TASEF-2 : A Computer Program for Temperature Analysis of Structures Exposed to Fire

Wickström, Ulf

1979

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
ULF WICKSTRÖM

TASEF-2 — A COMPUTER PROGRAM FOR TEMPERATURE ANALYSIS OF STRUCTURES EXPOSED TO FIRE
TASEF-2 - A Computer Program for Temperature Analysis of Structures Exposed to Fire

Ulf Wickström
ACKNOWLEDGEMENTS

The people who work in the Division of Structural Mechanics and Concrete Construction here in Lund have helped to create an atmosphere in which I could finish the work reported in this dissertation. For this they deserve my heartfelt thanks.

Professor Ove Pettersson has for many years been both in the fore- and background of my work. As head of this Division and as advisor for my doctoral work, his example has helped me to define my goals as a research engineer.

Professor Hans Petersson has graciously and carefully reviewed the manuscript of this work again and again as it has evolved. Hans has the gift of making one believe that requests for advice are never an imposition and that he gains as much from any interchange as does the student. While I cannot believe this to be true in my case, I thank him both for his criticisms, but especially for his patience and time.

I also wish to acknowledge Judith Sanders for her proficient assistance in editing this report and her patience in encouraging and supporting me in my endeavor.

Tarja Aunola and Bo Zadig typed the report and drew the figures, and advised me on the design of the tables and figures.

The project has been partly sponsored by the National Swedish Council for Building Research.
TABLE OF CONTENTS

ACKNOWLEDGEMENT

TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>INTRODUCTION</td>
<td>5</td>
</tr>
<tr>
<td>2.</td>
<td>HEAT TRANSFER ANALYSIS</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>Basic Equations</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Initial and Boundary Conditions</td>
<td>8</td>
</tr>
<tr>
<td>3.</td>
<td>FINITE ELEMENT APPROXIMATION</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>Solution Techniques</td>
<td>12</td>
</tr>
<tr>
<td>3.2</td>
<td>Basic Approximations</td>
<td>12</td>
</tr>
<tr>
<td>3.3</td>
<td>Matrix Equilibrium Equations for Transient Heat Conduction</td>
<td>13</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Conductivity Matrix</td>
<td>15</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Heat Capacity and Volume Matrices</td>
<td>17</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Internally Generated Heat</td>
<td>18</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Boundary Heat Flow</td>
<td>18</td>
</tr>
<tr>
<td>3.4</td>
<td>Time Integration</td>
<td>21</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Forward Differences</td>
<td>22</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Critical Time Increment</td>
<td>24</td>
</tr>
<tr>
<td>4.</td>
<td>COMPUTER PROGRAM</td>
<td>27</td>
</tr>
<tr>
<td>5.</td>
<td>EXAMPLES</td>
<td>31</td>
</tr>
<tr>
<td>5.1</td>
<td>Example I - Square Plate Subjected to Heat Transfer from Surrounding Gas</td>
<td>31</td>
</tr>
<tr>
<td>5.2</td>
<td>Steel Beams Embedded in Concrete</td>
<td>32</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Material Properties and Boundary Conditions</td>
<td>32</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Example II - Wide-Flange I-beam</td>
<td>42</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Example III - Box Girder</td>
<td>43</td>
</tr>
<tr>
<td>6.</td>
<td>SUMMARY AND CONCLUSIONS</td>
<td>53</td>
</tr>
<tr>
<td>6.1</td>
<td>Present Study</td>
<td>53</td>
</tr>
<tr>
<td>6.2</td>
<td>Future Development</td>
<td>54</td>
</tr>
</tbody>
</table>
7. REFERENCES

APPENDIX A - USER'S MANUAL

APPENDIX B - PROGRAM LISTING OF TASEF-2

APPENDIX C - EXAMPLE INPUT
1. INTRODUCTION

A nonlinear heat flow equation must be solved to predict the distribution of temperature in a structure exposed to fire. Since analytical solutions of such equations exist only for idealized cases, numerical schemes that incorporate either the finite element or finite difference method have generally been employed to approximate heat conduction [1-5].

Ödeen computed temperature distribution in homogeneous concrete cross-sections exposed to fire [1] using a program based on the finite difference method. Latent heat due to evaporation of water was considered in the calculation, but only structures with simple geometries were analyzed. Based on work by Wilson et al. [2,3] the finite element programs FIRES-T [4] and later FIRES-T3 [5] were developed for analyzing thermal response of structures exposed to fire. An implicit backward difference time integration scheme is used in these programs. Computation therefore often becomes unnecessarily expensive, and materials with latent heat - for instance humid concrete - cannot be analyzed accurately.

In this report TASEF-2 (Temperature Analysis of Structures Exposed to Fire - Two Dimensional Version) a computer program based on the finite element method is described. Structures comprised of one or more materials and structures that enclose voids can be analyzed. Heat transferred by convection and radiation at the boundaries can be modeled. The explicit forward difference time integration scheme used in TASEF-2 facilitates consideration of latent heat in the calculation of temperature in materials such as humid concrete. The maximum length of the time increment that can be used without inducing numerical instability is discussed, and some procedures to avoid very short time steps are suggested. In the present version of the program two-dimensional rectangular elements are used; input of the geometry and generation of the finite element mesh have been automated.
In the report, the theoretical model and solution techniques are derived, the organization of the computer program is explained, and a commentary on practical aspects of using the program is made. Several examples are analyzed using TASEF-2 and calculated temperatures are in some cases compared to experimental results. The report contains fully annotated input instructions, and a listing of the program.
2. HEAT TRANSFER ANALYSIS

2.1 Basic Equations

The governing equations for heat conduction are the heat balance equilibrium equation

\[ \nabla q + \dot{e} - Q = 0 \quad (2.1) \]

and the Fourier law

\[ q = -k \nabla T \quad (2.2) \]

where \( q \) is the heat flow vector, \( \dot{e} = \frac{\partial e}{\partial t} \) the rate of specific volumetric enthalpy change, \( Q \) the rate of internally generated heat per unit volume, \( k \) a symmetric positive definite thermal conductivity matrix, \( T \) temperature, and \( t \) time. For isotropic materials

\[ k = k I \quad (2.3) \]

where \( k \) is thermal conductivity, and \( I \) the identity matrix. The gradient operator \( \nabla \) is defined as

\[ \nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \quad (2.4) \]

where \( x, y, \) and \( z \) are Cartesian coordinates. Equation (2.2) is substituted into Equation (2.1) to yield the transient heat flow equation

\[ -T^T (k \nabla T) + \dot{e} - Q = 0 \quad (2.5) \]
Specific volumetric enthalpy is by definition

\[ e = \int_{T_0}^{T} c_p dT + \sum \ell_i \]  \hspace{1cm} (2.6)

where \( T_0 \) is a reference temperature, usually zero, \( c_p \) specific heat, \( \rho \) density, and \( \ell_i \) latent volumetric heat due to phase changes at various temperature levels. The time derivative of

\[ \dot{e} = c_p \dot{T} \]  \hspace{1cm} (2.7)

where \( \dot{T} = \frac{\partial T}{\partial t} \) is rate of temperature change. Substitution of Equation (2.7) into Equation (2.5) yields the conventional form of the transient heat flow equation

\[ -\nabla \cdot (k \nabla T) + c_p \dot{T} - Q = 0 \]  \hspace{1cm} (2.8)

Nominal specific volumetric heat \( \overline{c_p} \) will be defined by the equation

\[ e = \overline{c_p} T \]  \hspace{1cm} (2.9)

In Figure 2.1 specific volumetric enthalpy is plotted versus temperature for a material with latent heat indicated by a step \( \ell \) in the curve. The tangential and secantial or nominal volumetric specific heats, \( c_p \) and \( \overline{c_p} \), respectively, are then as shown in Figure 2.1. Note that at the temperature \( T_\ell \), where the enthalpy curve is stepped, the value of \( c_p \) is undefined while the value of \( \overline{c_p} \) is always finite.

2.2 Initial and Boundary Conditions

Initial and boundary conditions must be specified in order to solve Equations (2.5) or (2.8). An initial condition is given by specifying the distribution of temperature in a body at a reference time zero. Boundary con-
Figure 2.1. Definitions of specific volumetric heat conditions are prescribed as temperature or heat flow on parts of the boundary $\partial V_T$ and $\partial V_q$, respectively. The total boundary is then defined by

$$\partial V = \partial V_T + \partial V_q$$ (2.10)

Temperature on the boundary $\partial V_T$ of a body is specified as

$$T = T(x, y, z, t)$$ (2.11)

Heat flow normal to a surface must satisfy the heat balance equation

$$q_n = n^T q = -n^T k \nabla T$$ (2.12)
where $\mathbf{n}$ is the outward normal to the surface. Specified heat flow on $\partial V_q$ therefore is

$$\hat{q}_n = -\mathbf{n} \cdot \mathbf{T} - k \nabla T$$  \hspace{1cm} (2.13)

where $\hat{q}_n$ is prescribed heat flow.

At free surfaces heat is transferred by convection and radiation. These phenomena are complex and difficult to model, but approximate formula can be used. Convection heat transfer is thus calculated as

$$\hat{q}_n^c = \beta (T_s - T_g)^\gamma$$  \hspace{1cm} (2.14)

where $\hat{q}_n^c$ is the rate of heat transferred by convection, $\beta$ and $\gamma$ are the convection factor and power, respectively, and $T_s$ and $T_g$ are the surface and surrounding gas temperatures, respectively.

Radiation heat flux from a surface is approximated by

$$\hat{q}_n^r = \varepsilon_r \sigma (T_s^4 - T_g^4)$$  \hspace{1cm} (2.15)

where $\sigma$ is the Stefan-Boltzmann constant, and $T_s$ and $T_g$ are absolute surface temperature and absolute surrounding gas temperature, respectively. Resultant emissivity $\varepsilon_r$ varies with surface properties and geometric configuration. If the surface considered is small compared with a surrounding environment at uniform temperature $T_g$, resultant emissivity will be equal to surface emissivity $\varepsilon_s$ [6]. When assessing radiation between flames and structures in fire engineering design, resultant emissivity is sometimes calculated assuming radiation between two infinitely long parallel planes [7]; thus,

$$\varepsilon_r = \frac{1}{1/\varepsilon_s + 1/\varepsilon_g^{-1}}$$  \hspace{1cm} (2.16)
where $\varepsilon_q$ is appropriate gas or flame emissivity.

The total heat flux at a boundary is calculated by adding the contributions of convection and radiation:

$$\hat{q}_n = \hat{q}_n^c + \hat{q}_n^r \quad (2.17)$$
3. FINITE ELEMENT APPROXIMATION

3.1 Solution Techniques

Since analytical solutions of heat transfer problems are feasible only for linear applications with simple geometries and boundary conditions, a numerical method is used to solve the heat balance equation stated in Chapter 2 for temperature distribution in structural elements. The finite element method is used since it is general with respect to geometry, material properties, and boundary conditions. Nonlinear boundary conditions and the temperature dependence of material properties can be considered when the finite element method described in this chapter is used to analyze temperature distribution in fire-exposed structural elements.

3.2 Basic Approximations

In the finite element method of analysis a solid continuum is idealized by an assemblage of discrete elements. These elements may be of variable size and shape, and connected at a finite number of nodal points. The element boundaries are often linear, although if isoparametric elements are used, curved boundaries can be considered.

The temperature field within each element is approximated by a set of interpolation or shape functions \( N_i \), chosen so as to define temperature uniquely within each finite element in terms of its nodal temperatures \( T_i \). Temperature is thus approximated as

\[
T = \sum_i N_i(x,y,z) T_i(t) = \mathbf{N} \mathbf{T}
\]

(3.1)

The time differentiation of the temperature is

\[
\dot{T} = \mathbf{N} \dot{\mathbf{T}}
\]

(3.2)
Each shape function $N_\alpha$ is constructed so that it has the value 1 at node $i$ and is zero at all other nodes. In elements adjacent to node $i$, $N_\alpha$ takes values less than unity, and in other elements it vanishes [8].

3.3 Matrix Equilibrium Equations for Transient Heat Conduction

The heat balance equilibrium equation for transient heat conduction in matrix form can be derived by various methods. The method of weighted residuals will be used here. Thus, Equation (2.9) is substituted into the heat balance equation, Equation (2.5); the resulting expression is multiplied by a weighting function $v$ and integrated over the body [8]:

$$
\int_V v(-\nabla^T \kappa \nabla T + \frac{\partial}{\partial t}(\bar{c}\rho T) - Q) dV = 0 \tag{3.3}
$$

The first term is integrated by parts (Green's formula):

$$
\int_V v(-\nabla^T \kappa \nabla T) dV = -\int_{\partial V} \frac{\partial}{\partial n} \kappa \nabla T dS + \int_V (v \nabla T)^T \kappa \nabla T dV \tag{3.4}
$$

where $n$ is the outward normal to the boundary $\partial V$. A set of weighting functions $v_i$ equal to the shape functions $N_\alpha$ (the Galerkin method) is then chosen, i.e.

$$
v_i = N_\alpha \tag{3.5}
$$

Equations (3.1, 3.2, 3.4, 3.5) are substituted into Equation (3.3), yielding the matrix heat balance equation

$$
\int_V \left[ (v N)^T \kappa \nabla N dV \right] T + \frac{\partial}{\partial t} \left[ \int_V \bar{c}\rho N dV T \right] = \\
= \int_V N^T Q dV + \int_{\partial V} N^T n^T \kappa \nabla T dS \tag{3.6}
$$
or

\[ F_T + \frac{\partial}{\partial t}(E) = F_Q + F_q \]  \hspace{1cm} (3.7)

where \( F_T \), \( F_Q \) and \( F_q \) are vectors of nodal heat flow due to conduction, enthalpy or heat stored in elements adjacent to nodes, rate of internally generated heat per unit volume, and rate of heat flow supplied at the boundary, respectively. The vector of internal heat flow due to conduction is

\[ F_T = K T \]  \hspace{1cm} (3.8)

where \( K \) is the heat conductivity matrix.

Equation (3.8) is substituted into Equation (3.7) to yield

\[ K T + \frac{\partial}{\partial t}(E) = F \]  \hspace{1cm} (3.9)

where

\[ F = F_Q + F_q \]  \hspace{1cm} (3.10)

The nodal enthalpy vector is

\[ E = C T \]  \hspace{1cm} (3.11)

where \( C \) is the nominal heat capacity matrix. This expression is substituted into Equation (3.9)

\[ K T + \frac{\partial}{\partial t}(C T) = F \]  \hspace{1cm} (3.12)

Alternatively, the heat balance equation can be expressed in terms of nodal enthalpy rather than in terms of temperature

\[ K^* E + \frac{\partial}{\partial t}(E) = F \]  \hspace{1cm} (3.13)
where

\[ K^* = K C^{-1} \]  \hspace{1cm} (3.14)

The integrals in Equation (3.6) are evaluated over all elements \( m \) and boundary elements \( \partial m \). Thus

\[ K = \sum_m K^m \]  \hspace{1cm} (3.15)

\[ C = \sum_m C^m \]  \hspace{1cm} (3.16)

\[ F = \sum_m F^m + \sum_{\partial m} F^{\partial m} \]  \hspace{1cm} (3.17)

where

\[ K_{ij}^m = \int_{V^m} (\nabla N_i)^T k(\nabla N_j) dV \]  \hspace{1cm} (3.18)

\[ C_{ij}^m = \int_{V^m} \bar{\sigma} \nabla N_j dV \]  \hspace{1cm} (3.19)

\[ F_{Qi}^m = \int_{V^m} N_i Q dV \]  \hspace{1cm} (3.20)

and

\[ F_{Qi}^{\partial m} = \int_{\partial V^m} N_i \nabla \cdot k \nabla T dS \]  \hspace{1cm} (3.21)

\( V^m \) and \( \partial V^m \) are element volumes and boundary element surfaces, respectively.

The integrals of Equations (3.18-3.21) are often solved numerically by Gaussian quadrature. Explicit expressions can be derived for simple rectangular two dimensional elements as used in TASEF-2, as will be shown in the following sections.

3.3.1 Conductivity Matrix

In this section the element conductivity matrix \( K^m \) for the simple two-dimensional rectangular element used in
Figure 3.1. Rectangular finite element

Program TASEF-2 will be derived. Consider the rectangular element with sides parallel with the axes and of lengths a and b as shown in Figure 3.1. Make the variable substitutions

\[ \xi = \frac{(x - x_0)}{a} \]  

and

\[ \eta = \frac{(y - y_0)}{b} \]  

where \( \xi \) and \( \eta \) are dimensionless coordinates in a local system. A set of allowable shape functions is then

\[ N_i = \frac{(1 + \xi \xi_i)(1 + \eta \eta_i)}{4} \]  

where \( i \) takes values from 1 to 4, and
\[
\n\begin{align*}
\n\n\n\n\begin{bmatrix}
\frac{\partial N_i}{\partial x} \\
\frac{\partial N_i}{\partial y}
\end{bmatrix} = \begin{bmatrix}
\frac{1}{a} & \frac{1}{b}
\end{bmatrix}
\begin{bmatrix}
\frac{\partial N_i}{\partial x} \\
\frac{\partial N_i}{\partial y}
\end{bmatrix} = 1/4 \begin{bmatrix}
\frac{\xi_i}{a} (1 + n \eta_i) \\
\frac{\eta_i}{b} (1 + \xi \xi_i)
\end{bmatrix}
\end{align*}
\]
\]
\[
\text{Equation (3.25) is substituted into Equation (3.18) and constant thickness } d \text{ and conductivity } k \text{ are assumed for the element. Thus the local element conductivity matrix } K_i \text{ is, after evaluation of simple integrals,}
\]
\[
K_i = \frac{1}{3} \frac{kd}{ab} \begin{bmatrix}
\frac{a^2 + b^2}{2} - b^2 & \frac{a^2}{2} - \frac{b^2}{2} - a^2 \\
\frac{a^2}{2} - \frac{b^2}{2} - a^2 & \frac{b^2}{2} - \frac{a^2}{2} \\
\frac{a^2 + b^2}{2} - b^2 & \frac{b^2}{2} - \frac{a^2}{2}
\end{bmatrix}
\]
\[
\text{If conductivity } k \text{ for a particular application varies with temperature, } k \text{ at average nodal temperature is used in the calculation.}
\]

3.3.2 Heat Capacity and Volume Matrices

The computation of the element heat capacity matrix $C_i$ as given in Equation (3.19) results in a fully populated matrix identical in form to the element conductivity matrix $K_i$. The assembled heat capacity matrix $C$ is symmetric, positive-definite, and has the same nonzero structure as the system conductivity matrix $K$. The element heat capacity matrix $C_i$ can, however, be approximated by a lumped diagonal matrix with no loss of accuracy. The lumping eliminates the coupling between the time rate-of-change of temperature at adjacent nodes and results in a diagonal heat capacity matrix $C$. Such an approximation facilitates solution of the heat balance equation as will be shown in Section 3.4.
The lumped element heat capacity matrix $C_{ii}^m$ is formed in TASEF-2 as:

$$C_{ii}^m = \bar{c}_p(T_i) w_{ii}$$  \hspace{1cm} (3.27)

where $\bar{c}_p(T_i)$ is nominal specific volumetric heat capacity at nodal temperature $T_i$ and $w_{ii}$ the volume of element $m$ associated with node $i$. For rectangular 4-node elements the volume associated with each node is a quarter of an element. If all elements connected at a node $i$ are of the same material, the lumped heat capacity matrix can be stated as:

$$C_{ii} = \bar{c}_p(T_i) w_{ii}$$  \hspace{1cm} (3.28)

where

$$w_{ii} = \sum_m w_{ii}^m$$  \hspace{1cm} (3.29)

defines the global diagonal volume matrix.

3.3.3 Internally Generated Heat

Internally generated heat is calculated elementwise using the volume matrix $W$. Thus for a node $i$

$$F_{Qi} = \sum_m Q_i^m w_{ii}^m$$  \hspace{1cm} (3.30)

where $Q_i^m$ is the rate of heat generated per unit volume at node $i$ in element $m$ and $w_{ii}^m$ is the volume adjacent to node $i$ of element $m$. In TASEF-2 the rate of internally generated heat is input as a function of temperature.

3.3.4 Boundary Heat Flow

Either heat flow $F_{Qi}$ or temperature $T_i$ are prescribed for all nodes $i$. On that part $\partial V_q$ of the boundary where heat
flow is prescribed, nodal heat flow is calculated by substituting Equation (2.13) into Equation (3.21):

\[
F_{qi}^{am} = -\int N_i \hat{q}_n \, dS \quad \gamma^m \partial_q
\]

(3.31)

The shape functions \( N_i \) are linear along the boundaries. Thus for a boundary element \( \varepsilon^m \) with lengths \( s \) and thickness \( d \) as shown in Figure 3.2 the nodal heat flow to an adjacent node \( i \) is

\[
F_{qi}^{am} = -\frac{1}{6} sd(2q_{ni}^{am} + \hat{q}_{nj}^{am})
\]

(3.32)

Figure 3.2. Heat flow to a boundary element

Equations (2.14 and 2.15) are then used to yield

\[
\hat{q}_{ni}^{am} = -\left[ \epsilon_r \sigma (\bar{T}_{gi}^4 - \bar{T}_i^4) + \beta (\bar{T}_{gi} - \bar{T}_i)^\gamma \right]
\]

(3.33)

where \( \bar{T}_{gi} \) and \( \bar{T}_i \) are absolute gas and surface temperature at node \( i \), respectively; \( \epsilon_r \) is resultant emissivity, and \( \beta \) and \( \gamma \) are convection factor and power, respectively, for a boundary element \( \varepsilon^m \). Equation (3.33) is substituted into Equation (3.32) to yield

\[
F_{qi}^{am} = B_{rii}^{am} T_i^4 + B_{rij}^{am} T_j^4 + B_{ci}^{am} T_i^4 + B_{cj}^{am} T_j^4
\]

(3.34)
where \( i \) and \( j \) are nodes adjacent to a boundary element \( \delta m \) and

\[
B_{rii} = \frac{1}{3} \frac{s d e_r}{c} \quad (3.35)
\]

\[
B_{rij} = \frac{1}{2} B_{rii} \quad (3.36)
\]

\[
B_{cii} = \frac{1}{3} \frac{s d \beta}{c} \quad (3.37)
\]

\[
B_{cij} = \frac{1}{2} B_{cii} \quad (3.38)
\]

\[
T_{ri} = T_{gi}^4 - T_i^4 \quad (3.39)
\]

and

\[
T_{ci} = (T_{gi} - T_i)\gamma \quad (3.40)
\]

External heat flow to all boundary nodes is assembled in matrix form to

\[
F_q = B_r T_r + B_c T_c \quad (3.41)
\]

where \( T_r \) and \( T_c \) are vectors of modified nodal temperature as defined by Equations (3.39 and 3.40), respectively, and \( B_r \) and \( B_c \) are boundary radiation and convection matrices, respectively, where

\[
B_{rij} = \sum_{\delta m} B_{\delta m} \quad (3.42)
\]

and

\[
B_{cij} = \sum_{\delta m} B_{\delta m} \quad (3.43)
\]

Summation need be carried out only for the two boundary elements adjacent to a node \( i \) as only these contribute to the external heat flow to that node.
In TASEF-2 boundary nodes must be input sequentially around the boundary. The boundary matrices $B_r$ and $B_c$ then become tri-diagonal, i.e. only elements in the diagonal and adjacent to the diagonal have nonzero values, and since they are symmetric only two column matrices need be stored. The boundary matrices will remain constant and need be established only once when emissivity $\varepsilon_r$ and convection factor $\beta$ are assumed constant.

3.4 Time Integration

The heat flow equilibrium equation in matrix form may be solved by directly integrating the coupled differential equation step-by-step. If nodal enthalpy and external heat flow are assumed to vary linearly within each time step, Equation (3.13) can be approximated as

$$\begin{align*}
\epsilon t^+ \Delta t = & \frac{\epsilon t^+ \Delta t}{\Delta t} + (1-\theta)K^*E_t + \frac{(E_{t+\Delta t} - E_t)}{\Delta t} + \\
= & \epsilon t^+ \Delta t + (1-\theta)E_t
\end{align*}
$$

where the indices indicates time, and where $\theta$ is an arbitrary parameter in the range

$$0 \leq \theta \leq 1
$$

If different values are assigned to $\theta$ various time integration schemes are defined. Thus for $\theta = 0$, 0.5, and 1, the wellknown forward-, mid-, and backward-difference methods, respectively, are obtained. While for linear problems the latter two methods are unconditionally stable, i.e. for any time increment $\Delta t$ used solutions will not diverge, the forward-difference method will converge only if the time increment $\Delta t$ is less than a critical value $\Delta t_{CR}$. The value of this critical time increment depends on element size, material properties, and boundary conditions. If $\mathbf{C}$ is a diagonal (lumped) matrix the solution for $T_{t+\Delta t}$ is straight forward; each value can be computed directly from its precursor without the need to solve
simultaneous equations. Thus the forward-difference method is explicit while the mid- and backward-difference methods are implicit and require an equation system to be solved at each time step. Although such solutions can be very costly for nonlinear problems, implicit methods are often used because they are unconditionally stable with respect to length of time increment.

Time increments are, however, also limited by the requirement that variations in boundary conditions and material properties be adequately followed. Therefore in many problems in fire engineering, short time increments must be used even if implicit methods are employed; the magnitude of the critical time steps for the explicit Euler method is thus often the same as that required to follow changes in boundary conditions. Since during each time-step explicit methods require less computation, the forward-difference method becomes favourable. In the following section it will also be shown that the Euler method is particularly advantageous when specific heat for a material varies with temperature or when energy-consuming phase changes occur.

3.4.1 Forward Differences

For $\theta = 0$ in Equation (3.44) the explicit forward difference formula is

$$E_{t+\Delta t} = E_t + (P_t - K^* E_t) \Delta t$$  \hspace{1cm} (3.46)

or after substitution of Equation (3.11) and (3.14)

$$E_{t+\Delta t} = E_t + (F_t - K T_t) \Delta t$$  \hspace{1cm} (3.47)

Equation (3.11) is then used to obtain temperature at a node $i$:
If $C_{ii}$ varies with temperature the exact solution of Equation (3.48) is obtained by iteration. However, if all elements around a node $i$ are of the same material the specific volumetric enthalpy is calculated as

$$e_{i,t+\Delta t} = E_{i,t+\Delta t}/W_{ii}$$

and $T_{i,t+\Delta t}$ is obtained by using the temperature-specific volumetric enthalpy relation as shown in Figure 3.3. The latter method is computationally very fast and is therefore used whenever possible in TASEF-2. For nodes at interfaces between elements of different materials, the following iteration formula is used to calculate temperature.

![Figure 3.3. Translation of specific volumetric enthalpy into temperature](image)

The formula is:

$$T_{i,t+\Delta t} = C_{ii}^{-1}(T_{i,t+\Delta t})E_{i,t+\Delta t}$$

(3.48)
where \( j \) refers to iteration steps. For the first iteration step temperature from the previous time step is assumed. Iteration terminates when the difference between the nodal temperature from two successive iterations is less than a permissible value \( \delta \) expressed as
\[
\frac{T_{j+1, t+\Delta t} - T_{j, t+\Delta t}}{T_{j, t+\Delta t}} < \frac{\delta}{2}
\]

\( \delta \) is in TASEF-2 set equal 1%. Normally, convergence is achieved in a small number of iteration steps.

### 3.4.2 Critical Time Increment

To derive a simple expression by which the critical time increment \( \Delta t_{cr} \) can be estimated, the first steps in a modal solution of the heat flow equilibrium equation are shown below. If the nominal heat capacity matrix \( C \) is assumed to be time independent, Equation (3.12) is
\[
C \frac{\partial}{\partial t} T + K T = F
\]

At any time step, the righthand side of Equation (3.51) can be linearized at current temperature; thus, matrix \( K_p \) is defined by
\[
K_{ij} = \frac{dF_j}{dT_j}
\]

where \( i \) and \( j \) denote rows and columns. Thus the homogeneous part of Equation (3.52) is
\[
C \frac{\partial}{\partial t} T + \bar{K} T = 0
\]

where
\[
\bar{K} = K - K_p
\]
In case of homogeneous boundary conditions, solutions of Equation (3.54) have the form

\[ T = e^{-\lambda t} \phi \]  

(3.56)

where \( \phi \) is a vector independent of time \( t \). Multiply by the inverse of the diagonal matrix \( C \):

\[-\lambda e^{-\lambda t} \phi + C^{-1} \phi = 0\]  

(3.57)

Because \( e^{-\lambda t} \) can never be equal zero, the eigenvalue problem

\[(C^{-1}K)\phi = \lambda \phi \]  

(3.58)

arises. Equation (3.58) is an \( n \)th order equation where \( n \) is the number of temperature degrees of freedom in the system. There are \( n \) solutions of eigenvalues (thermal frequencies) \( \lambda_1, \lambda_2, ..., \lambda_n \) with corresponding eigenvectors (thermal modes) \( \phi_1, \phi_2, ..., \phi_n \).

The critical time increment for a forward difference scheme is now obtained [8]

\[ \Delta t_{cr} = \frac{2}{\lambda_{\text{max}}} \]  

(3.59)

where \( \lambda_{\text{max}} \) is the maximum eigenvalue.

Exact calculation of \( \lambda_{\text{max}} \) at every time step is very time consuming. The Gerschgorin's theorem [9], however, states that the maximum eigenvalue of a matrix with elements \( a_{ij} \) is

\[ \lambda \leq \max_{i} \left( a_{ii} + \sum_{j \neq i} |a_{ij}| \right) \]  

(3.60)

where \( i \) and \( j \) are rows and columns, respectively. The
diagonal elements of the heat conduction matrix $K$ are equal to the negative sum of the off-diagonal elements of the corresponding row, i.e.

$$K_{ii} = -\sum K_{ij}$$ (3.61)

Thus the maximum eigenvalue of Equation (3.58) is

$$\lambda_{\max} \leq \max_i \left[ C_{ii}^{-1} \left( 2K_{ii} + \sum_j K_{ij} \right) \right]$$ (3.62)

and an upper limit to the critical time increment is

$$\Delta t_{cr} = \min_i \left[ \frac{C_{ii}}{K_{ii} + \frac{1}{2} \sum_j K_{ij}} \right]$$ (3.63)

This approximation is used in TASEF-2 to update the critical time increment at each time step; time increments are thus continually adjusted to account for current conditions.

In Equation (3.63) it is implicit that for nodes for which the ratio of heat capacity to thermal conductance to adjacent nodes is small, the critical time increment will be very small. When possible without jeopardizing accuracy, thermal resistance between such nodes can then be neglected; the temperature of these nodes is set to the same value. All terms for these coupled nodes are combined. The resulting denominator in Equation (3.63) is reduced while $C_{ii}$ is increased; the resulting critical time step for this region is thus substantially increased. When calculating temperature in fire-exposed steel structures, for example, the difference in temperature between opposite sides of steel sheets will in most cases be negligible. Corresponding nodes can therefore be coupled without losing accuracy (see Example II and III in Section 5). At boundaries for which the heat transfer coefficient is high, short time increments may be avoided by prescribing surface temperature instead of heat transfer. This approximation is particularly useful when analyzing heat transfer in light insulating materials.
4. COMPUTER PROGRAM

The computer program TASEF-2 (Temperature Analysis of Structures Exposed to Fire - Two Dimensional Version) is developed for the analysis of thermal response of a variety of structures exposed to fire. It is coded on the basis of the theory presented in previous sections of this report. All subroutines are coded in Fortran V, while the main program is coded in NuAlgol in order to permit dynamic allocation of arrays. As all storage is in core, the number of nodes and elements in a structure is limited by available computer memory.

Input of geometric data to the current version of TASEF-2 has been automated. A structure is generated from a base rectangle with two sides that coincide with the x- and y-axes, and two at maximum x- and y-coordinates. A mesh is then generated by lines either at specified distances or at prescribed coordinates. Rectangular subregions either with elements of different material than that of the main region, or fictitious elements in voids or cut outs from the base rectangle, are defined in the input by their minimum and maximum x- and y-coordinates. Any structure that can be assembled of rectangular elements is therefore easily generated.

The material properties conductivity and specific volumetric enthalpy are assumed to vary piecewise linearly with temperature, and are input for each region as a number of temperature property-value pairs. As the conductivity of heated concrete in the cooling phase remains approximately as at maximum temperature, the user can specify that, for appropriate regions, conductivity in the cooling phase is to be calculated as a function of maximum instead of current temperature.
The critical time increment for nodes close to each other or separated by a material with high thermal diffusivity will be very short (see Section 3.4.2). Such nodes may be coupled to other adjacent nodes, i.e. their temperature will be prescribed to be equal. Errors thus introduced are negligible if the exact temperature at the coupled nodes differs little.

Nodes with common properties can be grouped to facilitate input and computation. Such groups may consist of nodes at boundaries with prescribed temperature or heat transfer conditions. Node groups are also used to define voids where heat transfer by convection and radiation occur. Emissivity and convection factor and power are assigned to node groups, where appropriate.

Heat exchange by convection and radiation between enclosure surfaces in structures with voids may be considered. The procedure is fully described in [10]. View factors between surfaces defined by the nodes on the enclosure surfaces are calculated automatically by the program. Convection is computed assuming that no exchange of enclosed air occurs and that heat stored in the air is negligible for the heat balance of the surrounding solid. Portions of enclosure surfaces are assigned heat transfer properties by using several node groups to define each void.

The temperature of boundary nodes or of the surrounding gas is defined as a constant ambient temperature or a time-dependent fire temperature. A fire temperature history is specified by a number of points on a time-temperature curve. Temperature between these points is obtained by linear interpolation. If the time-temperature relation specified for the ISO 834 standard fire resistance test [11] is assumed, the fire temperature $T_f$ may instead be calculated as

$$ T_f = T_0 + 345 \log_{10} (430 t + 1) \quad (4.1a) $$
for \( t \leq t_u \), when \( T_o \) is ambient temperature, and \( t \) and \( t_u \) are time and duration of heating phase, respectively, in hours. In the cooling phase the fire temperature decreases at the following rates:

\[
\begin{align*}
625 \degree C/h & \quad \text{if } t_u \leq 0.5 \text{ h} \\
250(3-t_u) \degree C/h & \quad \text{if } 0.5 \leq t_u \leq 2 \text{ h} \\
250 \degree C/h & \quad \text{if } t_u > 2 \text{ h}
\end{align*}
\]

The forward difference time integration scheme described in Section 3.4 is used. The conductivity matrix (Section 3.3.1) is symmetric and banded, and therefore only the lower half band including the diagonal of the matrix is formed. The heat capacity and volume matrices (Section 3.3.2) are diagonal (lumped) and are therefore stored as vectors. The conductivity matrix is updated either at each time step or at intervals specified in the input. Computer time is saved if updating can be avoided without sacrificing accuracy.

At each time step the length of the critical time increment is computed as described in Section 3.4.2 and a time increment is obtained by multiplying by a time increment factor specified by the user, usually in the range 0.75-1.0. Values greater than this range may cause numerical instability, while smaller values will prolong computation without necessarily increasing accuracy. If the critical time increment is very long in relation to the rate of change of boundary temperature, the user may specify an upper limit to the time increment.

Nodal temperature is printed at specified times. Maximum nodal temperatures obtained during an analysis are currently stored in a vector and are used in calculations of conductivity for certain elements as described above. When a nodal temperature begins to decrease, node number, maximum
temperature, and time of occurrence are printed. Finally, when the analysis is terminated at a time specified in the input, maximum nodal temperatures are printed.

A summary of the solution technique and detailed input instructions and a complete listing of the program are given in Appendices A and B, respectively.
5. EXAMPLES

Three examples were solved to demonstrate the use and verify the accuracy of TASEF-2. The solution of the first problem is compared to an analytical solution and results from the others to experimental data. Input cards for the examples are listed in Appendix C.

5.1 Example I - Square Plate Subjected to Heat Transfer from Surrounding Gas

A square plate with side lengths $2d$ as shown in Figure 5.1a initially at uniform temperature $T_0$ is suddenly subjected to an environment of uniform gas temperature $T_g$. Heat transfer $q$ at the boundary to the body surface is

$$q = h(T_g - T_s)$$

where $T_s$ is surface temperature and $h$ a heat transfer coefficient assumed constant. Conductivity $k$, density $\rho$, and specific heat capacity $c$ are assumed constant. The following dimensionless parameters are introduced for convenience:

$$\theta = (T - T_g)/(T_0 - T_g)$$

$$Fo = at/\ell^2$$

$$Bi = h\ell/k$$

where the thermal diffusivity $\alpha$ is

$$\alpha = k/c\rho$$

and $Fo$ and $Bi$ are the Fourier and Biot numbers, respectively. By separation of variables, analytical product
solutions as infinite sums are obtained to this problem [12]. The temperature at the center of the plate was calculated analytically and numerically by the program TASEF-2 for \( Bi = 1 \).

As the problem is symmetrical only one quadrant need be analyzed. Numerical solutions were obtained using a coarse mesh of 4 elements and a fine mesh of 16 elements as shown in Figure 5.1b. By assigning a time increment factor equal to one, time steps equal to \( 0.0667 \frac{a}{k^2} \) and \( 0.0200 \frac{a}{k^2} \) were calculated for the coarse and fine meshes, respectively.

Numerical results obtained with TASEF-2 and the exact analytical solution are given in Table 5.1. Errors in the numerical solutions are small even for the coarse mesh with only 4 elements.

5.2 Steel Beams Embedded in Concrete

5.2.1 Material Properties and Boundary Conditions

A wide-flange I-beam and a box girder of steel embedded in normal concrete, as shown in Figures 5.2 and 5.3, were exposed on one side to a model fire that approximately corresponded to the ISO 834 standard time-temperature curve in a test furnace. Steel and concrete temperatures were measured at several points over the cross section and compared to temperature predicted by the program TASEF-2.

Conductivity and specific volumetric heat of steel were assumed to vary with temperature as shown in Figures 5.4 and 5.5 [13]. Latent heat due to phase changes at temperature around 725\(^\circ\)C is considered. The thermal properties of concrete vary considerably with type of mix, moisture content, curing, age, etc. The assumed temperature-
Table 5.1. Comparison of analytically [11] and numerically calculated dimensionless temperature $\theta$ at center of square plate exposed to heat transfer from surrounding gas, $Bi=1$

<table>
<thead>
<tr>
<th>Dimensionless time $Po$</th>
<th>Exact solution</th>
<th>Coarse mesh $\Delta t=0.0667 \Delta t/l^2$</th>
<th>Fine mesh $\Delta t=0.0200 \Delta t/l^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Numerical solution</td>
<td>Error</td>
<td>Numerical solution</td>
</tr>
<tr>
<td>0.1</td>
<td>0.9864</td>
<td></td>
<td>0.993</td>
</tr>
<tr>
<td>0.2</td>
<td>0.9038</td>
<td>0.925</td>
<td>-0.021</td>
</tr>
<tr>
<td>0.4</td>
<td>0.6902</td>
<td>0.688</td>
<td>0.002</td>
</tr>
<tr>
<td>0.6</td>
<td>0.5147</td>
<td>0.505</td>
<td>0.009</td>
</tr>
<tr>
<td>0.8</td>
<td>0.3827</td>
<td>0.370</td>
<td>0.012</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2845</td>
<td>0.271</td>
<td>0.013</td>
</tr>
</tbody>
</table>

Number of time steps

15

50

Figure 5.1. Finite element model for calculating heat transfer in a square plate.
Figure 5.2. Wide-flange I-beam (HE100B) embedded in a concrete slab. The vertical sides were insulated during furnace test. Dimensions in mm.
Figure 5.3. Box girder embedded in a concrete slab. The vertical concrete sides were insulated during furnace test. Dimensions in mm.
Figure 5.4. Conductivity of steel

conductivity relation (Figure 5.6) was that measured by the Stålhane Pyk method for the type of concrete used in the test [14]. The assumed variation of specific volumetric enthalpy with temperature is based on measurements on dry concrete [15]. Enthalpy corresponding to heating and evaporation of moisture is then calculated and added. Thus the specific enthalpy $e$ for concrete with a moisture content $u$ is

$$e(T) = e^c(T) + u e^w(T)$$
Figure 5.5. Specific volumetric enthalpy of steel

where $e^C$ and $e^W$ are specific enthalpy for dry concrete and water, respectively. If the water is assumed to evaporate linearly in the temperature range of $T_1$ to $T_2$:

$$e^W(T) = c^W_T$$  \[\text{for } T < T_1\]

$$e^W(T_1) + \frac{1}{2} c^W(T_2 - T_1) + a^W$$  \[\text{for } T = T_2\]

$$e^W(T) = e^W(T_2)$$  \[\text{for } T > T_2\]
Figure 5.6. Thermal conductivity of concrete

where $c_w$ and $\Delta h_w$ are the specific heat and heat of evaporation, respectively, of water. The specific volumetric enthalpy $e_v$ is then obtained by multiplying by the density $\rho_c$ of concrete:

$$e_v(T) = \rho_c e(T)$$

As the test specimens were cured in an environment of 40% relative humidity for a month, a moisture content
of 1.5% by weight was assumed [16]. The specific volumetric enthalpy for dry and moist concrete, where moisture is assumed to evaporate in the temperature range of 100 to 115°C, is plotted versus temperature in Figure 5.7.

![Graph showing specific volumetric enthalpy of dry concrete and concrete containing 1.5% water by weight.](image)

**Figure 5.7.** Specific volumetric enthalpy of dry concrete and concrete containing 1.5% water by weight

The cool side surfaces of the specimens were small in comparison with surrounding surfaces at ambient temperature. The resultant emissivity \( \varepsilon_r \), as defined by Equation (2.15), was therefore chosen to be equal to the emissivity of appropriate material surfaces; i.e. 0.6 [17] and 0.8 [18]
for steel and concrete surfaces, respectively. On the fire-exposed side the same resultant emissivities were chosen, as radiation conditions in the furnace were to little known to justify any other values.

When assessing convection heat transfer factors ε and powers γ as defined by Equation (2.14), free convection was assumed and the formula [19]

\[ \text{Nu} = C(\text{Pr Gr})^m \]  

(a)

was employed. The Nusselt number Nu is defined by the equation

\[ \text{Nu} = \frac{q^c d}{k(T_s - T_g)} \]  

(b)

where \( q^c \) is heat transferred by convection, \( d \) characteristic length, and \( k \) gas conductivity. The Prandtl number \( \text{Pr} \) is approximately 0.7 and the Grashof number \( \text{Gr} \) is

\[ \text{Gr} = \frac{g \frac{1}{T_b} (T_s - T_g) d^3}{v^2} \]  

(c)

where \( g \) is the acceleration of gravity, \( T_b = (T_s + T_g)/2 \) the average absolute boundary layer temperature, and \( v \) kinematic viscosity. For horizontal plates the characteristic length is calculated from

\[ d = \frac{A}{P} \]

where \( A \) is the area and \( P \) is the perimeter of the surface. Conductivity and viscosity of air are functions of temperature \( T_b \) [20]:

\[ k = 13.75 \cdot 10^{-5} T_b^{0.92} \text{ [W/mK]} \]  

(d)

\[ v = 1.13 \cdot 10^{-9} T_b^{5/3} \text{ [m}^2\text{/s]} \]  

(e)
From Equations (a-e) the formula (SI-units)
\[ q^C = 13.75 \cdot 10^{-5} (5.48 \cdot 10^{18})^m C \left[ \frac{d^{3m-1}}{(\frac{13}{3} m - 0.92)} (T_s - T_g)^{m+1} \right] \]
is derived.

The characteristic length \( d \) of the beams considered is approximately 0.15 m. Substitution of Equation (e) into Equation (c) then gives \( \text{Gr} < 8 \cdot 10^6 \) for expected levels of temperature. Laminar convection is therefore expected on the cool side, and \( C \) and \( m \) are 0.54 and 0.25, respectively [21]. Equation (f) then yields
\[ q^C = 3.59 \left[ d^{0.25} T_b^{-0.16} (T_s - T_g)^{1.25} \right] \quad [\text{W/m}^2] \]
and by inserting \( d = 0.15 \) m and assuming \( T_b = 400 \) K, the convection factor \( \beta \) and power \( \gamma \) are identified as 2.2 W/m\(^2\)K\(^{1.25}\) and 1.25, respectively. At the fire exposed side the burners will cause turbulent conditions [22], and therefore \( C \) and \( m \) are 0.15 and 1/3 [21], respectively. Thus
\[ q^C = 36 \left[ T_b^{-0.52} (T_s - T_g)^{1.33} \right] \quad [\text{W/m}^2] \]
and if \( T_b \) is assumed to be 1000K, \( \beta \) and \( \gamma \) are identified as 1.0 W/m\(^2\)K\(^{1.33}\) and 1.33, respectively.

Convection heat transfer is only approximately modeled in TASEF-2. Errors in predicted temperature thus introduced are, however, negligible on the hot fire-exposed side, where radiation is dominant, while on the cool side they may be relatively great near the surface.

The beams were tested for one and a half hours; the furnace temperature approximately followed the ISO 834 standard time gas temperature curve for one hour and then it cooled off for half an hour.
Temperature in the furnace was measured with bare thermocouples. That is, however, not sufficient to determine accurately the heat transfer by radiation and convection from the furnace to the specimens [22]. The gas time-temperature curves assumed in the calculation were therefore adjusted so that calculated and measured temperature matched at the center of the fire-exposed flanges.

5.2.2 Example II - Wide-flange I-beam (Figure 5.2)

The finite element mesh shown in Figure 5.8 was employed to predict temperature in the wide-flange I-beam. Since the steel elements are small and the thermal diffusivity

\[ E_r = 0.6 \]

\[ E_r = 0.8 \]

\[ T_{amb} = 25^\circ C \quad \beta = 2.2 \text{ W/m}^2\text{K}^{1.25} \quad \gamma = 1.25 \]

\[ T = T_f \quad \beta = 1.0 \text{ W/m}^2\text{K}^{1.33} \quad \gamma = 1.33 \]

Figure 5.8. Finite element mesh of I-beam embedded in concrete
of steel is high, temperature differences between two nodes on opposite sides of a steel sheet will be negligible. Such nodes were therefore coupled as shown in Figure 5.8, and required to reach the same temperature. Critical time increments are thus substantially increased without introducing any noticeable error (see Section 3.4.2).

Assumed furnace gas temperature and measured and calculated temperature histories at the center of the top and bottom flanges, and along a vertical line 140 mm from the center-line of the steel cross section are plotted in Figures 5.9 and 5.10, respectively. Measured and calculated temperature distributions at selected times along the steel beam flanges and web are plotted in Figures 5.11 and 5.12, respectively.

While predicted steel temperature is accurate, such good agreement cannot be expected for concrete temperature since the effect of moisture migration is neglected. The characteristic plateau in the time-temperature curve at about 100°C, when water evaporates does, however, appear in the calculated temperature curve (Figure 5.10). Better agreement could be achieved if more accurate data on conductivity and specific enthalpy were available for the type of concrete tested.

5.2.3 Example III - Box girder (Figure 5.13)

The finite element mesh employed in the analysis of the box girder is shown in Figure 5.13. Only one half of the symmetrical cross section need be analyzed.

Heat transfer in the void by convection and radiation was considered as described in [10]. Convection heat transfer between the enclosure surfaces is assessed by neglecting heat capacity of enclosed air and assuming that no air
Figure 5.9. Assumed furnace gas temperature in Example II and measured and calculated temperature histories of top and bottom flanges at the centerline.
Figure 5.10. Calculated and measured concrete temperature at 140 mm from the centerline of steel beam and at distances d equal to 3.5 and 5.0 cm from fire-exposed surface and on cool surface, respectively.
Figure 5.12. Calculated and measured temperature distributions along flanges of I-beam at selected times.
Figure 5.13. Finite element mesh of box girder embedded in concrete.
flows along the beam. Convection factors \( \beta \) and powers \( \gamma \) for the enclosure surfaces are estimated by assuming free convection between two horizontal plates of uniform temperature. Thus \( \beta \) and \( \gamma \) equal to \( 1.6 \text{ W/m}^2 \text{ K}^{4/3} \) and \( 1.33 \), respectively, were obtained by assuming a temperature level of \( 500 \text{ K} \) \[19\]. This estimation of the convective heat transfer is very rough, but any error introduced will be small as radiation increasingly dominates for increasing temperature. When calculating radiation heat transfer between the enclosure surfaces, temperature between adjacent nodal points is assumed to be uniform; by considering view factors and emissivities, the net radiation to each surface is calculated and then distributed to adjacent nodes \[10\]. The emissivity of the enclosure steel surfaces was assumed to be 0.6 \[17\].

In Figure 5.14 assumed furnace gas temperature and measured and predicted temperature of the center of the upper and lower flanges of the box girder and the cool upper concrete surface at the line of symmetry are plotted versus time. Distributions of measured and predicted temperature in the flanges and webs are plotted in Figures 5.15 and 5.16, respectively, at selected times.
Figure 5.14. Assumed furnace gas temperature in Example III and measured and calculated temperature of top and bottom flanges of box girder and cool concrete surface at the centerline.
Figure 5.15. Calculated and measured temperature distributions along flanges of box girder at selected times.
Figure 5.16. Calculated and measured temperature distributions along webs of box girder at selected times.
6. SUMMARY AND CONCLUSIONS

6.1 Present Study

TASEF-2 is particularly well suited for the analysis of temperature in fire-exposed structures; the program is simple to use; rectangular finite element meshes are generated automatically with a minimum of input; nonlinearities due to the temperature dependence of material properties and boundary conditions can be considered; and heat transfer by radiation and convection in voids can be calculated using an algorithm described in [10].

The forward difference step-by-step time integration scheme used in TASEF-2 is a very efficient means of solving problems where materials having nonlinear temperature-specific heat relations - due to for instance evaporation of water - must be considered. A technique has been developed by which the critical time increment at which the applied step-by-step method will become numerically unstable can be estimated. Time increments are then calculated as a user-specified fraction of the critical time increment at each time step.

To avoid unnecessarily short time increments, and thus lengthy computations, nodes expected to attain approximately the same temperature are coupled and required to attain equal temperature. The technique has been successfully applied to composite cross sections of concrete and steel exposed to fire.

Three problems were analyzed in order to assess the accuracy and efficiency of TASEF-2. The solution of the first problem was compared to an analytical solution. The accuracy was good even for relatively coarse finite element meshes and long time increments. In the other two problems predicted temperature in composite steel and concrete beams was compared
to temperature measured during laboratory tests. One of the beams enclosed a void where heat transfer by radiation and convection was considered. The analysis proved to be accurate particularly for steel temperature. An equally good accuracy was not possible for concrete temperature because the thermal properties at elevated temperature of concrete are not as accurately known and the influence of mass transfer of water is not considered in the model. Heat of vaporization is, however, accounted for by stepping the temperature-specific enthalpy curve in the temperature range when water in the concrete evaporates (Figure 2.1). The total heat balance for a body heated to a temperature above the range at which evaporation occurs is therefore correct, and thus predicted temperature can be expected to be more accurate at high temperatures.

6.2 Future Development

In present version of TASEF-2, rectangular two-dimensional elements are available. Various one-, two-, and three-dimensional elements could be relatively easily introduced, but input would then be more complicated.

Heat transfer due to mass transfer in porous materials is not considered in TASEF-2. An extension of the model to include such phenomena would substantially complicate the analysis; in addition material data on mass diffusivity at high temperature are difficult to obtain. Results may, however, be improved with the present model if material properties determined at transient conditions accomplished by exposing specimens to boundary conditions similar to those in a fire were used in the analysis. A finite number of parameters by which the variation with temperature of one of the thermal properties - conductivity or specific enthalpy - could be described would then determine iteratively. Estimated parametric values are first used in such an analysis; calculated and measured temperature are then
compared. A new set of parameters is then chosen and new temperatures calculated. The procedure is repeated until the difference between measured and calculated temperature is minimized; computer programs are available by which the iterative search for parametric values can be accomplished. An optimal set of input data to the numerical model can thus be obtained for a given material, exposed to a similar fire; the influence of moisture migration will be indirectly considered in such an analysis.

The explicit forward difference time integration scheme used in TASEF-2 is very efficient for the nonlinear problem considered in this report. Nodes for which heat capacity is low and which are separated from adjacent nodes by little thermal resistance may, however, require that very small time increments be used if numerical stability is to be ensured. In the present version of TASEF-2, such critical nodes can be coupled and restrained to equal temperature, as illustrated in Example II and III in Section 5. If the error thus introduced is unacceptable, algorithms could be employed so that unconditionally stable implicit methods could be used for critical nodes. Such mixed implicit-explicit procedures have successfully been used in structural dynamics [23].
7. REFERENCES


[22] Paulsen, O., Om varmeovergang i Brandprøveovne (On Heat Transfer in Fire Test Furnaces), Danmarks Tekniske Højskole, 1975, (in Danish).

APPENDIX A - USER'S MANUAL

The solution technique employed in TASEF-2 is summarized in Table A:1. Variable names are as in the program and differ occasionally from that used in previous sections. Table A:2 contains all subroutines in TASEF-2 with corresponding routine references, input variables, and common blocks; a chart of all routines is shown in Figure A:1. Detailed input instructions are given in Table A:3. Except for title cards, all input is read in free field format; input fields are then separated by a comma, or one or more blanks. Sequential commas are recognized as zero input values. Input variables may be given in any consistent unit system. Default values are, however, in SI-units and the expression for the ISO 834 standard test curve assumes time in hours and temperature in Kelvin or degree Celsius.
TABLE A:1

SUMMARY OF SOLUTION METHOD
(Variable names as in TASEF-2)

INITIAL CALCULATION:

1. Input geometry and dimension system arrays.

2. Define coupled nodes.

3. Input material data.

4. Input initial and ambient temperature.

5. Form node groups and input appropriate heat transfer data.

6. Define boundary node groups for prescribed heat transfer, and form heat transfer matrices BR and BC.

7. Define boundary node groups for prescribed temperature.

8. Define void node groups and form internal heat exchange matrices E and H.

9. Input time data.

10. Form node volume vector W.

FOR EACH FIRE TO BE ANALYZED:

11. Input new fire time temperature curve, or terminate program.

12. Initialize nodal temperature T and enthalpy EN vectors. Set the time variable TIME and the time step counter KTIME equal to zero.
TABLE A:1 (Cont.)

13. Set the time increment $\text{DELTI}$ equal to zero and execute first time step for calculating first time increment only.

FOR EACH TIME STEP:

14. Increment time step: $\text{TIME}=\text{TIME}+\text{DELTI}$ and $\text{KTIME}=\text{KTIME}+1$.

15. Form new conduction matrix, and compute nodal heat flow $F$ from nodes by internal conduction.

16. Find fire temperature.

17. Compute nodal heat flow by heat transfer at boundaries and in voids and internally generated.

18. a) For nodes surrounded by elements of one material, compute new nodal specific volumetric enthalpy $\text{EN}=\text{EN}+\text{DELTI}*(\text{FLOW}-F)/W$ and obtain new temperature from the temperature specific volumetric enthalpy relation of the material.

b) For nodes at interfaces between materials, compute new nodal enthalpy $\text{EN}=\text{EN}+\text{DELTI}*(\text{FLOW}-F)$ and get new temperature by iteration $T_{i+1}=\text{EN}/P$ where the total heat capacity of a node $P$ is a function of temperature.

19. Set prescribed temperature to appropriate boundary nodes.

20. Print nodal temperature if required.

21. Test for more time steps:
   YES: Calculate new time increment $\text{DELTI}$. Go to step 14.
   NO : Print maximum temperature obtained during the process. Go to step 11.
<table>
<thead>
<tr>
<th>NR</th>
<th>ROUTINE</th>
<th>REFERENCED IN</th>
<th>REFERENCES</th>
<th>INPUT VARIABLES</th>
<th>COMMON BLOCKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ACOUPL</td>
<td>FEM2</td>
<td></td>
<td></td>
<td>UNIT</td>
</tr>
<tr>
<td>2</td>
<td>AMB</td>
<td>PROG2</td>
<td></td>
<td>TINIT,TAMB,SIGMA,TABS</td>
<td>UNIT</td>
</tr>
<tr>
<td>3</td>
<td>ASSA2</td>
<td>FEM2</td>
<td>COND2</td>
<td></td>
<td>RGEO</td>
</tr>
<tr>
<td>4</td>
<td>ASSP2</td>
<td>FEM2</td>
<td>INTF,XVERSY</td>
<td></td>
<td>RMAT</td>
</tr>
<tr>
<td>5</td>
<td>ASSW2</td>
<td>PROG2</td>
<td></td>
<td></td>
<td>RGEO</td>
</tr>
<tr>
<td>6</td>
<td>BFIRED</td>
<td>INIT</td>
<td></td>
<td>TITFIR;NFP;TIM,TB</td>
<td>UNIT,FIRE</td>
</tr>
<tr>
<td>7</td>
<td>BRBCA</td>
<td>FQBNDA</td>
<td></td>
<td></td>
<td>BND</td>
</tr>
<tr>
<td>8</td>
<td>BRBCB</td>
<td>FQBNDB</td>
<td></td>
<td></td>
<td>RMAT</td>
</tr>
<tr>
<td>9</td>
<td>COND2</td>
<td>ASSA2</td>
<td>XVERSY</td>
<td>NCPLG;NCOUPL</td>
<td>COUPLE</td>
</tr>
<tr>
<td>10</td>
<td>COUPLA</td>
<td>PROG2</td>
<td></td>
<td></td>
<td>COUPLE</td>
</tr>
<tr>
<td>11</td>
<td>COUPLB</td>
<td>FEM2</td>
<td></td>
<td></td>
<td>COUPLE</td>
</tr>
<tr>
<td>12</td>
<td>COUPLC</td>
<td>FEM2</td>
<td></td>
<td></td>
<td>COUPLE</td>
</tr>
<tr>
<td>13</td>
<td>CTEMP</td>
<td>FEM2</td>
<td></td>
<td></td>
<td>TOUT</td>
</tr>
<tr>
<td>14</td>
<td>DTIME</td>
<td>FEM2</td>
<td></td>
<td>ENRAD1,ENCON1</td>
<td>BND,ENCLOS,ENRAD, ENCON,DIM,UNIT</td>
</tr>
<tr>
<td>15</td>
<td>ENCL01</td>
<td>PROG2</td>
<td>ENRAD2,ENCON2</td>
<td>CONTRO;NENC;XSYM, YSYM,IGREN</td>
<td>ENCLOS</td>
</tr>
<tr>
<td>16</td>
<td>ENCL02</td>
<td>FEM2</td>
<td>ENRAD2,ENCON2</td>
<td>HTRANS</td>
<td>BND,ENCLOS,ENCON, DUMMY</td>
</tr>
<tr>
<td>17</td>
<td>ENCON1</td>
<td>ENCL01</td>
<td></td>
<td>VIEWFC,INVER,MULT, ETRANS</td>
<td>ENCON,ENCLOS,DUMMY</td>
</tr>
<tr>
<td>18</td>
<td>ENCON2</td>
<td>ENCL02</td>
<td></td>
<td>RADVEC</td>
<td>BNOD,ENCLOS,ENRAD, UNIT,ENCON,ENCL02,DUMMY</td>
</tr>
<tr>
<td>19</td>
<td>ENRAD1</td>
<td>ENCL01</td>
<td></td>
<td></td>
<td>ENCON,ENCLOS,DUMMY</td>
</tr>
<tr>
<td>20</td>
<td>ENRAD2</td>
<td>ENCL02</td>
<td></td>
<td></td>
<td>BNOD,ENCLOS,ENRAD, UNIT,ENCON,ENCL02,DUMMY</td>
</tr>
<tr>
<td>21</td>
<td>ETRANS</td>
<td>ENRAD1</td>
<td></td>
<td>INIT,ASSP2,XVERSY, MINTP,ASSA2,MPACKV, FQBNDB,FQGEN,ENCL02, COUPLB,ACOUPC,CTEMP, HTEMP,PTBNDB,COUPLC, OUT2,MAXCO,DTIME, OUTM2</td>
<td>RMAT,FIRE</td>
</tr>
<tr>
<td>22</td>
<td>FEM2</td>
<td>PROG2</td>
<td></td>
<td>NFQNG;FA1,ING1</td>
<td>UNIT,BNOD</td>
</tr>
<tr>
<td>23</td>
<td>FQBNDA</td>
<td>PROG2</td>
<td>BRBCA</td>
<td></td>
<td>FQB,BNOD,UNIT</td>
</tr>
<tr>
<td>24</td>
<td>FQBNDB</td>
<td>FEM2</td>
<td>BRBCB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NR</td>
<td>ROUTINE</td>
<td>REFERENCED IN</td>
<td>REFERENCES</td>
<td>INPUT VARIABLES</td>
<td>COMMON BLOCKS</td>
</tr>
<tr>
<td>----</td>
<td>---------</td>
<td>---------------</td>
<td>------------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>25</td>
<td>FQGEN</td>
<td>FEM2</td>
<td></td>
<td></td>
<td>RMAT</td>
</tr>
<tr>
<td>26</td>
<td>GEOCO2</td>
<td>PROG2</td>
<td></td>
<td></td>
<td>ELFICT</td>
</tr>
<tr>
<td>27</td>
<td>HTEMP</td>
<td>FEM2</td>
<td></td>
<td></td>
<td>RMAT</td>
</tr>
<tr>
<td>28</td>
<td>HTRANS</td>
<td>ENCON1</td>
<td></td>
<td>XVERSY</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>INIT</td>
<td>FEM2</td>
<td></td>
<td>BFIRE</td>
<td>TOUT,UNIT,FI</td>
</tr>
<tr>
<td>30</td>
<td>INTERF</td>
<td>PROG2</td>
<td></td>
<td></td>
<td>RGE0,COPPL</td>
</tr>
<tr>
<td>31</td>
<td>INTP</td>
<td>ASSP2</td>
<td></td>
<td>XVERSY</td>
<td>RMAT</td>
</tr>
<tr>
<td>32</td>
<td>INVER</td>
<td>ENRAD1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>MAIN2</td>
<td>PROG2</td>
<td></td>
<td>NET2, MESH2, PROG2</td>
<td>RMAT,RGE0</td>
</tr>
<tr>
<td>34</td>
<td>MAT</td>
<td>PROG2</td>
<td></td>
<td>MAT; CCC, NTC, NTE, NQE, ET; TC, C; TE, ENT; TQ, QE</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>MAXCO</td>
<td>FEM2</td>
<td></td>
<td></td>
<td>TOUT</td>
</tr>
<tr>
<td>36</td>
<td>MINTP</td>
<td>FEM2</td>
<td></td>
<td></td>
<td>COUPLE</td>
</tr>
<tr>
<td>37</td>
<td>MESH2</td>
<td>MAIN2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>MPACKV</td>
<td>FEM2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>MULT</td>
<td>ENRAD1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>NET2</td>
<td>MAIN2</td>
<td></td>
<td>TITLE; AXIAL, XMAX, XMAX, RBOX, YBOX, NR, NX, NY; ELFICT, SRIAC; XA, YA</td>
<td>DUMMY</td>
</tr>
<tr>
<td>41</td>
<td>NGROUP</td>
<td>PROG2</td>
<td></td>
<td>NGROUP; NCHECK, NUMB, EPSG, BETA, GAMMA; NBOUND</td>
<td>BNOD, ENRAD, ENCON</td>
</tr>
<tr>
<td>42</td>
<td>OUT2</td>
<td>FEM2</td>
<td></td>
<td></td>
<td>TOUT</td>
</tr>
<tr>
<td>43</td>
<td>OUTMA2</td>
<td>FEM2</td>
<td></td>
<td></td>
<td>FIRE</td>
</tr>
<tr>
<td>44</td>
<td>PROG2</td>
<td>MAIN2</td>
<td></td>
<td>REG2, COPLA, INTERP, MAT, GEOCO2, AMB, NGROUP, FQBNDA, PTBNDA, ENCLO1, TIME, TIME, ASSW2, COUPLB, FEM2</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>PTBNDA</td>
<td>PROG2</td>
<td></td>
<td>NPTNG; FA1, ING1</td>
<td>PTB, BNOD</td>
</tr>
<tr>
<td>46</td>
<td>PTBNDB</td>
<td>FEM2</td>
<td></td>
<td></td>
<td>BNOD, UNIT</td>
</tr>
<tr>
<td>47</td>
<td>RADVEC</td>
<td>ENRAD2</td>
<td></td>
<td></td>
<td>RGE0</td>
</tr>
<tr>
<td>48</td>
<td>REG2</td>
<td>PROG2</td>
<td></td>
<td>NT, TIMMAX, DMAX, TIMPAC, KMAX, KUPDA, TOUT</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>TIME</td>
<td>PROG2</td>
<td></td>
<td></td>
<td>BNOD, ENCLOS</td>
</tr>
<tr>
<td>50</td>
<td>VIEWFC</td>
<td>ENRAD1</td>
<td></td>
<td>ASSP2, COND2, FEM2, HTEMP, INTF</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>XVERSY</td>
<td>ASSP2, COND2, FEM2, HTEMP, INTF</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Input Instructions

Input variables are given appropriate default values if zero is input.

A. TITLE CARD, FORMAT (20A4)

**TITLE**

Input appropriate title for labeling output.

B. GEOMETRY

1. Main geometry card

A base rectangle is generated between the coordinate axes and the lines \( x = XMAX \) and \( y = YMAX \). A number of subregions are defined by their minimum \( x- \) and \( y- \) coordinates, and maximum \( x- \) and \( y- \) coordinates. Lines parallel with the axes are generated at increments \( XBOX \) and \( YBOX \) or at reduced distances if subregions or specified lines to refine the mesh are present.

**AXIAL,XMAX,YMAX,XBOX,YBOX,NR,NX,NY**

**AXIAL** .TRUE. or .FALSE. if axisymmetric or plane problem, respectively. (In present version of TASEF-2 axisymmetric problems cannot be analyzed)

**XMAX** maximum x-coordinate of base structure

**YMAX** maximum y-coordinate of base structure

**XBOX** maximum distance between two x-lines (lines parallel to the y-axis)
TABLE A:3 (Cont.)

YBOX  maximum distance between two y-lines (lines parallel to the x-axis)

NR  number of regions ($NR \geq 1$). A structure is composed of a main region and a number of subregions of differing thickness or material properties. Fictitious subregions are used to specify voids and cut outs. Subregions will be defined by the following cards. (In current version $NR \leq 10$)

NX  number of specified x-lines for refining of element mesh

NY  number of specified y-lines for refining of element mesh

2. Subregion specifications

(NR-1) cards

ELFICT,SRDIAC(4)

ELFICT  .TRUE. if the subregion is a void or cut out of the base structure. Otherwise .FALSE.

SRDIAC(4)  minimum x- and y-coordinates, and maximum x- and y-coordinates of subregion

3. Specified x-lines

If $NX=0$ omit this card

XA(NX)

XA(NX)  coordinates of specified x-lines
TABLE A:3 (Cont.)

4. Specified y-lines

If NY=0 omit this card

YA(NY)

YA(NY) coordinates of specified y-lines

As an example the structure in Figure A.2a is divided into a finite element mesh as shown in Figure A.2b by the following input cards: (The variable names are given within parentheses)

F,10.,6.,2.,1.5.,3,1,2 (AXIAL,XMAX,YMAX,XBOX,
YBOX,NR,NX,NY)

F,3.,2.,6.,3.5 (ELFICT,SRDIAC(4))

T,6.,4.5,10.,6. (ELFICT,SRDIAC(4))

4.5 (XA(1))

1.0,5.25 (YA(1),YA(2))

C. COUPLED NODES

1. Number of groups of coupled nodes

NCPLG

2. Each group

NCOUPL(8)

NCOUPL(8) coupled nodes. Each card must have 8 numbers. If fewer nodes in a group, fill with commas
(a) Structure to be analyzed

(b) Generated mesh with node numbers

Figure A.2. Example of mesh generation
D. MATERIAL DATA

For each nonfictitious region the following cards are read starting with the main region and followed by the subregion in the order as defined at B.2. Conductivity and enthalpy are input as sequential groups where each individual property is described as a piecewise linearized function of temperature.

1. Each material

   a. Identification card

      MAT

      MAT arbitrary test to be written on output list

   b. Material description

      CCC,NTC,NTE,NQE,ET

      CCC .TRUE. if conductivity is constant in cooling phase

      NTC number of points associated with temperature conductivity function (NTC≤20)

      NTE number of points associated with temperature specific volumetric enthalpy function (NTC≤20)

      NQE number of points associated with temperature rate of heat generated per unit volume function (NQE≤20)

      ET thickness of elements; default 1
TABLE A:3 (Cont.)

c. Data card for temperature conductivity function

$TC, C, TC, C, TC, C, ...$ (NTC pairs)

The input is given in ordered pairs describing each point (temperature, function value)

d. Data card for temperature specific volumetric enthalpy function

$TE, ENT, TE, ENT, TE, ENT, ...$ (NTE pairs)

(same as c above)

e. Data card for time rate of heat generated per unit volume function

(If NQE=0 omit this card)

$TQ, QE, TQ, QE, TQ, QE, ...$ (NQE pairs)

E. INITIAL AND AMBIENT TEMPERATURE, AND UNIT SYSTEM DEPENDENT CONSTANTS

$TINIT, TAMB, SIGMA, TABS$

$TINIT$ initial uniform temperature of structure

$TAMB$ ambient temperature

$SIGMA$ Stefan-Boltzmann constant; default $SIGMA=5.67 \cdot 10^{-8}$

$TABS$ shift for absolute temperature; default 273
TABLE A:3 (Cont.)

F. NODE GROUPS

Groups of nodes with common conditions are defined and, if appropriate, heat transfer properties at boundaries are specified.

1. Number of node groups

\[ \text{NGROUP} \]

\((\text{NGROUP} \leq 10)\)

If NGROUP equal zero omit next card

2. Each node group

a. Properties

\[ \text{NCHECK,NUMB,EP5G,BETA,GAMMA} \]

\text{NCHECK} \quad \text{node group number in sequential order starting from 1}

\text{NUMB} \quad \text{number of nodes of a group (NUMB} \leq 30)\)

\text{EPSG} \quad \text{emissivity}

\text{BETA} \quad \text{convection factor}

\text{GAMMA} \quad \text{convection power}

b. Node numbers

\[ \text{NBOUND(NUMB)} \]
G. PRESCRIBED HEAT FLUX BOUNDARY

Boundary conditions are defined by node groups and their heat transfer properties. Surrounding gas is either at fire or ambient temperature.

1. Number of boundary node groups with prescribed heat flux

\[ NFQNG \]

If \( NFQNG \) equals zero omit next card

2. Each boundary node group

\[ FA1,ING1 \]

\[ FA1 \] .TRUE. if specified boundary temperature varies with time, e.g. fire temperature history

.FALSE. if specified boundary temperature is the constant ambient temperature TAMB

\[ ING1 \] node group number

H. PRESCRIBED TEMPERATURE BOUNDARY

Node groups with prescribed temperature are input.

1. Number of boundary node groups with prescribed temperature

\[ NPTNG \]

If \( NPTNG \) equals zero omit next card
TABLE A:3 (Cont.)

2. Each boundary node group

FA1,ING1

Same as G.2

I. VOIDS

One or two voids with heat exchange between the enclosure surfaces may be defined by surrounding node groups. Heat exchange between enclosure surfaces is described by properties assigned to the node groups. Halves or quarters of voids may be analyzed if they are symmetrical around one or both of the coordinate axes.

1. Control card

CONTRO

If no voids exist insert arbitrary card, and omit the following cards. Otherwise input VOID.

2. Number of voids

NENC

(NENC≤2)

3. For each void

a. XSYN,YSYM,IGREN(4)

XSYM .TRUE. if void is symmetric around the x-axis
TABLE A:3 (Cont.)

YSYM .TRUE. if void is symmetric around the y-axis

IGREN(4) node groups surrounding a void. If less than 4, fill with commas

J. TIME

1. Time control card

NT,TIMMAX,DTMAX,TIMFAC,KTMAX,KUPDA

NT number of specified times for printing out of temperature

TIMMAX maximum time of analysis

DTMAX maximum time increment; default TIMMAX

TIMFAC time increment factor; default 0.8

KTMAX maximum number of time steps; default 1000

KUPDA number of time steps between updating of heat conduction matrix; default 1

2. Specified times for temperature output

TOUT(NT)

K. FIRE TEMPERATURE HISTORY

1. Control card

TITFIR
TITFIR is printed for identification of assumed fire. If TITFIR = 'ISO 834' the time temperature relation according to ISO 834 fire resistance standard test is assumed, and the next two cards are omitted. Terminate analysis by a blank card.

2. Fire temperature

A fire temperature history is input by a number of points on the time temperature curve. Temperature between these points are obtained by linear interpolation.

a. Number of points

NFP

(NFP\leq50)

b. Data card

TIM,TB,TIM,TB,TIM,TB,... (NFP pairs)

3. Go back to K1 and begin analysis with new fire, or terminate program by inserting blank card.
APPENDIX B - Listing of TASEF-2

The subroutines are listed in alphabetic order. The main program MAIN2 is coded in ALGOL and all other routines in FORTRAN V. Definitions of major variables are given in subroutine PROG2. Although the program has been tested, no warranty is made regarding its accuracy or reliability, and no responsibility is assumed in this respect.

```fortran
SUBROUTINE ACOUPL(A, DT, NN, MAXJ)
  C-----ADD SLAVE NODE QUANTITIES TO MASTER NODE QUANTITIES
  C-----FOR EACH COUPLING NODE GROUP
  DIMENSION AMX(MAXJ, DT, NN)
  PARAMETER NCP=60
  COMMON /COUPLE/ NCPL(NCP,R), NCPL
  C-----
  IF (NCPLG.EQ.0) RETURN
  DO 50 M=1,NN
    AMX=M, M=1,NN
  DO 40 J=1,MAXJ
    IF (NODJ(J, M).EQ.0) GO TO 50
    IF (NODF(J, M).EQ.0) GO TO 40
    NODF(J, M)=NODF(J, M)+1
  DO 50 M=1,NN
    IF (NNUM.GT.1) AMX=AMX+AMX
  RETURN
END

SUBROUTINE AIR
  C-----READ INITIAL AND AMBIENT TEMPERATURE,
  C-----END UNIT DEPENDENT CONSTANTS
  COMMON/UNIT/SIGMA, TAB, T, A
  DATA SIGMA, TAB, T, A = (5.777 + 273.15)
  PRINT 200
  RETURN
END

SUBROUTINE ASSA2(M, N, T, xi, AL, M, AX, A)
  C-----THIS SUBROUTINE COMPUTES THE GLOBAL HEAT CONDUCTION MATRIX A
  DIMENSION NNUM, N, NODES, MAX, NOD, NNOD, MAXJ, AMX
  PARAMETER N=20, MAX=100, NOD=200, NNOD=200
  COMMON /ROPO/ ET, X, Y, Z, T, TMAX, TMAX, A
  PRINT 200
  RETURN
END
```
DIMENSION X(NN), Y(NN), T(NN), TT(NN), Tmax(NN), Ktop4, NE),
1 A(K1,MAX), ELA(4, NE)
LOGICAL TRAX, ELFICT, AXIAL
DO 5 J=1, NN
5 A(J,J)=C
DO 10 K=1, NE
10 A(K,J)=C
CONTINUE
RETURN
END

SUBROUTINE ASSW1(NN, N, Y, T, TT, MAX, EV4, NODEL, NODEL, PI, W)
CREATE GLOBAL VECTOR
DIMENSION X(NN), Y(NN), Ktop4, NE, EV4(NN), NODEL(NN), W(NN), N(NE)
PARAMETER '1001
COMMON/RGEO/ELFICT(MNR)
LOGICAL ELFICT
1=1, N
DO 20 I=1, N
20 IF (MODE1(I).LT.0) GOTO 20
IF (MODE1(I).GT.0) GOTO 20
CONTINUE
RETURN
END

SUBROUTINE ASSW2(NN, N, KTOP, X, Y, EV, AXIAL, W)
CREATES GLOBAL VECTOR
DIMENSION X(NN), Y(NN), KTOP(NE), NE, EV4(NN), NODEL(NN), W(NN)
PARAMETER '1001
COMMON/RGEO/ELFICT(MNR)
LOGICAL AXIAL, ELFICT
1=1, N
DO 30 J=1, NN
30 W(J)=C
SUBROUTINE SRCAL(BR,BC,EPSIG,SET,PAR,NUM1,ING1)

C-----FORM BOUNDARY RADIATION AND CONVECTION MATRICES

DIMENSION BR(NUM1,2),BC(NUM1,2),AR(NUM1)

BR(1,1)=.5555555*AR(ING1,1)
NUM1=NUM1-1
IF(NUM1.EQ.1) GOTO 2C
DO 10 I=2,NUM1
    BR(I,1)=.16666667*AR(I-1,1)
    BR(I,2)=.33333333*(AR(I-1,1)+AR(I-1,2))
    CONTINUE
2C CONTINUE

BR(NUM1,1)=.16666667*AR(NUM1,NUM1)
BR(NUM1,2)=.33333333*AR(NUM1,NUM1)

DO 16 I=1,NUM1
16 BC(I,1)=SET*BR(I,1)
17 BC(I,2)=EPSIG*BR(I,2)
DO 18 J=1,2
18 CONTINUE
RETURN
END

SUBROUTINE SRCAL2(BR,BC,EPSIG,SET,PAR,NUM1,ING1)

C-----FORM BOUNDARY RADIATION AND CONVECTION MATRICES

DIMENSION BR(NUM1,2),BC(NUM1,2),AR(NUM1)

BR(1,1)=.5555555*AR(ING1,1)
NUM1=NUM1-1
IF(NUM1.EQ.1) GOTO 2C
DO 10 I=2,NUM1
    BR(I,1)=.16666667*AR(I-1,1)
    BR(I,2)=.33333333*(AR(I-1,1)+AR(I-1,2))
    CONTINUE
2C CONTINUE

BR(NUM1,1)=.16666667*AR(NUM1,NUM1)
BR(NUM1,2)=.33333333*AR(NUM1,NUM1)

DO 16 I=1,NUM1
16 BC(I,1)=SET*BR(I,1)
17 BC(I,2)=EPSIG*BR(I,2)
DO 18 J=1,2
18 CONTINUE
RETURN
END
C----INTERMEDIATE NODES

C

NUM1=NUM1+1
IF(NUM1 EQ.1) GOTO 22
DO 10 T=2,NUM1
CORDER=ORDER(2*V,1)
T1=T*2
T2=T*2
T3=T*2
T4=T*2
C1=TC2
C2=TC2
C3=TC2
C4=TC2
TC1=TC2
TC2=TC2
TC3=TC2
TC4=TC2
DA=DA+CR2*TRD2+BC2*TCD2
DA=DA+CR1*TRD1+BC1*TCD1

DTA(NODE)=DTA(NODE)+.5*DA
FLOW(NODE)=FLOW(NODE)+FLW
CONTINUE

C----LAST NODE

C

NODE=ORDER(4*V,1)
T5=T*2
T6=T*2
T7=T*2
T8=T*2
TC5=TC2
TC6=TC2
TC7=TC2
TC8=TC2
DA=DA+BR2*TRD2+BC2*TCD2

DA=DA+BR1*TRD1+BC1*TCD1

DO 20 I=1,2
C

RETURN
END

SUBROUTINE COND(T1,T2,T3,T4,TT1,TT2,TT3,TT4,TMAX1,TMAX2,TMAX3,
TMAX,MT,AXIAL,CE)

1
C-----GET ELEMENT CONDUCTIVITY

LOGICAL TK1,TK2,TK3,TK4,AXIAL
PARAMETER MAX=10,MR=10
COMMON/MAXCC(MAXR,MR-10),CC(MAXR,MR),CR(MAXR,MR),
1 TMV(MAXR,MR),TR(MAXR,MR),TR(MAXR,MR),GE(MNV,MR),LO(MR)

LOGICAL CC,LC
IF(CCC(N1)) TM=(TTT+TTT+TTT+TTT)/4.
IF(.NOT..CCC(N1)) TM=(TTT+TTT+TTT+TTT)/4.
CALL .NAMES)(C,MAXV,MAXR,MAXR)
RETURN
END

SUBROUTINE COUPLA(NODCPL,NN,NODINT)

C-----READ COUPLED NODES AND FORM CONTROL VECTOR VCOUPL
PARAMETER NCP=5
COMMON/COUPL/NCP,NCP,L
DIMENSION NODCPL(NCP,NCP),NCP,L

C-----NODCPL = -1 NODE UNCOUPLED
C-----NODCPL = 0 SLAVE NODE
C-----NODCPL = 1 MASTER NODE

DO 5 I=1,NN
5 NODCPL(I)=1
PRINT 200
C---- READ 100,NCP,L
C---- IF(NCP.LT.5) GOTO 30
PRINT 203
DO 40 I=1,NCP,L
40 CONTINUE
C---- READ 120,NODCPL(J,J)=NCP
C---- COUPLED NODES ARE ALWAYS INTERFACE NODES
NODCPL(J,J)=1
DO 30 J=1,NCP,L
30 CONTINUE
C---- IF(NODCPL(J,J).GT.0) GOTO 10
NODCPL(J,J)=NCP
C---- IF(NODCPL(J,J).LT.1) GOTO 20
C---- CONTINUE
20 CONTINUE
C---- RETURN
RETURN
END

C-----ADD SLAVE NODE QUANTITIES TO MASTER NODE QUANTITIES
C-----FOR EACH COUPLING NODE GROUP
DIMENSION V(1)
PARAMETER NCP=5
COMMON /COUPLE/ VCOUPL(NCP,NCP,L)

C---- IF(NCP.LT.5) RETURN
SUBROUTINE COUPLE(I)
C-----UPDATE SLAVE NODE TEMPERATURE
DIMENSION TC(I)
PARAMETER NCP=5
COMMON/Couple/NCoupl(NCP,8),NCPL6
IF(NCPL6.EQ.0)RETURN
DO 26 I=1,NCPL6
  NOD=NCoupl(I,1)
  T(NOD)=TC(NOD)
DO 26 J=2,8
  NOO=NCoupl(I,J)
  IF(NO0.EQ.0)GOTO 26
  T(NOD)=T(NOD)+T(NO0)
CONTINUE
RETURN
END

SUBROUTINE TEMP(NODE,T,P,Eh,Flow,F,DELT1,NODEL,NOOE,N4,NCPL)
C-----CALCULATE TEMPERATURE OF INTERFACE NODES
PARAMETER NCP=50
COMMON/Couple/NCoupl(NP,50),NCPL
DIMENSION PI(NODE),EN(NODE),FLOW(NODE),F(NODE),NODEL(NODEL),NOOE(NODEL),N4(NODEL),NCPL
DATA EPS/.0001
EN(NODE)=EN(NODE)+(FLOW(NODE)-F(NODE))*DELT1
PI=P(NODE)
DO 48 J=1,5
  T=EN(J)/RI
  P1=C.
  CALL INTP(MODEL(NODEL),NODEL(NODEL),N4,EN(T),P1)
  IF(NOC.N.G)GOTO 48
  DO 26 I=2,8
    NOD=NCoupl(NODEL,I)
    IF(ND(NOD).G)GOTO 26
    CALL INTP(MODEL(NODEL),NODEL(NODEL),N4,EN(T),P1)
    ERR=(P1-P1)/PI
    IF(ABS(ERR).LT.EPS)GOTO 50
    PI=(PI*PI)/2.
CONTINUE
PIINT 2DC,NDDE,T,ERR
2DC FORMAT(U' CONVERGENCE NOT ACHIEVED FOR NODEN',15,' TEMPERATURE)',6F9.3,
1 ERR=4.9.2)
RETURN
END

SUBROUTINE DTME(NP,D,TAM,MAX,NODENT,NCPL,TIME,DELT1,NODE)
C-----THIS ROUTINE CALCULATES TIME INCREMENT
SUROIINE ENCl01(X,Y)

C---- THIS IS THE FIRST OF A SET OF ROUTINES FOR CALCULATION OF
C---- THE RATE OF CONVECTION AND RADIATION HEAT EXCHANGE IN Voids
C---- INBEDDED IN SOLIDS. THE SAME SURFACE NODES AS FOR THE SOLID STATE
C---- FINITE ELEMENT ANALYSIS ARE EMPLOYED.
C----
C---- PROGRAMMED BY
C---- ULF WICKSTROM
C---- JUNE 1977
C---- REVISED FEB 1979
C----
C---- MAJOR VARIABLES,
C----
C---- NMBR - NUMBER OF NODES IN THE NODE GROUPS
C---- NOWN - NODE NUMBERS IN THE NODE GROUPS
C---- AREA - AREA BETWEEN NODES. THIRD DIMENSION ASSUMED UNITY
C---- NENE - NUMBER OF NODES
C---- NEVNC - NUMBER OF NODE GROUPS SURROUNDING EACH VOID
C---- IGREN - NODE GROUP NUMBERS SURROUNDING EACH VOID
C---- NNODEN - NUMBER OF NODES SURROUNDING EACH VOID
C---- NNODEN - ALL NODE NUMBERS IN ALL Voids
C---- EPSG - EPSISSIVITY OF NODE GROUP ZONES
C---- H - CONVECTION VECTORS
C---- NMBF - CONVECTION FACTORS OF THE NODE GROUP ZONES
C---- TAIR - VOID AIR TEMPERATURE
C----
C---- PARAMETER NS=TC,NMBR=2,NBE=4,NNB
C---- COMMON/NOMD/NUMBD,NOWN(NB,NNB),AREA(NB,NNB),
C---- EPSG(NB),ETAM(NB),EPSG(NB),FA(NB)
C---- COMMON/ENCLERS/KENT,NEVNC(2),IGREN(2,4),NNODEN(2),NNODEN(100),
C---- AREA(NB),EPSG(2),EPSG(2)
C---- COMMON/ENRDC(E(1000)
C---- COMMON/ENCND/KS77,TAIR(2)
C---- COMMON/UNIT/SIGMA,TAMB,TAMX,TAMB
C---- DATA MAXC,MAXND(4,25),
C---- LOGICAL LEN
C---- LOGICAL XSYM,YSYM,SYM,LOUM
C---- INTEGER EN
C---- PRINT TPC
C----
C---- READ CONTROL CARD
C---- IF NO Voids IN STRUCTURE RETURN
C---- READ SC,CONTRO
C---- IF(CONTRO.EQ.4,4,HVOID) GOTO 1C
C---- PRINT TPC
C---- RETURN
C---- CONTINUE
C---- TAIR=INIT
C---- LENS=.TRUE.
CROUTINE ENCLOSURE

C-----READ AND ESTABLISH VOID GROUP DATA
C-----READ NUMBER OF VOIDS
READ IEC,NEC
PRINT 200,NEC

C-----READ EACH VOID
DO 45 EC=1,NEC

C-----READ SYMMETRY PROPERTIES AND NODE GROUPS THAT DEFINES THE VOID
C-----READ SYMMETRY PROPERTIES AND NODE GROUPS THAT DEFINES THE VOID
READ 120,XSYM(EC),YSYM(EC),(IGRE(Y,EN,J),J=1,MAXG)
SYM=XSYM(EC).OR.YSYM(EC)
IND=IND+1
INDEX(IND)=GROUP(Y,EN,J)
NODE=NODE(IND)
END

C-----READ EACH NODE GROUP
DO 20 IG=1,MAXG
IND=IND+1
CONTINUE
PRINT 210,EN,INDEX(IG),INDEX(IG),IND,IG
IF(ISYM(EC)) PRINT 220,EN
NENCNC(EC)=I
IF(ISYM(EC)) PRINT 230,EN
NENCNC(EC)=I
END=IND-1
CONTINUE
NODE(EN)=IND
CONTINUE
CALL ENRAD1(X,Y)
CALL ENC3111
RETURN
FORATT4)
FORMAT(/"VOIDS'/"*"*")
FORMAT(/"THIS STRUCTURE HAS NO VOIDS'/")
220 FORMAT(/"NUMBER OF VOIDS'=",I2/
230 FORMAT(/"VOID NUMBER',I2,' IS SURROUNDED BY THE FOLLOWING /
1 'NODE GROUPS'=",I2(3)
240 FORMAT(/"VOID NUMBER',I2,' IS SYMMETRICAL AROUND THE X-AXIS /
1 ')'250 FORMAT(/"VOID NUMBER',I2,' IS SYMMETRICAL AROUND THE Y-AXIS /
1 ')'260 FORMAT(/"SURROUNDING NODEGROUPS NOT COMPATIBLE'/" ENT=',I3,
1 'IND=',I3(3)
270 FORMAT(/"FIRST AND LAST NODE ARE NOT IDENTICAL FOR /
1 'VOID NUMBER',I2//' FIRST NODE=',I4//' SECOND NODE=',I4)
END

SURROUTINE ENCLOSURE(T,FLOW)
C-----THIS ROUTINE IS CALLED FROM THE BASIC FINITE ELEMENT PROGRAM
C-----TO CALCULATE THE RATE OF HEAT EXCHANGE BETWEEN ENCLOSED SURFACES
SUBROUTINE ENRAD2(T,FLOW)

---CALCULATE RATE OF RADIATION HEAT EXCHANGE
ENRAD2(T,FLOW)
RETURN
END

SUBROUTINE ENCON2(T,FLOW)

---CALCULATE RATE OF CONVECTION HEAT EXCHANGE
ENCON2(T,FLOW)
RETURN
END

SUBROUTINE ENCON1

---THIS ROUTINE FORMS CONVECTION ARRAY H
SUBROUTINE ENCON1

---FORM ONE CONVECTION ARRAY
DO 150 EN=NENC
SY=MVISY(EN).OR.VVISY(EN)
VVIS=VISY(EN)
END
---FORM NODE CONVECTION ARRAY
CALL HTRANS(H1,H(IND),IN,SYM)
---RETURN
END

---FORM NODE CONVECTION ARRAY
CALL HTRANS(H1,H(IND),IN,SYM)
---RETURN
END

---RETURN
END

---RETURN
END

---RETURN
END

---RETURN
END

---RETURN
END

---RETURN
C-----STORE APPROPRIATE NODAL TEMPERATURES IN DUMMY VECTOR TEN
DO 11 NODE=INDN(INDT+1)
10 TEN(I)=T(NODE)
C-----CALCULATE THE AIR TEMPERATURE TA BY ITERATION
C-----USE STARTING VALUE FROM FORMER TIME STEP
TA=TANW(EN)
DO 50 ITPR=1,INDT
50 SHBAR=SHBT
51 SWAY
C-----FOR EACH NODE
52 DO 53 I=1,INDT
53 TDIF=ABS(T(I)-TA)
54 IF(TDIF.LT.CE) HBAR(I)=1.0
55 IF(TDIF.LT.CE) GO TO 2C
56 HBAR(HBAR(I))=TDIF**1.5
57 SHBAR=SHBAR+HBAR(I)
58 SHHT=SHBAR+HBAR(I)*T(I)
59 CONTINUE
60 IF(SHHT.GT.25,58)
61 TANEW=SHHT/SHBAR
C-----CONVERGENCE CHECK OF AIR TEMPERATURE
62 IF(ABS(TANEW-TA)/(TANEW+TA).LT.PE) GO TO 6:
63 IF(ITP.8.1) GO TO 3C
64 TA=TA
65 TY=TA
66 TAE=TANEW
67 GOTO 5C
68 DO 59 I=1,INDT
69 D=X-TA
70 DYT=TA
71 D=Y-DX
72 IF(D.GT.70,60)
73 DN=X+D*TY-TA
74 TA=TA
75 TY=TANEW
76 C-----USE LINEAR INTERPOLATION TO SPEED UP CONVERGENCE
77 TA=DY/D
78 CONTINUE
79 PRINT *,'TA,CTOT
80 STOP
81 CONTINUE
82 DO 53 I=1,INDT
83 DX=T-X
84 DY=TA
85 IF(DY.GT.70,60)
86 D=X+D*TA
87 C-----USE LINEAR INTERPOLATION TO IMPROVE THE CALCULATED TEMPERATURE
88 TA=(X+D*TA)/2.0
89 TC=T(AE)+TA
90 C-----CALCULATE CONVECTION HEAT FLOW AND ADD TO THE GLOBAL HEAT FLOW
91 GQTOT=CT
92 DO 95 I=1,INDT
93 Q=HBAR(I)*TA
94 GQTOT=GQTOT+Q
95 CONTINUE
96 Continue
97 C-----VECTOR FLOW
98 IF(INDT.GT.1) PRINT 230,TA,GQTOT
99 IND=IND+1
100 CONTINUE
101 RETURN
102 FORMAT('** CONVERGENCE NOT ACHIEVED FOR THE AIR TEMPERATURE',*,13)
103 1 ' IN ENCLOSURE NUMBER',13)
SUBROUTINE EURAD(X,Y)

C-----FOR EACH VOID AND STORE THEM IN
C-----THE VECTOR E
C-----CALCULATE VIEW=FACTOR MATRIX VIEW AND ZONE AREA VECTOR D
DIMENSION X(19),Y(19),AC(25,25),H(25,25)
PARAMETER N=17, NHB=30, NHB2=2*NHB
COMMON/HECDUM/NUMC(25),NDUMC(25),HB(25,25),A(25,25)
COMMON/EPSG(NHB),EPSB(NHB),TA(25)
COMMON/UNITSIGMA,TARS
COMMON/UNITSIGMA,TARS
COMMON/DUMMY/AC(25),NDUMC(25)
DIMENSION VIEW(25,25)
EQUIVALENCE (AC,J)
DATA N02,2E7,12
LOGICAL LEN
LOGICAL SYM,SYM
INTEGER EN

C-----FOR EACH VOID
DO 152 EN=1,NVC
CALL VIEWFAC(X,Y,EN,VIEW,MAXNOD)
C-----FORM THE MATRICES A AND B
NENG=NENCNE(EN)
C-----FOR EACH NODE GROUP
DO 122 IG=1,NENG
IF(IGRE(EN,IG).EQ.1)
NUMI=NUMI+1
C-----FOR EACH ZONE
DO 126 I=2,NUMI
IN=I-1
JN=I
DO 126 I=1,IGRE(I)
J=IGRE(I)
NUMJ=NUMJ+1
EPSJ=SIGMA(J)
DO 126 J=2,NUMJ
IF(JNE(JN)) GOTO 126
IN=IN+1
IN=I-1
JN=I
DO 126 J=2,NUMJ
W(IN,JN)=W(IN,JN)*SIGMA(J)
A(IN,JN)=VIEW(IN,JN)*EPSJ/EPSP(JN)
IF(IN.EQ.JN) GO TO 126
B(IN,JN)=SIGMA(IN,JN)
A(IN,JN)=1.E0/EPSP(JN)*EPSB(JN)
120 CONTINUE
152 CONTINUE
C-----INVERT A AND STORE RESULT IN A
CALL INVER(A,N,MAXNOD)
C-----MULTIPLY A AND B AND STORE RESULT IN A
CALL MULT(A,B,A,N,MAXNOD)
SYM=.FALSE.
IF(SYM(EN).AND.SYM(EN)) SYM=.TRUE.
154 CONTINUE
C-----TRANSFORM THE LOCAL RADIATION MATRICES A AND STORE THE RESULT IN
C-----VECTOR E
C-----A IS EMPLOYED AS A DUMMY MATRIX
CALL ETRAN(E,B,LL(EN),N2,SYM,MAXNOD)
156 CONTINUE
END
SUBROUTINE ENCSPACE(1, FLOW)
C------This routine calculates the radiation heat flow to each node of a
C------enclosure surface and adds the result to the global heat flow
C------VECTOR FLOW
    DIMENSION T(1), FLOW(1)
    COMMON ENCSPACE(NENC, NENC, NENC(2), NENC(4), NNODE(2), NNODE(100),
    1 RATE(2), RATE(2))
    COMMON UNIT(3), SIGMA, TABS
    COMMON DUMMYETA(25, 25)
    LOGICAL LYN
    INTEGER EN

    IF = 1
    CONTINUE
    RETURN
END

SUBROUTINE ENCSPACE(1, FLOW)
C------Each void
    DD IS: EN+1, NENC
    NENC(N)
    C------Calculate absolute temperatures to the fourth power for each
    C------node of the enclosure surface
    DO 1 T = 1, N
    NODE = INODE(IND + 1)
    DUM = T(NODE) * TABS
    DUM = DUM + DUM
    ETA(1) = DUM
    CALL DUM(1, ETA, 1, 1)
    ETA(1) = ETA(1) ** 4

    DO 1, T = 1, N
    NODE = INODE(IND + I)
    ETA(1) = ETA(1) ** 4

    CONTINUE
    RETURN
END

SUBROUTINE ETRANS(A, B, E, N, NZ, SYM, MAX)
C------This routine transforms the zone radiation matrix A to a
C------radiation matrix and store the result in E
C------If symmetry is present expand radiation matrix
    DIMENSION E(N, N), A(MAX, MAX), B(MAX, MAX)
    LOGICAL SYM
    E = S1 + A
    DO 10 I = 2, N
    DO 10 J = 1, I
        B(I, J) = A(I - 1, J) + A(I, J)
    CONTINUE

    IF (SYM) GO TO 10
    DO 50 I = 1, NZ
    DO 50 J = 1, I
        E(I, J) = A(I, J)
        50 CONTINUE

    E = E ** 4
    DO 55 I = 1, N
    DO 55 J = 1, NZ
        E(I, J) = 0.25 * (A(I, J) + A(I, J - 1))
    CONTINUE
    E = E ** 4
    RETURN
END

SUBROUTINE ETRANS(A, B, E, N, NZ, SYM, MAX)
C------This routine calculates the zone radiation heat flow to each node of a
C------enclosure surface and adds the result to the global heat flow
C------VECTOR FLOW
    DIMENSION T(1), FLOW(1)
    COMMON ENCSPACE(NENC, NENC, NENC(2), NENC(4), NNODE(2), NNODE(100),
    1 RATE(2), RATE(2))
    COMMON UNIT(3), SIGMA, TABS
    COMMON DUMMYETA(25, 25)
    LOGICAL LYN
    INTEGER EN

    IF = 1
    CONTINUE
    RETURN
END

SUBROUTINE ENCSPACE(1, FLOW)
C------Each void
    DD IS: EN+1, NENC
    NENC(N)
    C------Calculate absolute temperatures to the fourth power for each
    C------node of the enclosure surface
    DO 1 T = 1, N
    NODE = INODE(IND + 1)
    DUM = T(NODE) * TABS
    DUM = DUM + DUM
    ETA(1) = DUM
    CALL DUM(1, ETA, 1, 1)
    ETA(1) = ETA(1) ** 4

    DO 1, T = 1, N
    NODE = INODE(IND + I)
    ETA(1) = ETA(1) ** 4

    CONTINUE
    RETURN
END

SUBROUTINE ETRANS(A, B, E, N, NZ, SYM, MAX)
C------This routine transforms the zone radiation matrix A to a
C------radiation matrix and store the result in E
C------If symmetry is present expand radiation matrix
    DIMENSION E(N, N), A(MAX, MAX), B(MAX, MAX)
    LOGICAL SYM
    E = S1 + A
    DO 10 I = 2, N
    DO 10 J = 1, I
        B(I, J) = A(I - 1, J) + A(I, J)
    CONTINUE

    IF (SYM) GO TO 10
    DO 50 I = 1, Nz
    DO 50 J = 1, I
        E(I, J) = A(I, J)
        50 CONTINUE

    E = E ** 4
    DO 55 I = 1, Nz
    DO 55 J = 1, Nz
        E(I, J) = 0.25 * (A(I, J) + A(I, J - 1))
    CONTINUE
    E = E ** 4
    RETURN
END
SUBROUTINE FEX(X,Y,N,H,H,Y,TOP,Y,T,TMAX,ELA,EV4,FMAX,FLO)
E(I,J)=0.25*(E(I,J)+E(J,I))
I=I+1
RETURN
END

C----- THIS ROUTINE INITIALIZES SYSTEM ARRAYS AND
C----- CONTROLS TIME INTEGRATION
DIMENSION N(I,J),KTOP(N,J),X,NN,N,NMAX,NMAX
I=I+1
IF(NODE(I,NN),NODINT(NN),NODEL(NN),DATA(NN))
PARAMETER NVD=20,MAX=10
COMMON/FIRE/TIMES(N,TC),TSTATUS,
COMMON/FIRE(TMU,MMR),C(NN,NN),TE(NN,NN),ENT(NN,NN),
COMMON/TOUT/I,TTOUT,DTOUT,DTMAX,DTMAX,DTMAX,KUPDA
LOGICAL TIME,AXIAL
LOGICAL FIN,CON,YODI,UPDA
DATA FIN,FALSE,
C----- INITIALIZE NODAL TEMPERATURES
CALL INIT(NN,T,TT,TMAX,NODINT)
C----- INPUT FIRE BOUNDARY TEMPERATURE
CALL FIRE_FWD
C----- FIRST TIME INCREMENT FOR CALCULATING INCREMENT LENGTH ONLY
DEL=3.
C----- IF FIN=.TRUE., ANALYZE NEW FIRE
C----- IF FIN=.TRUE., TERMINATE RUN
IF(FIN)GOTO 1:DC

C----- INITIALIZE NODAL ENTHALPY VECTOR EN
C----- HOMOGENEOUS NODES - EN = ENTHALPY(HEAT) PER UNIT VOLUME
C----- INTERFACE NODES - EN = ENTHALPY(HEAT)
DO 10 I=1,NN
NE=NODINT(I)
IF(NODECPL(I),EQ.J.OR.NE.LT.C)GOTO 10
IF(NE.GT.0)CALL XVERSY(TE,ENT,'llNV,NE,T(I),EN(I))
C----- MASTER NODES AND INTERFACE NODES
IF(NODECPL(I),EQ.J.OR.NODI.EQ.J)CALL MINTP(I,NODCPL(I),P)
IF(NODI.EQ.J)EN(I)=P(I)*T(I)
10 CONTINUE

C----- START TIME INTEGRATION LOOP
C-----
C----- CONTINUE
SUM=FLOAT(KTIME)/FLOAT(KUPDA)
SUM=INT(SUM)
UPDA=DUM1+SUM,DUM2,OR,KTIME,END.
C----- CALCULATE INTERNAL HEAT FLOW BY CONDUCTION
IF(KUPDA)
1 CALL ASSA2(NN,NE,N,KTOP,X,Y,ELA,T,TMAX,AXIAL,FMAX)
DO 20 I=1,NN
20 DT(i)=AE(I,MAX)
CALL MACRY(V,T,F,MAX,NN)
C----- GET FIRE TEMPERATURE
CALL X85ERT(TIM,TO,ST,TIME,TFIRE)

C ---CALCULATE BOUNDARY HEAT FLOW
CALL FQ6RDT(FlOW,DTA,NN,MAX,TFIRE)

C ---CALCULATE INTERNALLY GENERATED HEAT FLOW
CALL FQ6FN(NN,NE,XTOP,EV4,T,FLOW)

C ---CALCULATE ENCLOSURE (VOID) HEAT FLOW
CALL ENCLUD(FLOW)

C ---CALCULATE HEAT CAPACITY MATRIX AT CURRENT TEMPERATURE
CALL ASSP2(NN,NAX,T,TT,TMAX,EV4,MODEL,NNODE1,MODEL,P,AXIAL)

C ---SUM APPROPRIATE QUANTITIES OF COUPLED NODES
CALL COUPLE(F)

C ---CALCULATE NEW NODAL TEMPERATURES
DO 5 SC = 1,NN
NOD=NODECl(I)
NODIN=NODINT(I)
IF(NODI.EQ.0) GOTO 950

C ---HOMOGENEOUS NODES ONLY
NODIN=NODI
IF(NODICALL M(TPK(I),K(I),EN(I),FLOW(I),F(I),NOD,DELT))
IF(NODI) GOTO 950

C ---INTERFACE NODES
IF(NODECl(I).LT.0) CALL MTPK(I,NODECl(I),P)
CALL CTTPK(I,T(I),P,EN,FlOW,F,DELT,MODEL,MODEL,N,EV4,NBC)

5C CONTINUE

C ---SET PRESCRIBED NODAL TEMPERATURES
CALL PF6RDT(TFIRE)

C ---CALL COUPLE

C ---PRINT CURRENT NODAL AND VOID AIR TEMPERATURES
CALL OUT4(SJ,NN,AF,X,TIME,TIME,DELT,I,T,TMAX,FOw,TFIRE,
1 NOD,AXIAL)

C ---SET CON=.FALSE. TO TERMINATE TIME INTEGRATION
CALL MAXO(NN,TMAX,TT,TIME,CON)

C ---CALCULATE NEW TIME INCREMENT DELTI
CALL DTIME(NN,DTA,MAX,NODINT,NODECl,MODEL,TIME,NODT)

C ---TIME TIME=DELT
IF(CON=.TRUE.) GOTO 950

C ---END TIME INTEGRATION LOOP
C ---END TIME INTEGRATION
C ---PRINT MAXIMUM TEMPERATURE OBTAINED DURING ANALYSIS
CALL OUT42(SJ,IY,NN,NE,X,Y,TIME,TIME,TT,TMAX,FOw,AXIAL)

945 GOTO 5

943 TCL RETURN

944 END

945

SUBROUTINE FQNDA

C ---THIS ROUTINE FORMS RADIATION AND CONVECTION MATRICES BR AND BC
COMMON/UNIT21,TEMPS,TIME,TAMB,THAB
COMMON/UNIT31,TTMAX,TTMIN,TAMB,THAB
PARAMETER (N=10,NN=30,NB=2*N)
COMMON/TAB(2*N+1,2*N),TAB(N),(2*N)
COMMON/ND(2*N),(2*N),ND(N)
COMMON/NDG(2*N),(2*N),NDG(N)
COMMON/NDG1,NODST(N),NBOUND(N),AREA(N3,N3).
1 EPSG(NB),ETR(NB),CP(ND),FA(N)

946 CALL LOGICAL(FA,FAS)

C ---READ NUMBER OF BOUNDARY NODES GROUPS
READ 10C,NGONG

947 IF(NGONG.EQ.0) RETURN

948 PRINT 100
C———EACH BOUNDARY FLOW NODE GROUP
962 DO 10 13=1,NFGNG
963 C———
964 C———READ TOO,FAGN1
965 C———IF FAF=TRUE, FIRE BOUNDARY ELSE AMBIENT TEMPERATURE
966 C———ING1 = NODE GROUP NUMBER
967 FAFING1=FA1
968 NFAFING1=ING1
969 NUM=NUMBING1
970 RET=DATAING1
971 EPSIG=EPSIGING1*SIGMA
972 IF(EPSIG.EQ.0.AND.RET.EQ.0) PRINT 330
973 IF(EPSIG.EQ.0.AND.RET.EQ.4) STOP
974 CALL ARGCORING1,BC(ING1),EPSIG,RET,BAREA,NUMI,NB,ING1
975 IND=IND+*NUM1
976 IF(FA1) PRINT 21C,ING1
977 IF(FAIL) PRINT 22C,ING1
978 10 CONTINUE
979 1C FORMAT()
980 2C FORMAT/* PRESCRIBED FLOW BOUNDARY*/1X,24(1H*)/ 981 1 /* NODE GROUPS AND TYPES OF BOUNDARIES*/
982 2C FORMAT/* NODE GROUP',I3,' FIRE BOUNDARY*/
983 2C FORMAT/* NODE GROUP',I3,' AMBIENT BOUNDARY*/
984 3C FORMAT/* BOTH EMITTIVITY AND CONVECTION FACTOR ZERO*/
985 RETURN
986 END
987
988 SUBROUTINE FGNDB3(T,FLOW,DTA,NN,MAX,TFIRE)
989 C———THIS ROUTINE PREPARES CALCULATION OF PRESCRIBED BOUNDARY FLOW
990 DIMENSION T(NN),DTA(NN),FLOW(NN)
991 PARAMETER NB=10,NMB=30,NMB2=2*NB
992 COMMON/FQINFQNG,INGF(NB),TR(NN3),TC(NN8)
993 3R(NN2),3C(NN82),TRD(NN1),TCO(NN8)
994 COMMON/BOUND,NBOUND(NB,NMB),BAREA(NB,NMB),
995 EPSG(NB),MTR(NB),CGF(NB),FA(NB)
996 COMMON/UNIT/SIGMA,TARS,TINIT,TAMB
997 LOGICAL FA
998 C———NULL FLOW VECTOR
999 DO 777 I=1,NN
1000 777 FLOW(I)=0.
1001 C———RETURN IF NO PRESCRIBED BOUNDARY FLOW
1002 IF(NFGNG.EQ.0) RETURN
1003 777 CONTINUE
1004 1C FORMAT/
1005 C———EACH BOUNDARY FLOW NODE GROUP
1006 DO 13 I=1,NFGNG
1007 1C FORMAT/
1008 ING=NFGING
1009 TRG=TMFB
1010 TNG=TMG
1011 IF(ING1) TNG=TFIRE
1012 IF(ING1) TRG=TFIRE
1013 NUM=NUMBING1
1014 CP=CGFING1
1015 DO 13 I=1,NUM
1016 NODE=NBOUNDING1,1)
1017 TNG=TNG+TNG
1018 13 CONTINUE
1019 C———RADIATION
1020 TRG=TRG+TNG**2
1021 TNG=TNG**4
1022 C———CONVECTION
1023 TNG=TMG
1024 TCD=CP*ABS(DWXY)*(CP-1.)
SUBROUTINE FQGEN(NN, NE, N, KTOP, EV4, T, FLOW)
!
C-----CALCULATE INTERNALLY GENERATED HEAT
!
DIMENSION N(NE), KTOP(4, N), EV4(NE), T(NN), FLOW(NN)

PARAMETER NN=20, MNR=10

COMMON/FMAT/C(MNR), T(MNV, MNR), C(MNV, MNR), T(MNV, MNR),

1 EN(T(MNV, MNR), CT(MNV, MNR), TC(MNV, MNR), BC(MNV, MNR), EL(MNR)

LOGICAL LOG, LOG

DO IC = 1, NN
HT=N(I)
IF(NCLT LOG) GOTO 100:
NOD=KTOP(4, I)
CALL XVERSY(TQ, GE, MNV, N1, T(NOD), FGEN)

15 FLOW(NOD)=FLOW(NOD)+EV4(I)*FGEN
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
SUBROUTINE HTRANS(HZ,H,N,SYM)
C-----THIS ROUTINE TRANSFORMS THE ZONE CONVECTION ARRAY H TO A NODE
C-----CONVECTION ARRAY STORED IN THE SAME ARRAY
C-----IF SYMMETRY IS PRESENT EXPAND CONVECTION VECTOR
DIMENSION HZ(1),H(N)
LOGICAL SYM
DO 1 I=1,N
HZ(I)=.5*(HZ(1-I)+HZ(I))
1 CONTINUE
HZ(1)=.5*(HZ(1)+HZ(N))
HZ(N+1)=.5*HZ(N)
3 CONTINUE
RETURN
END

SUBROUTINE INIT(NN,T,TT,TMAX,NODINT)
C-----SET INITIAL NODEAL TEMPERATURE
DIMENSION T(NN),TT(NN),TMAX(NN),NODINT(NN)
LOGICAL TRAX
COMMON/TOUN,TTOUN,GNMAX,GNMAXSY
COMMON/UNIT/SIGMA,TAPS,TINIT,TMAX,TAMS
IM=1
DO 1 I=1,NN
IF(NODINT(I).LT.0) GOTO 1
1 CONTINUE
IF(NR.EQ.1) GOTO 1
JX1=NX-1
NY1=NY-1
IF(NY.EQ.2) GOTO 25
DO 20 I=1,NX1
INY=(I-1)*NY1

20 CONTINUE
RETURN
END

SUBROUTINE INTERF(NN,NE,NR,NX,NY,KTOP,N,NODINT,NODCPL)
C-----THIS ROUTINE FORMS VECTOR FOR IDENTIFICATION OF
C-----INTERFACE AND FICTITIOUS NODES
C-----NODINT= 1 = FICTITIOUS NODE
C-----NODINT= 0 = INTERFACE NODE
C-----NODINT= 1 = HOMOGENEOUS NODE
COMMON/NINT,Q,NCP50
COMMON/CPL2CP,CP2CPL,CP2CPL2,CP2CPL3
COMMON/CPL2CPL2,CPL2CPL3
COMMON/CPL2CPL2,CPL2CPL3
DIMENSION KTOP(4,NE),N(NE),NODINT(NN),NODCPL(NN)
LOGICAL ELLIP
PRINT 200
DO 5 I=1,NN
5 CONTINUE
If(NR,EQ.1) GOTO 90
5 CONTINUE
NIT=NIT+1
5 CONTINUE
If(NR,EQ.2) GOTO 25
5 CONTINUE
Do 20 I=1,NN
10 CONTINUE
INT=(I-1)*NY
INT=INT+1
INT=INT+1
INT=INT+1
20 CONTINUE
RETURN
END
1152  N1=N1+1
1153  C------
1154  DO 20 J=2,N1
1155  IF(IE1.EQ.N1) GOTO 15
1156  DO 10 I=1,13
1157  IE2=IE2+(J-1)*N1
1158  IF(IE1.EQ.IE2.OR.IP1.EQ.IE2) GOTO 15
1159  N1=N1+1
1160  IF(IE1.EQ.N1) GOTO 15
1161  IF(N1.EQ.N2) GOTO 10
1162  IF(JUM.EQ.-2) GOTO 10
1163  NOD=INT(J)+1
1164  NODINT(NOD)=0
1165  NOD=NOD+NY
1166  NODINT(NOD)=G
1167  C------
1168  9. CONTINUE
1169  15 CONTINUE
1170  IE1=IE1+1
1171  M=INT(E1)
1172  C------
1173  2L CONTINUE
1174  25 CONTINUE
1175  C------
1176  IF(NX.EQ.2) GOTO 5G
1177  DO 4C I=1,N1
1178  INX=I-1
1179  JLX=I
1180  Y1=INT(E1)
1181  C------
1182  DO 4C J=2,N1
1183  C------
1184  IE2=IE2+NY1
1185  N2=INT(E2)
1186  IF(IE1.NE.IE2) GOTO 10
1187  IF(N1.EQ.N2) GOTO 10
1188  NOD=(J-1)*NY1
1189  NODINT(NOD)=C
1190  NOD=NOD+1
1191  NODINT(NOD)=0
1192  C------
1193  3C CONTINUE
1194  IE1=IE2
1195  M=IE2
1196  C------
1197  4C CONTINUE
1198  5C CONTINUE
1199  C------
1200  DO 6C J=1,N6
1201  M=INT(J)
1202  IF(IE1.EQ.M) GOTO 10
1203  DO 6C J=1,4
1204  NOD=INT(J)
1205  60 IF(NODINT(NOD).EQ.-1) NODINT(NOD)=M
1206  7D CONTINUE
1207  C------IF ONE NODE IN A COUPLED GROUP IS AN INTERFACE NODE
1208  C------ALL NODES IN THE GROUP ARE CONSIDERED INTERFACE NODES
1209  DO 9C J=1,4
1210  IF(NCP1.EQ.0) GOTO 10
1211  DO 8C J=1,4
1212  NOD=INT(J)+1
1213  IF(NOD.EQ.LD) GOTO 10
1214  IF(NODINT(NOD).EQ.0) GOTO 9C
1215  8S CONTINUE
DO 110 J=1,NX
II=(I-1)*NY
CONTINUE
END

SUBROUTINE INTP(NODE,NODEL,N,EV4,TI,PI)
C-----CALCULATE HEAT CAPACITY OF INTERFACE NODES
DIMENSION NODEL(4),N(1),EV4(1)
PARAMETER MV=2G, NR=1r
COMMON/RMAT(1NV,1NR),C(MV,1NR),FE(MV,1NR),ENT(MV,1NR)
DO 1(J=1,1r)
IE=NODEL(J)
E=EV4(IE)
DO 25(J=1,1r)
PRINT 21J,(NODINT(II+J),J=1,NY)
25 CONTINUE
END
SUBROUTINE INVER(A,M,MAX)
C-----THIS ROUTINE INVERTS THE MATRIX A AND STORES THE RESULT IN THE
C-----SAME MATRIX
DIMENSION A(MAX,1)
DO 200 N=1,M
A(N,N)=1/C
DO 290 J=1,N
IF N.EQ.J GOTO 290
A(N,J)=A(N,J)/C
290 CONTINUE
IF J.EQ.N GOTO 140
A(I,J)=A(I,J)+A(I,N)*A(N,J)
140 CONTINUE
RETURN
END

COMMENT MAIN PROGRAM CODED IN NULGOL FOR DYNAMIC ALLOCATION OF ARRAYS
FOR INFORMATION ABOUT ARRAYS SEE SUBROUTINE PROG2;
BEGIN
INTEGER NN,NE, NR,IX,IY,MAX;
REAL ARRAY XL,YL,XA,YA(1:100);
BOOLEAN AXIAL;
EXTERNAL FORTRAN PROCEDURE NET2,DIM2;
BEGIN
INTEGER MAX(1:NN);
INTEGER ARRAY KTOP(1:4, 1:4), X(1:NE), Nmodel(1:NN, 1:4), Nmodel,
KTOP(1:4, 1:NE), Y(1:NE), Nmodel(1:NN, 1:4), Nmodel,
REAL ARRAY X, Y, IT, IT, P, EN, F, Flow, NDCPL, Eta(1:NN, 1:4),
ELAC(1:4, 1:4), NModel(1:4, 1:4), X, Y, IT, IT, IMAX, Eta,
EN, P, EN, F, Flow, AX, NDCPL, NDDINT, ETA;
END;
SUBROUTINE MAT(NR)
C-----THIS ROUTINE READS MATERIAL INPUT
C-----MATERIAL NAME FOR IDENTIFICATION
READ 95, MAT
C-----INPUT MATERIAL AND ELEMENT PROPERTIES FOR EACH REGION
READ 95, ECC(1, NTE, NVE, EN, IT(1))
IF (NC(1), 1, 0.9, NC, 1.2, 0.1, 0.9) GOTO 1200
IF (NC(1), 1, 0.9, NC, 1.2, 0.1, 0.9) GOTO 1300
MAXTEM=MAXTEM(MAXTEM, NTE)
IF (IT(1), 1, 0.9, NC, 1.2, 0.1, 0.9) PRINT 97, IT(1), MAT
IF (NC(1), 1, 0.9, NC, 1.2, 0.1, 0.9) PRINT 102, NC(1), 1, 0.9, NC,
MAXTEM=MAXTEM(MAXTEM, NTE)
IF (IT(1), 1, 0.9, NC, 1.2, 0.1, 0.9) PRINT 97, IT(1), MAT
IF (NC(1), 1, 0.9, NC, 1.2, 0.1, 0.9) PRINT 102, NC(1), 1, 0.9, NC,
MAXTEM=MAXTEM(MAXTEM, NTE)
IF (IT(1), 1, 0.9, NC, 1.2, 0.1, 0.9) PRINT 97, IT(1), MAT
IF (NC(1), 1, 0.9, NC, 1.2, 0.1, 0.9) PRINT 102, NC(1), 1, 0.9, NC,
MAXTEM=MAXTEM(MAXTEM, NTE)
IF (IT(1), 1, 0.9, NC, 1.2, 0.1, 0.9) PRINT 97, IT(1), MAT
IF (NC(1), 1, 0.9, NC, 1.2, 0.1, 0.9) PRINT 102, NC(1), 1, 0.9, NC,
MAXTEM=MAXTEM(MAXTEM, NTE)
IF (IT(1), 1, 0.9, NC, 1.2, 0.1, 0.9) PRINT 97, IT(1), MAT
IF (NC(1), 1, 0.9, NC, 1.2, 0.1, 0.9) PRINT 102, NC(1), 1, 0.9, NC,
MAXTEM=MAXTEM(MAXTEM, NTE)
IF (IT(1), 1, 0.9, NC, 1.2, 0.1, 0.9) PRINT 97, IT(1), MAT
IF (NC(1), 1, 0.9, NC, 1.2, 0.1, 0.9) PRINT 102, NC(1), 1, 0.9, NC,
MAXTEM=MAXTEM(MAXTEM, NTE)
IF (IT(1), 1, 0.9, NC, 1.2, 0.1, 0.9) PRINT 97, IT(1), MAT
IF (NC(1), 1, 0.9, NC, 1.2, 0.1, 0.9) PRINT 102, NC(1), 1, 0.9, NC,
SUBROUTINE ANXCM(T, AX)
  CONTINUE
  IF (CON.EQ.0) GOTO 45
  PRINT 14C, (TV(K,1), OE(K,1), K=1, NNE)
  CONTINUE
  RETURN

SUBROUTINE MECH2(XL, YL, IX, IY, X, Y, KTOP)
  DIMENSION XL(1), YL(1), X(1), Y(1), KTOP(1)
  DO 5 I=1, IX
    DO 5 J=1, IY
      KK=J*Y(I-1)
      X(KK)=XL(I)
      Y(KK)=YL(J)
      CONTINUE
  END

  SUBROUTINE MECH2(XL, YL, IT, T)
    DIMENSION XL(1), YL(1), IT(1)
    CONTINUE
    RETURN
KTOP(J,IE)=KTOP(J,IE)+1
CONTINUE
RETURN
END

SUBROUTINE MINTP1(NODCPL,P)

KTOP(4,IE)=-KTOP(3,IE)+1
RETURN
END

RETURN
E=

SUBROUTINE ~INTP(I,NODCPL,P)

KTOP(1,IE)=KTOP(1,IE)+1

RETURN
E=

SUBROUTINE PACKV(A,X,R,M1,NN)

PARAMETER NCP=-5!
COMMON/COUPLE/COUPL(NCP,~),NCPLG

DO 3: J=2,1'
COUPL(NODCPL,J)
R(NOD.EQ.C)
RETURN
END

SUBROUTINE NET2(XL,YL,IX,IX,NR,AXIAL,XA,YA,NN,NE,MAX)

C-----INPUT GEOMETRICAL DATA AND GENERATE LINES PARALLEL WITH AXIS
C-----AND CALCULATE NUMBER OF GENERATED NODES AND ELEMENTS
C-----AXIAL .TRUE. IF AXI-SYMMETRIC PROBLEM

C----- NN NUMBER OF NODES
PARAMETER NMR=10,
COMMON/REGION,ELFICT(MNR),ET(MNR),SRDIA(4,MNR)
LOGICAL AXIAL,ELFICT
DIMENSION XL(1),YL(1),XAI(1),YAI(1),HEAD(20)

***MACHINE DEPENDENT STATEMENT + IN LDC*LIB.SEG + REQUIRED IN MAP ELEMENT
CALL SEG('TASEF')
PRINT 30

***INPUT TITLE OF RUN
READ 116,HEAD
PRINT 225,HEAD

***INPUT MAIN GEOMETRICAL DATA
READ 125,AXIAX,XMAX,YMAX,XBOX,YBOX,NA,NY
PRINT 210,XMAX,YMAX
PRINT 220,XMAX,YBOX

IF(NR.GE.2) NA=2
NP=NR+1
EPS=XMAX/1000.

***INPUT SUBREGION LIMITS
***READ THE DIAGONAL COORDINATES FOR EACH SUBREGION

IF(NR.GE.1) GO TO 5

PRINT 25,C
READ 10G,(ELFICT(J),SRDIA(I,J),I=1,4),J=2,NR1
PRINT 24G,(SRDIA(I,J),I=1,4),ELFICT(J),J=2,NR1
CONTINUE

5
CONTINUE

***INPUT SPECIFIED X - LINES

PRINT 230
READ 10C,(XA(I),I=1,NA)
PRINT 210,(XAMAX),I=1,NA
CONTINUE

6
CONTINUE

7
CONTINUE

***IF AN AXI-SYMMETRIC PROBLEM INPUT INNER RADIUS

8
CONTINUE

***GENERATE X-LINES

XL(1)=XAI(1)
DO 15 IX=2,10C
XL(IX)=XL(IX-1)+XBOX
15 CONTINUE

***CONTROL OF SPECIFIED X-LINES
DO 10 IX=1,IX
   IF(XL(IX-1).LT.(XA(I)-EPS)) XL(IX)=MIN(XL(IX),XA(I))
10   CONTINUE
C-----CONTROL OF SUBREGION LIMITS
   IF(XL(I).LT.XA(I)) GOTO 12
   DO 11 IX=1,IX
      IF(XL(IX-1).LT.(SROIAC(I,1)-EPS)) XL(IX)=MIN(XL(IX),SROIAC(I,1))
   11 CONTINUE
   CONTINUE
C-----CONTROL OF XMAX
   XL(I)=MIN(XL(I),XMAX)
   IF(XL(IX-1).LT.(PLMAX(I),IX),XMAX) GOTO 16
   CONTINUE
C-----CONTROL OF SPECIFIED X-LINES
   IF(YL(IY-1).LT.(YA(I)-EPS)) YL(IY)=MIN(YL(IY),YA(I))
   CONTINUE
17   CONTINUE
C-----CONTROL OF SUBREGION LIMITS
   IF(YL(IY).LT.(YMAX)-EPS)) YL(IY)=MIN(YL(IY),YMAX)
   CONTINUE
19   CONTINUE
C-----CONTROL OF YMAX
   YL(IY)=MIN(YL(IY),YMAX)
   IF(YL(IY-1).LT.(YMAX)-EPS)) YL(IY)=MIN(YL(IY),YMAX)
   CONTINUE
21   CONTINUE
C-----PRINT COORDINATES OF X- AND Y-LINES
   PRINT 31C,IX,XL(I),I=1,IX
   PRINT 32C,IY,YL(I),I=1,IY
   31 FORMAT(1H1)
   32 FORMAT(2CA4)
   TITLE OF RUN : ',20A4)
   TITLE = ',20A4)
   PRINT 33D,NR-,NR
   33 FORMC(1H1)
   NTRY=IR+IJ-1
   MAX=IT+4
   PRINT 34C,IX,IX}
   34 FORMC(D8.4)
   DO 15 IY=1,IN
      IF(YL(IY-1).LT.(YMAX)-EPS)) YL(IY)=MIN(YL(IY),YMAX)
      CONTINUE
   15 CONTINUE
C-----TEMPERATURE ANALYSIS OF STRUCTURES EXPOSED TO FIRE
C "SOLVES NON LINEAR TRANSIENT FIELD PROBLEMS"
C "TWO DIMENSIONAL VERSION "
C "PROGRAMMED BY Ulf Wickstrom" LUND FEB 1979"
SUBROUTINE NGRID(X,Y)
C---- THIS ROUTINE READS AND FORMS NODE GROUP DATA
DIMENSION X(1),Y(1)
COMMON/NOD/NB-1,NB+2*NB,NS,NNB
COMMON/AREA(10),NB,NNB,EPXG(NB),BETA(NB),CPG(NB),FAX(NB)
COMMON/CEX(10),TAI(2)
LOGICAL FA
C----
PRINT 22O
READ 10C,NGROUP
C---- DO 10 I=1,NGROUP
C---- READ 10C,NCHCK,NUMI(I),EPSG(I),BETA(I),CPG(I)
C---- IF(I.NE.NCHCK) GO TO 1CCG
NUM1=NUMI(I)
C---- 
READ 10C,(NBOUND(I,J),J=1,NUMI)
MODI=NBOUND(I,1)
DO 10 J=2,NUMI
MODJ=NBOUND(I,J)
DAREA(I,J)=SQR((X(MODI)-X(MODJ))**2+(Y(MODI)-Y(MODJ))**2)
MODI=MODJ
10 CONTINUE
C----PRINT INPUT DATA
C-- DO IS IS+1,NGROUP
C-- PRINT 23O,IS
C-- IF (EPSG(IS).GT.6.0) AND (EPSG(IS).EQ.10.0) GOTO 22
C-- PRINT 22O,EPSG(IS),BETA(IS),CPG(IS)
C-- ZC PRINT 230,(NBOUND(I,J),J=1,NUMI)
C-- 210 FORMAT(* NODE GROUP=[1,11(1M)])
C-- 21O FORMAT(* NODE GROUP=[13])
C-- 220 FORMAT(* EMISSIVITY=[1.0,5], CONVECTION FACTOR=[1.0,5])
C-- 230 FORMAT(* NODES=1,15(11),15)
GOTO 101
SUBROUTINE OUT2(IX, IY, NN, NE, X, Y, TIME, KTIME, I, TT, Tmax, FLOW, AXIAL)
C-----THIS ROUTINE PRINTS MAXIMUM CALCULATED NODAL TEMPERATURES
COMMON/FIRE/TIM(50), TB(50), TITFIR
INTEGER TITFIR (18)
LOGICAL TMAX, AXIAL
DIMENSION X(NN), Y(NN), TT(NN), Tmax(NN), Flow(NN)
PRINT 220, TITFIR, X(NN), Y(NN)
1 DO 1 = 1, IX
1 DO 2 = 1, IDUM + 1
1 DO 3 = 1, IDUM + 2
IF((IY).LE.7) PRINT 210, (J, TT(J), J = IDUM1, IDUM2)
IF((IY).GT.7) PRINT 230, (TT(J), J = IDUM1, IDUM2)
CONTINUE
PRINT 220, TIME, KTIME
2 FORMAT(///1A, 75(1HF)/2H F	MAXIMAL TEMPERATURES"/", 1.5A4)/
3 FORMAT(1F15.5, 1F13.0, 1F12H, "TMAX=", 1F12H, "TIME=", 1F12H, "FACTOR=", 1F12H, "KUPDA"
4 CONTINUE
PRINT 220, I, TIME, KTIME, TFIRE, NODT
IF(TMAX(I).GT.0.10T(I)) GOTO 70
PRINT 100, I, I, NENC
DO 10 I = 1, NENC
1 PRINT 300, I, TIME, I, NENC
IF(TMAX(I).GT.0.10T(I)) GOTO 70
CONTINUE
C-----IF THE NODAL TEMPERATURE DECREASES SET TMAX=.TRUE. AND PRINT
70 CONTINUE
10 RETURN
END
SUBROUTINE OUT2(IX, IY, NN, NE, X, Y, TIME, KTIME, I, TT, Tmax, FLOW, AXIAL)
C-----THIS ROUTINE PRINTS NODAL TEMPERATURES AND VOID AIR TEMPERATURES
COMMON X(NN), Y(NN), TT(NN), Tmax(NN), Flow(NN)
LOGICAL TMAX, AXIAL, I, X, Y, TIME, DELTA, I, TT, Tmax, Flow
TIME = TIME + DELTA
DO 5 I = 1, NN
5 IF(TMAX(I).GT.0.10T(I)) GOTO 1
C-----IF THE NODAL TEMPERATURE DECREASES SET TMAX=.TRUE. AND PRINT
1 CONTINUE
C-----MAX TEMPERATURE TT
C-----IF TIME=0 PRINT ALL TEMPERATURES
IF(TIME+TOUT(I) = I.E-4) GOTO 70
PRINT 100, I, TIME, TIME, TFIRE, NODT
IF(NOT, LEN) GOTO 70
PRINT 100, I, TIME, TIME, TFIRE, NODT
DO 20 I = 1, NENC
20 PRINT 300, I, TIME, TIME, TFIRE, NODT
CONTINUE
10 RETURN
END
IF(LDUM) PRINT 210,(J,T(J),J=IUM1,IUM2)
IF(.NOT.LDUM) PRINT 220,(T(J),J=IUM1,IUM2)

10 CONTINUE
70 CONTINUE
100 FORMAT(/1Y,6(1H*)/6H TIME+FR.72X,6(1H*)/' INCREMENT',
173 1 * NUMBER*16,
173 22X17(1H*)/FIRE TEMPERATURE'F.+T.0+2X,6(1H*)
173 3+2X* TIME INCREMENT LIMITING N0ME*15/)
178 200 FORMAT(//4H NOD+14:5X:BNMAX TEMP,6.0+5X,TIME'+.10,4S,5X,
178 179 1 SD0ELT,010,4/X,70(1H*))
178 210 FORMAT(13F15,F5.0)
178 220 FORMAT(18F7.0)
300 FORMAT(' FNCLOSE AIR TEMP.F.'RATURE')
310 FORMAT(I10)
RETURN
END

SUBROUTINE PPOG2(IX,IY,NN,NE,NR,N,KT0P,NOnFL,MNOn~L,x,Y,T,TT,TMAX,
ELA,EV4,A.MAX,P,W,EN,F,FLOW.AXIAL,NOOCPL,NOnINT,DTA)

C-------------------------------------------------------------------------
C-----TEMPERATURE ANALYSIS OF STRUCTURES EXPOSED TO FIRE
C-----FINITE ELEMENT PROGRAM FOR ANALYSIS OF TRANSIENT NONLINEAR
C-----HEAT TRANSFER PROBLEMS
C-----PROGRAMMED BY
C-----ULF WICKSTROM
C-----LUND INSTITUTE OF TECHNOLOGY
C-----MARCH 1979
C-----THIS IS THE MAIN控制 ROUTINE
C-----DEFINITIONS OF VARIABLES
C-----IX, IY NUMBER OF X- AND Y- LINES
C-----NN NUMBER OF NODES IN BASE STRUCTURE
C-----NE NUMBER OF ELEMENTS IN BASE STRUCTURE
C-----N VECTOR OF REGION NUMBERS
C-----KTOP NODES ADJACENT TO EACH ELEMENT
C-----NODEL ELEMENTS ADJACENT TO EACH NODE
C-----X,Y NODE COORDINATES
C-----T MAXIMUM NODAL TEMPERATURES
C-----TMAX TRUE IF MAXIMUM NODAL TEMPERATURE OBTAINED
C-----A HEAT CONDUCTION MATRIX
C-----P HEAT CAPACITY VECTOR
C-----W NODAL VOLUME VECTOR
C-----EN NODAL ENTHALPY VECTOR
C-----FLOW INTERNAL NODAL HEAT FLOW VECTOR
C-----AXIAL TRUE IF AXISYMMETRIC PROBLEM
C-----NOOCPL INDICATES COUPLED NODES
C-----NOINT INDICATES INTERFACE NODES
C-----DTA DUMMY VECTOR FOR CRITICAL TIME INCREMENT CALCULATION
C-----PARAMETER CONSTANTS
C-----NB MAXIMUM NUMBER OF NODE GROUPS
C----HNB MAXIMUM NUMBER OF NODES IN ONE NODE GROUP
C----HCP MAXIMUM NUMBER OF COUPLED GROUPS OF NODES
C----HNR MAXIMUM NUMBER OF REGIONS
C----HNV MAXIMUM NUMBER OF VALUE PAIRS
C-----
C-----COMMON FIELDS
C-----
C-----COUPLE DATA ON COUPLED NODES
1580 C----- NCPLG NUMBER OF COUPLED GROUPS OF NODES
1581 C----- NCPLG MATRIX OF COUPLED NODES
1582 C-----DIM DIMENSIONS OF CERTAIN ARRAYS
1583 C----- MAXA MAXIMUM NUMBER OF NODE GROUPS DEFINING ONE ENCLOSURE
1584 C----- MAXN MAXIMUM NUMBER OF NODES AROUND ONE ENCLOSURE
1585 C-----DUMMY DUMMY MATRICES
1586 C-----ENCLOS ENCLOSURE DATA
1587 C----- LTN TRUE IF STRUCTURE CONTAINS VOID OR ENCLOSURE
1588 C----- NENC NUMBER OF ENCLOSURES
1589 C----- NENCNS VECTOR OF NUMBER OF NOOD GROUPS
1590 C----- IGFN MATRIX OF NODES
1591 C----- NNOGFN NUMBER OF NODES SURROUNDING VOID
1592 C----- XSYM TRUE IF VOID SYMMETRICAL AROUND X-AXIS
1593 C----- YSYM TRUE IF VOID SYMMETRICAL AROUND Y-AXIS
1594 C----- ENCON ENCLOSURE CONVECTION DATA
1595 C----- H ARRAY OF ENCLOSURE CONVECTION VECTORS
1596 C----- TAIN ENCLOSURE AIR TEMPERATURE
1597 C----- ENCRA ENCLOSED RADIATION DATA
1598 C----- E ARRAY OF ENCLOSURE RADIATION MATRICES
1599 C----- FIRE FIRE TEMPERATURE DATA
1600 C----- TMTF TIME - FIRE TEMPERATURE PAIRS
1601 C----- FID FIRE IDENTIFIER
1602 C----- FB8 PRESCRIBED HEAT FLOW DATA
1603 C----- NFB8M NUMBER OF NODE GROUPS DEFINING PRESCRIBED FLOW BOUNDARIES
1604 C----- NF8VE VECTOR OF NODE GROUPS DEFINING PRESCRIBED FLOW
1605 C----- TP+TC VECTORS OF MODIFIED TEMPERATURE
1606 C----- BR+BC RADIATION AND CONVECTION BOUNDARY MATRICES
1607 C----- NOG DATA ON NODE GROUPS
1608 C----- NUMB VECTOR OF NUMBER OF NODES IN THE NODE GROUPS
1609 C----- NBOUND MATRIX OF NODE NUMBERS IN THE NOOD GROUPS
1610 C----- BNTR MATRIX OF DISTANCES BETWEEN NODES
1611 C----- EPSO VECTOR OF EMISSIVITY OF NOOD GROUPS
1612 C----- BETA VECTOR OF CONVECTION FACTORS OF NOOD GROUPS
1613 C----- CMp VECTOR OF CONVECTION POWERS OF NOOD GROUPS
1614 C----- FA TRUE FOR FIRE BOUNDARY NODE GROUPS
1615 C----- PTB PRESCRIBED TEMPERATURE
1616 C----- NPTB NUMBER OF NODE GROUPS DEFINING PRESCRIBED TEMPERATURE
1617 C----- NPTO VECTOR OF NODE GROUPS DEFINING PRESCRIBED TEMPERATURES
1618 C----- RGEO GEOMETRIC DATA
1619 C----- ELFICT TRUE FOR FICTITIOUS ELEMENTS
1620 C----- ET ELEMENT THICKNESS
1621 C----- SSTDG SUBREGION DIAGONAL DATA
1622 C----- MMDT MATERIAL DATA
1623 C----- CCC TRUE IF CONDUCTIVITY IS FUNCTION MAXIMUM TEMPERATURE
1624 C----- TC+C TEMPERATURE - CONDUCTIVITY PAIRS
1625 C----- TF+NT TEMPERATURE - SPECIFIC VOLUME FNPHALY PAIRS
1626 C----- NR NOMINAL SPECIFIC VOLUME HPR
1627 C----- TG+GE TEMPERATURE - INTERNALLY GENERATED HEAT PAIRS
1628 C----- LG TRUE IF INTERNAL HEAT IS GENERATED
1629 C----- TOUT TIME DATA
1630 C----- II COUNTER
1631 C----- TOUT VECTOR OF PRINT OUT TIMES
1632 C----- TIMFX MAXIMUM TIM
1633 C----- DTMAX MAXIMUM TIME INCREMENT
1634 C----- TIMFA TIME INCREMENT FACTOR
1635 C----- KTIME MAXIMUM NUMBER TIME INCREMENTS
C-----UNIT  KNUM  NUMBER OF TIME STEPS BETWEEN UPDATING CONDUCTION MATRIX
C-----UNIT  SDNUM  UNIT DEPENDENT CONSTANTS
C-----UNIT  SIGMA  STEFAN-BOLTZMANN CONSTANT
C-----UNIT  TABS  ABSOLUTE TEMPERATURE SHIFT
C-----UNIT  TINT  INITIAL TEMPERATURE
C-----DIMENSION N(N),KTOP(N,N),X(N),Y(N),T(N),TT(T),TMAY(T),TMAX(T),TMIN(T),NUM(T)
C-----DIMENSION NODCPL(N),NODINT(N),MODE(N),MODEL(N),DETA(N),DODATA(N)
C-----LOGICAL MAXAXIAL
C-----FOR EACH VECTOR N
C-----READ NPTNG, NPTG(N)
C-----IF(NPTNG.~Q.O) RETURN
C-----EACH PRESCRIBED TEMPERATURE AROUND NODE GROUP
DO 20 I=1,NPTNG
C-----READ INO,FAL(INO)
C-----IF(INO.~Q.O) THEN TEMPAT = FAL(INO)
C-----ELSE IF(INO.~Q.O) THEN TEMPAT = INO
C-----FA(I)= NODE GROUP NUMBER
C-----READ INIT(INO)
C-----IF(INIT.~Q.O) THEN
C-----DIMENSION NSTEPS,STEPS(4),STEPS1(4),STEPS2(4)
C-----ABSOLUTE TEMPERATURE START TIME
C-----STEPS(1)= INIT(INO)
C-----IF(STEPS(1).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE END TIME
C-----STEPS(2)= INIT(INO)
C-----IF(STEPS(2).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE HISTOGRAM TEMPAT
C-----STEPS(3)= TEMPAT
C-----IF(STEPS(3).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE INITIAL TEMPAT
C-----STEPS(4)= TEMPAT
C-----IF(STEPS(4).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(1)= STEPS(INO)
C-----IF(STEPS(1).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(2)= STEPS(INO)
C-----IF(STEPS(2).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(3)= STEPS(INO)
C-----IF(STEPS(3).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(4)= STEPS(INO)
C-----IF(STEPS(4).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(1)= STEPS(INO)
C-----IF(STEPS(1).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(2)= STEPS(INO)
C-----IF(STEPS(2).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(3)= STEPS(INO)
C-----IF(STEPS(3).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(4)= STEPS(INO)
C-----IF(STEPS(4).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(1)= STEPS(INO)
C-----IF(STEPS(1).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(2)= STEPS(INO)
C-----IF(STEPS(2).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(3)= STEPS(INO)
C-----IF(STEPS(3).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(4)= STEPS(INO)
C-----IF(STEPS(4).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(1)= STEPS(INO)
C-----IF(STEPS(1).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(2)= STEPS(INO)
C-----IF(STEPS(2).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(3)= STEPS(INO)
C-----IF(STEPS(3).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(4)= STEPS(INO)
C-----IF(STEPS(4).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(1)= STEPS(INO)
C-----IF(STEPS(1).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(2)= STEPS(INO)
C-----IF(STEPS(2).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(3)= STEPS(INO)
C-----IF(STEPS(3).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(4)= STEPS(INO)
C-----IF(STEPS(4).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(1)= STEPS(INO)
C-----IF(STEPS(1).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(2)= STEPS(INO)
C-----IF(STEPS(2).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(3)= STEPS(INO)
C-----IF(STEPS(3).~Q.O) THEN
C-----ABSOLUTE TEMPERATURE TIME
C-----STEPS(4)= STEPS(INO)
1920  NODE=ROUND(NING1,J)
1921  IF (NODE=ROUND(NING1,J)) PRINT 310,NODE
1922  ENDIF
1923  STOP
1924  CONTINUE
1925  CONTINUE
1926  CONTINUE
1927  CONTINUE
1928  CONTINUE
1929  CONTINUE
1930  CONTINUE
1931  CONTINUE
1932  CONTINUE
1933  CONTINUE
1934  CONTINUE
1935  CONTINUE
1936  CONTINUE
1937  CONTINUE
1938  CONTINUE
1939  CONTINUE
1940  CONTINUE
1941  CONTINUE
1942  CONTINUE
1943  CONTINUE
1944  CONTINUE
1945  CONTINUE
1946  CONTINUE
1947  CONTINUE
1948  CONTINUE
1949  CONTINUE
1950  CONTINUE
1951  CONTINUE
1952  CONTINUE
1953  CONTINUE
1954  CONTINUE
1955  CONTINUE
1956  CONTINUE
1957  CONTINUE
1958  CONTINUE
1959  CONTINUE
1960  CONTINUE
1961  CONTINUE
1962  CONTINUE
1963  CONTINUE
1964  CONTINUE
1965  CONTINUE
1966  CONTINUE
1967  CONTINUE
1968  CONTINUE
1969  CONTINUE
1970  CONTINUE
1971  CONTINUE
1972  CONTINUE
1973  CONTINUE
1974  CONTINUE
1975  CONTINUE
1976  CONTINUE
1977  CONTINUE
1978  CONTINUE
1979  CONTINUE
1980  CONTINUE
1981  CONTINUE
1982  CONTINUE
1983  CONTINUE

C-----SET PRESCRIBED NODAL BOUNDARY TEMPERATURE
C-----DIMENSION T(I)
C-----PARAMETER NB=10,NMR=30,NNB=2*NB
C-----COMMON/PNS/NPTNG,NPTG(NB)
C-----COMMON/BND/NMB(NB),RNBBOUND(NB,NBB),TN(100),
C-----1 EPS(10,10),ETA(10),T(15,N)
C-----COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
C-----LOGICAL FA
C-----IF(NPTNG.EQ.0) RETURN
C-----EACH PRESCRIBED TEMPERATURE BOUNDARY NODE GROUP
DO 10 IB=1,NPTNG
TG=TAFIRE
ING1=NPTG(IB)
IF(FA(ING1).EQ.TAFIRE) TNODE=TN(ING1)
DO 10 J=1,N
10 TNODE=TNODE+T(J)
CONTINUE
RETURN
END

C-----THIS ROUTINE FORMS THE LOCAL ENCLOSURE SURFACE RADIATION HEAT
C-----DIMENSION G(I,J),ETA(I,J),(N,N)
GTOTAL=0.
DO 20 I=1,N
G(TOT)=G(TOT)+G(I,J)
20 CONTINUE
RETURN
END

C-----THIS ROUTINE FORMS VECTOR OF REGION NUMBERS N OF EACH ELEMENT
C-----DIMENSION X(IN),Y(IN),N(M,NE),KTOP(IN,NE),NOD(X(IN),N(N,NE)),NOD(100,100,100)
C-----PARAMETER NMR=10
C-----COMMON/ROF/ELFICT(NMR),ET(10,10),S3DIAC(10,10)
C-----LOGICAL ELFICT
EPS=1.E-7
DO 10 I=1,NE
1984  N(I) = 1
1985  IF(NR.EQ.1) GO TO 10
1986  ND2 = KTOP(1,1)
1987  ND2 = KTOP(4,1)
1988  DO 5 J = 2, NR
1989  IF((XND1 - SQRTAC(3, J)) / GT - EPS) GOTO 10
1990  IF((YND1 - SQRTAC(4, J)) / GT - EPS) GOTO 10
1991  IF((XND2 - SQRTAC(3, J)) / LT - EPS) GOTO 10
1992  IF((YND2 - SQRTAC(4, J)) / LT - EPS) GOTO 10
1993  N(I) = J
1994  5 CONTINUE
1995  10 CONTINUE
1996  DO 40 I = 1, NN
1997  10 0
1998  DO 30 IE = 1, NT
1999  IF(EFCTCNIJ) GOTO 30
2000  20 CONTINUE
2001  N(IE) = IE
2002  IF((KTOP(1,IE),NE.1) GOTO 20
2003  10 CONTINUE
2004  RETURN
2005  END
2006  SUBROUTINE TIME
2007  COMMON/TOUT/II,TOUT(100),TIMMAX,DTMAX,KTMAX,KUPDA
2008  PRINT 200
2009  READ 100,(TOUT(I),I=1,NT)
2010  PRINT 210,
2012  CRITICAL TIME INCREMENT FACTOR=2.G8.3/
2013  MAXIMUM NUMBER OF TIME INCREMENTS=2.IE/
2014  NUMBER OF STEPS BETWEEN UPDATING OF CONDUCTION MATRIX=2.ISJ
2015  RETURN
2016  END
2017  SUBROUTINE VIEWFCCX,Y,D,EN,VIEW,MAXNOD)
2018  DIMENSION X(l), Y(l), O(l), VIEW(MAXNOD,MAXNOD)
2019  COMMON/ENCLOS/lEN, ENC,PEN, ENCEN, ENCEN(2), ENCEN(2)
2020  COMMON/ENCLOS/LEN, ENC, ENCEN, ENCEN(2), ENCEN(2)
2021  COMMON/ENCLOS/LEN, ENC, ENCEN, ENCEN(2), ENCEN(2)
2022  COMMON/ENCLOS/LEN, ENC, ENCEN, ENCEN(2), ENCEN(2)
2023  COMMON/ENCLOS/LEN, ENC, ENCEN, ENCEN(2), ENCEN(2)
2024  COMMON/ENCLOS/LEN, ENC, ENCEN, ENCEN(2), ENCEN(2)
2025  COMMON/ENCLOS/LEN, ENC, ENCEN, ENCEN(2), ENCEN(2)
2026  COMMON/ENCLOS/LEN, ENC, ENCEN, ENCEN(2), ENCEN(2)
2046 LOGICAL LFN
2047 LOGICAL LDUM, XSYM, YSYM, DSM
2048 INTEGER EN
2051 DO 15 I = 1, MAXNO
2052 DO 15 J = 1, MAXNO
2053 15 VIEW(I, J) = 0.
2054 C-----COMPUTE VIEW-FACTORS USING HOTTEL'S CROSSED-STRING METHOD
2055 NENG = NENG(NENG, EN)
2056 SIGNX2 = 1.
2057 SIGNY2 = 1.
2058 IN=0
2059 C-----EACH NODE GROUP
2060 DO 100 I = 1, NENG
2061 DO 100 J = 1, MAXNO
2062 C-----COMPUTE VIEW-FACTORS USING HOTTEL'S CROSSED-STRING METHOD
2063 NUMI = NUMI + 1
2064 C-----EACH NODE GROUP
2065 DO 100 I = 1, NENG
2066 C-----EACH NODE GROUP
2067 DO 100 J = 1, NENG
2068 C-----EACH NODE GROUP
2069 C-----EACH NODE GROUP
2070 C-----EACH NODE GROUP
2071 DO 100 I = 1, NENG
2072 C-----EACH NODE GROUP
2073 DO 100 J = 1, NENG
2074 C-----EACH NODE GROUP
2075 C-----FORM THE ZONE AREA VECTOR D
2076 D(IN) = 0.1
2077 JN=0
2078 C-----FORM THE ZONE AREA VECTOR D
2079 C-----FORM THE ZONE AREA VECTOR D
2080 C-----FORM THE ZONE AREA VECTOR D
2081 C-----FORM THE ZONE AREA VECTOR D
2082 C-----FORM THE ZONE AREA VECTOR D
2083 C-----FORM THE ZONE AREA VECTOR D
2084 C-----FORM THE ZONE AREA VECTOR D
2085 C-----FORM THE ZONE AREA VECTOR D
2086 C-----FORM THE ZONE AREA VECTOR D
2087 C-----FORM THE ZONE AREA VECTOR D
2088 C-----FORM THE ZONE AREA VECTOR D
2089 C-----FORM THE ZONE AREA VECTOR D
2090 C-----FORM THE ZONE AREA VECTOR D
2091 CONTINUE
2092 IF (IN.GE.JN) GOTO 100
2093 IF (JN.GE.IN) GOTO 100
2094 80 CONTINUE
2095 X3 = XSYM(EN, NUMI)
2096 Y3 = YSYM(EN, NUMI)
2097 X4 = XSYM(EN, NUMI)
2098 Y4 = YSYM(EN, NUMI)
2099 DO 1505 (I = 1, X3) = (I - Y3) ** 2
2010 DO 1505 (I = 1, X4) = (I - Y4) ** 2
2011 DO 1505 (I = 1, X4) = (I - Y4) ** 2
2012 DO 1505 (I = 1, X3) = (I - Y3) ** 2
2013 C-----HOTTEL'S CROSSED-STRING METHOD
2014 C-----HOTTEL'S CROSSED-STRING METHOD
2015 C-----HOTTEL'S CROSSED-STRING METHOD
2016 C-----HOTTEL'S CROSSED-STRING METHOD
2017 C-----HOTTEL'S CROSSED-STRING METHOD
2018 C-----HOTTEL'S CROSSED-STRING METHOD
2019 C-----HOTTEL'S CROSSED-STRING METHOD
2020 C-----HOTTEL'S CROSSED-STRING METHOD
2021 C-----HOTTEL'S CROSSED-STRING METHOD
2022 C-----HOTTEL'S CROSSED-STRING METHOD
2023 C-----HOTTEL'S CROSSED-STRING METHOD
2024 C-----HOTTEL'S CROSSED-STRING METHOD
2025 C-----HOTTEL'S CROSSED-STRING METHOD
2026 C-----HOTTEL'S CROSSED-STRING METHOD
2027 C-----HOTTEL'S CROSSED-STRING METHOD
2028 C-----HOTTEL'S CROSSED-STRING METHOD
2029 C-----HOTTEL'S CROSSED-STRING METHOD
2030 C-----HOTTEL'S CROSSED-STRING METHOD
2031 C-----HOTTEL'S CROSSED-STRING METHOD
2032 C-----HOTTEL'S CROSSED-STRING METHOD
2033 C-----HOTTEL'S CROSSED-STRING METHOD
2034 C-----HOTTEL'S CROSSED-STRING METHOD
2035 C-----HOTTEL'S CROSSED-STRING METHOD
2036 C-----HOTTEL'S CROSSED-STRING METHOD
2037 C-----HOTTEL'S CROSSED-STRING METHOD
2038 C-----HOTTEL'S CROSSED-STRING METHOD
2039 C-----HOTTEL'S CROSSED-STRING METHOD
2040 C-----HOTTEL'S CROSSED-STRING METHOD
2041 C-----HOTTEL'S CROSSED-STRING METHOD
2042 C-----HOTTEL'S CROSSED-STRING METHOD
2043 C-----HOTTEL'S CROSSED-STRING METHOD
2044 C-----HOTTEL'S CROSSED-STRING METHOD
2045 C-----HOTTEL'S CROSSED-STRING METHOD
2046 C-----HOTTEL'S CROSSED-STRING METHOD
2047 C-----HOTTEL'S CROSSED-STRING METHOD
2048 C-----HOTTEL'S CROSSED-STRING METHOD
2049 C-----HOTTEL'S CROSSED-STRING METHOD
2050 C-----HOTTEL'S CROSSED-STRING METHOD
2051 C-----HOTTEL'S CROSSED-STRING METHOD
2052 C-----HOTTEL'S CROSSED-STRING METHOD
2053 C-----HOTTEL'S CROSSED-STRING METHOD
2054 C-----HOTTEL'S CROSSED-STRING METHOD
2055 C-----HOTTEL'S CROSSED-STRING METHOD
2056 C-----HOTTEL'S CROSSED-STRING METHOD
2057 C-----HOTTEL'S CROSSED-STRING METHOD
2058 C-----HOTTEL'S CROSSED-STRING METHOD
2059 C-----HOTTEL'S CROSSED-STRING METHOD
2060 C-----HOTTEL'S CROSSED-STRING METHOD
2061 C-----HOTTEL'S CROSSED-STRING METHOD
2062 C-----HOTTEL'S CROSSED-STRING METHOD
2063 C-----HOTTEL'S CROSSED-STRING METHOD
2064 C-----HOTTEL'S CROSSED-STRING METHOD
2065 C-----HOTTEL'S CROSSED-STRING METHOD
2066 C-----HOTTEL'S CROSSED-STRING METHOD
2067 C-----HOTTEL'S CROSSED-STRING METHOD
2068 C-----HOTTEL'S CROSSED-STRING METHOD
2069 C-----HOTTEL'S CROSSED-STRING METHOD
2070 C-----HOTTEL'S CROSSED-STRING METHOD
2071 C-----HOTTEL'S CROSSED-STRING METHOD
2072 C-----HOTTEL'S CROSSED-STRING METHOD
2073 C-----HOTTEL'S CROSSED-STRING METHOD
2074 C-----HOTTEL'S CROSSED-STRING METHOD
2075 C-----HOTTEL'S CROSSED-STRING METHOD
2076 C-----HOTTEL'S CROSSED-STRING METHOD
2077 C-----HOTTEL'S CROSSED-STRING METHOD
2078 C-----HOTTEL'S CROSSED-STRING METHOD
2079 C-----HOTTEL'S CROSSED-STRING METHOD
2080 C-----HOTTEL'S CROSSED-STRING METHOD
2081 C-----HOTTEL'S CROSSED-STRING METHOD
2082 C-----HOTTEL'S CROSSED-STRING METHOD
2083 C-----HOTTEL'S CROSSED-STRING METHOD
2084 C-----HOTTEL'S CROSSED-STRING METHOD
2085 C-----HOTTEL'S CROSSED-STRING METHOD
2086 C-----HOTTEL'S CROSSED-STRING METHOD
2087 C-----HOTTEL'S CROSSED-STRING METHOD
2088 C-----HOTTEL'S CROSSED-STRING METHOD
2089 C-----HOTTEL'S CROSSED-STRING METHOD
2090 C-----HOTTEL'S CROSSED-STRING METHOD
2091 C-----HOTTEL'S CROSSED-STRING METHOD
2092 C-----HOTTEL'S CROSSED-STRING METHOD
2093 C-----HOTTEL'S CROSSED-STRING METHOD
2094 C-----HOTTEL'S CROSSED-STRING METHOD
2095 C-----HOTTEL'S CROSSED-STRING METHOD
2096 C-----HOTTEL'S CROSSED-STRING METHOD
2097 C-----HOTTEL'S CROSSED-STRING METHOD
2098 C-----HOTTEL'S CROSSED-STRING METHOD
2099 C-----HOTTEL'S CROSSED-STRING METHOD
2100 C-----HOTTEL'S CROSSED-STRING METHOD
2101 C-----HOTTEL'S CROSSED-STRING METHOD
2102 C-----HOTTEL'S CROSSED-STRING METHOD
2103 C-----HOTTEL'S CROSSED-STRING METHOD
2104 C-----HOTTEL'S CROSSED-STRING METHOD
2105 C-----HOTTEL'S CROSSED-STRING METHOD
2106 C-----HOTTEL'S CROSSED-STRING METHOD
2107 C-----HOTTEL'S CROSSED-STRING METHOD
2108 C-----HOTTEL'S CROSSED-STRING METHOD
2109 C-----HOTTEL'S CROSSED-STRING METHOD
2110 C-----HOTTEL'S CROSSED-STRING METHOD
2111 C-----HOTTEL'S CROSSED-STRING METHOD

SUBROUTINE XVPSY(X,Y,N,M,XS,YS)
C-----FIND YS AS FUNCTION OF XS BY LINEAR INTERPOLATION
C-----IN TABLE OF X- AND Y-VALUES
C-----IN TABLE OF X- AND Y-VALUES
DIMENSION Y(N+1),X(N+1)
DO 10 I=2,N
  IF(XS.GE.X(I,M))GOTO 10
  YS=Y(I-1,M)+(XS-X(I-1,M))*(Y(I,M)-Y(I-1,M))/(X(I,M)-X(I-1,M))
  GOTO 11
10 CONTINUE
11 RETURN
END
APPENDIX C - Example input

Input cards used in examples I-III in Section 5.

Example I

1 SQUARE PLATE
2 F+1,1-125,.125-...
3 0
4 UNIT MATERIAL DATA
5 F+2.2-...
6 --10000:1-
7 ..10000:10000-
8 .00001:10000...
9 i
10 1+17,1:1-
11 9 18 27 36 45 54 63 72 81 90 79 78 77 76 75 74 73
12 1
13 F:1
14 0
15 NOVOID
16 20:1:1:1...-
17 .05 .10 .15 .20 .25 .30 .35 .40 .45 .50 .55 .60 .65 .70 .75 .80 .90 .95 1-
18 DUMMY TEMPERATURE
19 2
20 +1000...
21
Example II

1  I BEAM EMBEDDED IN CONCRETE
2  F1.14,1.05,1.06,1.07
3  F1.05,0.01
4  F1.03,0.09
5  F1.09,0.01
6  T1.05,
7  T1.09,
8  0.15,
9  0.03,
10  0.07,
11  0.02,
12  0.08,
13  0.12,
14  0.13,
15  0.14,
16  0.15,
17  0.16,
18  0.17,
19  0.18,
20  0.19,
21  0.20,
22  0.21,
23  BETONG
24  T 7.7
25  24,127,115,130,124,117,140,117,164,82,89,85,185,150,185,
26  100,65,115,91,100,200,124,100,600,30,700,180,500,100,200
27  F 3.7
28  10,800,27,200,27
29  10,200,21,700,400,466,600,600,75,800,700,92,700,200,119,200,1200,176,600
30  F 3.7
31  10,800,27,200,27
32  10,200,21,700,400,466,600,600,75,800,700,92,700,200,119,200,1200,176,600
33  STAL
34  10,800,27,200,27
35  10,200,21,700,400,466,600,600,75,800,700,92,700,200,119,200,1200,176,600
36  10,800,27,200,27
37  10,200,21,700,400,466,600,600,75,800,700,92,700,200,119,200,1200,176,600
38  10,800,27,200,27
39  10,200,21,700,400,466,600,600,75,800,700,92,700,200,119,200,1200,176,600
40  1
41  1
42  17,25,31,31
43  2
44  2
45  34,62,50,50
46  3,6,6,2,2,2,2,2,2,2,2,2,2,2
47  3,6,6,2,2,2,2,2,2,2,2,2,2,2
48  39,47,55,55
49  4
50  T 1
51  T 2
52  T 3
53  T 4
54  0
55  NOVOID
56  15,1,3,4,5,6,7,8,9,1,0,1,1,2,1,3,1,4,1,5
57  HE1000FI
58  9
59  25,0,05,045,0,16,0,3,725,0,6,8,0,900,0,1,25,0,600,0,1,4,75,0,1,500,0,34,000,0
Example III

| 1 | BOX GIRDER EMBEDDED IN CONCRETE |
| 2 | P: 14,15.11,12.5,6 |
| 3 | P: 0.075,0.04 |
| 4 | P: 0.345,0.04,0.0875,0.112 |
| 5 | P: 0.112,0.0375,0.12 |
| 6 | T: 0.04,0.0345,0.112 |
| 7 | 0.05,0.07,0.11 |
| 8 | 0.015,0.029,0.044,0.056,0.085,0.13 |
| 9 | 14 |
| 10 | 1 2,6,7,8,9 |
| 11 | 12 13 14 15 |
| 12 | 23 24 34 35 36 |
| 13 | 45 46 47 |
| 14 | 56 57 58 59 |
| 15 | 68 69 70 71 |
| 16 | 72 73 74 75 |
| 17 | 76 77 78 79 |
| 18 | 80 81 82 83 |
| 19 | 84 85 86 87 |
| 20 | 88 89 90 91 |
| 21 | 92 93 94 95 |
| 22 | 96 97 98 99 |
| 23 | 100 101 102 103 |
| 24 | BETONG |
| 25 | T: 7,7,7 |
| 26 | 24.5,1.75,1.15,1.28,1.24,1.17,1.4,1.01,1.17,1.43,1.92,0.95,0.85,1.94,0.89, |
| 27 | 1.01,5660,1.15,91700,2.00,129400,2.00,397200,1.00,66970,1.50,1.10,0.00, |
| 28 | UTAL |
| 29 | P: 37, |
| 30 | 60,600,27,2000,27 |
| 31 | r: 200,2.1700,4.00,4.6000,6.00,758000,7.00,927000,8.00,1192000,9.00,1766000 |
| 32 | STAL |
| 33 | P: 37, |
| 34 | 60,600,27,2000,27 |
| 35 | r: 200,2.1700,4.00,4.6000,6.00,758000,7.00,927000,8.00,1192000,9.00,1766000 |
| 36 | STAL |
| 37 | P: 37, |
| 38 | 60,600,27,2000,27 |
| 39 | r: 200,2.1700,4.00,4.6000,6.00,758000,7.00,927000,8.00,1192000,9.00,1766000 |
| 40 | 25,22,0001, |
| 41 | 1,7,6,99,1.33 |
| 42 | 1.15,13,34,8,5,56,67, |
| 43 | 3,4,6,89, |
| 44 | 67,78,69 |
| 45 | 3,9,0,2,20,1.25 |
| 46 | 11,22,33,54,55,66,77,88,99 |
| 47 | 4,12,6,1,6,1, |
| 48 | 2,13,24,25,26,27,28,29,30,19,8 |
| 49 | 1,7, |
| 50 | 1,7, |
| 51 | 1,7, |
| 52 | 1,7, |
| 53 | P: 3 |
| 54 | 0, VOID |
| 55 | 0, PTONDA |
| 56 | 0, |
| 57 | 0, |
| 58 | 0, |
| 59 | 0, |
| 60 | 0, |
| 61 | 12 |
| 62 | 25,0,5,225,2,765,3,900,5,925,7,980,1,1015,1,105,6,7,1,1,600,1,2,500, |
| 63 | 1.50,1,350, |