,3)=CT
      GKH(3,2)=CT
      GKH(3,3)=2.0*CT
      FQP(2)=Q(I)*SEJK/2.0
      FQP(3)=FQP(2)
      FH(2)=H(I)*SEJK*AT(I)/2.0
      FH(3)=FH(2)
      GO TO 30
40    SEIJ=SQRT((X2-X1)**2+(Y2-Y1)**2)
C
      CT=H(I)*SEIJ/6.0
      GKH(1,1)=2.0*CT
      GKH(1,2)=CT
      GKH(2,1)=CT
      GKH(2,2)=2.0*CT
      FQP(1)=Q(I)*SEIJ/2.0
      FQP(2)=FQP(1)
      FH(1)=H(I)*SEIJ*AT(I)/2.0
      FH(2)=FH(1)
      GO TO 30
50    SEKI=SQRT((X1-X3)**2+(Y1-Y3)**2)
      CT=H(I)*SEKI/6.0
      GKH(1,1)=2.0*CT
      GKH(1,3)=CT
      GKH(3,1)=CT
      GKH(3,3)=2.0*CT
      FQP(1)=Q(I)*SEKI/2.0
      FQP(3)=FQP(1)
      FH(1)=H(I)*SEKI*AT(I)/2.0
      FH(3)=FH(1)
```

```
30      CONTINUE
C
C.....ASSEMBLE THE GLOBAL PROPERTY MATRICES
C
        DO 60 M=1,3
        IM=NENN(I,M)
        GF(IM)=GF(IM)-FQP(M)+FH(M)+FQ(M)
        DO 60 N=1,3
        IN=NENN(I,N)-IM+1
        IF (IN.LE.0) GO TO 60
        GCM(IM,IN)=GCM(IM,IN)+GK(M,N)+GKH(M,N)
60      CONTINUE
100     CONTINUE
C
C.....INTRODUCE THE DIRICHLET BOUNDARY CONDITIONS
C
        DO 70 M=1,NNODE
        IF(IDBC(M).EQ.0) GO TO 70
        ST=PT(M)
C
C.....CALL SUBROUTINE DIRBC
C
        CALL DIRBC (GCM,GF,NNODE,NBW,M,ST)
70      CONTINUE
C
C.....SOLVE THE SYSTEM OF EQUATIONS
C
        DO 80 M=1,NNODE
80      TEM(M)=GF(M)
C
C
        CALL DECOM (NNODE,NBW,GCM)
        CALL CHOLE (NNODE,NBW,GCM,TEM)
C
        RETURN
        END
```

```
C***********************************************************************
C*      THIS SUBROUTINE MODIFIES THE CONDUCTION MATRIX (GCM) BY        *
C*      INTRODUCING THE SPECIFIED NODAL TEMPERATURES,IE DIRICHLET      *
C*      BOUNDARY CONDITIONS.                                           *
C***********************************************************************
        SUBROUTINE DIRBC(GCM,GF,NNODE,NBW,M,PT)
        DIMENSION GCM(NNODE,NBW),GF(NNODE)
        DO 10 K=2,NBW
        I1=M-K+1
        I2=M+K-1
        IF(I1.GE.1) GF(I1)=GF(I1)-GCM(I1,K)*PT
10      IF(I2.LE.NNODE) GF(I2)=GF(I2)-GCM(M,K)*PT
        GF(M)=PT
        DO 20 J=1,NBW
        I1=M-J+1
        IF  (I1.GE.1) GCM(I1,J)=0.0
20      GCM(M,J)=0.0
        GCM(M,1)=1.0
        RETURN
        END
```

```
C**********************************************************************
C*    THIS SUBROUTINE PERFORMS THE DECOMPOSITION OF THE GLOBAL        *
C*    CONDUCTION MATRIX INTO AN UPPER TRIANGULAR MATRIX.              *
C*    COPIED FROM REFERENCE 9, WITH SOME MODIFICATIONS.               *
C**********************************************************************
C
      SUBROUTINE DECOM(NNODE,NBW,A)
      DIMENSION A(NNODE,NBW)
      DOUBLE PRECISION D
      A(1,1)=SQRT(A(1,1))
      DO 10 I=2,NBW
10    A(1,I)=A(1,I)/A(1,1)
      DO 20 I=2,NNODE
      I1=I+1
      I2=I-1
      D=A(I,1)
      DO 30 J=1,I2
      I3=I+1-J
      IF (I3.GT.NBW) GO TO 30
      D=D-A(J,I3)**2
30    CONTINUE
      A(I,1)=DSQRT(D)
      DO 40 IJ=2,NBW
      IF(I+IJ-1.GT.NNODE) GO TO 20
      D=A(I,IJ)
      DO 50 J=1,I2
      I3=I+1-J
      I4=I-J+IJ
      IF (I4.GT.NBW)GO TO 50
      IF (I3.GT.NBW)GO TO 50
      D=D-A(J,I3)*A(J,I4)
50    CONTINUE
40    A(I,IJ)=D/A(I,1)
20    CONTINUE
      RETURN
      END


C**********************************************************************
C*    THIS SUBROTINE WRITES THE FINAL SOLUTION.
C*                                                                    *
C**********************************************************************
      SUBROUTINE FEAOUT(NNODE,TEM)
      DIMENSION TEM(NNODE)
C
      WRITE(6,10)
10    FORMAT (///' NODE NUMBER',10X,' TEMPERATURE'//)
      DO 20 I=1,NNODE
20    WRITE(6,30)I,TEM(I)
30    FORMAT(4X,I5,13X,F13.6)
C
      RETURN
      END
```

```
C*********************************************************************
C*    THIS SUBROUTINE PERFORMS THE SOLUTION OF THE SYSTEM EQUATIONS    *
C*    USING THE UPPER TRIANGULAR MATRIX WHICH IS OBTANIED BY "DECOM".  *
C*    COPIED FROM REFERENCE 9, WITH SOME MODIFICATIONS.                *
C*********************************************************************
C
      SUBROUTINE CHOLE (NNODE,NBW,A,B)
      DIMENSION A(NNODE,NBW),B(NNODE)
      DOUBLE PRECISION D
C
C
      B(1)=B(1)/A(1,1)
      DO 10 I=2,NNODE
      D=B(I)
C
C
      DO 20 J=2,NBW
      I1=I+1-J
      I2=I+1-I1
      IF (I1.LT.1) GO TO 20
      IF (I2.GT.NBW) GO TO 20
      D=D-A(I1,I2)*B(I1)
20    CONTINUE
      B(I)=D/A(I,1)
10    CONTINUE
      B(NNODE)=B(NNODE)/A(NNODE,1)
      DO 30 I=2,NNODE
      I3=NNODE+1-I
      D=B(I3)
      DO 40 J=2,NBW
      I4=I3-1+J
      IF(I4.GT.NNODE) GO TO 40
      D=D-A(I3,J)*B(I4)
40    CONTINUE
      B(I3)=D/A(I3,1)
30    CONTINUE
      RETURN
      END
```

THE ABSORBER PLATE WITH 9 NODES AND 8 ELEMENTS } Section I

Section II

COORDINATES X	Y	PRES. TEMP.	FLAG
0.0	0.0	50.0	1
0.0	0.30	0.0	0
0.0	0.6	53.0	1
0.075	0.0	50.0	1
0.075	0.30	0.0	0
0.075	0.6	53.0	1
0.15	0.0	50.0	1
0.15	0.30	0.0	0
0.15	0.6	53.0	1

Section III

ELEMENT CONEC. I	J	K	FLAG	SIDE I	SIDE J	HEAT FLUX	HEAT GENER.	CONVECTION COEFFICIENT	AMBIENT TEMP.
1	4	2	0	0	0	0.0	139433.07	0.0	0.0
2	4	5	0	0	0	0.0	139433.07	0.0	0.0
3	2	5	0	0	0	0.0	139433.07	0.0	0.0
3	5	6	0	0	0	0.0	139433.07	0.0	0.0
5	4	7	0	0	0	0.0	139433.07	0.0	0.0
7	8	5	1	7	8	0.0	139433.07	304.39	51.00
5	8	6	0	8	9	0.0	139433.07	304.39	51.50
8	9	6	1	8	9	0.0	139433.07	304.39	51.0

DMSLI0740I EXECUTION BEGINS...
**** SOLAR PLATE, 9 NODES AND 8 ELEMENTS

TOTAL NUMBER OF NODES = 9
TOTAL NUMBER OF ELEMENTS = 8
NUMBER OF BANDWIDTH = 4
THERMAL CONDUCTIVETY = 236.000000

NODE NUMBER	X_COORDINATE	Y_COORDINATE	PRES. TEMPERATURE	IDBC
1	0.0	0.0	50.00000	1
2	0.0	0.30000	0.0	0
3	0.0	0.60000	53.00000	1
4	0.07500	0.0	50.00000	1
5	0.07500	0.30000	0.0	0
6	0.07500	0.60000	53.00000	1
7	0.15000	0.0	50.00000	1
8	0.15000	0.30000	0.0	0
9	0.15000	0.60000	53.00000	1

E L E M E N T   C O N E C T I V I T Y   A N D   P R O P E R T I E S

ELMT	I	J	K	INBC	IBCON I	IBCON J	Q	QD	H	AT
1	1	4	2	0	0	0	0.0	139433.062	0.0	0.0
2	2	4	5	0	0	0	0.0	139433.062	0.0	0.0
3	3	2	5	0	0	0	0.0	139433.062	0.0	0.0
4	3	5	6	0	0	0	0.0	139433.062	0.0	0.0
5	5	4	7	0	0	0	0.0	139433.062	0.0	0.0
6	7	8	5	1	7	8	0.0	139433.062	304.38989	51.00000
7	5	8	6	0	8	9	0.0	139433.062	304.38989	51.50000
8	8	9	6	1	8	9	0.0	139433.062	304.38989	51.00000

NODE NUMBER	TEMPERATURE
1	50.000000
2	73.121613
3	53.000000
4	50.000000
5	72.811249
6	53.000000
7	50.000000
8	71.841431
9	53.000000

BIBLIOGRAPHY

1. Clough, R.W. "The Finite Element Method in Plane Stress Analysis", Proceedings of 2nd ASCE Conference on Electronic Computation, Pittsburgh, Pa., September 8 and 9, 1960.

2. Turner, M.J., et al." Stiffness and Deflection Analysis of Complex Structure", Journal of Aeronautical Sciences, 23, 1956.

3. Zienkiewicz, O.C. and Cheung, Y.K. "Finite Element in the Solution of Field problems", The Engineer, vol. 220, 1965.

4. Szabo, A. Barnce and Lee, George C. "Derivation of Stiffness Matrices for Problems in Plane Elasticity by Galerkin's Method", International Journal of Numerical Methods in Engineering, vol. 1, 1969.

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

6. Kreith, frank and Kreider, Jan F. Principles of Solar Engineering, Mcgraw-Hill, London, 1978, pp. 204-208.

7. Huebner, Kenneth A. The Finite Element Method for Engineers, John wiley and Sons, New York, 1975, p. 6.

8. Brebbia, C.A. and Ferrante, A.J. Computational Methods for the Solution of Engineering Problems, Pentech press, New York, 1978, p. 123.

9. Rao, S.S. The Finite Element Method in Engineering, Pergmon Press, New York, 1982, pp. 119-121.

10. Segerlind, L.J. Applied Finite Element Analysis, second edition, John Wiley and Sons, New York, 1984, pp. 73-77

11. Segerlind, L.J. Applied Finite Element Analysis, second edition, John Wiley and Sons, New York, 1984, pp. 77-78

12. Stasa, Frank L. Applied Finite Element Analysis For Engineers, CBS College Publishing, 1985, pp. 121-129.

13. Kreith, Frank and Black, William Z. Basic Heat Transfer, Harper and Row, New York, 1980, pp. 243-245.

14. Brebbia, C.A. and Ferrante, A.J. Computational Methods for the Solution of Engineering Problems, Pentech press, New York, 1978, pp. 57-61

REFERENCES

Akin, J.E. Application and Implementation of Finite Element Method. New York : Academic Press, 1982.

Hetter, Robert L. and Prawel, Sherwood P. Modern methods of Engineering Computations. New York : McGraw-Hill, 1969.

Rao, S.S. The Finite Element Method in Engineering. New York : Pergman Press, 1982.

Segrelind, L.J. Applied Finite Element Analysis, Second Edition. New York : John Wiley and Sons, 1984.

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