LUND UNIVERSITY

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

Wickström, Ulf

1979

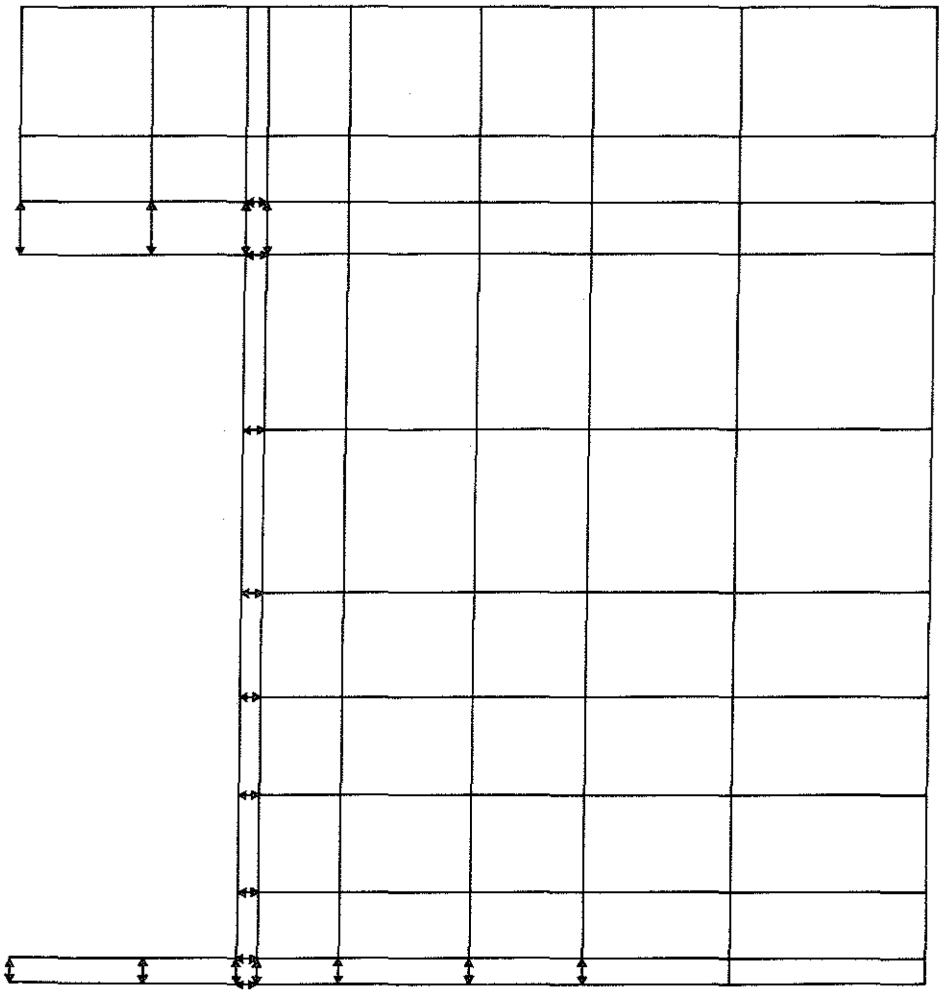
Citation for published version (APA):
Wickström, U. (1979). *TASEF-2 : A Computer Program for Temperature Analysis of Structures Exposed to Fire*. (Report / Lund Institute of Technology, Lund, Sweden, Department of Structural Mechanics; Vol. 79-2). Lund Institute of Technology.

Total number of authors:
1

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

LUND INSTITUTE OF TECHNOLOGY · LUND · SWEDEN
DEPARTMENT OF STRUCTURAL MECHANICS
REPORT NO. 79 - 2



ULF WICKSTRÖM

TASEF-2 – A COMPUTER PROGRAM FOR TEMPERATURE ANALYSIS OF STRUCTURES EXPOSED TO FIRE

LUND INSTITUTE OF TECHNOLOGY   LUND   SWEDEN
DEPARTMENT OF STRUCTURAL MECHANICS

REPORT No. 79-2

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

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 employd 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

$$\underline{\nabla}\ \underline{q} + \dot{e} - Q = 0 \tag{2.1}$$

and the Fourier law

$$\underline{q} = -\underline{k}\ \underline{\nabla}\ T \tag{2.2}$$

where $\underline{q}$ 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, $\underline{k}$ a symmetric positive definite thermal conductivity matrix, T temperature, and t time. For isotropic materials

$$\underline{k} = k\ \underline{I} \tag{2.3}$$

where k is thermal conductivity, and $\underline{I}$ the identity matrix. The gradient operator $\underline{\nabla}$ is defined as

$$\underline{\nabla} = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \tag{2.4}$$

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

$$-\underline{\nabla}^T(\underline{k}\ \underline{\nabla}\ T) + \dot{e} - Q = 0 \tag{2.5}$$

Specific volumetric enthalpy is by definition

$$e = \int_{T_o}^{T} c\rho dT + \sum_i \ell_i \qquad (2.6)$$

where T_o is a reference temperature, usually zero, c specific heat, ρ density, and ℓ_i latent volumetric heat due to phase changes at various temperature levels. The time derivative of

$$\dot{e} = c\rho\dot{T} \qquad (2.7)$$

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

$$-\underline{\nabla}^T(\underline{k}\ \underline{\nabla}\ T) + c\rho\dot{T} - Q = 0 \qquad (2.8)$$

Nominal specific volumetric heat $\overline{c\rho}$ will be defined by the equation

$$e = \overline{c\rho}T \qquad (2.9)$$

In Figure 2.1 specific volumetric enthalpy is plotted versus temperature for a material with latent heat indicated by a step ℓ in the curve. The tangential and secantial or nominal volumetric specific heats, $c\rho$ and $\overline{c\rho}$, respectively, are then as shown in Figure 2.1. Note that at the temperature T_ℓ, where the enthalpy curve is stepped, the value of $c\rho$ is undefined while the value of $\overline{c\rho}$ 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-

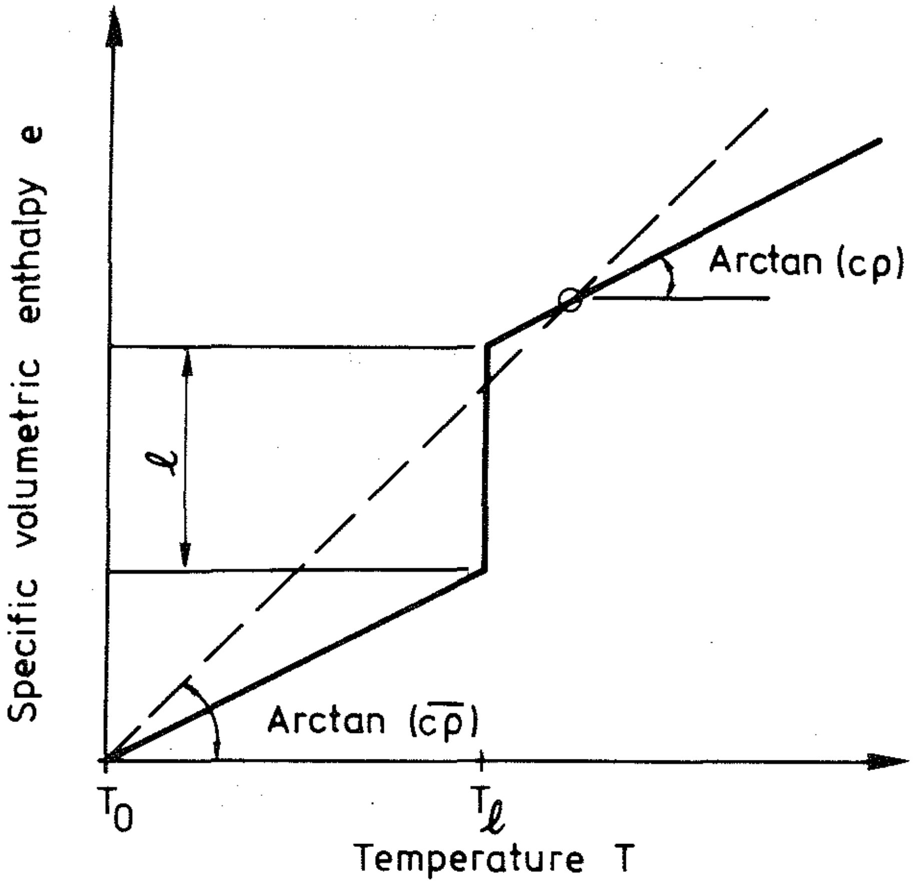


Figure 2.1. Definitions of specific volumetric heat

ditions are prescribed as temperature or heat flow on parts of the boundary ∂V_T and ∂V_q, respectively. The total boundary is then defined by

$$\partial V = \partial V_T + \partial V_q \tag{2.10}$$

Temperature on the boundary ∂V_T of a body is specified as

$$T = T(x, y, z, t) \tag{2.11}$$

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

$$q_n = \underline{n}^T \underline{q} = -\underline{n}^T \underline{k} \underline{\nabla} T \tag{2.12}$$

where $\underline{n}$ is the outward normal to the surface. Specified heat flow on ∂V_q therefore is

$$\hat{q}_n = -\underline{n}^T \underline{k} \underline{\nabla} T \tag{2.13}$$

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

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

where $\hat{q}_n^c$ is the rate of heat transferred by convection, β and γ 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

$$\hat{q}_n^r = \varepsilon_r \sigma(\bar{T}_s^4 - \bar{T}_g^4) \tag{2.15}$$

where σ is the Stefan-Boltzmann constant, and $\bar{T}_s$ and $\bar{T}_g$ are absolute surface temperature and absolute surrounding gas temperature, respectively. Resultant emissivity ε_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 ε_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,

$$\varepsilon_r = \frac{1}{1/\varepsilon_s + 1/\varepsilon_g - 1} \tag{2.16}$$

where ε_g is appropriate gas or flame emissivity.

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

$$\hat{q}_n = \hat{q}_n^c + \hat{q}_n^r \tag{2.17}$$

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

$$T = \sum_i N_i(x,y,z)\ T_i(t) = \underline{N}\ \underline{T} \qquad (3.1)$$

The time differentiation of the temperature is

$$\dot{T} = \underline{N}\ \dot{\underline{T}} \qquad (3.2)$$

Each shape function N_i 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_i 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]:

$$\int_V v(-\underline{\nabla}^T \underline{k}\, \underline{\nabla}\, T + \frac{\partial}{\partial t}(\overline{c\rho} T) - Q)\, dV = 0 \qquad (3.3)$$

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

$$\int_V v(-\underline{\nabla}^T \underline{k}\, \underline{\nabla}\, T)\, dV = -\int_{\partial V} v\, \underline{n}^T \underline{k}\, \underline{\nabla}\, T\, dS + \\ + \int_V (\underline{\nabla}\, v)^T \underline{k}\, \underline{\nabla}\, T\, dV \qquad (3.4)$$

where $\underline{n}$ is the outward normal to the boundary ∂V. A set of weighting functions v_i equal to the shape functions N_i (the Galerkin method) is then chosen, i.e.

$$v_i = N_i \qquad (3.5)$$

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

$$\int_V \left[(\underline{\nabla}\, \underline{N})^T \underline{k}\, \underline{\nabla}\, \underline{N}\, dV \right] \underline{T} + \frac{\partial}{\partial t} \left[\int_V \underline{N}^T \overline{c\rho} \underline{N}\, dV\, \underline{T} \right] = \\ = \int_V \underline{N}^T Q\, dV + \int_{\partial V} \underline{N}^T \underline{n}^T \underline{k}\, \underline{\nabla}\, T\, dS \qquad (3.6)$$

or

$$\underline{F}_T + \frac{\partial}{\partial t}(\underline{E}) = \underline{F}_Q + \underline{F}_q \tag{3.7}$$

where $\underline{F}_T$, $\underline{E}$, $\underline{F}_Q$ and $\underline{F}_q$ 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

$$\underline{F}_T = \underline{K}\ \underline{T} \tag{3.8}$$

where $\underline{K}$ is the heat conductivity matrix.

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

$$\underline{K}\ \underline{T} + \frac{\partial}{\partial t}(\underline{E}) = \underline{F} \tag{3.9}$$

where

$$\underline{F} = \underline{F}_Q + \underline{F}_q \tag{3.10}$$

The nodal enthalpy vector is

$$\underline{E} = \underline{C}\ \underline{T} \tag{3.11}$$

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

$$\underline{K}\ \underline{T} + \frac{\partial}{\partial t}(\underline{C}\ \underline{T}) = \underline{F} \tag{3.12}$$

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

$$\underline{K}^*\underline{E} + \frac{\partial}{\partial t}(\underline{E}) = \underline{F} \tag{3.13}$$

where

$$\underline{K}^* = \underline{K}\ \underline{C}^{-1} \tag{3.14}$$

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

$$\underline{K} = \sum_m \underline{K}^m \tag{3.15}$$

$$\underline{C} = \sum_m \underline{C}^m \tag{3.16}$$

$$\underline{F} = \sum_m \underline{F}_Q^m + \sum_{\partial m} \underline{F}_q^{\partial m} \tag{3.17}$$

where

$$K_{ij}^m = \int_{V^m} (\underline{\nabla}\ N_i)^T\ \underline{k}(\underline{\nabla}\ N_j)dV \tag{3.18}$$

$$C_{ij}^m = \int_{V^m} N_i\ \overline{c\rho}\ N_j\ dV \tag{3.19}$$

$$F_{Qi}^m = \int_{V^m} N_i\ Q\ dV \tag{3.20}$$

and

$$F_{qi}^{\partial m} = \int_{\partial V^m} N_i\ \underline{n}^T\ \underline{k}\ \underline{\nabla}\ T\ dS \tag{3.21}$$

V^m and ∂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 $\underline{K}^m$ for the simple two-dimensional rectangular element used in

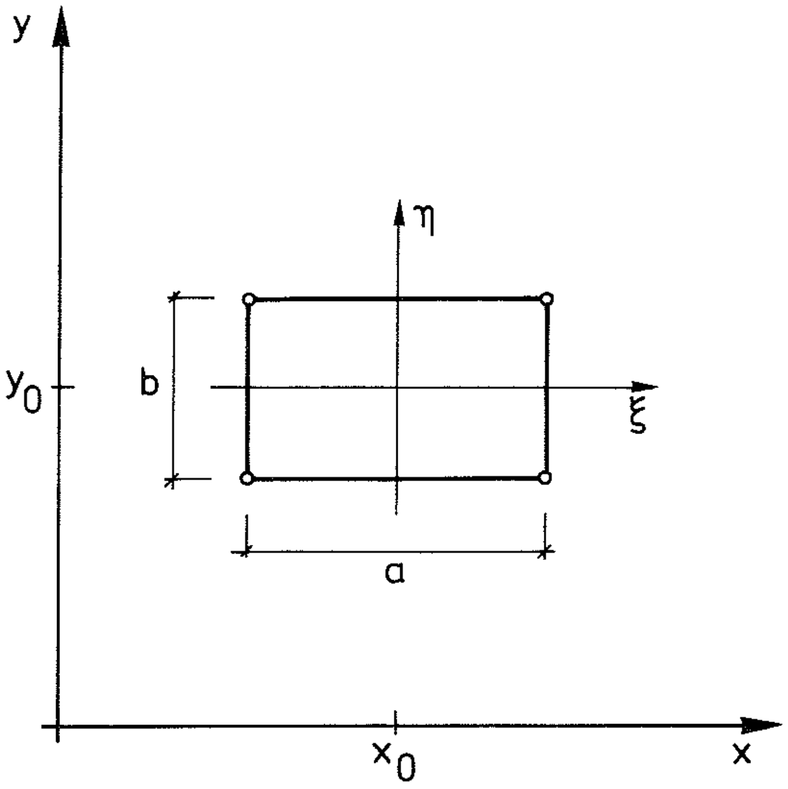


Figure 3.1. Rectangular finite element

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

$$\xi = (x - x_o)/a \tag{3.22}$$

and

$$\eta = (y - y_o)/b \tag{3.23}$$

where ξ and η are dimensionless coordinates in a local system. A set of allowable shape functions is then

$$N_i = (1 + \xi\xi_i)(1 + \eta\eta_i)/4 \tag{3.24}$$

where i takes values from 1 to 4, and

$$\underline{\nabla}\, N_i = \begin{bmatrix} \dfrac{\partial N_i}{\partial x} \\[2ex] \dfrac{\partial N_i}{\partial y} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{a}\,\dfrac{\partial N_i}{\partial \xi} \\[2ex] \dfrac{1}{b}\,\dfrac{\partial N_i}{\partial \eta} \end{bmatrix} = 1/4 \begin{bmatrix} \dfrac{\xi_i}{a}(1+\eta\eta_i) \\[2ex] \dfrac{\eta_i}{b}(1+\xi\xi_i) \end{bmatrix} \qquad (3.25)$$

Equation (3.25) is substituted into Equation (3.18) and constant thickness d and conductivity k are assumed for the element. Thus the local element conductivity matrix $\underline{K}^m$ is, after evaluation of simple integrals,

$$\underline{K}^m = \frac{1}{3}\,\frac{kd}{ab} \begin{bmatrix} a^2+b^2 & -b^2+\dfrac{a^2}{2} & -\dfrac{b^2}{2}-\dfrac{a^2}{2} & \dfrac{b^2}{2}-a^2 \\[2ex] & a^2+b^2 & \dfrac{b^2}{2}-a^2 & -\dfrac{b^2}{2}-\dfrac{a^2}{2} \\[2ex] & & a^2+b^2 & -b^2+\dfrac{a^2}{2} \\[2ex] & \text{sym.} & & a^2+b^2 \end{bmatrix} \qquad (3.26)$$

If conductivity k for a particular application varies with temperature, k 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 $\underline{C}^m$ as given in Equation (3.19) results in a fully populated matrix identical in form to the element conductivity matrix $\underline{K}^m$. The assembled heat capacity matrix $\underline{C}$ is symmetric, positive-definite, and has the same nonzero structure as the system conductivity matrix $\underline{K}$. The element heat capacity matrix $\underline{C}^m$ 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 $\underline{C}$. 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:

$$C_{ii}^m = \overline{c\rho}^m(T_i)W_{ii}^m \qquad (3.27)$$

where $\overline{c\rho}^m(T_i)$ is nominal specific volumetric heat capacity at nodal temperature T_i and W_{ii}^m 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:

$$C_{ii} = \overline{c\rho}(T_i)W_{ii} \qquad (3.28)$$

where

$$W_{ii} = \sum_m W_{ii}^m \qquad (3.29)$$

defines the global diagonal volume matrix.

3.3.3 Internally Generated Heat

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

$$F_{Qi} = \sum_m Q_i^m W_{ii}^m \qquad (3.30)$$

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 ∂V_q of the boundary where heat flow is prescribed, nodal heat flow is calculated by substituting Equation (2.13) into Equation (3.21):

$$F_{qi}^{\partial m} = -\int_{\partial V_q^m} N_i \; \hat{q}_n \; dS \qquad (3.31)$$

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

$$F_{qi}^{\partial m} = -1/6 \; sd(2\hat{q}_{ni}^{\partial m} + \hat{q}_{nj}^{\partial m}) \qquad (3.32)$$

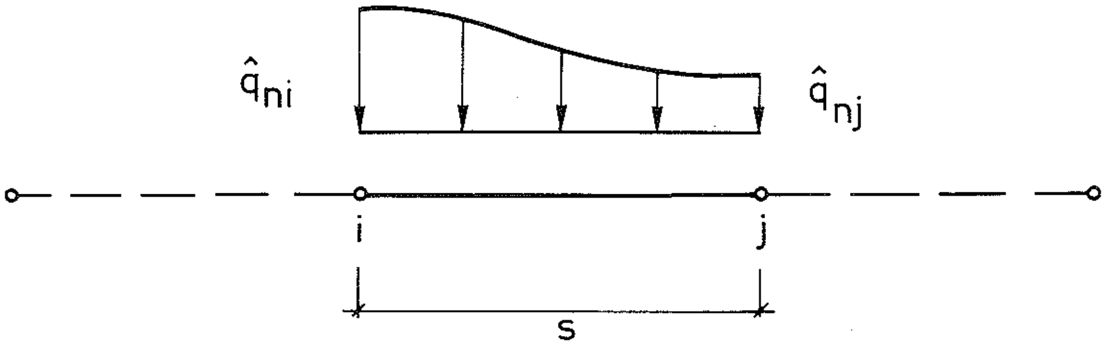


Figure 3.2. Heat flow to a boundary element

Equations (2.14 and 2.15) are then used to yield

$$\hat{q}_{ni}^{\partial m} = -\left[\varepsilon_r \sigma(\bar{T}_{gi}^4 - \bar{T}_i^4) + \beta(\bar{T}_{gi} - \bar{T}_i)^{\gamma}\right] \qquad (3.33)$$

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

$$F_{qi}^{\partial m} = B_{rii}^{\partial m} T_{ri} + B_{rij}^{\partial m} T_{rj} + B_{cii}^{\partial m} T_{ci} + B_{cij}^{\partial m} T_{cj} \qquad (3.34)$$

where i and j are nodes adjacent to a boundary element ∂m and

$$B^{\partial m}_{rii} = \frac{1}{3} sd\varepsilon_r \sigma \tag{3.35}$$

$$B^{\partial m}_{rij} = \frac{1}{2} B^{\partial m}_{rii} \tag{3.36}$$

$$B^{\partial m}_{cii} = \frac{1}{3} sd\beta \tag{3.37}$$

$$B^{\partial m}_{cij} = \frac{1}{2} B^{\partial m}_{cii} \tag{3.38}$$

$$T_{ri} = \bar{T}^4_{gi} - \bar{T}^4_i \tag{3.39}$$

and

$$T_{ci} = (\bar{T}_{gi} - \bar{T}_i)^\gamma \tag{3.40}$$

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

$$\underline{F}_q = \underline{B}_r \, \underline{T}_r + \underline{B}_c \, \underline{T}_c \tag{3.41}$$

where $\underline{T}_r$ and $\underline{T}_c$ are vectors of modified nodal temperature as defined by Equations (3.39 and 3.40), respectively, and $\underline{B}_r$ and $\underline{B}_c$ are boundary radiation and convection matrices, respectively, where

$$B_{rij} = \sum_{\partial m} B^{\partial m}_{rij} \tag{3.42}$$

and

$$B_{cij} = \sum_{\partial m} B^{\partial m}_{cij} \tag{3.43}$$

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 $\underline{B}_r$ and $\underline{B}_c$ 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 ε_r and convection factor β 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

$$\begin{aligned}&\Theta\underline{K}^*_{t+\Delta t}\underline{E}_{t+\Delta t}+(1-\Theta)\underline{K}^*_t\underline{E}_t+(\underline{E}_{t+\Delta t}-\underline{E}_t)/\Delta t = \\ &= \Theta\underline{F}_{t+\Delta t}+(1-\Theta)\underline{F}_t\end{aligned} \qquad (3.44)$$

where the indices indicates time, and where Θ is an arbitrary parameter in the range

$$0 \leq \Theta \leq 1 \qquad (3.45)$$

If different values are assigned to Θ 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 Δt used solutions will not diverge, the forward-difference method will converge only if the time-increment Δt is less than a critical value Δt_{cr}. The value of this critical time increment depends on element size, material properties, and boundary conditions. If $\underline{C}$ is a diagonal (lumped) matrix the solution for $\underline{T}_{t+\Delta t}$ 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

$$\underline{E}_{t+\Delta t} = \underline{E}_t + (\underline{F}_t - \underline{K}^*_t \, \underline{E}_t)\Delta t \qquad (3.46)$$

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

$$\underline{E}_{t+\Delta t} = \underline{E}_t + (\underline{F}_t - \underline{K}_t \underline{T}_t)\Delta t \qquad (3.47)$$

Equation (3.11) is then used to obtain temperature at a node i:

$$T_{i,t+\Delta t} = C_{ii}^{-1}(T_{i,t+\Delta t})E_{i,t+\Delta t} \tag{3.48}$$

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

$$e_{i,t+\Delta t} = E_{i,t+\Delta t}/W_{ii} \tag{3.49}$$

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 when ever possible in TASEF-2. For nodes at interfaces between elements of different materials, the following iteration formula is used to calculate temperature

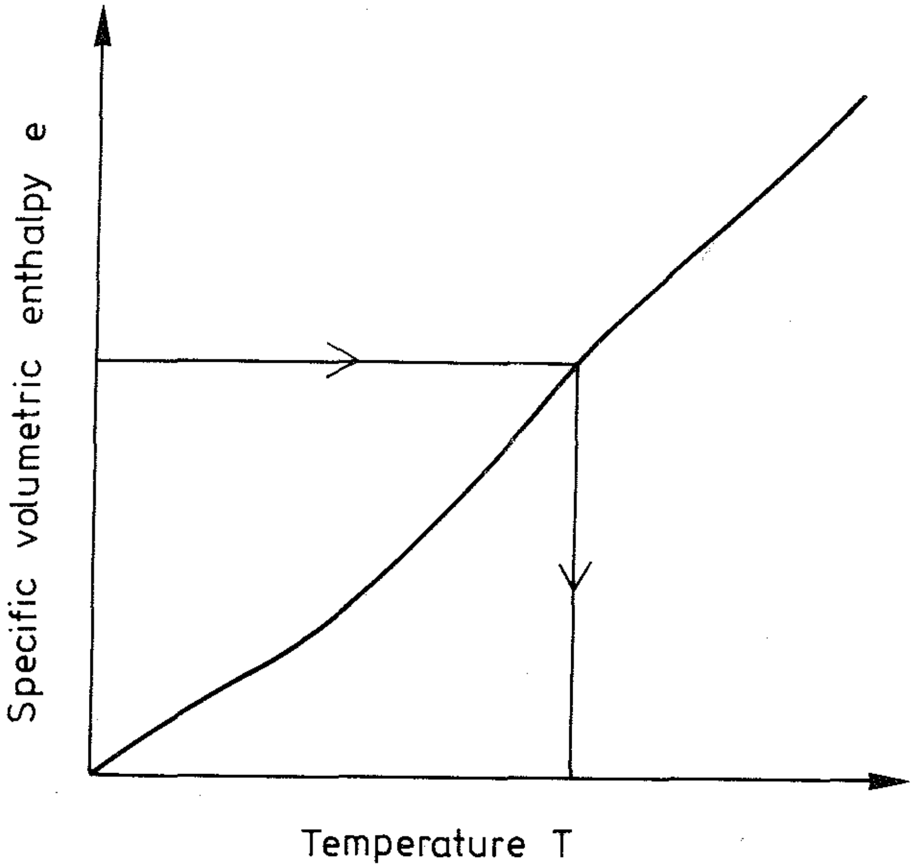


Figure 3.3. Translation of specific volumetric enthalpy into temperature

$$T^{j+1}_{i,t+\Delta t} = C^{-1}_{ii}(T^{j}_{i,t+\Delta t})E_{i,t+\Delta t} \tag{3.50}$$

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 δ expressed as

$$\frac{T^{j+1}_{i,t+\Delta t} - T^{j}_{i,t+\Delta t}}{T^{j+1}_{i,t+\Delta t} + T^{j}_{i,t+\Delta t}} < \delta/2 \tag{3.51}$$

δ 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 Δt_{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 $\underline{C}$ is assumed to be time independent, Equation (3.12) is

$$\underline{C} \frac{\partial}{\partial t} \underline{T} + \underline{K}\,\underline{T} = \underline{F} \tag{3.52}$$

At any time step, the righthand side of Equation (3.51) can be linearized at current temperature; thus, matrix $\underline{K}_F$ is defined by

$$K_{Fij} = \frac{dF_i}{dT_j} \tag{3.53}$$

where i and j denote rows and columns. Thus the homogeneous part of Equation (3.52) is

$$\underline{C} \frac{\partial}{\partial t} \underline{T} + \underline{\bar{K}}\,\underline{T} = \underline{0} \tag{3.54}$$

where

$$\underline{\bar{K}} = \underline{K} - \underline{K}_F \tag{3.55}$$

In case of homogeneous boundary conditions, solutions of Equation (3.54) have the form

$$\underline{T} = e^{-\lambda t}\underline{\phi} \qquad (3.56)$$

where $\underline{\phi}$ is a vector independent of time t. Multiply by the inverse of the diagonal matrix $\underline{C}$:

$$-\lambda\ e^{-\lambda t}\underline{\phi} + \underline{C}^{-1}\underline{\bar{K}}\ e^{-\lambda t}\underline{\phi} = 0 \qquad (3.57)$$

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

$$(\underline{C}^{-1}\underline{\bar{K}})\underline{\phi} = \lambda\ \underline{\phi} \qquad (3.58)$$

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 frequences) $\lambda_1, \lambda_2 \ldots \lambda_n$ with corresponding eigenvectors (thermal modes) $\phi_1, \phi_2 \ldots \phi_n$.

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

$$\Delta t_{cr} = \frac{2}{\lambda_{max}} \qquad (3.59)$$

where λ_{max} is the maximum eigenvalue.

Exact calculation of λ_{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

$$\lambda \leq \max_i (a_{ii} + \sum_j |a_{ij}|) \qquad j \neq i \qquad (3.60)$$

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

$$K_{ii} = -\Sigma\ K_{ij} \tag{3.61}$$

Thus the maximum eigenvalue of Equation (3.58) is

$$\lambda_{max} \leq \max_{i}\left[C_{ii}^{-1}(2K_{ii}+\sum_{j} K_{Fij})\right] \tag{3.62}$$

and an upper limit to the critical time increment is

$$\Delta t_{cr} = \min_{i}\left[\frac{C_{ii}}{K_{ii}+\frac{1}{2}\sum_{j} K_{Fij}}\right] \tag{3.63}$$

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 loosing 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

$$T_f = T_o + 345 \log_{10}(430t + 1) \qquad (4.1a)$$

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:

$$\begin{array}{lll} 625 \ ^oC/h & \text{if } t_u \leq 0.5 & h \\ 250(3-t_u) \ ^oC/h & \text{if } 0.5 \leq t_u \leq 2 & h \\ 250 \ ^oC/h & \text{if } t_u > 2 & h \end{array} \qquad (4.1b)$$

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 occurence 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 2ℓ as shown in Figure 5.1a initially at uniform temperature T_o 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 ρ, and specific heat capacity c are assumed constant. The following dimensionless parameters are introduced for convenience:

$$\Theta = (T - T_g)/(T_o - T_g)$$

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

$$Bi = h\ell/k$$

where the thermal diffusivity a is

$$a = k/c\rho$$

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 $0.0667\ a/\ell^2$ and $0.0200\ a/\ell^2$ 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

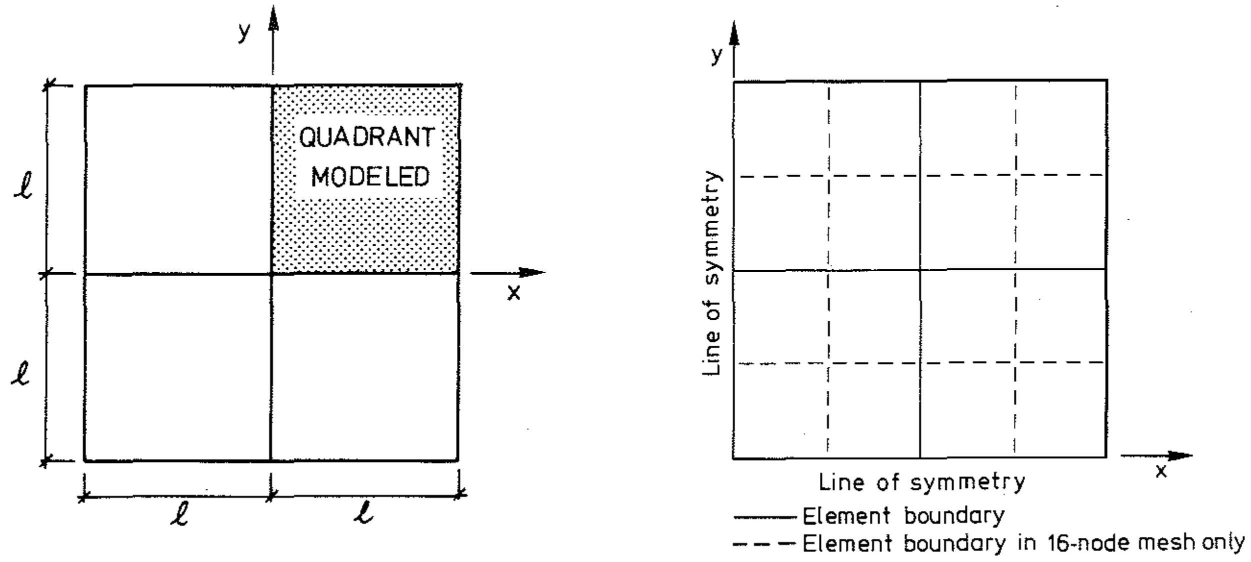
5.2.1 Material Properties and Boundary Conditions

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°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 Θ at center of square plate exposed to heat transfer from surrounding gas, Bi=1

Dimensionless time Fo	Exact solution	Coarse mesh 4 elements $\Delta t=0.0667\ at/\ell^2$		Fine mesh 16 elements $\Delta t=0.0200\ at/\ell^2$	
		Numerical solution	Error	Numerical solution	Error
0.1	0.9864			0.993	-0.007
0.2	0.9038	0.925	-0.021	0.909	-0.005
0.4	0.6902	0.688	0.002	0.690	-
0.6	0.5147	0.505	0.009	0.512	0.003
0.8	0.3827	0.370	0.012	0.379	0.004
1.0	0.2845	0.271	0.013	0.281	0.004
Number of time steps		15		50	



(a) Square plate

(b) Finite element meshes for square quadrant

Figure 5.1. Finite element model for calculating heat transfer in a square plate

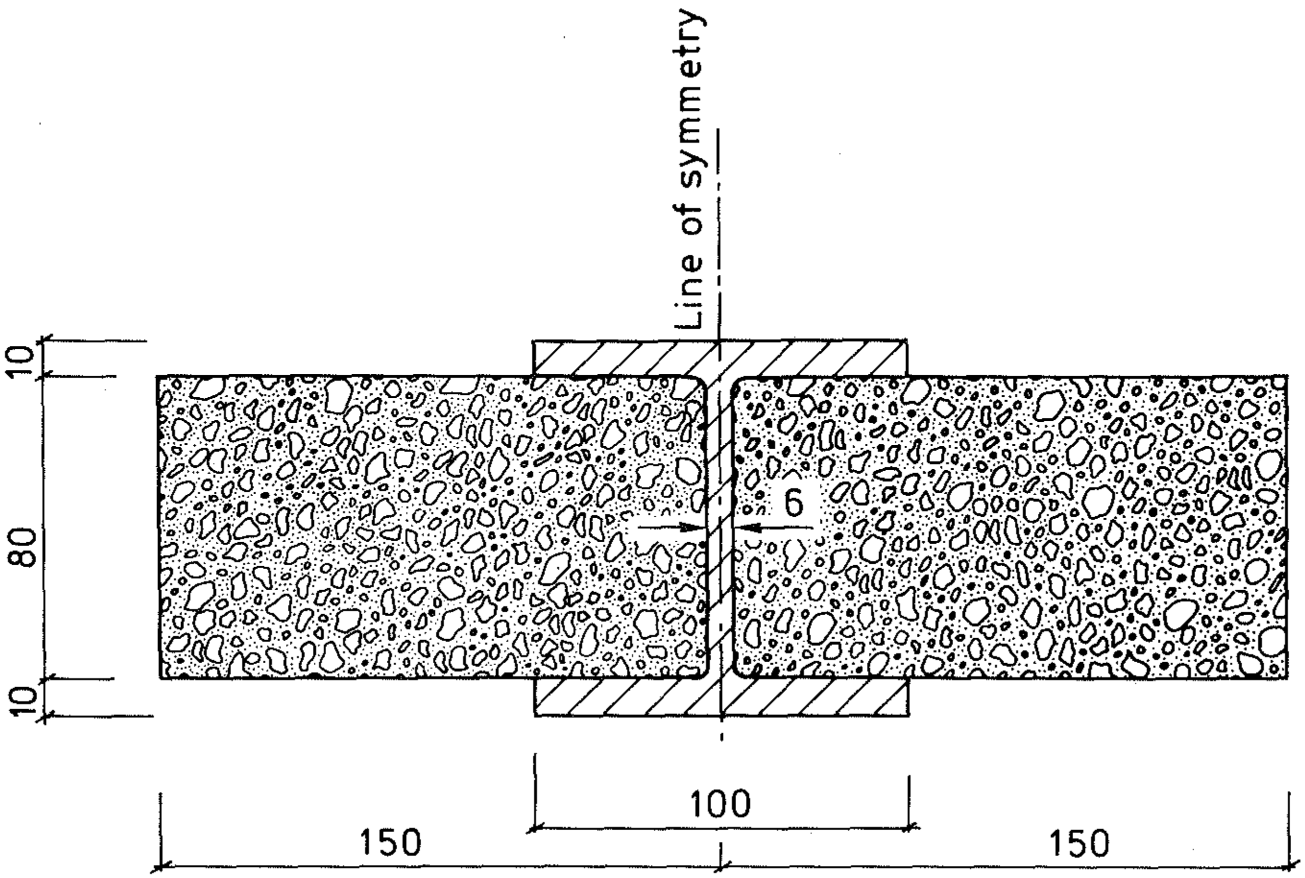


Figure 5.2. Wide-flange I-beam (HE100B) embedded in a concrete slab. The vertical sides were insolated during furnace test. Dimensions in mm

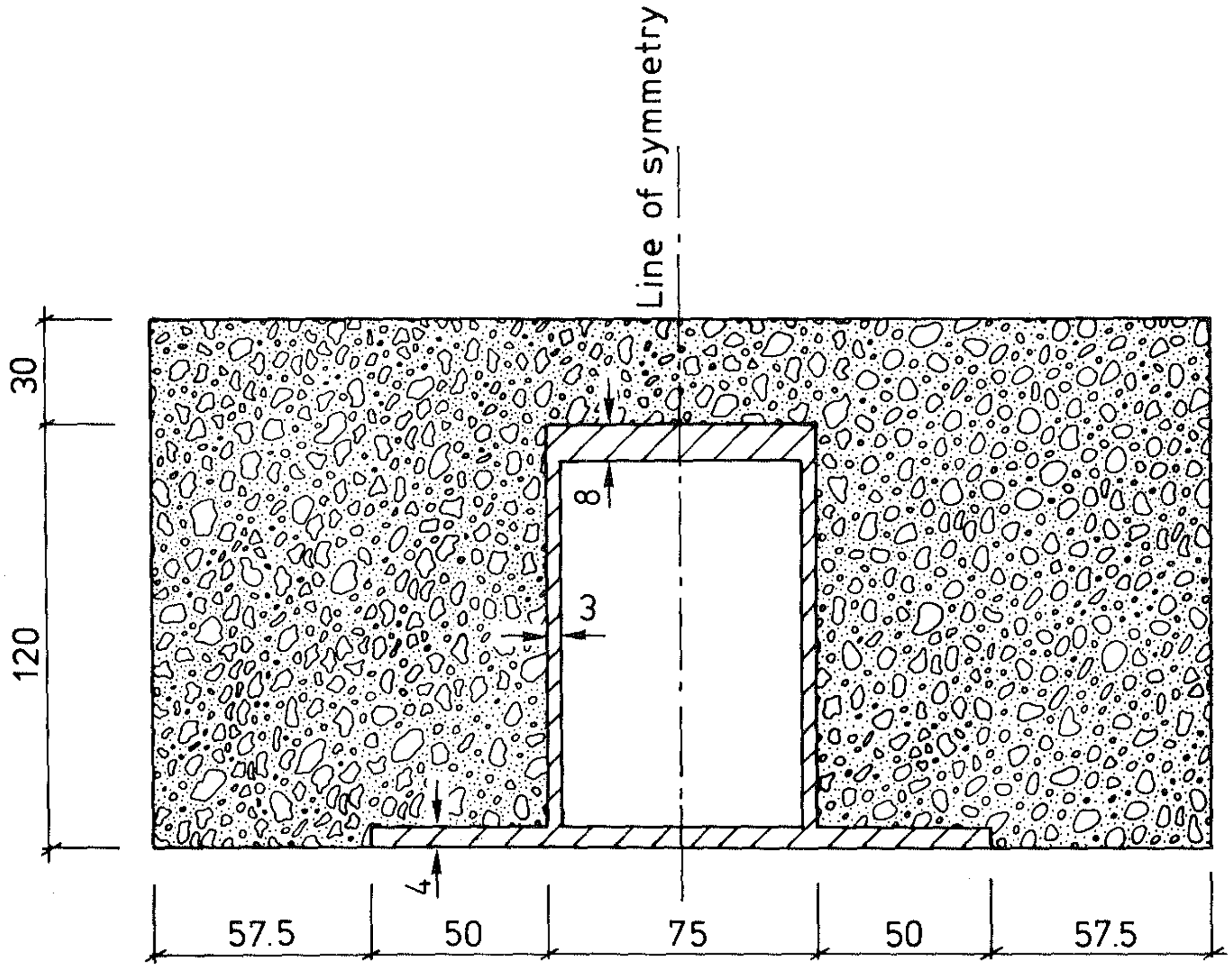


Figure 5.3. Box girder embedded in a concrete slab. The vertical concrete sides were insolated during furnace test. Dimensions in mm

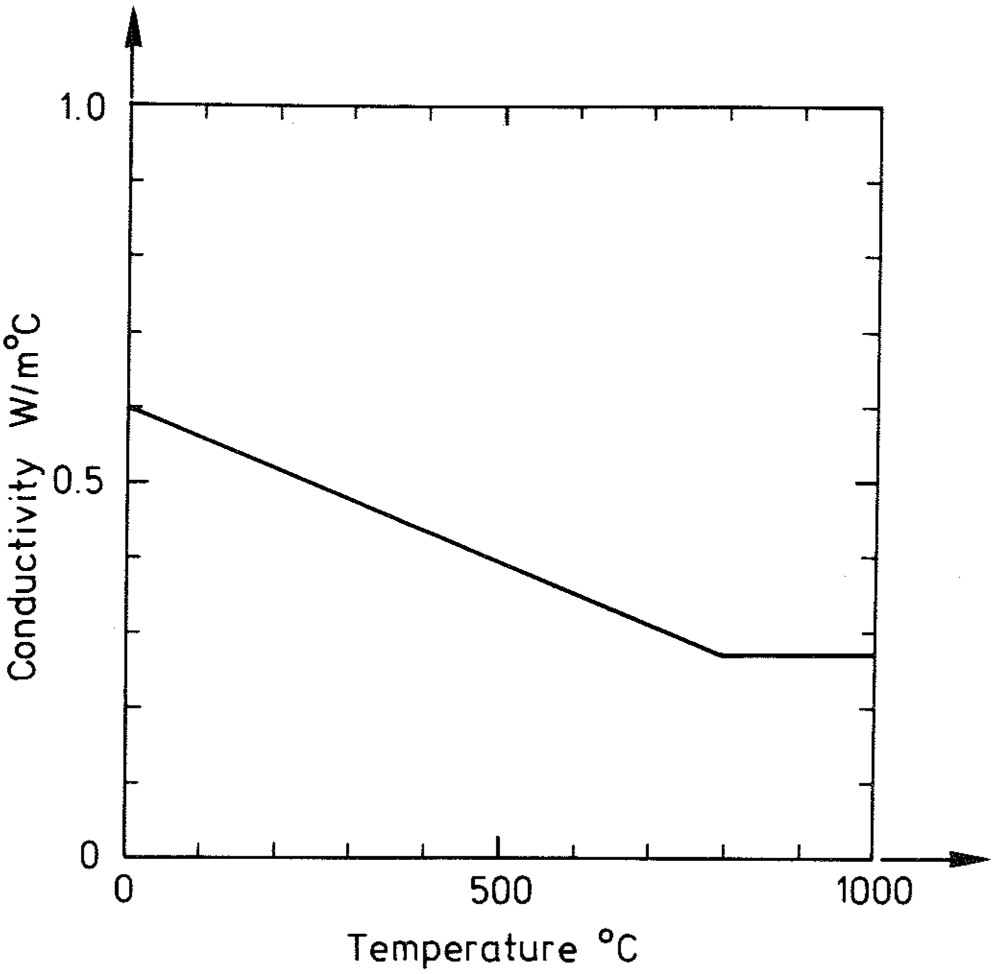


Figure 5.4. Conductivity of steel

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

$$e(T) = e^{c}(T) + u\, e^{w}(T)$$

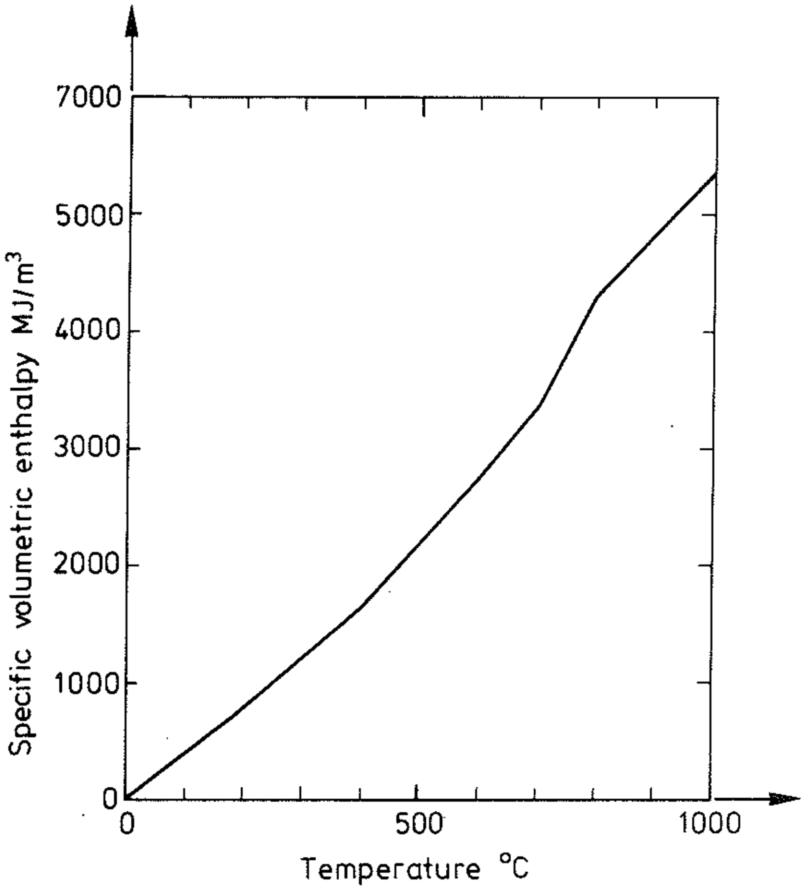


Figure 5.5. Specific volumetric enthalpy of steel

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:

$$e^w(T) = c^w T \qquad \text{for } T < T_1$$

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

$$e^w(T) = e^w(T_2) \qquad \text{for } T > T_2$$

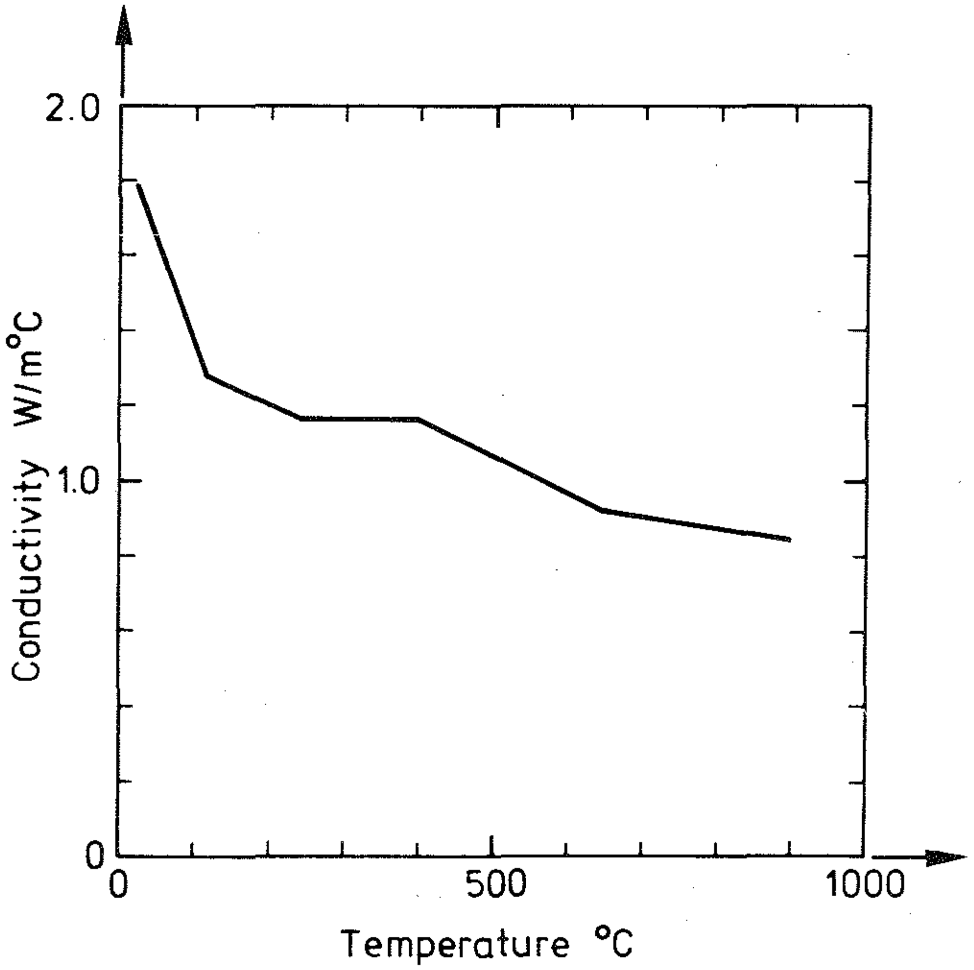


Figure 5.6. Thermal conductivity of concrete

where c_w^w and a_w^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 ρ^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.

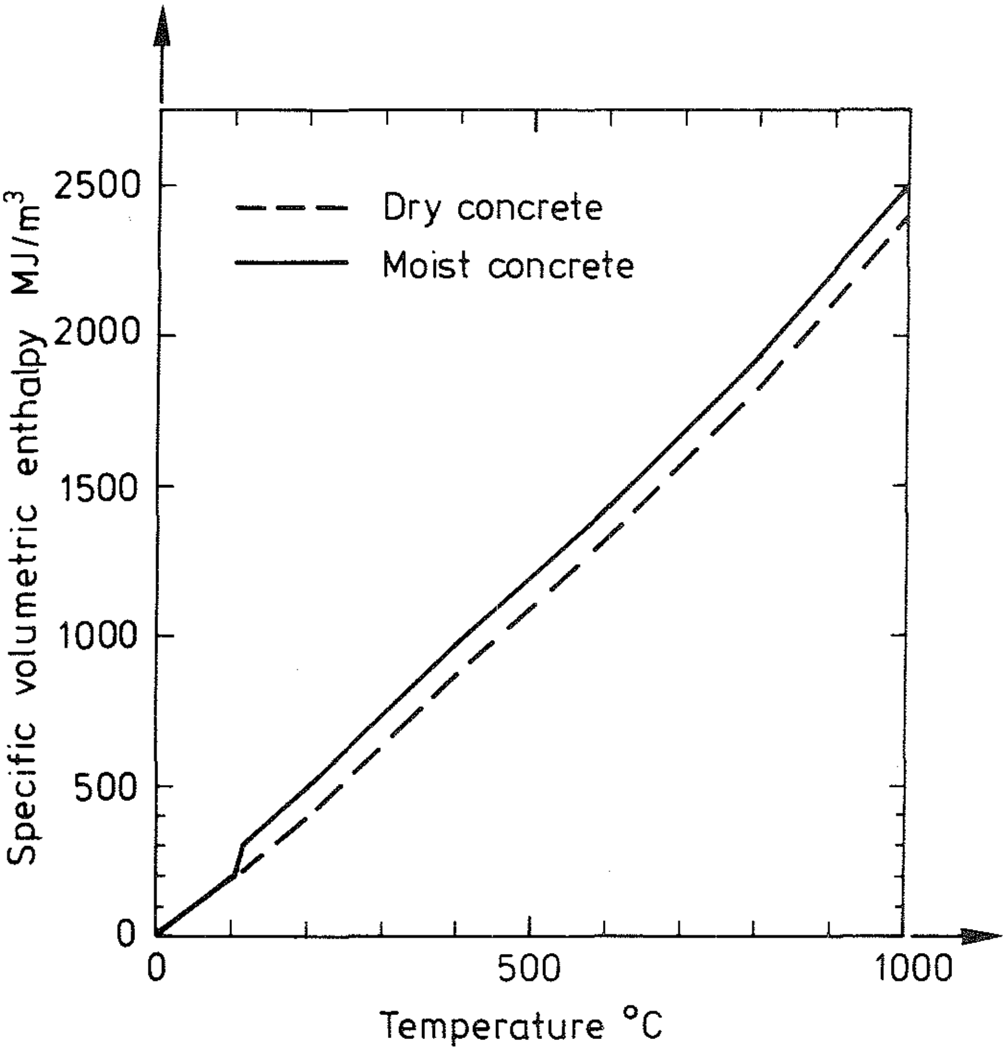


Figure 5.7. Specific volumetric enthalpy of dry concrete and concrete containing 1.5% water by weight

The cool side surfaces of the specimens were small in comparison with surrounding surfaces at ambient temperature. The resultant emissivity ε_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 β and powers γ as defined by Equation (2.14), free convection was assumed and the formula [19]

$$Nu = C(Pr\ Gr)^m \qquad (a)$$

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

$$Nu = \frac{q^c d}{k(T_s - T_g)} \qquad (b)$$

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

$$Gr = \frac{g \frac{1}{T_b} (T_s - T_g) d^3}{\nu^2} \qquad (c)$$

where g is the acceleration of gravity, $T_b = (T_s + T_g)/2$ the average absolute boundary layer temperature, and ν 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]:

$$k = 13.75 \cdot 10^{-5}\ T_b^{0.92} \quad [W/mK] \qquad (d)$$

$$\nu = 1.13 \cdot 10^{-9}\ T_b^{5/3} \quad [m^2/s] \qquad (e)$$

From Equations (a-e) the formula (SI-units)

$$q^c = 13.75\cdot10^{-5}(5.48\cdot10^{18})^m C \frac{d^{3m-1}}{T_b^{(\frac{13}{3}m-0.92)}}(T_s - T_g)^{m+1} \quad (f)$$

is derived.

The characteristic length d of the beams considered is approximately 0.15 m. Substitution of Equation (e) into Equation (c) then gives $Gr < 8\cdot10^6$ 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

$$q^c = 3.59\ d^{-0.25}\ T_b^{-0.16}(T_s - T_g)^{1.25} \qquad [W/m^2]$$

and by inserting d = 0.15 m and assuming T_b = 400 K, the convection factor β and power γ are identified as 2.2 $W/m^2K^{1.25}$ 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

$$q^c = 36\ T_b^{-0.52}\ (T_s - T_g)^{1.33} \qquad [W/m^2]$$

and if T_b is assumed to be 1000K, β and γ are identified as 1.0 $W/m^2K^{1.33}$ 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

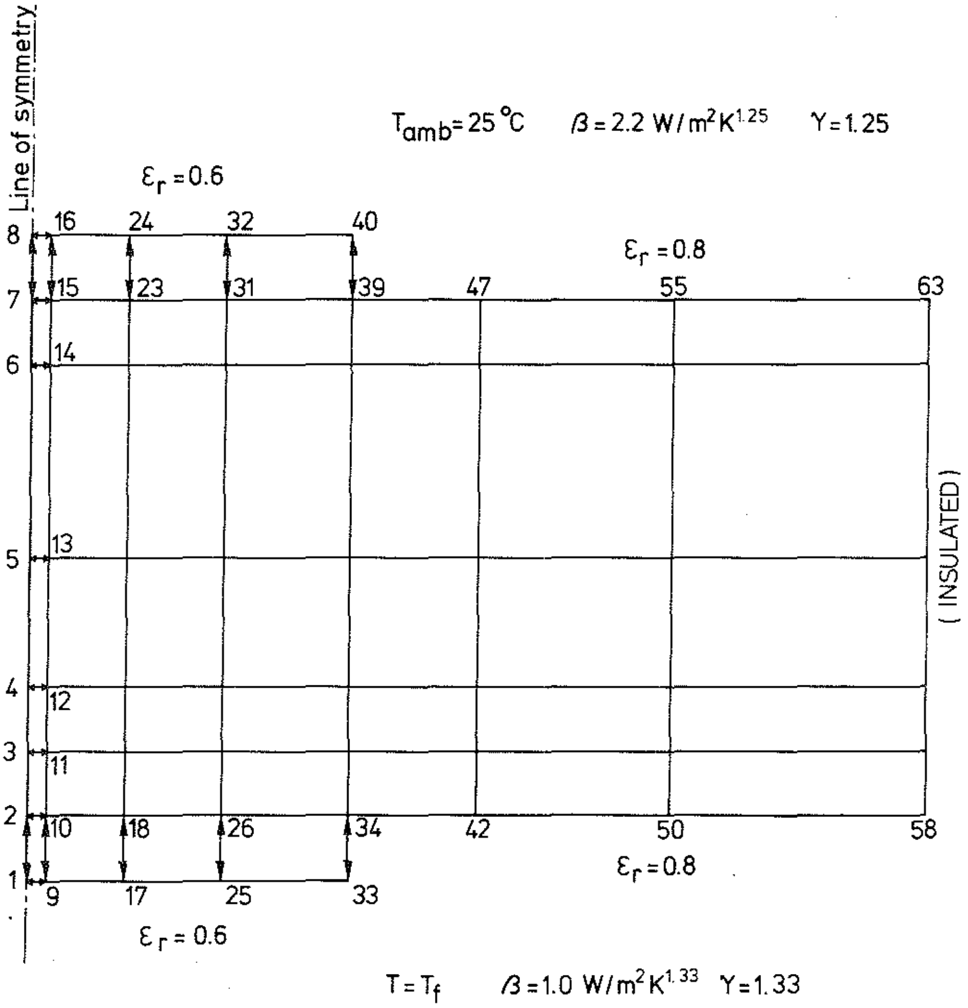


Figure 5.8. Finite element mesh of I-beam embedded in concrete

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 centerline 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 5.3)

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

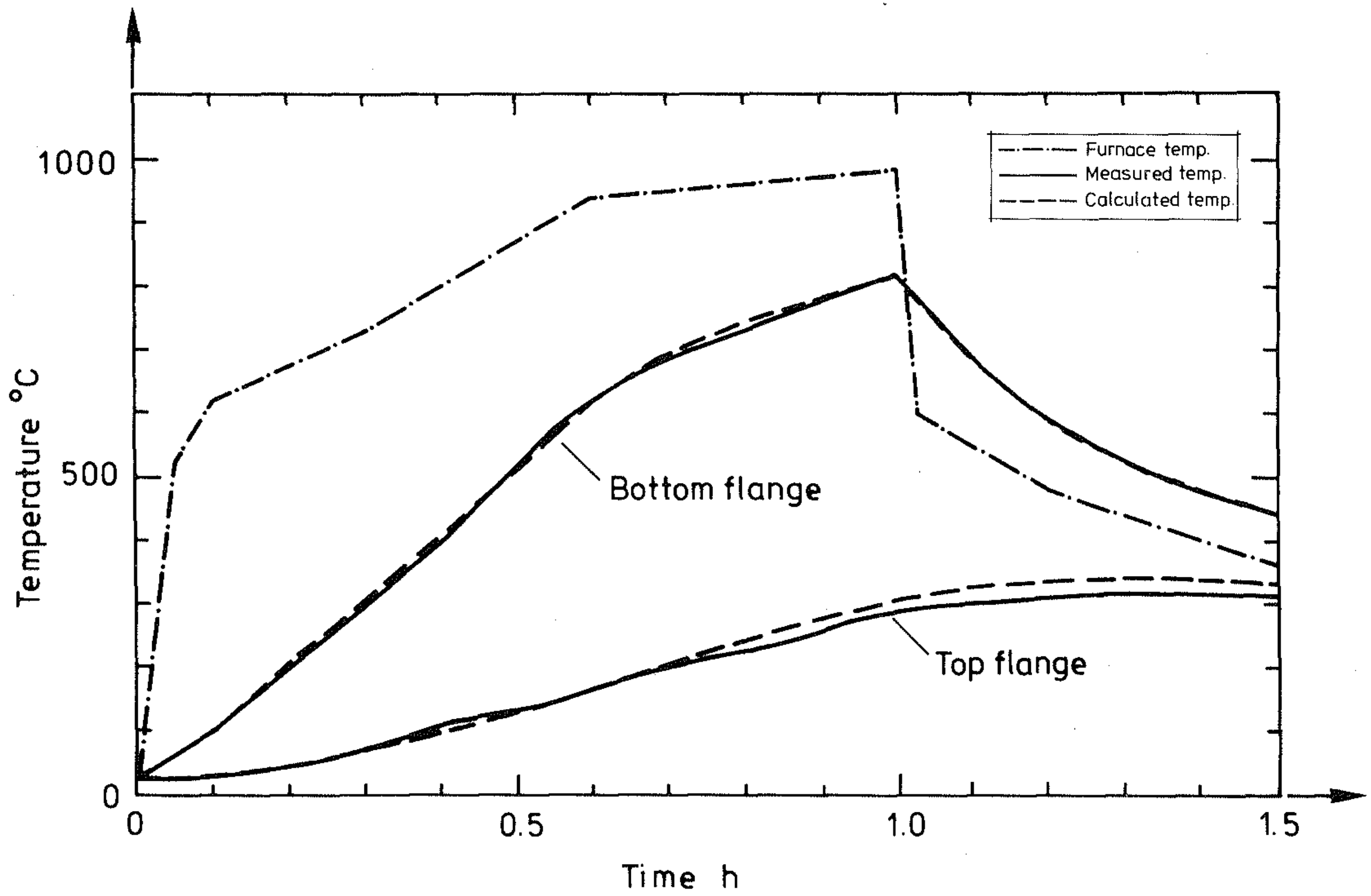


Figure 5.9. Assumed furnace gas temperature in Example II and measured and calculated temperature histories of top and bottom flanges at the centerline

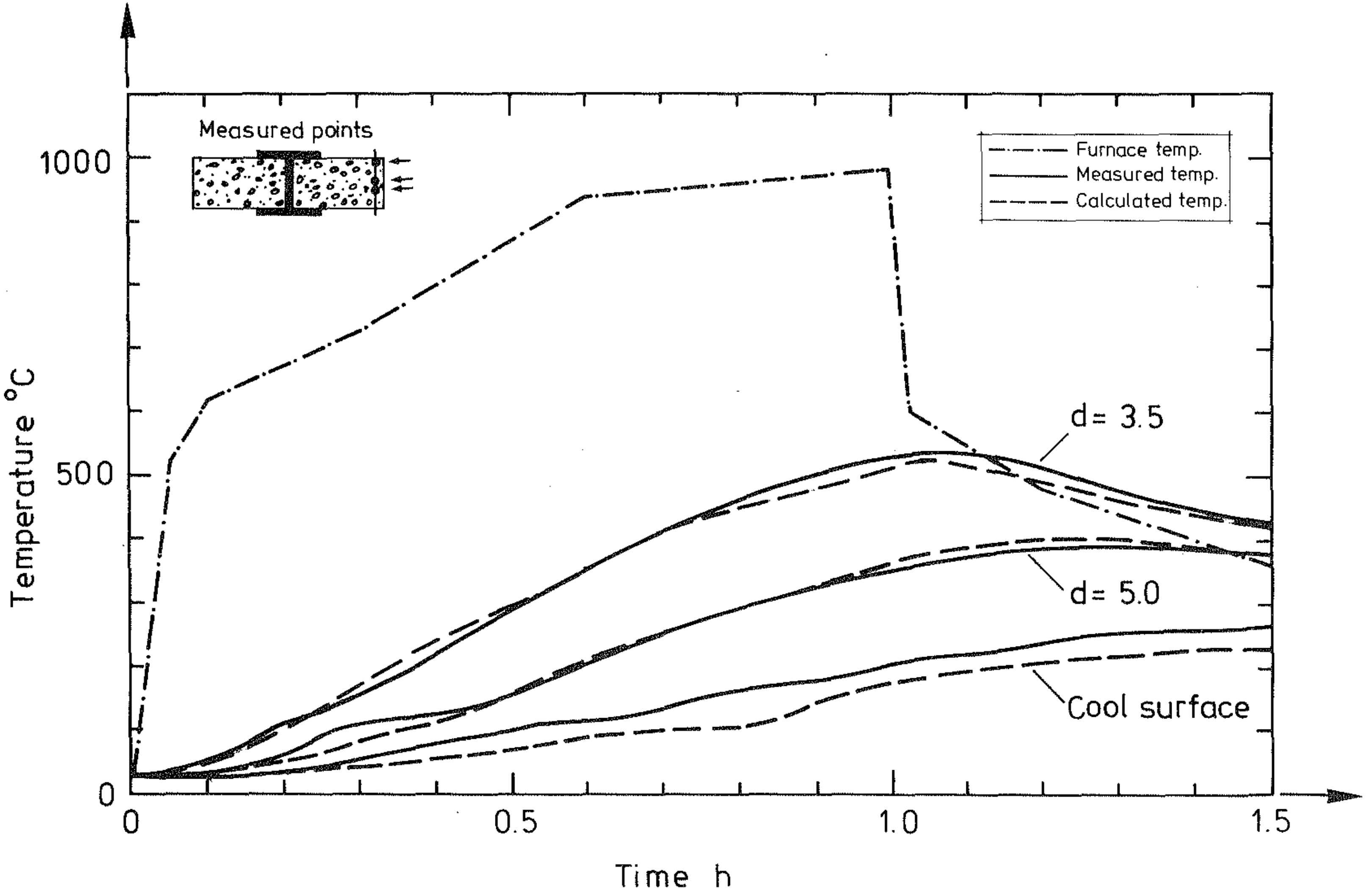


Figure 5.10. Calculated and measured concrete temperature at 140 mm from the centerline of steel beam and at distances d equal to 3.5 and 5.0 cm from fire-exposed surface and on cool surface, respectively

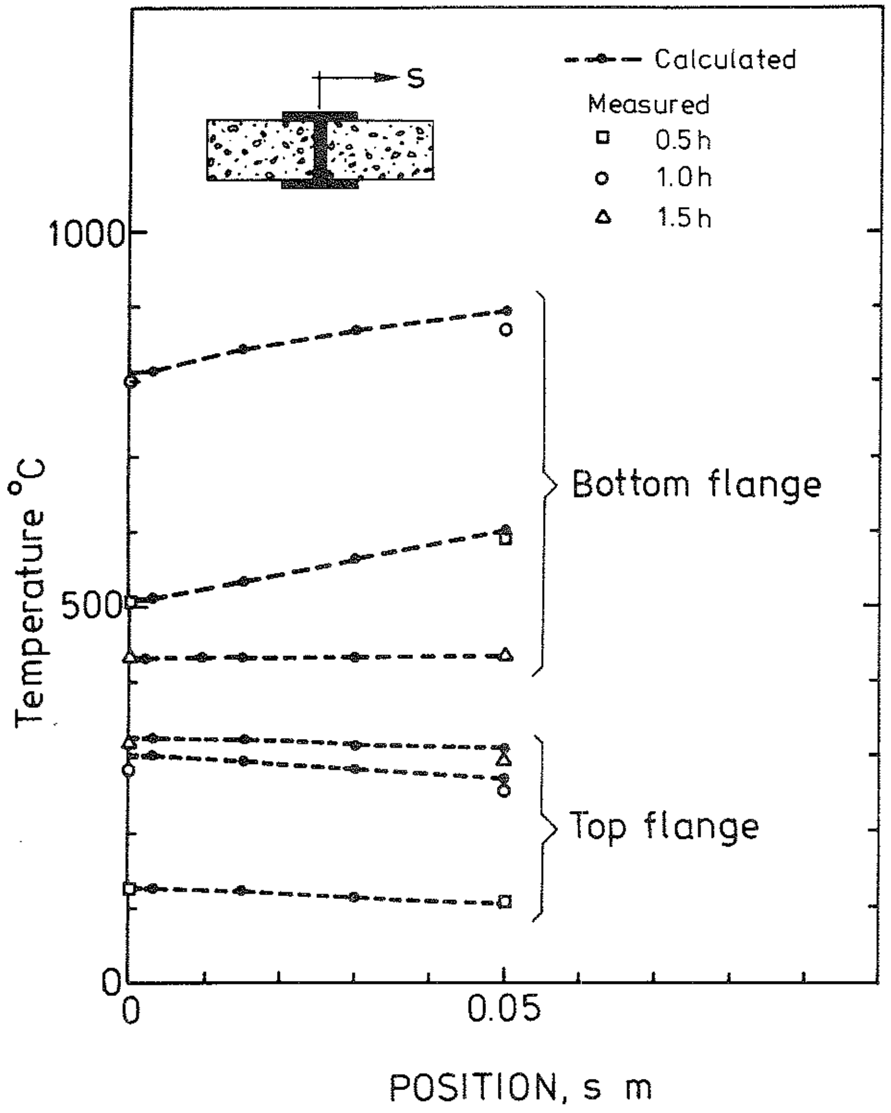


Figure 5.12. Calculated and measured temperature distributions along flanges of I-beam at selected times

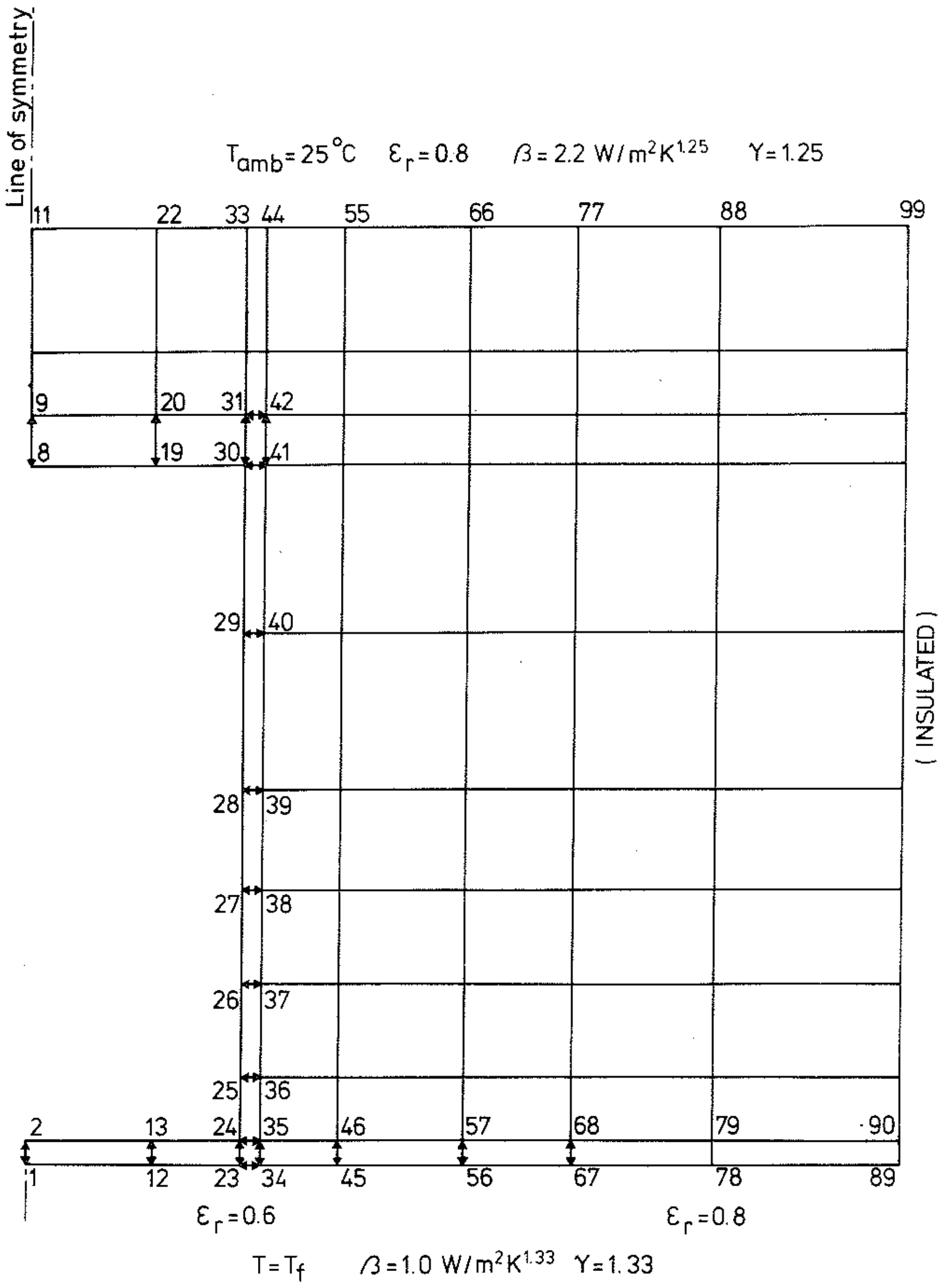


Figure 5.13. Finite element mesh of box girder embedded in concrete

flows along the beam. Convection factors β and powers γ for the enclosure surfaces are estimated by assuming free convection between two horizontal plates of uniform temperature. Thus β and γ equal to 1.6 $W/m^2\ K^{4/3}$ and 1.33, respectively, were obtained by assuming a temperature level of 500 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.

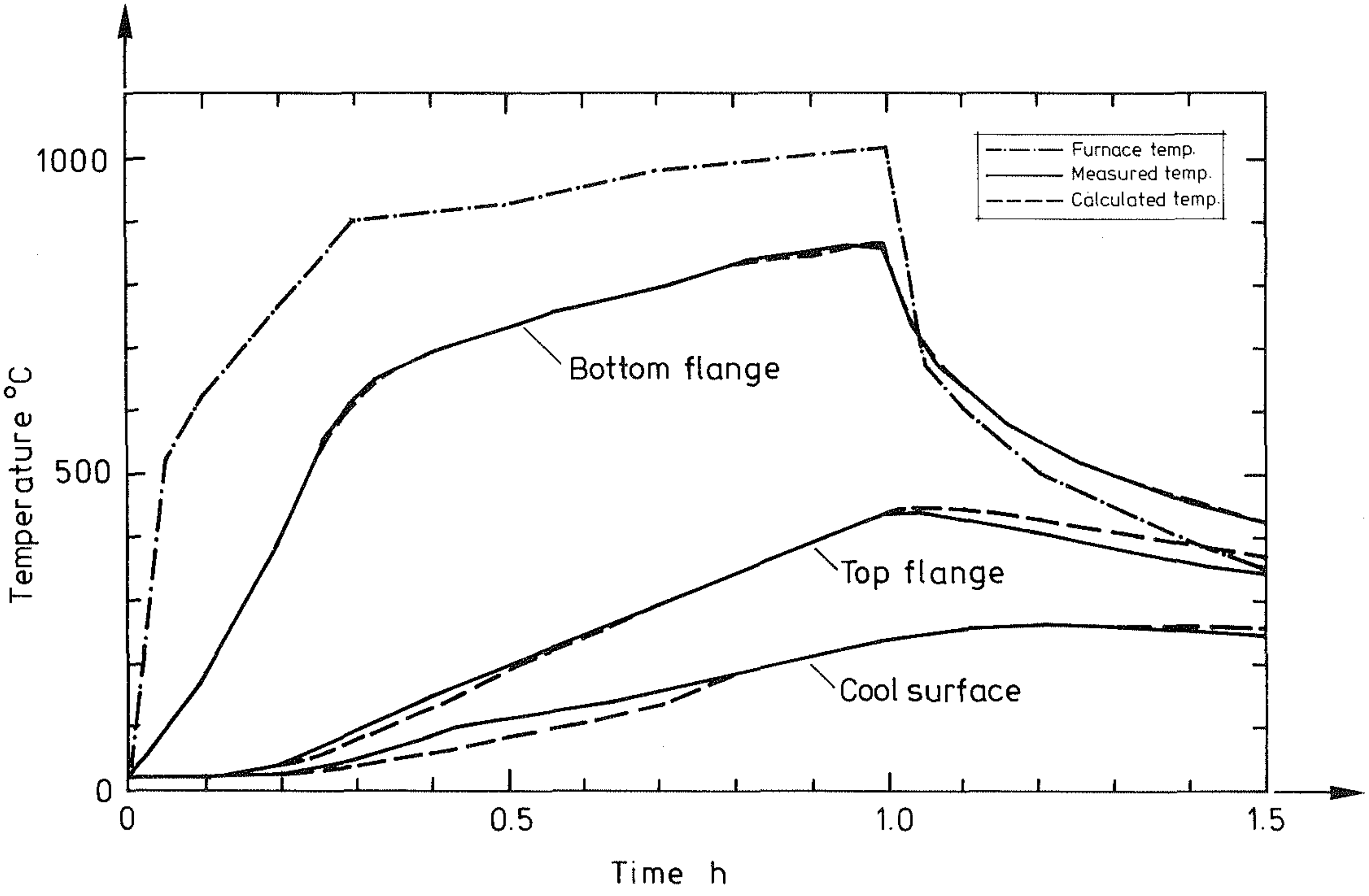


Figure 5.14. Assumed furnace gas temperature in Example III and measured and calculated temperature of top and bottom flanges of box girder and cool concrete surface at the centerline

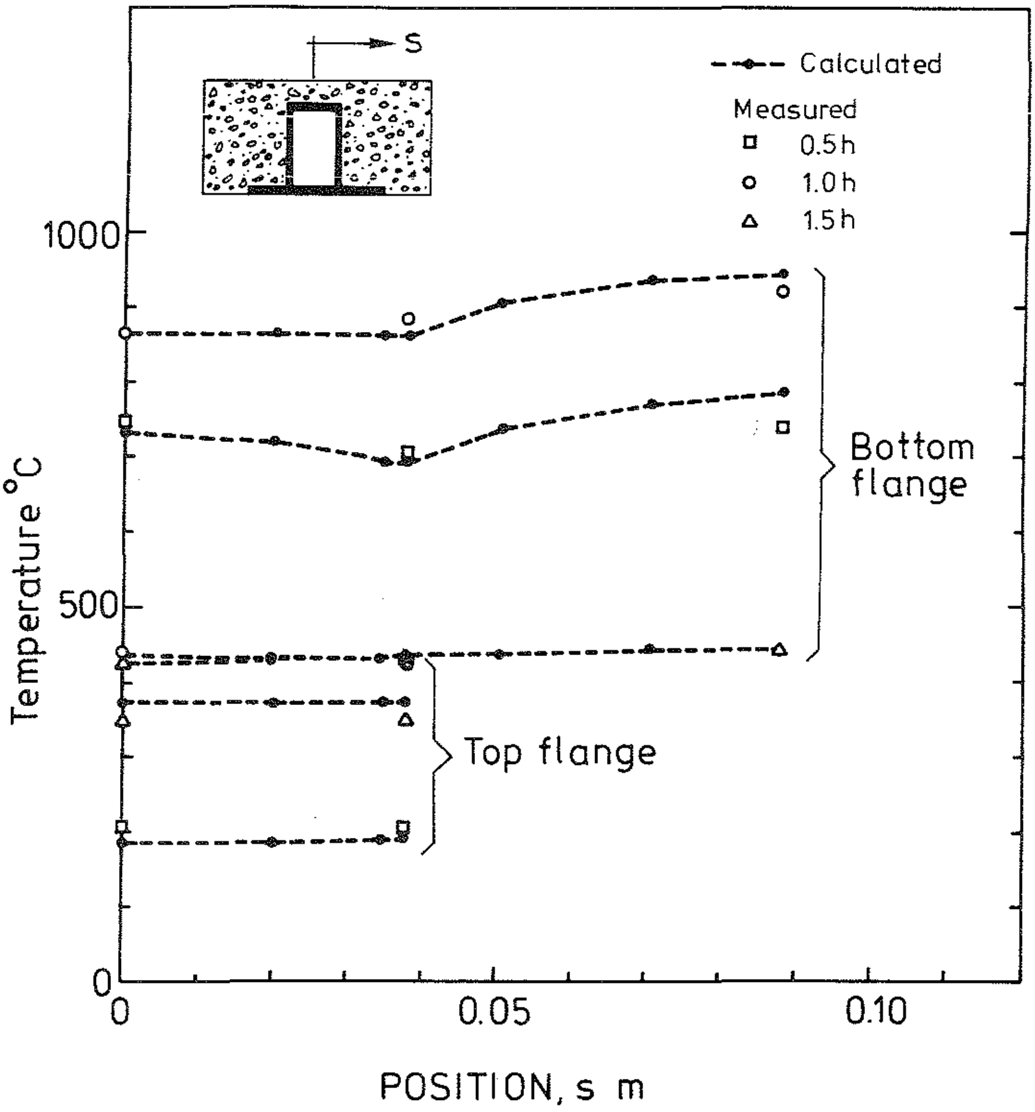


Figure 5.15. Calculated and measured temperature distributions along flanges of box girder at selected times

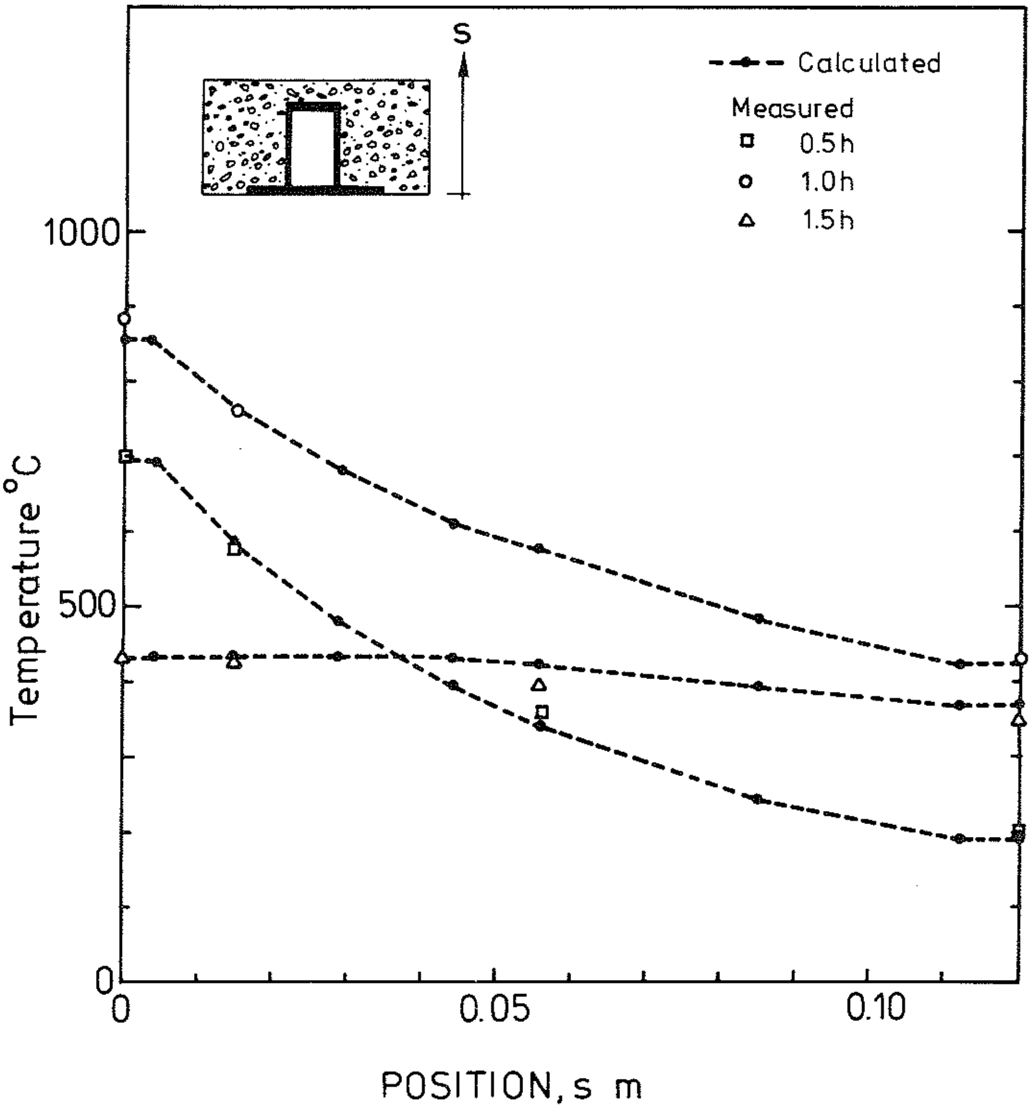


Figure 5.16. Calculated and measured temperature distributions along webs of box girder at selected times

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 lengthly computations, nodes expected to attain approximately the same temperature are coupled and required to attain equal temperature. The technique has been successively 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

[1] Ödeen, K.,"Fire Resistance of Prestressed Concrete Double T Units", Acta Polytechnica Scandinavica, Ci 48, Stockholm, 1968.

[2] Wilson, E.L. and Nickell, R.E., "Applications of the Finite Element Method to Heat Conduction Analysis", Nuclear Engineering and Design, Vol. 4, 1966, pp. 276-286.

[3] Wilson, E.L. and Farhoomand, I., "Non-Linear Heat Transfer Analysis of Axisymmetric Solids", Structural Engineering Laboratory, Report No. 71-6, University of California, Berkeley, April 1971.

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

[5] Iding, R., Bresler, B. and Nizamuddin, Z., "FIRES-T3 - A Computer Program for the Fire Response of Structures - Thermal (Three Dimensional Version)", Report No. UCB FRG 77-15, Fire Research Group, Structural Engineering and Structural Mechanics, Department of Civil Engineering, University of California, Berkeley, 1977.

[6] Rohsenow, W.M. and Hartnett, J.P. , Handbook of Heat Transfer , McGraw-Hill, New York, 1973.

[7] Pettersson, O., Magnusson, S.E. and Thor, J., "Fire Engineering Design of Steel Structures", Bulletin 52, Division of Structural Mechanics and Concrete Construction, Lund Institute of Technology, Lund, 1976.

[8] Zienkiewicz, O.C., The Finite Element Method in Structural and Continuum Mechanics, 3rd edition, McGraw-Hill, London, 1977.

[9] Strang, G. and Fix, J.F., An Analysis of the Finite Element Method, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.

[10] Wickström, U., "A Numerical Procedure for Calculating Temperature in Hollow Structures Exposed to Fire", Report No. UCB FRG 77-9, Fire Research Group, Structural Engineering and Structural Mechanics, Department of Civil Engineering, University of California, Berkeley, 1974. - see also "A Numerical Procedure for Calculating Temperature in Hollow Structures Exposed to Fire", Report No. 79:2, Department of Structural Mechanics, Lund Institute of Technology, Lund, 1979.

[11] ISO, Fire-Resistance Tests - Elements of Building Construction, International Standard 834, 1975-11-01.

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

[13] Alpsten, G., Egenspänningar i varmvalsade stålprofiler (Residual Stresses in Hot-rolled Steel Sections), Institutionen för Brobyggnad, Kungliga Tekniska Högskolan, Stockholm, 1967 (In Swedish).

[14] Anderberg, Y., "Fire-Exposed Hyperstatic Concrete Structures - An Experimental and Theoretical Study", Bulletin 55, Division of Structural Mechanics and Concrete Construction, Lund Institute of Technology, Lund, 1976.

[15] Ödeen, K. and Nordström, Å., "Termiska egenskaper hos betong vid höga temperaturer (Thermal Properties of Concrete at Elevated Temperature)", Cement och Betong 1972:1, Stockholm, 1972 (in Swedish).

[16] Ahlgren, L., "Fuktfixering i porösa byggnadsmaterial (Moisture Fixation in Porous Building Materials)", Report 36, Division of Building Technology, Lund Institute of Technology, Lund, 1972 (Swedish).

[17] Schmidt, E. and Eckert, E., "Über die Richtungsverteilung der Wärmestrahlung von Oberflächen", Forsch.Ing.-Wes. 6, 1935.

[18] Thor, J., "Strålningspåverkan på oisolerade eller undertaksisolerade stålkonstruktioner vid brand", Bulletin 29, Division of Structural Mechanics and Concrete Construction, Lund Institute of Technology, Lund, 1972 (in Swedish).

[19] Holman, J.P. , Heat Transfer , 4th edition, McGraw-Hill, 1976.

[20] Hadvig, S. , Stökiometri og Stofvaerdier (Stochiometry and Material Properties) , Danmarks Tekniske Højskole, 1972 (in Danish).

[21] Fujii, T. and Imura, H., "Natural Convection Heat Transfer from a Plate with Arbitrary Inclination", International Journal of Heat and Mass Transfer, Vol. 15, pp. 755, 1972.

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

[23] Hughes, T.J.R. and Liu, W.K., "Implicit-Explicit Finite Elements in Transient Analysis: Stability Theory", Journal of Applied Mechanics, Vol. 45, June 1978.

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 DELTI equal to zero and execute first time step for calculating first time increment only.

FOR EACH TIME STEP:

14. Increment time step: TIME=TIME+DELTI and KTIME=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
    EN=EN+DELTI*(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
    EN=EN+DELTI*(FLOW-F)
    and get new temperature by iteration
    T^{i+1}=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 DELTI. Go to step 14.
    NO : Print maximum temperature obtained during the process. Go to step 11.

TABLE A:2. Subroutines of TASEF-2

NR	ROUTINE	REFERENCED IN	REFERENCES	INPUT VARIABELS	COMMON BLOCKS
1	ACOUPL	FEM2			
2	AMB	PROG2		TINIT,TAMB,SIGMA,TABS	UNIT
3	ASSA2	FEM2	COND2		RGEO
4	ASSP2	FEM2	INTP,XVERSY		RMAT
5	ASSW2	PROG2			RGEO
6	BFIRE	INIT		TITFIR;NFP;TIM,TB	UNIT,FIRE
7	BRBCA	FQBNDA			
8	BRBCB	FQBNDB			BNOD
9	COND2	ASSA2	XVERSY		RMAT
10	COUPLA	PROG2		NCPLG;NCOUPL	COUPLE
11	COUPLB	FEM2			COUPLE
12	COUPLC	FEM2			COUPLE
13	CTEMP	FEM2			COUPLE
14	DTIME	FEM2			TOUT
15	ENCLO1	PROG2	ENRAD1,ENCON1	CONTRO;NENC;XSYM,YSYM,IGREN	BNOD,ENCLOS,ENRAD,ENCON,DIM,UNIT
16	ENCLO2	FEM2	ENRAD2,ENCON2		ENCLOS
17	ENCON1	ENCLO1	HTRANS		BNOD,ENCLOS,ENCON,DUMMY
18	ENCON2	ENCLO2			ENCON,ENCLOS,DUMMY
19	ENRAD1	ENCLO1	VIEWFC,INVER,MULT,ETRANS		BNOD,ENCLOS,ENRAD,UNIT,DIM,DUMMY
20	ENRAD2	ENCLO2	RADVEC		ENCLOS,ENRAD,UNIT,DUMMY
21	ETRANS	ENRAD1			
22	FEM2	PROG2	INIT,ASSP2,XVERSY,MINTP,ASSA2,MPACKV,FQBNDB,FQGEN,ENCLO2,COUPLB,ACOUPL,CTEMP,HTEMP,PTBNDB,COUPLC,OUT2,MAXCO,DTIME,OUTMA2		RMAT,FIRE
23	FQBNDA	PROG2	BRBCA	NFQNG;FA1,ING1	UNIT,BNOD
24	FQBNDB	FEM2	BRBCB		FQB,BNOD,UNIT

<u>TABLE A:2.</u> (Cont.)

NR	ROUTINE	REFERENCED IN	REFERENCES	INPUT VARIABELS	COMMON BLOCKS
25	FQGEN	FEM2			RMAT
26	GEOCO2	PROG2			ELFICT
27	HTEMP	FEM2	XVERSY		RMAT
28	HTRANS	ENCON1			
29	INIT	FEM2	BFIRE		TOUT,UNIT,FIRE
30	INTERF	PROG2			RGEO,COUPLE
31	INTP	ASSP2	XVERSY		RMAT
32	INVER	ENRAD1			
33	MAIN2		NET2,MESH2,PROG2		
34	MAT	PROG2		MAT;CCC,NTC,NTE,NQE, ET;TC,C;TE,ENT;TQ,QE	RMAT,RGEO
35	MAXCO	FEM2			TOUT
36	MINTP	FEM2			COUPLE
37	MESH2	MAIN2			
38	MPACKV	FEM2			
39	MULT	ENRAD1			DUMMY
40	NET2	MAIN2		TITLE;AXIAL,XMAX,YMAX, XBOX,YBOX,NR,NX,NY; ELFICT,SRDIAC;XA,YA	RGEO
41	NGROUP	PROG2		NGROUP;NCHECK,NUMB, EPSG,BETA,GAMMA; NBOUND	BNOD,ENRAD,ENCON
42	OUT2	FEM2			TOUT
43	OUTMA2	FEM2			FIRE
44	PROG2	MAIN2	REG2,COUPLA,INTERF, MAT,GEOCO2,AMB,NGROUP, FQBNDA,PTBNDA,ENCLO1, TIME,TIME,ASSW2, COUPLB,FEM2		
45	PTBNDA	PROG2		NPTNG;FA1,ING1	PTB,BNOD
46	PTBNDB	FEM2			PTB,BNOD,UNIT
47	RADVEC	ENRAD2			
48	REG2	PROG2			RGEO
49	TIME	PROG2		NT,TIMMAX,DTMAX,TIMFAC, KTMAX,KUPDA;TOUT	
50	VIEWFC	ENRAD1			BNOD,ENCLOS
51	XVERSY	ASSP2,COND2, FEM2,HTEMP, INTP			

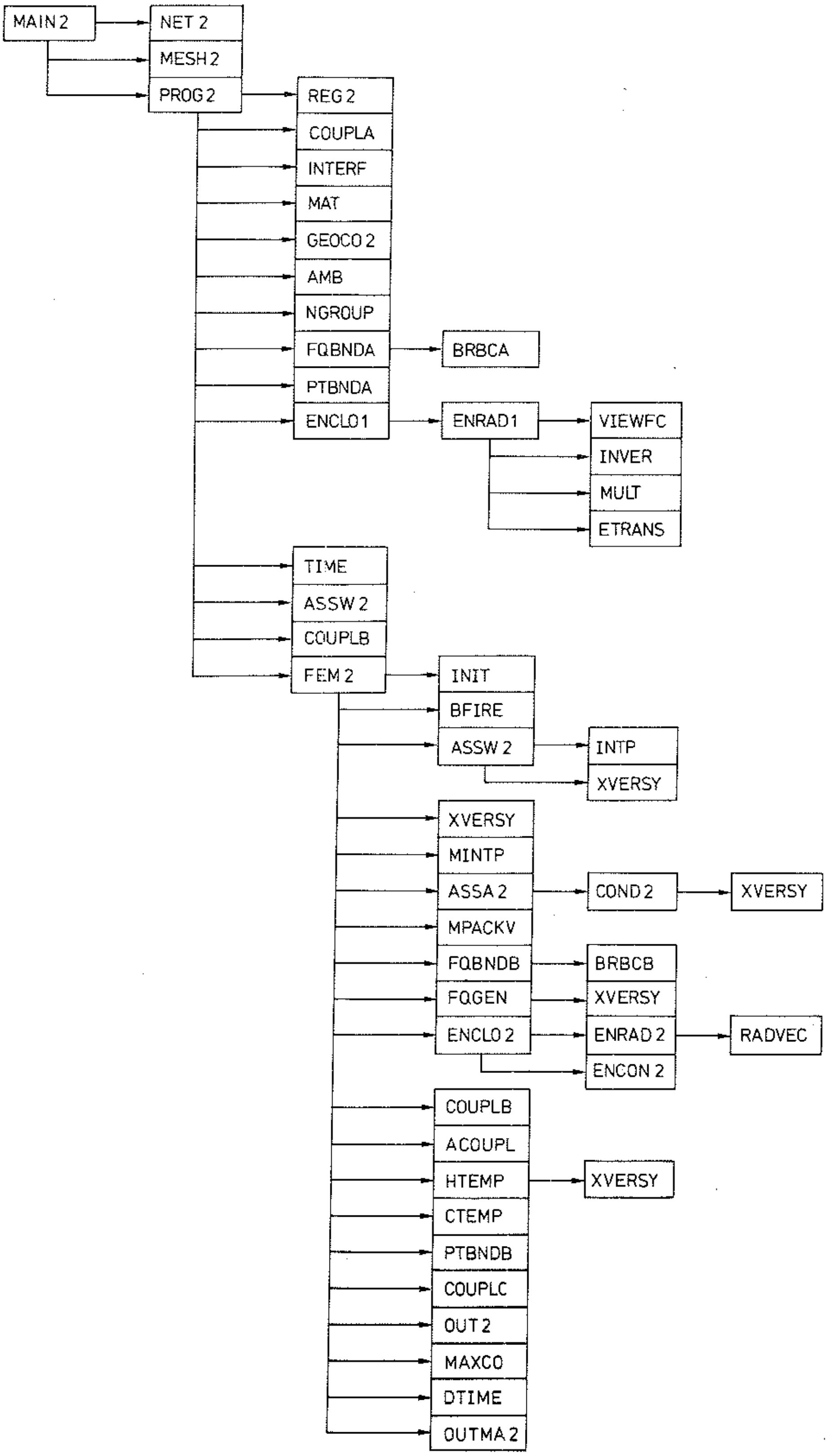


FIG A:1. Chart of subroutines in TASEF-2

TABLE A:3

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

<u>ELFICT,SRDIAC(4)</u>

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

<u>XA(NX)</u>

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

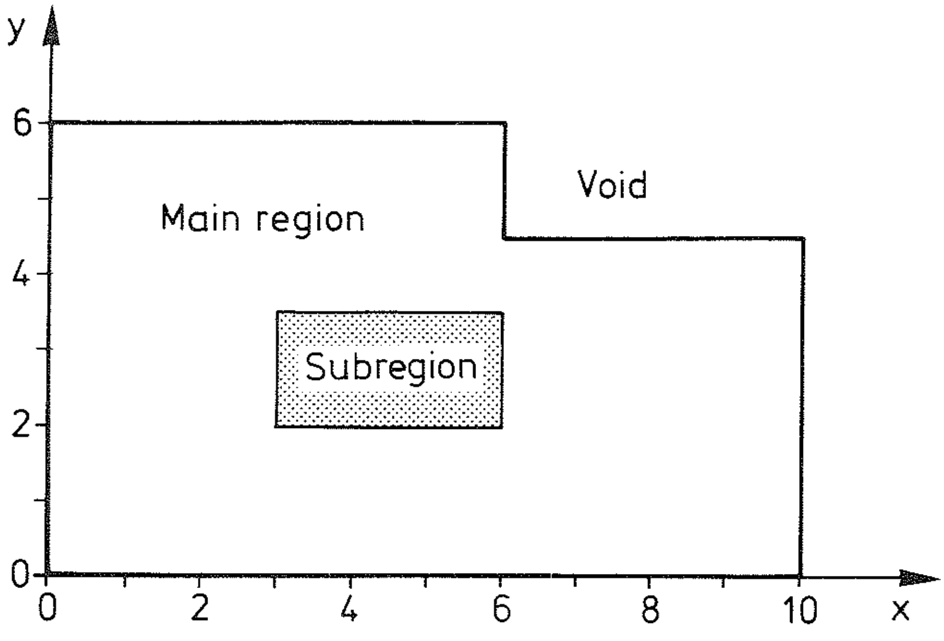
1. Number of groups of coupled nodes

NCPLG

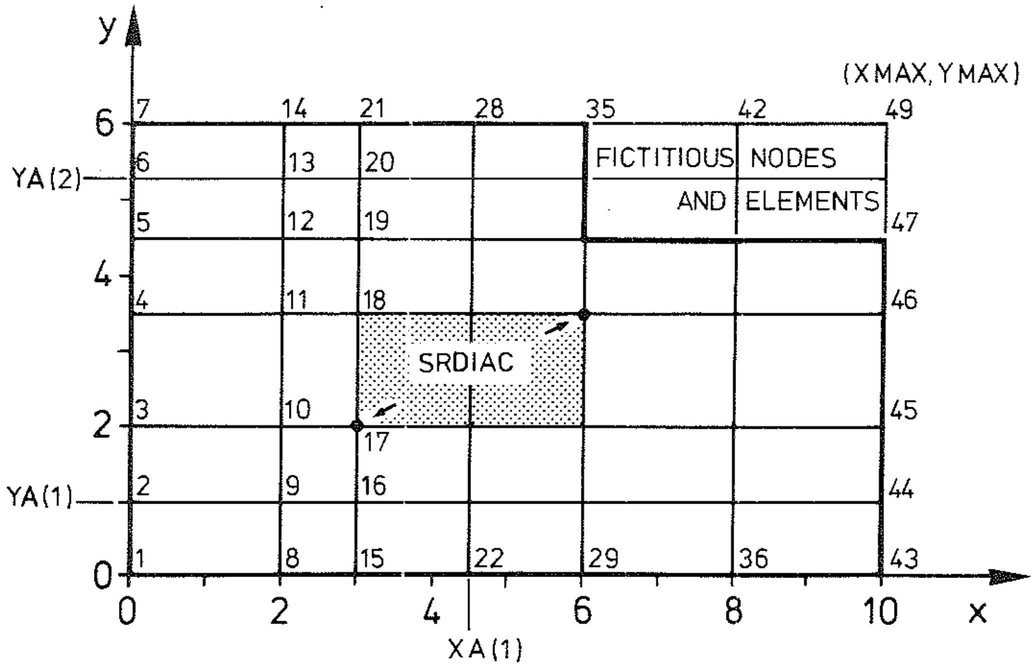
2. Each group

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 ($NTC \leq 20$)
NTE	number of points associated with temperature specific volumetric enthalpy function ($NTC \leq 20$)
NQE	number of points associated with temperature rate of heat generated per unit volume function ($NQE \leq 20$)
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=$5.67\cdot10^{-8}$
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

<u>NGROUP</u>

(NGROUP$\leq$10)

If NGROUP equal zero omit next card

2. Each node group

a. Properties

<u>NCHECK,NUMB,EPSG,BETA,GAMMA</u>

NCHECK	node group number in sequential order starting from 1
NUMB	number of nodes of a group (NUMB$\leq$30)
EPSG	emissivity
BETA	convection factor
GAMMA	convection power

b. Node numbers

<u>NBOUND(NUMB)</u>

TABLE A:3 (Cont.)

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

<u>FA1,ING1</u>

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

<u>CONTRO</u>

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

2. Number of voids

<u>NENC</u>

($NENC \leq 2$)

3. For each void

a. <u>XSYN,YSYM,IGREN(4)</u>

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

TABLE A:3 (Cont.)

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

<u>NFP</u>

($NFP \leq 50$)

b. Data card

<u>TIM,TB,TIM,TB,TIM,TB,...</u> (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.

```
      SUBROUTINE ACOUPL(A,DTA,NN,MAX)
C-----ADD SLAVE NODE QUANTITIES TO MASTER NODE QUANTITIES
C-----FOR EACH COUPLING NODE GROUP
      DIMENSION A(NN,MAX),DTA(NN)
      PARAMETER NCP=50
      COMMON /COUPLE/ NCOUPL(NCP,8),NCPLG
C-----
      IF(NCPLG.EQ.0)RETURN
      DO 50 I=1,NCPLG
      MNOD=NCOUPL(I,1)
      AMX=0.
      DO 40 J=1,8
      NODJ=NCOUPL(I,J)
      IF(NODJ.EQ.0) GOTO 50
      DO 30 K=1,8
      NODK=NCOUPL(I,K)
      IF(NODK.EQ.0) GOTO 40
      NDUM=NODJ-NODK
      IF(NDUM) 30,30,60
60    CONTINUE
      IF(NDUM.GE.MAX) GOTO 30
      AMX=AMX+2.*A(NODJ,MAX-NDUM)
30    CONTINUE
C-----
40    AMX=AMX+A(NODJ,MAX)
50    DTA(MNOD)=DTA(MNOD)+AMX-A(MNOD,MAX)
      RETURN
      END

      SUBROUTINE AMB
C-----READ INITIAL AND AMBIENT TEMPERATURE,
C-----AND UNIT DEPENDENT CONSTANTS
      COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
      DATA SIGM,TAB/5.77E-8,273./
      PRINT 200
C-----
      READ 100,TINIT,TAMB,SIGMA,TABS
C-----
      IF(SIGMA.LT.1.E-20) SIGMA=SIGM
      IF(TABS.LT.1.E-20) TABS=TAB
      PRINT 210,TINIT,TAMB,SIGMA,TABS
      TAMB4=(TAMB+TABS)**4
      RETURN
100   FORMAT()
200   FORMAT(//' INITIAL DATA'/1X,12(1H*))
210   FORMAT(/' INITIAL TEMPERATURE=',G9.3/
     1    ' AMBIENT TEMPERATURE=',G9.3/
     2    ' STEFAN-BOLTZMANN CONSTANT=',G9.3/
     3    ' ABSOLUTE TEMPERATURE SHIFT=',G9.3)
      END

      SUBROUTINE ASSA2(NN,NE,N,KTOP,X,Y,FLA,T,TT,TMAX,AXIAL,MAX,A)
C-----THIS SUBROUTINE COMPUTES THE GLOBAL HEAT COWDUCTION MATRIX A
C-----           NN        NUMBER OF NODES
C-----           NE        NUMBER OF ELEMENTS
C-----           N         ELEMENT REGION NUMBER
C-----           ELA       GEOMETRIC DUMMY VECTOR
C-----           T         TEMPERATURE VECTOR
C-----           TT        TEMPERATURE HISTORY VECTOR
C-----           TMAX      .TRUE. IF MAXIMUM TEMPERATURE IS OBTAINED
C-----           A         GLOBAL HEAT CONDUCTION MATRIX
      PARAMETER MNR=10
      COMMON/RGEO/ELFICT(MNR),ET(MNR),SRDIAC(4,MNR)
```

```
      DIMENSION N(NE),X(NN),Y(NN),T(NN),TT(NN),TMAX(NN),KTOP(4,NE),
     1    A(NN,MAX),FLA(4,NE)
      LOGICAL TMAX,ELFICT,AXIAL
      DO 5 I=1,NN
      DO 5 J=1,MAX
5     A(I,J)=0.
      DO 10 I=1,NE
      N1=N(I)
      IF(ELFICT(N1)) GOTO 10
      K1=KTOP(1,I)
      K2=KTOP(2,I)
      K3=KTOP(3,I)
      K4=KTOP(4,I)
      CALL COND2(T(K1),T(K2),T(K3),T(K4),TT(K1),TT(K2),TT(K3),TT(K4),
     1   TMAX(K1),TMAX(K2),TMAX(K3),TMAX(K4),N(I),AXIAL,C)
      A(K1,MAX)=A(K1,MAX)+C*ELA(1,I)
      A(K2,MAX)=A(K2,MAX)+C*ELA(1,I)
      A(K3,MAX)=A(K3,MAX)+C*ELA(1,I)
      A(K4,MAX)=A(K4,MAX)+C*ELA(1,I)
      A(K2,MAX-1)=A(K2,MAX-1)+C*ELA(2,I)
      A(K3,MAX-K3+K1)=A(K3,MAX-K3+K1)+C*ELA(3,I)
      A(K3,MAX-K3+K2)=A(K3,MAX-K3+K2)+C*ELA(4,I)
      A(K4,MAX-K4+K1)=A(K4,MAX-K4+K1)+C*ELA(4,I)
      A(K4,MAX-K4+K2)=A(K4,MAX-K4+K2)+C*ELA(3,I)
      A(K4,MAX-1)=A(K4,MAX-1)+C*ELA(2,I)
10    CONTINUE
      RETURN
      END

      SUBROUTINE ASSP2(NN,N,X,Y,T,TT,TMAX,EV4,NODEL,MNODEL,P,W,
     1   NODINT,AXIAL)
C-----FORM HEAT CAPACITY MATRIX
      DIMENSION N(1),X(1),T(1),TT(1),TMAX(1),EV4(4,1),NODEL(4,NN),
     1   MNODEL(1),P(1),W(1),NODINT(1)
      PARAMETER MNV=20,MNR=10
      COMMON/RMAT/CCC(MNR),TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR),
     1    ENT(MNV,MNR),CR(MNV,MNR),TQ(MNV,MNR),QE(MNV,MNR),LQ(MNR)
      LOGICAL CCC,LQ
      LOGICAL AXIAL,TMAX
C
      DO 30 I=1,NN
      NODINI=NODINT(I)
      IF(NODINI.LT.0) GOTO 30
      TI =T(I)
      IF(NODINI.GT.0) GOTO 20
      P(I)=0.
      CALL INTP(MNODEL(I),NODEL(1,I),N,EV4,TI,P(I))
      GOTO 30
20    CONTINUE
      CALL XVERSY(TE,ENT,MNV,NODINI,TI,ENI)
      CRA=ENI/TI
      P(I)=W(I)*CRA
30    CONTINUE
      RETURN
      END

      SUBROUTINE ASSW2(NN,NE,N,KTOP,X,Y,EV4,AXIAL,W)
C-----THIS SUBROUTINE COMPUTES THE GLOBAL VOLUME VECTOR W
      DIMENSION X(NN),Y(NN),KTOP(4,NE),EV4(NN),W(NN),N(NE)
      PARAMETER MNR=10
      COMMON/RGEO/ELFICT(MNR)
      LOGICAL AXIAL,ELFICT
      DO 5 I=1,NN
5     W(I)=0.
```

```
      DO 10 I=1,NE
      N1=N(I)
      IF(ELFICT(N1)) GOTO 10
      K1=KTOP(1,I)
      K2=KTOP(2,I)
      K3=KTOP(3,I)
      K4=KTOP(4,I)
      W(K1)=W(K1)+EV4(I)
      W(K2)=W(K2)+EV4(I)
      W(K3)=W(K3)+EV4(I)
      W(K4)=W(K4)+EV4(I)
 10   CONTINUE
      RETURN
      END

      SUBROUTINE BFIRE(FIN)
C-----THIS ROUTINE FORMS VECTORS FOR FIRE BOUNDARY
C-----TIME TEMPERATURE RELATION
      COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
      COMMON/FIRE/TIM(50),TB(50),TITFIR
      LOGICAL FIN
      INTEGER TITFIR(19),BLANK(19),ISO(2)
      DATA BLANK/19*4H    /,ISO/4HISO ,4H834 /
      FIN=.FALSE.
C-----INPUT TITLE OF FIRE BOUNDARY
C-----IF - BLANK -      TERMINATE RUN
C-----IF - ISO 834 -    STANDARD FIRE ASSUMED
C-----ELSE INPUT TIME TEMPERATURE PAIRS
      READ 110,TITFIR
      DO 10 I=1,19
 10   IF(TITFIR(I).NE.BLANK(I)) GOTO 20
      FIN=.TRUE.
      RETURN
 20   CONTINUE
      PRINT 200
      PRINT 210,TITFIR
      DO 30 I=1,2
 30   IF(TITFIR(I).NE.ISO(I)) GOTO 50
C-----ISO STANDARD CURVE
      DT=.05
      TI=0.
      DO 40 I=1,20
      TIM(I)=TI
      TB(I)=TINIT+1325.-430.*EXP(-.2*TI)-270.*EXP(-1.7*TI)-
     1  625.*EXP(-19.*TI)
 40   TI=TI+DT
      TI=TI-DT
      DT=5./29.
      TI=TI+DT
      DO 45 I=21,50
      TIM(I)=TI
      TB(I)=TINIT+1325.-430.*EXP(-.2*TI)-270.*EXP(-1.7*TI)-
     1  625.*EXP(-19.*TI)
 45   TI=TI+DT
      RETURN
 50   CONTINUE
C-----ARBITRARY FIRES
C-----INPUT NUMBER OF TIME TEMPERATURE PAIRS
      READ 100,NFP
      IF(NFP.EQ.0) FIN =.TRUE.
      IF (FIN) RETURN
C-----INPUT TIME TEMPERATURE PAIRS
      READ 100,(TIM(I),TB(I),I=1,NFP)
      PRINT 220,(TIM(I),TB(I),I=1,NFP)
```

```
 100  FORMAT()
 110  FORMAT(20A4)
 200  FORMAT(//' FIRE BOUNDARY TEMPERATURE'/1X,25(1H*))
 210  FORMAT(/1X,20A4)
 220  FORMAT(/' FIRE BOUNDARY TIME - TEMPERATURE INPUT PAIRS'//
     1 '  TIME',5X,'TEMPERATURE'//(G10.3,G10.3))
      RETURN
      END

      SUBROUTINE BRBCA(BR,BC,EPSIG,BET,BAR,NUMI,NB,ING1)
C-----FORM BOUNDARY RADIATION AND CONVECTION MATRICES
      DIMENSION BR(NUMI,2),BC(NUMI,2),BAR(NB,NUMI)
      BR(1,1)=0.
      BR(1,2)=.33333333*BAR(ING1,2)
      NUM1=NUMI-1
      IF(NUM1.EQ.1) GOTO 20
      DO 10 I=2,NUM1
      BR(I,1)=.16666667*BAR(ING1,I)
      BR(I,2)=.3333333*(BAR(ING1,I)+BAR(ING1,I+1))
 10   CONTINUE
 20   CONTINUE
      BR(NUMI,1)=.16666667*BAR(ING1,NUMI)
      BR(NUMI,2)=.33333333*BAR(ING1,NUMI)
C
      DO 30 I=1,NUMI
      DO 30 J=1,2
      BC(I,J)=BET*BR(I,J)
      BR(I,J)=EPSIG*BR(I,J)
 30   CONTINUE
      RETURN
      END

      SUBROUTINE BRBCB(BR,BC,TR,TC,TRD,TCD,NUMI,DTA,NN,MAX,FLOW,TG,
     1   T,ING1)
C-----THIS ROUTINE CALCULATES EXTERNAL HEAT FLOW BY RADIATION  AND CONVECTION
C-----AND ADDS THE CORRESPONDING CONTRIBUTIONS TO THE VECTOR DTA FOR
C-----CALCULATION OF CRITICAL TIME INCREMENT
      DIMENSION BR(NUMI,2),BC(NUMI,2),DTA(NN),FLOW(NN),T(NN)
     1     ,TR(NUMI),TC(NUMI),TRD(NUMI),TCD(NUMI)
      PARAMETER NB=10,NNB=30,NNB2=2*NNB
      COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),
     1 EPSG(NB),BETA(NB),CPG(NB),FA(NB)
      LOGICAL FA
C
C-----FIRST NODE
C
      NODE=NBOUND(ING1,1)
      TR2=TR(1)
      TC2=TC(1)
      TR3=TR(2)
      TC3=TC(2)
      TRD2=TRD(1)
      TCD2=TCD(1)
      TRD3=TRD(2)
      TCD3=TCD(2)
      BR2=BR(1,2)
      BC2=BC(1,2)
      BR3=BR(2,1)
      BC3=BC(2,1)
      FLW=BR2*TR2+BC2*TC2
      FLW=FLW+BR3*TR3
      FLW=FLW+BC3*TC3
      DA=BR2*TRD2+BC2*TCD2
      DA=DA+BR3*TRD3
```

```
      DA=DA+BC3*TCD3
      DTA(NODE)=DTA(NODE)+.5*DA
      FLOW(NODE)=FLOW(NODE)+FLW
C
C-----INTERMEDIATE NODES
C
      NUM1=NUMI-1
      IF(NUM1.EQ.1)GOTO 20
      DO 10 I=2,NUM1
      NODE=NBOUND(ING1,I)
      TR1=TR2
      TR2=TR3
      TR3=TR(I+1)
      TC1=TC2
      TC2=TC3
      TC3=TC(I+1)
      TRD1=TRD2
      TRD2=TRD3
      TRD3=TRD(I+1)
      TCD1=TCD2
      TCD2=TCD3
      TCD3=TCD(I+1)
      BR1=BR3
      BR2=BR(I,2)
      BR3=BR(I+1,1)
      BC1=BC3
      BC2=BC(I,2)
      BC3=BC(I+1,1)
      FLW=BR2*TR2+BC2*TC2
      FLW=FLW+BR1*TR1+BR3*TR3
      FLW=FLW+BC1*TC1+BC3*TC3
      DA=BR2*TRD2+BC2*TCD2
      DA=DA+BR1*TRD1+BR3*TRD3
      DA=DA+BC1*TCD1+BC3*TCD3
      DTA(NODE)=DTA(NODE)+.5*DA
10    FLOW(NODE)=FLOW(NODE)+FLW
20    CONTINUE
C
C-----LAST NODE
C
      NODE=NBOUND(ING1,NUMI)
      TR1=TR2
      TR2=TR3
      TC1=TC2
      TC2=TC3
      TRD1=TRD2
      TRD2=TRD3
      TCD1=TCD2
      TCD2=TCD3
      BR1=BR3
      BR2=BR(NUMI,2)
      BC1=BC3
      BC2=BC(NUMI,2)
      FLW=BR2*TR2+BC2*TC2
      FLW=FLW+BR1*TR1+BC1*TC1
      DA=BR2*TRD2+BC2*TCD2
      DA=DA+BR1*TRD1+BC1*TCD1
      DTA(NODE)=DTA(NODE)+.5*DA
      FLOW(NODE)=FLOW(NODE)+FLW
      RETURN
      END

      SUBROUTINE COND2(T1,T2,T3,T4,TT1,TT2,TT3,TT4,TMAX1,TMAX2,TMAX3,
     1           TMAX4,N1,AXIAL,CE)
```

```
C-----GET ELEMENT CONDUCTIVITY
      LOGICAL TMAX1,TMAX2,TMAX3,TMAX4,AXIAL
      PARAMETER MNV=20,MNR=10
      COMMON/RMAT/CCC(MNR),TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR),
     1    ENT(MNV,MNR),CR(MNV,MNR),TQ(MNV,MNR),QE(MNV,MNR),LQ(MNR)
      LOGICAL CCC,LQ
      IF(CCC(N1)) TM=(TT1+TT2+TT3+TT4)/4.
      IF(.NOT.CCC(N1)) TM=(T1+T2+T3+T4)/4.
      CALL XVERSY(TC,C,MNV,N1,TM,CE)
      RETURN
      END

      SUBROUTINE COUPLA(NODCPL,NN,NODINT)
C-----READ COUPLED NODES AND FORM CONTROL VECTOR NCOUPL
      PARAMETER NCP=50
      COMMON/COUPLE/ NCOUPL(NCP,8),NCPLG
      DIMENSION NODCPL(NN),NODINT(NN)
C-----NODCPL = -1   NODE UNCOUPLED
C-----NODCPL =  0   SLAVE NODE
C-----NODCPL = NCOUPL(1,J)  MASTER NODE
C-----
      DO 5 I=1,NN
5     NODCPL(I)=-1
      PRINT 200
C-----
      READ 100,NCPLG
C-----
      IF(NCPLG.EQ.0) GOTO 30
      PRINT 205
      DO 20 I=1,NCPLG
C-----
      READ 100,(NCOUPL(I,J),J=1,8)
C-----
      II=NCOUPL(I,1)
C-----COUPLED NODES ARE ALWAYS INTERFACE NODES
      NODCPL(II)=I
      DO 10 J=2,8
      II=NCOUPL(I,J)
      IF(II.EQ.0) GOTO 10
      NODINT(II)=0
      NODCPL(II)=0
      JJ=J
10    CONTINUE
      PRINT 210,(NCOUPL(I,J),J=1,JJ)
20    CONTINUE
      RETURN
30    CONTINUE
      PRINT 220
100   FORMAT()
200   FORMAT(//' COUPLED NODES'/1X,13(1H*)/)
205   FORMAT(' MASTER    SLAVES'/)
210   FORMAT(I3,6X,7I3)
220   FORMAT(' NO COUPLED NODES')
      RETURN
      END

      SUBROUTINE COUPLB(V)
C-----ADD SLAVE NODE QUANTITIES TO MASTER NODE QUANTITIES
C-----FOR EACH COUPLING NODE GROUP
      DIMENSION V(1)
      PARAMETER NCP=50
      COMMON /COUPLE/ NCOUPL(NCP,8),NCPLG
C-----
      IF(NCPLG.EQ.0)RETURN
```

```
      DO 30 I=1,NCPLG
      MNOD=NCOUPL(I,1)
      VMN=V(MNOD)
      DO 10 J=2,8
      NOD=NCOUPL(I,J)
      IF(NOD.EQ.0) GOTO 20
      VMN=VMN+V(NOD)
   10 CONTINUE
C-----
   20 CONTINUE
      V(MNOD)=VMN
   30 CONTINUE
      RETURN
      END

      SUBROUTINE COUPLC(T)
C-----UPDATE SLAVE NODE TEMPERATURE
      DIMENSION T(1)
      PARAMETER NCP=50
      COMMON/COUPLE/NCOUPL(NCP,8),NCPLG
      IF(NCPLG.EQ.0)RETURN
      DO 20 I=1,NCPLG
      MNOD=NCOUPL(I,1)
      TMNOD=T(MNOD)
      DO 10 J=2,8
      NOD=NCOUPL(I,J)
      IF(NOD.EQ.0) GOTO 20
   10 T(NOD)=TMNOD
   20 CONTINUE
      RETURN
      END

      SUBROUTINE CTEMP(NODE,T,P,EN,FLOW,F,DELTI,NODEL,MNODEL,N,EV4,NDC)
C-----CALCULATE TEMPERATURE OF INTERFACE NODES
      PARAMETER NCP=50
      COMMON/COUPLE/NCOUPL(NCP,8),NCPLG
      DIMENSION P(1),EN(1),FLOW(1),F(1),NODEL(4,1),MNODEL(1),N(1),EV4(1)
      DATA EPS/.005/
      EN(NODE)=EN(NODE)+(FLOW(NODE)-F(NODE))*DELTI
      ENI=EN(NODE)
      PI=P(NODE)
      DO 40 J=1,5
      T=ENI/PI
      P1=0.
      CALL INTP(MNODEL(NODE),NODEL(1,NODE),N,EV4,T,P1)
      IF (NDC.LT.0) GOTO 30
      DO 20 I=2,8
      NOD=NCOUPL(NDC,I)
      IF (NOD.EQ.0) GOTO 30
   20 CALL INTP(MNODEL(NOD),NODEL(1,NOD),N,EV4,T,P1)
   30 ERR=(PI-P1)/(PI+P1)
      IF(ABS(ERR).LT.EPS) GOTO 50
      PI=(P1+PI)/2.
   40 CONTINUE
      PRINT 200,NODE,T,ERR
  200 FORMAT(/' CONVERGENCE NOT ACHIEVED FOR NODE',I5,'  TEMP=',G9.3,
     1     '  ERR=',G9.2)
   50 T=ENI/P1
      P(NODE)=P1
      RETURN
      END

      SUBROUTINE DTIME(NN,P,DTA,MAX,NODINT,NODCPL,TIME,DELTI,NODT)
C-----THIS ROUTINE CALCULATES TIME INCREMENT
```

```
      DIMENSION P(NN),NODINT(NN),NODCPL(NN),DTA(NN)
      COMMON/TOUT/II,TOUT(100),TIMMAX,DTMAX,TIMFAC,KTMAX,KUPDA
      DELTI=DTMAX
      DO 10 I=1,NN
      IF (NODCPL(I).EQ.0) GOTO 10
      IF (NODINT(I).LT.0) GOTO 10
      DUM=TIMFAC*P(I)/DTA(I)
      DELTI=AMIN1(DELTI,DUM)
      IF(DELTI.EQ.DUM) NODT=I
   10 CONTINUE
      DUM=TOUT(II)-TIME
      IF(DUM.GT.0) DELTI=AMIN1(DELTI,DUM)
      RETURN
      END

      SUBROUTINE ENCLO1(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-----REVICED FEB 1979
C-----
C-----MAJOR VARIABLES,
C-----     NUMB   - NUMBER OF NODES IN THE NODE GROUPS
C-----     NBOUND - NODE NUMBERS IN THE NODE GROUPS
C-----     BAREA  - AREA BETWEEN NODES. THIRD DIMENSION ASSUMED UNITY
C-----     NENC   - NUMBER OF VOIDS
C-----     NENCNG - NUMBER OF NODE GROUPS SURROUNDING EACH VOID
C-----     IGREN  - NODE GROUP NUMBERS SURROUNDING EACH VOID
C-----     NNODEN - NUMBER OF NODES SURROUNDING EACH VOID
C-----     INODEN - ALL NODE NUMBERS IN ALL VOIDS
C-----     E      - NODE RADIATION MATRICES
C-----     EPSG   - EMISSIVITY OF NODE GROUP ZONES
C-----     H      - CONVECTION VECTORS
C-----     BETA   - CONVECTION FACTORS OF THE NODE GROUP ZONES
C-----     TAIR   - VOID AIR TEMPERATURE
C-----
      PARAMETER NB=10,NNB=30,NNB2=2*NNB
      COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),
     1  EPSG(NB),BETA(NB),CPG(NB),FA(NB)
      COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100),
     1   XSYM(2),YSYM(2)
      COMMON/ENRAD/E(1000)
      COMMON/ENCON/H(50),TAIR(2)
      COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
      COMMON/DIM/MAXNG,MAXNOD
      DATA MAXNG,MAXNOD/4,25/
      LOGICAL LEN
      LOGICAL XSYM,YSYM,SYM,LDUM
      INTEGER EN
      PRINT 180
C-----READ CONTROL CARD
C-----IF NO VOIDS IN STRUCTURE RETURN
      READ 90,CONTRO
      IF(CONTRO.EQ.4HVOID) GOTO 10
      PRINT 190
      RETURN
   10 CONTINUE
      TAIR=TINIT
      LEN=.TRUE.
```

```
C-----READ AND ESTABLISH NODE GROUP DATA
C-----READ NUMBER OF VOIDS
      READ 100,NENC
      PRINT 200,NENC
      IND=0
C-----EACH VOID
      DO 150 EN=1,NENC
C-----READ SYMMETRI PROPERTIES AND NODE GROUPS THAT DEFINES THE
C-----VOID
      READ 100.XSYM(EN),YSYM(EN),(IGREN(EN,J),J=1,MAXNG)
      SYM=XSYM(EN).OR.YSYM(EN)
      IND=IND+1
      I1=IGREN(EN,1)
      INODEN(IND)=NBOUND(I1,1)
      NODE1=INODEN(IND)
C-----EACH NODE GROUPE
      DO 20 IG=1,MAXNG
      I1=IGREN(EN,IG)
      IF(I1.EQ.0) GOTO 30
      K1=IG
      NUMI=NUMB(I1)
      LDUM=INODEN(IND).NE.NBOUND(I1,1)
      IF(LDUM) PRINT 500,EN,IND,I1
      IF(LDUM) STOP
C-----EACH ZONE
      DO 20 I=2,NUMI
      IND=IND+1
20    INODEN(IND)=NBOUND(I1,I)
30    CONTINUE
      PRINT 210,EN,(IGREN(EN,J),J=1,K1)
      IF(XSYM(EN)) PRINT 250,EN
      IF(YSYM(EN)) PRINT 260,EN
      NENCNG(EN)=K1
      IF(SYM) GOTO 40
      LDUM=INODEN(IND).NE.NODE1
      IF(LDUM) PRINT 510,EN,NODE1,INODEN(IND)
      IF(LDUM) STOP
      IND=IND-1
40    CONTINUE
      NNODEN(EN)=IND
150   CONTINUE
      CALL ENRAD1(X,Y)
      CALL ENCON1
      RETURN
90    FORMAT(A4)
100   FORMAT()
150   FORMAT(//' VOIDS'/' *****')
190   FORMAT(/' THIS STRUCTURE HAS NO VOIDS'/)
200   FORMAT(/' NUMBER OF VOIDS=',I2/)
210   FORMAT(' VOID NUMBER',I2,' IS SURROUNDED BY THE FOLLOWING '
     1  ,'NODE GROUP(S)',4I3)
250   FORMAT(' VOID NUMBER',I2,' IS SYMMETRICAL AROUND THE X-AXIS
     1     ')
260   FORMAT(' VOID NUMBER',I2,' IS SYMMETRICAL AROUND THE Y-AXIS
     1    ')
500   FORMAT(///' SURROUNDING NODEGROUPS NOT COMPATIBLE'//' EN=',I3,
     1      '     IND=',I3,'    I1=',I3)
510   FORMAT(///' FIRST AND LAST NODE ARE NOT IDENTICAL FOR '
     1     ,'VOID NUMBER',I3//' FIRST NODE=',I4/' SECOND NODE=',I4)
      END

      SUBROUTINE ENCLO2(T,FLOW)
C-----THIS ROUTINE IS CALLED FROM THE BASIC FINITE ELEMENT PROGRAM
C-----TO CALCULATE THE RATE OF HEAT EXCHANGE BETWEEN ENCLOSURE SURFACES
```

```
C-----AS AFUNCTION OF CURRENT TEMPERATURE
      DIMENSION T(1),FLOW(1)
      COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100),
     1   XSYM(2),YSYM(2)
      LOGICAL LEN
      IF(.NOT.LEN) RETURN
C-----CALCULATE RATE OF RADIATION HEAT EXCHANGE
      CALL ENRAD2(T,FLOW)
C-----CALCULATE RATE OF CONVECTION HEAT EXCHANGE
      CALL ENCON2(T,FLOW)
      RETURN
      END

      SUBROUTINE ENCON1
C-----THIS ROUTINE FORMS CONVECTION ARRAY H
      PARAMETER NB=10,NNB=30,NNB2=2*NNB
      COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),TH(NB),
     1  EPSG(NB),BETA(NB),CPG(NB),FA(NB)
      COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100),
     1   XSYM(2),YSYM(2)
      COMMON/ENCON/H(50),TAIR(2)
      COMMON/DUMMY/HZ(25),DUM2(25)
      LOGICAL LEN
      LOGICAL XSYM,YSYM,SYM
      INTEGER EN
      IND=1
C-----FORM ZONE CONVECTION ARRAY
C-----EACH VOID
      DO 150 EN=1,NENC
      SYM=XSYM(EN).OR.YSYM(EN)
      IN=0
      NENG=NENCNG(EN)
C-----EACH NODE GROUP
      DO 10 IG=1,NENG
      I1=IGREN(EN,IG)
      NUMI=NUMB(I1)
      BE=BETA(I1)
C-----EACH ZONE
      DO 10 I=2,NUMI
      IN=IN+1
10    HZ(IN)=BE*BAREA(I1,I)
C-----FORM NODE CONVECTION ARRAY
      CALL HTRANS(HZ,H(IND),IN,SYM)
      N=IN
      IF(SYM) N=N+1
      IND=IND+N
150   CONTINUE
      RETURN
      END

      SUBROUTINE ENCON2(T,FLOW)
C-----THIS ROUTINE CALCULATES THE AIR TEMPERATURE AND CONVECTIVE HEAT
C-----FLOW IN EACH ENCLOSURE
      DIMENSION T(1),FLOW(1)
      COMMON/ENCON/H(50),TAIR(2)
      COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100),
     1   XSYM(2),YSYM(2)
      COMMON/DUMMY/HBAR(25),TEN(25)
      LOGICAL LEN
      INTEGER EN
      DATA PE/.0001/
      IND=0
C-----EACH VOID
      DO 150 EN=1,NENC
```

```
      N=NNODEN(EN)
C-----STORE APPROPRIATE NODAL TEMPERATURES IN DUMMY VECTOR TEN
      DO 10 I=1,N
      NODE=INODEN(IND+I)
10    TEN(I)=T(NODE)
C-----CALCULATE THE AIR TEMPERATURE TA BY ITERATION
C-----USE STARTING VALUE FROM FORMER TIME STEP
      TA=TAIR(EN)
      DO 50 ITER=1,10
      SHBAR=0.
      SHBT=0.
C-----EACH NODE
      DO 20 I=1,N
      TDIF=ABS(TEN(I)-TA)
      IF(TDIF.LT..0001) HBAR(I)=0.
      IF(TDIF.LT..0001) GOTO 20
      HBAR(I)=H(I)*TDIF**.33
      SHBAR=SHBAR+HBAR(I)
      SHBT=SHBT+HBAR(I)*TEN(I)
20    CONTINUE
      IF(SHBAR) 25,90,25
25    TANEW=SHBT/SHBAR
C-----CONVERGENCE CHECK OF AIR TEMPERATURE
      IF(ABS((TANEW-TA)/(TANEW+TA)).LT.PE) GOTO 60
      IF(ITER.GT.1) GOTO 30
      TX=TA
      TY=TANEW
      TA=(TANEW+TA)*.5
      GOTO 50
30    DX=TX-TA
      DY=TY-TANEW
      D=DY-DX
      IF(D)40,70,40
40    DN=TX*DY-TY*DX
      TX=TA
      TY=TANEW
C-----USE LINEAR INTERPOLLATION TO SPEED UP CONVERGENCE
      TA=DN/D
50    CONTINUE
      PRINT 200,EN
      STOP
60    CONTINUE
      DX=TX-TA
      DY=TY-TANEW
      D=DY-DX
      IF(D)65,70,65
C-----USE LINEAR INTERPOLLATION TO IMPROVE THE CALCULATED TEMPERATURE
65    TA=(TX*DY-TY*DX)/D
70    TAIR(EN)=TA
C-----CALCULATE CONVECTION HEAT FLOW AND ADD TO THE GLOBAL HEAT FLOW
C-----VECTOR FLOW
      QTOT=0.
      DO 80 I=1,N
      NODE=INODEN(IND+I)
      Q=HBAR(I)*(TA-TEN(I))
      QTOT=QTOT+Q
80    FLOW(NODE)=FLOW(NODE)+Q
90    CONTINUE
CC      PRINT 220,TA,QTOT
      IND=IND+N
150   CONTINUE
      RETURN
200   FORMAT(///' CONVERGENCE NOT ACHIEVED FOR THE AIR TEMPERATURE',
     1        ' IN ENCLOSURE NUMBER',I3)
```

```
      220   FORMAT(1H+,50X,'AIR TEMPERATURE',F7.1,5X,'TOTAL CONVECTIVE HEAT',
           1     ' EXCHANGE',E10.3)
            END

            SUBROUTINE ENRAD1(X,Y)
      C-----FORM RADIATION MATRICES FOR EACH VOID AND STORE THEM IN
      C-----THE VECTOR E.
      C-----CALCULATE VIEW-FACTOR MATRIX VIEW AND ZONE AREA VECTOR D
            DIMENSION X(1),Y(1),A(25,25),B(25,25)
            PARAMETER NB=10,NNB=30,NNB2=2*NNB
            COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),
           1  EPSG(NB),BETA(NB),CPG(NB),FA(NB)
            COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100),
           1   XSYM(2),YSYM(2)
            COMMON/ENRAD/E(1000)
            COMMON/UNIT/SIGMA,TABS
            COMMON/DIM/MAXNG,MAXNOD
            COMMON/DUMMY/D(25),DUM2(25)
            DIMENSION VIEW(25,25)
            EQUIVALENCE (A(1),VIEW(1))
            DATA IND,IE/0,1/
            LOGICAL LEN
            LOGICAL XSYM,YSYM,SYM
            INTEGER EN
      C-----EACH VOID
            DO 150 EN=1,NENC
            CALL VIEWFC(X,Y,D,EN,VIEW,MAXNOD)
      C-----FORM THE MATRICES A AND B
            NENG=NENCNG(EN)
            IN=0
      C-----EACH NODE GROUP
            DO 120 IG=1,NENG
            I1=IGREN(EN,IG)
            NUMI=NUMB(I1)
      C-----EACH ZONE
            DO 120 I=2,NUMI
            IN=IN+1
            JN=0
            DO 120 JG=1,NENG
            J1=IGREN(EN,JG)
            NUMJ=NUMB(J1)
            EPSJ=EPSG(J1)
            DO 120 J=2,NUMJ
            JN=JN+1
            B(IN,JN)=VIEW(IN,JN)*SIGMA
            A(IN,JN)=-VIEW(IN,JN)*(1.-EPSJ)/EPSJ/D(JN)
            IF(IN.NE.JN) GOTO 120
            B(IN,JN)=-SIGMA+B(IN,JN)
            A(IN,JN)=1./EPSJ/D(JN)+A(IN,JN)
      120   CONTINUE
            N=IN
      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,N,MAXNOD)
            SYM=.FALSE.
            IF(XSYM(EN).OR.YSYM(EN)) SYM=.TRUE.
            NZ=N
            IF(SYM) N=N+1
      C-----TRANSFORM THE LOCAL RADIATION MATRICE A AND STORE THE RESULT IN
      C-----VECTOR E
      C-----B IS EMPLOYED AS A DUMMY MATRIX
            CALL ETRANS(A,B,E(IE),N,NZ,SYM,MAXNOD)
            IE=IE+N*N
```

```
      IND=IND+N
150   CONTINUE
      RETURN
      END

      SUBROUTINE ENRAD2(T,FLOW)
C-----THIS ROUTINE CALCULATES THE PADIATION 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/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100),
     1   XSYM(2),YSYM(2)
      COMMON/ENRAD/E(1000)
      COMMON/UNIT/SIGMA,TABS
      COMMON/DUMMY/ETA(25),Q(25)
      LOGICAL LEN
      INTEGER EN
      IE=1
      IND=0
C-----EACH VOID
      DO 150 EN=1,NENC
      N=NNODEN(EN)
C-----CALCULATE ABSULUTE TEMPERATURES TO THE FOURTH POWER FOR FOR THE
C-----NODES OF THE ENCLOSURE SURFACE
      DO 10 I=1,N
      NODE=INODEN(IND+I)
      DUM=T(NODE)+TABS
      DUM=DUM*DUM
10    ETA(I)=DUM*DUM
C-----CALCULATE ENCLOSURE SURFACE RADIATION HEAT EXCHANGE VECTOR Q=E*ETA
      CALL RADVEC(E(IE),ETA,N,Q)
C-----ADD TO GLOBAL HEAT FLOW VECTOR FLOW
      DO 20 I=1,N
      NODE=INODEN(IND+I)
20    FLOW(NODE)=FLOW(NODE)+Q(I)
      IE=IE+N*N
      IND=IND+N
150   CONTINUE
      RETURN
      END

      SUBROUTINE ETRANS(A,B,E,N,NZ,SYM,MAX)
C-----THIS ROUTINE TRANSFORMS THE ZONE RADIATION MATRIX A TO A NODE
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
C-----E=SAT*A*SA
C-----B=SAT*A
      DO 10 I=2,NZ
      DO 10 J=1,NZ
10    B(I,J)=A(I-1,J)+A(I,J)
      DO 30 J=1,NZ
      IF(SYM) GOTO 20
      B(1,J)=A(1,J)+A(NZ,J)
      GOTO 30
20    B(1,J)=A(1,J)
      B(N,J)=A(NZ,J)
30    CONTINUE
C-----E=B*SA
      DO 50 I=1,N
      DO 50 J=2,NZ
50    E(I,J)=.25*(B(I,J)+B(I,J-1))
      DO 70 I=1,N
```

```
      IF(SYM) GOTO 60
      E(I,1)=.25*(B(I,1)+B(I,N))
      GOTO 70
60    E(I,1)=.25*B(I,1)
      E(I,N)=.25*B(I,NZ)
70    CONTINUE
      RETURN
      END

      SUBROUTINE FEM2(IX,IY,NN,NE,NR,N,KTOP,X,Y,T,TT,TMAX,ELA,EV4,A,MAX,
     1    P,W,EN,F,FLOW,AXIAL,NODCPL,NODINT,NODEL,MNODEL,DTA)
C-----THIS ROUTINE INITIALIZES SYSTEM ARRAYS AND
C-----CONTROLS TIME INTEGRATION
      DIMENSION N(NE),KTOP(4,NE),X(NN),Y(NN),T(NN),TT(NN),TMAX(NN),
     1     ELA(4,NE),EV4(NE),A(NN,MAX),P(NN),W(NN),EN(NN),F(NN),FLOW(NN)
     2    ,NODCPL(NN),NODINT(NN),NODEL(4,NN),MNODEL(NN),DTA(NN)
      PARAMETER MNV=20,MNR=10
      COMMON/FIRE/TIM(50),TB(50),TITFIR
      COMMON/RMAT/TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR),ENT(MNV,MNR),
      COMMON/TOUT/II,TOUT(100),TIMMAX,DTMAX,TIMFAC,KTMAX,KUPDA
      LOGICAL TMAX,AXIAL
      LOGICAL FIN,CON,NODI,UPDA
      DATA FIN/.FALSE./
C-----
5     CONTINUE
      KTIME=0
      CON=.TRUE.
C-----INITIALIZE NODAL TEMPERATURES
      CALL INIT(NN,T,TT,TMAX,NODINT)
C-----INPUT FIRE BOUNDARY TEMPERATURE
      CALL BFIRE(FIN)
C-----FIRST TIME INCREMENT FOR CALCULATING INCREMENT LENGTH ONLY
      DELTI=0.
C-----IF FIN=.TRUE. ANALIZE NEW FIRE
C-----IF FIN=.TRUE. TERMINATE RUN
      IF(FIN)GOTO 1000
      CALL ASSP2(NN,N,X,Y,T,TT,TMAX,EV4,NODEL,MNODEL,P,W,NODINT,
     1  AXIAL)
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
      NDI=NODINT(I)
      IF(NODCPL(I).EQ.0.OR.NDI.LT.0) GOTO 10
      IF(NDI.GT.0)  CALL XVERSY(TE,ENT,MNV,NDI,T(I),EN(I))
C-----MASTER NODES AND INTERFACE NODES
      IF(NODCPL(I).GT.0.AND.NDI.EQ.0) CALL MINTP(I,NODCPL(I),P)
      IF(NDI.EQ.0) EN(I)=P(I)*T(I)
10    CONTINUE
C-----
C-----START TIME INTEGRATION LOOP
C-----
700   CONTINUE
      DUM1=FLOAT(KTIME)/FLOAT(KUPDA)
      DUM2=AINT(DUM1)
      UPDA=DUM1.EQ.DUM2.OR.KTIME.EQ.1
      KTIME=KTIME+1
C-----CALCULATE INTERNAL HEAT FLOW BY CONDUCTION
      IF(UPDA)
     1   CALL ASSA2(NN,NE,N,KTOP,X,Y,ELA,T,TT,TMAX,AXIAL,MAX,A)
      DO 20 I=1,NN
20    DTA(I)=A(I,MAX)
      CALL MPACKV(A,T,F,MAX,NN)
C-----GET FIRE TEMPRATURE
```

```
      CALL XVERSY(TIM,TB,50,1,TIME,TFIRE)
C-----CALCULATE BOUNDARY HEAT FLOW
      CALL FQBNDB(T,FLOW,DTA,NN,MAX,TFIRE)
C-----CALCULATE INTERNALLY GENERATED HEAT FLOW
      CALL FQGEN(NN,NE,N,KTOP,EV4,T,FLOW)
C-----CALCULATE ENCLOSURE (VOID) HEAT FLOW
      CALL ENCLO2(T,FLOW)
C-----CALCULATE HEAT CAPACITY MATRIX AT CURRENT TEMPERATURE
      CALL ASSP2(NN,N,X,Y,T,TT,TMAX,EV4,NODEL,MNODEL,P,W,NODINT,
     1   AXIAL)
C-----SUM APPROPIATE QUANTITIES OF COUPLED NODES
      CALL COUPLB(F)
      CALL COUPLB(FLOW)
      CALL ACOUPL(A,DTA,NN,MAX)
C-----CALCULATE NEW NODAL TEMPERATURES
      DO 50 I=1,NN
      NDC=NODCPL(I)
      NDI=NODINT(I)
      IF(NDI.LT.0.OR.NDC.EQ.0) GOTO 50
C-----HOMOGENEOUS NODES ONLY
      NODI=NDI.GT.0
      IF(NODI)CALL HTEMP(T(I),W(I),EN(I),FLOW(I),F(I),NDI,DELTI)
      IF(NODI) GOTO 50
C-----INTERFACE NODES
      IF(NODCPL(I).GT.0) CALL MINTP(I,NODCPL(I),P)
      CALL CTEMP(I,T(I),P,EN,FLOW,F,DELTI,NODEL,MNODEL,N,EV4,NDC)
50    CONTINUE
C-----SET PRESCRIBED NODAL TEMPERATURES
      CALL PTBNDB(T,TFIRE)
      CALL COUPLC(T)
C-----PRINT CURRENT NODAL AND VOID AIR TEMPERATURES
      CALL OUT2(IX,IY,NN,NE,X,Y,TIME,KTIME,DELTI,T,TT,TMAX,FLOW,TFIRE,
     1       NODT,AXIAL)
C-----SET CON=.FALSE. TO TERMINATE TIME INTEGRATION
      CALL MAXCO(NN,TMAX,TT,T,TIME,KTIME,CON)
C-----CALCULATE NEW TIME INCREMENT DELTI
      CALL DTIME(NN,P,DTA,MAX,NODINT,NODCPL,TIME,DELTI,NODT)
      TIME=TIME+DELTI
      IF(CON) GOTO 700
C-----
C-----END TIME INTEGRATION LOOP
C-----
720   CONTINUE
      TIME=TIME-DELTI
C-----PRINT MAXIMUM TEMPERATURE OBTAINED DURING ANALYSIS
      CALL OUTMA2(IX,IY,NN,NE,X,Y,TIME,KTIME,T,TT,TMAX,FLOW,AXIAL)
      GOTO 5 .
1000  RETURN
      END

      SUBROUTINE FQBNDA
C-----THIS ROUTINE FORMS RADIATION AND CONVECTION MATRICES BR AND BC
      COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
      PARAMETER NB=10,NNB=30,NNB2=2*NNB
      COMMON/FQB/NFQNG,NFQG(NB),TR(NNB),TC(NNB)
     1  ,BR(NNB2),BC(NNB2)
      COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),
     1  EPSG(NB),BETA(NB),CPG(NB),FA(NB)
      LOGICAL FA,FA1
C-----READ NUMBER OF BOUNDARY NODES GROUPS
      READ 100,NFQNG
C-----
      IF(NFQNG.EQ.0) RETURN
      PRINT 200
```

```
      IND=1
C-----EACH BOUNDARY FLOW NODE GROUP
      DO 10 IB=1,NFQNG
C-----
      READ 100,FA1,ING1
C-----IF FA1=.TRUE. FIRE BOUNDARY ELSE AMBIENT TEMPERATURE
C-----ING1 = NODE GROUP NUMBER
      FA(ING1)=FA1
      NFQG(IB)=ING1
      NUMI=NUMB(ING1)
      BET=BETA(ING1)
      EPSIG=EPSG(ING1)*SIGMA
      IF(EPSIG.EQ.0.AND.BET.EQ.0) PRINT 300
      IF(EPSIG.EQ.0.AND.BET.EQ.0) STOP
      CALL BRBCA(BR(IND),BC(IND),EPSIG,BET,BAREA,NUMI,NB,ING1)
      IND=IND+2*NUMI
      IF(FA1) PRINT 210,ING1
      IF(.NOT.FA1) PRINT 220,ING1
10    CONTINUE
100   FORMAT()
200   FORMAT(//' PRESCRIBED FLOW BOUNDARY'/1X,24(1H*)/
     1 /' NODE GROUPS AND TYPES OF BOUNDARIES'/)
210   FORMAT(' NODE GROUP',I3,' FIRE BOUNDARY')
220   FORMAT(' NODE GROUP',I3,' AMBIENT BOUNDARY')
300   FORMAT(/' BOTH EMISSIVITY AND CONVECTION FACTOR ZERO')
      RETURN
      END

      SUBROUTINE FQBNDB(T,FLOW,DTA,NN,MAX,TFIRE)
C-----THIS ROUTINE PREPARES CALCULATION OF PRESCRIBED BOUNDARY FLOW
      DIMENSION T(NN),DTA(NN),FLOW(NN)
      PARAMETER NB=10,NNB=30,NNB2=2*NNB
      COMMON/FQB/NFQNG,NFQG(NB),TR(NNB),TC(NNB)
     1  ,BR(NNB2),BC(NNB2),TRD(NNB),TCD(NNB)
      COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),
     1  EPSG(NB),BETA(NB),CPG(NB),FA(NB)
      COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
      LOGICAL FA
C-----NULL FLOW VECTOR
      DO 777 I=1,NN
777   FLOW(I)=0.
C-----RETURN IF NO PRESCRIBED BOUNDARY FLOW
      IF(NFQNG.EQ.0) RETURN
      TF4=(TFIRE+TABS)**4
      IND=1
C-----EACH BOUNDARY FLOW NODE GROUP
      DO 30 IB=1,NFQNG
      TG4=TAMB4
      TG=TAMB
      ING1=NFQG(IB)
      IF(FA(ING1)) TG=TFIRE
      IF(FA(ING1)) TG4=TF4
      NUMI=NUMB(ING1)
      CP=CPG(ING1)
      DO 20 I=1,NUMI
      NODE=NBOUND(ING1,I)
      TNODE=T(NODE)
      TNABS=TNODE+TABS
C-----RADIATION
      TRD(I)=4.*TNABS**3
      TR(I)=TG4-TNABS**4
C-----CONVECTION
      DUM=TG-TNODE
      TCD(I)=CP*ABS(DUM)**(CP-1.)
```

```
      IF(DUM) 5,20,10
5     TC(I)=-(-DUM)**CP
      GOTO 20
10    TC(I)=DUM**CP
20    CONTINUE
C-----
      CALL BRGCB(BR(IND),BC(IND),TR,TC,TRD,TCD,NUMI,DTA,NN,
     1   MAX,FLOW,TG,T,ING1)
      IND=IND+2*NUMI
30    CONTINUE
      RETURN
      END

      SUBROUTINE FQGEN(NN,NE,N,KTOP,EV4,T,FLOW)
C-----CALCULATE INTERNALLY GENERATED HEAT
      DIMENSION N(NE),KTOP(4,NE),EV4(NE),T(NN),FLOW(NN)
      PARAMETER MNV=20,MNR=10
      COMMON/RMAT/CCC(MNR),TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR),
     1    ENT(MNV,MNR),CR(MNV,MNR),TQ(MNV,MNR),QE(MNV,MNR),LQ(MNR)
      LOGICAL CCC,LQ
      DO 20 I=1,NE
      N1=N(I)
      IF(.NOT.LQ(N1)) GOTO 20
      DO 10 K=1,4
      NOD=KTOP(K,I)
      CALL XVERSY(TQ,QE,MNV,N1,T(NOD),FGEN)
10    FLOW(NOD)=FLOW(NOD)+EV4(I)*FGEN
20    CONTINUE
      RETURN
      END

      SUBROUTINE GEOCO2(NN,NE,N,KTOP,X,Y,AXIAL,ELA,EV4)
C-----THIS SUBROUTINE COMPUTES ELEMENT GEOMETRICAL CONSTANTS
C-----
C-----          NN        NUMBER OF NODES
C-----          NE        NUMBER OF ELEMENTS
C-----          N         ELEMENT REGION NUMBER
C-----          X,Y       NODE COORDINATES
C-----          AXIAL     TRUE IF AXIAL SYMMETPIC PROBLEM
C-----          ELA       ELEMENT GEOMERTIC CONSTANTS
C-----          EV4       ELEMENT VOLUME/4
      DIMENSION X(NN),Y(NN),EV4(NE),ELA(4,NE),KTOP(4,NE),N(NE)
      PARAMETER MNR=10
      COMMON/RGEO/ELFICT(MNR),ET(MNR),SRDIAC(4,MNR)
      LOGICAL AXIAL,ELFICT
      DO 5 I=1,NE
      N1=N(I)
      K1=KTOP(1,I)
      K4=KTOP(4,I)
      A=X(K4)-X(K1)
      B=Y(K4)-Y(K1)
      ET1=ET(N1)
      ELA(1,I)=ET1*(A*A+B*B)/3./A/B
      ELA(2,I)=ET1*(-2*A*A+B*B)/6./A/B
      ELA(3,I)=ET1*(A*A-2.*B*B)/6./A/B
      ELA(4,I)=-ET1*(A*A+B*B)/6./A/B
      EV4(I)=ET1*A*B/4.
5     CONTINUE
      RETURN
      END

      SUBROUTINE HTEMP(T,W,EN,FLOW,F,N1,DELTI)
C-----CALCULATE TEMPERATURE FOR HOMOGENEOUS NODES
      PARAMETER MNV=20,MNR=10
```

```
      COMMON/RMAT/TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR),ENT(MNV,MNR),
     1             CR(MNV,MNR)
      D=(FLOW-F)*DELTI
      EN=EN+D/W
      CALL XVERSY(ENT,TE,MNV,N1,EN,T)
      RETURN
      END

      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(1)
      LOGICAL SYM
      DO 10 I=2,N
      H(I)=.5*(HZ(I-1)+HZ(I))
10    CONTINUE
      IF(SYM) GOTO 20
      H(1)=.5*(HZ(N)+HZ(1))
      GOTO 30
20    CONTINUE
      H(1)=.5*HZ(1)
      H(N+1)=.5*HZ(N)
30    RETURN
      END

      SUBROUTINE INIT(NN,T,TT,TMAX,NODINT)
C-----SET INITIAL NODAL TEMPERATURE
      DIMENSION T(NN),TT(NN),TMAX(NN),NODINT(NN)
      LOGICAL TMAX
      COMMON/TOUT/II,TOUT(100),TIMMAX
      COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
      II=1
      DO 1 I=1,NN
      IF(NODINT(I).LT.0) GOTO 1
      T(I)=TINIT
      TT(I)=T(I)
1     TMAX(I)=.FALSE.
      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
      PARAMETER MNR=10,NCP=50
      COMMON/RGEO/ELFICT(MNR)
      COMMON/COUPLE/NCOUPL(NCP,8),NCPLG
      DIMENSION KTOP(4,NE),N(NE),NODINT(NN),NODCPL(NN)
      LOGICAL ELFICT
      PRINT 200
      DO 5 I=1,NN
5     NODINT(I)=-1
      IF(NR.EQ.1) GOTO 50
      NX1=NX-1
      NY1=NY-1
      IF(NY.EQ.2)GOTO 25
C-----
      DO 20 I=1,NX1
      INY=(I-1)*NY
      INY1=(I-1)*NY1
      IE1=INY1+1
```

```
      N1=N(IE1)
C-----
      DO 20 J=2,NY1
      IF(ELFICT(N1)) GOTO 15
      DO 10 IDUM=1,3
      IE2=IE1+(IDUM-2)*NY1+1
      IF(IE2.LE.0.OR.IE2.GT.NE)GOTO 10
      N2=N(IE2)
      IF(ELFICT(N2)) GOTO 10
      IF(N1.EQ.N2) GOTO 10
      IF(IDUM.NE.2)GOTO 10
      NOD=INY+J
      NODINT(NOD)=0
      NOD=NOD+NY
      NODINT(NOD)=0
C-----
10    CONTINUE
15    CONTINUE
      IE1=IE1+1
      N1=N(IE1)
C-----
20    CONTINUE
25    CONTINUE
C-----
      IF(NX.EQ.2)GOTO 50
      DO 40 I=1,NY1
      INX=I-1
      IE1=I
      N1=N(IE1)
C-----
      DO 40 J=2,NX1
C-----
      IE2=IE1+NY1
      N2=N(IE2)
      IF(ELFICT(N1).OR.ELFICT(N2)) GOTO 30
      IF(N1.EQ.N2) GOTO 30
      NOD=(J-1)*NY+I
      NODINT(NOD)=0
      NOD=NOD+1
      NODINT(NOD)=0
C-----
30    CONTINUE
      IE1=IE2
      N1=N2
C-----
40    CONTINUE
50    CONTINUE
C-----
      DO 70 I=1,NE
      N1=N(I)
      IF(ELFICT(N1)) GOTO 70
      DO 60 J=1,4
      NOD=KTOP(J,I)
 60   IF(NODINT(NOD).EQ.-1) NODINT(NOD)=N1
 70   CONTINUE
C-----IF ONE NODE IN A COUPLED GROUP IS AN INTERFACE NODE
C-----ALL NODES IN THE GROUP ARE CONSIDERED INTERFACE NODES
      IF (NCPLG.EQ.0) GOTO 120
      DO 110 I=1,NCPLG
      DO 85 J=1,8
      NOD=NCOUPL(I,J)
      IF(NOD.EQ.0) GOTO 110
      IF(NODINT(NOD).EQ.0) GOTO 90
85    CONTINUE
```

```
90    CONTINUE
      DO 100 J=1,8
      NOD=NCOUPL(I,J)
      IF(NOD.EQ.0) GOTO 110
      NODINT(NOD)=0
100   CONTINUE
110   CONTINUE
120   CONTINUE
      PRINT 205
      DO 80 I=1,NX
      II=(I-1)*NY
80    PRINT 210,(NODINT(II+J),J=1,NY)
205   FORMAT(/' -1 - FICTITIOUS NODE'/'  0 - INTERFACE NODE'/
     1 '  1 - HOMOGENEOUS NODE')
200   FORMAT(//' INTERFACE NODES'/16(1H*))
210   FORMAT(/10X,50I2)
      RETURN
      END

      SUBROUTINE INTP(MND,NODEL,N,EV4,TI,PI)
C-----CALCULATE HEAT CAPACITY OF INTERFACE NODES
      DIMENSION NODEL(4),N(1),EV4(1)
      PARAMETER MNV=20,MNR=10
      COMMON/RMAT/TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR),ENT(MNV,MNR)
      DO 10 J=1,MND
      IE=NODEL(J)
      N1=N(IE)
      EV4IE=EV4(IE)
      CALL XVERSY(TE,ENT,MNV,N1,TI,ENI)
      CRA=ENI/TI
      PI=PI+EV4IE*CRA
10    CONTINUE
      RETURN
      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
      D=A(N,N)
      DO 100 J=1,M
  100 A(N,J)=-A(N,J)/D
      DO 150 I=1,M
      IF(N.EQ.I) GOTO 150
      DO 140 J=1,M
      IF(J.EQ.N) GOTO 140
      A(I,J)=A(I,J)+A(I,N)*A(N,J)
  140 CONTINUE
  150 A(I,N)=A(I,N)/D
  200 A(N,N)=1.0/D
      RETURN
      END

COMMENT MAIN PROGRAM CODED IN NUALGOL 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;
NET2(XL,YL,IX,IY,NR,AXIAL,XA,YA,NN,NE,MAX);
          BEGIN
          BOOLEAN ARRAY TMAX(1:NN);
```

```
          INTEGER ARRAY KTOP(1:4,1:NE),N(1:NE),NODEL(1:NN,1:4),MNODEL,
          NODINT(1:NN);
          REAL ARRAY X,Y,T,TT,P,W,EN,F,FLOW,NODCPL,DTA(1:NN),EV4(1:NE),
                     ELA(1:4,1:NE),A(1:NN,1:MAX);
          EXTERNAL FORTRAN PROCEDURE MESH2,PROG2;
          MESH2(XL,YL,IX,IY,X,Y,KTOP);
          PROG2(IX,IY,NN,NE,NR,N,KTOP,NODEL,MNODEL,X,Y,T,TT,TMAX,ELA,
           EV4,A,MAX,P,W,EN,F,FLOW,AXIAL,NODCPL,NODINT,DTA);
          END;
END

      SUBROUTINE MAT(NR)
C-----THIS ROUTINE READS MATERIAL INPUT
C-----          NR        NUMBER OF REGIONS
C-----COMMON    RMAT      REGIONAL MATERIAL DATA
C-----                    CCC        TRUE IF CONDUCTIVITY IS TAKEN AT
C-----                               MAXIMUM TEMPERATURE
C-----                    TC,C       TEMPERATURE VERSUS CONDUCTIVITY PAIRS
C-----                    TE,ENT,CR  TEMPERATURE VERSUS SPECIFIC ENTHALPY
C-----                               AND SPECIFIC ENTHALPY / TEMPERATURE
C-----                    TQ,QE      TEMPERATURE VERSUS INTERNALLY
C-----                               GENERATED HEAT
C-----                    LQ         TRUE IF HEAT IS GENERATED INTERNALLY
C-----COMMON    RGEO      REGIONAL GEOMETRICAL DATA
C-----                    ET         ELEMENT THICKNESS
C-----                    XR         SUBREGION LIMMITS
C-----                    SRDIAC     DIAGONAL COORDINATES OF SUBREGIONS
C-----PARAMETER
C-----          MNV       MAX NUMBER OF VALUE PAIRS
C-----          MNR       MAX NUMBER OF REGIONS
      DIMENSION MAT(20)
      PARAMETER MNV=20,MNR=10
      COMMON/RGEO/ELFICT(MNR),ET(MNR),SRDIAC(4,MNR)
      COMMON/RMAT/CCC(MNR),TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR),
     1    ENT(MNV,MNR),CR(MNV,MNR),TQ(MNV,MNR),QE(MNV,MNR),LQ(MNR)
      LOGICAL CCC,LQ
      LOGICAL ELFICT
      DATA CCC,LQ/MNR*.FALSE.,MNR*.FALSE./
      PRINT 95
      NRR=NR
      DO 2 I=1,NR
      IF(ELFICT(I)) NRR=NRR-1
      IF(ELFICT(I)) GOTO 2
      PRINT 100,I
C-----INPUT NAME OF REGION FOR IDENTIFICATION
      READ 200,MAT
C-----INPUT MATERIAL AND ELEMENT PROPERTIES FOR EACH REGION
      READ 1,CCC(I),NTC,NTE,NQE,ET(I)
      IF(NTC.LE.1.OR.NTE.LE.1.OR.NQE.EQ.1) GOTO 1000
      IF(NQE.GE.2) LQ(I)=.TRUE.
      MAXNTE=MAX0(MAXNTE,NTE)
      IF(ET(I).EQ..0) ET(I)=1.
      IF(.NOT.CCC(I)) PRINT 101,ET(I),MAT
      IF(CCC(I)) PRINT 102,ET(I),MAT
      NT=MAX0(NTC,NTE)
C-----INPUT TEMPERETURE CONDUCTIVITY PAIRS
      READ 1,(TC(K,I),C(K,I),K=1,NTC)
C-----INPUT TEMPERATURE SPECIFIC VOLUMETRIC HEAT PAIRS
      READ 1,(TE(K,I),ENT(K,I),K=1,NTE)
C-----INPUT TEMPERATURE INTERNALLY GENERATED HEAT PAIRS
      IF(LQ(I)) READ 1,(TQ(K,I),GE(K,I),K=1,NQE)
      DO 38 K=1,NTE
38    IF(K.NE.1) CR(K,I)=(ENT(K,I)-ENT(1,I))/(TE(K,I)-TE(1,I))
      CR(1,I)=CR(2,I)
```

```
      PRINT 110
      DO 39 K=1,NT
      PRINT 115
      IF(C(K,I).GT..00001) PRINT 120,TC(K,I),C(K,I)
      IF(CR(K,I).GT..00001) PRINT 130,TE(K,I),ENT(K,I),CR(K,I)
39    CONTINUE
      IF(NGE.EQ.0) GOTO 45
      PRINT 140,(TQ(K,I),QE(K,I),K=1,NQE)
45    CONTINUE
2     CONTINUE
      RETURN
1     FORMAT()
95    FORMAT(//1X,'MATERIAL DATA'/1X,13(1H*))
100   FORMAT(/1X,'REGION NUMBER',I3)
101   FORMAT(1H+,30X,'THICKNESS',F9.3/1X,20A4)
102   FORMAT(1H+,30X,'THICKNESS',F9.3/1X,20A4/
     1   ' CONDUCTIVITY IS KEPT CONSTANT AFTER REACHING MAXIMUM'
     2   ,' TEMPERATURE')
110   FORMAT(/1X,'TEMP',6X,'CONDUCTIVITY',13X,'TEMP',8X,'ENTALPHY',6X,
     1 'ENT/TEMP'/)
115   FORMAT(1X)
120   FORMAT(1H+,F6.0,E15.4)
130   FORMAT(1H+,33X,F6.0,E15.4,E15.4)
140   FORMAT(/' INTERNALLY GENERATED HEAT'/' TEMP',6X,'HEAT'
     1 //(1X,F6.0,E15.4))
200   FORMAT(20A4)
1000  PRINT 1200,NTC,NTE,NQE
      STOP
1200  FORMAT(///'      PROGRAM TERMINATED WHEN READING MATERIAL INPUT'
     1 //'NTC=',I3,'  NTE=',I3,'  NQE=',I3)
      END

      SUBROUTINE MAXCO(NN,TMAX,TT,T,TIME,KTIME,CON)
C-----SET CON=.FALSE. TO TERMINATE TIME INTEGRATION
      DIMENSION TMAX(NN),TT(NN),T(NN)
      LOGICAL TMAX
      LOGICAL CON
      COMMON/TOUT/II,TOUT(100),TIMMAX,DTMAX,TIMFAC,KTMAX
      IF(TIME.GE..9999*TIMMAX) CON=.FALSE.
      IF (KTIME.GT.KTMAX) PRINT 200,KTIME
      IF (KTIME.GT.KTMAX) CON=.FALSE.
200   FORMAT(//' TERMINATED AT MAXIMUM NUMBER OF TIME '
     1   ,'INCREMENTS   KTIME='I5)
      RETURN
      END

      SUBROUTINE MESH2(XL,YL,IX,IY,X,Y,KTOP)
C-----THIS SUBROUTINE COMPUTES COORDINATES AND TOPOLOGY
      DIMENSION XL(1),YL(1),X(1),Y(1),KTOP(4,1)
C-----COMPUTE X AND Y COORDINATES
      DO 5 I=1,IX
      DO 5 J=1,IY
      KK=J+IY*(I-1)
      X(KK)=XL(I)
      Y(KK)=YL(J)
5     CONTINUE
C-----COMPUTE THE TOPOLOGY MATRIX KTOP
      IX1=IX-1
      IY1=IY-1
      DO 10 I=1,IX1
      DO 10 J=1,IY1
      IE=IY1*(I-1)+J
      KTOP(1,IE)=IY*(I-1)+J
      KTOP(2,IE)=KTOP(1,IE)+1
```

```
      KTOP(3,IE)=KTOP(1,IE)+IY
      KTOP(4,IE)=KTOP(3,IE)+1
10    CONTINUE
      RETURN
      END

      SUBROUTINE MINTP(I,NODCPL,P)
C-----SUM HEAT CAPACITY OF COUPLED NODES
      PARAMETER NCP=50
      COMMON/COUPLE/NCOUPL(NCP,8),NCPLG
      DIMENSION P(1)
      DO 30 J=2,8
      NOD=NCOUPL(NODCPL,J)
      IF(NOD.EQ.0) RETURN
30    P(I)=P(I)+P(NOD)
      RETURN
      END

      SUBROUTINE MPACKV(A,X,R,MI,NN)
C-----THIS ROUTINE MULTIPLIES BANDED AND PACKED SYMMETRIC MATRIX
C-----WITH VECTOR   A * X = R
C-----A  -  MATRIX WITH DIAGONAL ELEMENTS IN RIGHT HAND SIDE COLUMN
      DIMENSION A(NN,MI),R(NN),X(NN)
      DO 3 I=1,NN
      R(I)=0.
      DO 2 J=1,MI
2     IF((I+J-MI).GT.0)
     *R(I)=R(I)+A(I,J)*X(I+J-MI)
      IF(I.EQ.NN) GO TO 3
      I1=MIN((MI-1),(NN-I))
      DO 1 J=1,I1
1     R(I)=R(I)+A(I+J,MI-J)*X(I+J)
3     CONTINUE
      RETURN
      END

      SUBROUTINE MULT(A,B,N,MAX)
C-----THIS ROUTINE MULTIPLIES THE MATRICES A AND B AND STORE THE
C-----RESULT IN A WITH CHANGED SIGNS
      DIMENSION A(MAX,MAX),B(MAX,MAX)
      COMMON/DUMMY/ETA(25),DUM2(25)
      DO 20 I=1,N
      DO 10 J=1,N
      ETA(J)=0.
      DO 10 K=1,N
10    ETA(J)=ETA(J)+A(I,K)*B(K,J)
      DO 20 J=1,N
20    A(I,J)=ETA(J)
      RETURN
      END

      SUBROUTINE NET2(XL,YL,IX,IY,NR,AXIAL,XA,YA,NN,NE,MAX)
C-----INPUT GEOMETRICAL DATA AND GENERATE LINES PARALLELL WITH AXIS
C-----AND CALCULATE NUMBER OF GENERATED NODES AND ELEMENTS
C-----
C-----    XL        COORDINATES OF X-LINES
C-----    YL        COORDINATES OF Y-LINES
C-----    XA        COORDINATES OF SPECIFIED  X-LINES
C-----    YA        COORDINATES OF SPECIFIED  Y-LINES
C-----    IX        NUMBER OF X-LINES
C-----    IY        NUMBER OF Y-LINES
C-----    NR        NUMBER OF REGIONS
C-----    AXIAL     .TRUE. IF AXI-SYMMETRIC PROBLEM
C-----    NN        NUMBER NODES
```

```
C-----     NE          NUMBER OF ELEMENTS
C-----     NX          NUMBER OF SPECIFIED X-LINES
C-----     NY          NUMBER OF SPECIFIED Y-LINES
C-----
      PARAMETER MNR=10
      COMMON/RGEO/ELFICT(MNR),ET(MNR),SRDIAC(4,MNR)
      LOGICAL AXIAL,ELFICT
      DIMENSION XL(1),YL(1),XA(1),YA(1),HEAD(20)
      PRINT 199
C*****MACHINE DEPENDENT STATEMENT  *  IN LDC*LIB.SEQ  *  REQUIRED IN MAP ELEMENT
      CALL SEQ('TASEF')
      PRINT 200
C-----INPUT TITLE OF RUN
      READ 110,HEAD
      PRINT 205,HEAD
C-----
C-----INPUT MAIN GEOMETRICAL DATA
C-----
      READ 100,AXIAL,XMAX,YMAX,XBOX,YBOX,NR,NX,NY
      PRINT 210,XMAX,YMAX
      PRINT 220,XBOX,YBOX
      IF(NR.EQ.0) NR=1
      NR1=NR-1
      EPS= XMAX/10000.
C-----
C-----INPUT SUBREGION LIMITS
C-----READ THE DIAGONAL COORDINATES FOR EACH SUBREGION
C-----
      IF(NR.EQ.1) GOTO 5
      PRINT 230
      READ 100,(ELFICT(J),(SRDIAC(I,J),I=1,4),J=2,NR)
      PRINT 240,NR1,((SRDIAC(I,J),I=1,4),ELFICT(J),J=2,NR)
5     CONTINUE
      IF(NX.EQ.0) GOTO 6
C-----
C-----INPUT SPECIFIED X - LINES
C-----
      PRINT 250
      READ 100,(XA(I),I=1,NX)
      PRINT 260,(XA(I),I=1,NX)
6     CONTINUE
      IF(NY.EQ.0) GOTO 7
C-----INPUT SPECIFIED Y - LINES
      PRINT 270
      READ 100,(YA(I),I=1,NY)
      PRINT 280,(YA(I),I=1,NY)
7     CONTINUE
C-----
C-----IF AN AXI-SYMMETRIC PROBLEM INPUT INNER RADIUS
C-----
      NX=NX+1
      IF(.NOT.AXIAL) GOTO 8
      READ 100,XA(1)
      PRINT 300,XA(1)
8     CONTINUE
C-----
C-----GENERATE X-LINES
C-----
      XL(1)=XA(NX)
      DO 15 IX=2,100
      XL(IX)=XL(IX-1)+XBOX
C-----
C-----CONTROL OF SPECIFIED X-LINES
C-----
```

```
      DO 10  I=1,NX
      IF(XL(IX-1).LT.(XA(I)-EPS)) XL(IX)=AMIN1(XL(IX),XA(I))
10    CONTINUE
C-----
C-----CONTROL OF SUBREGION LIMITS
C-----
      IF(NR.EQ.1) GOTO 12
      DO 11  I=2,NR
      IF(XL(IX-1).LT.(SRDIAC(1,I)-EPS)) XL(IX)=AMIN1(XL(IX),SRDIAC(1,I))
      IF(XL(IX-1).LT.(SRDIAC(3,I)-EPS)) XL(IX)=AMIN1(XL(IX),SRDIAC(3,I))
11    CONTINUE
12    CONTINUE
C-----
C-----CONTROL OF XMAX
C-----
      XL(IX)=AMIN1(XL(IX),XMAX)
      IF(ABS(XL(IX)-XMAX).LT.EPS) GOTO 16
15    CONTINUE
16    CONTINUE
C-----
C-----GENERATE Y-LINES
C-----
      DO 20 IY=2,100
      YL(IY)=YL(IY-1)+YBOX
C-----
C-----CONTROL OF SPECIFIED Y-LINES
C-----
      DO 17 I=1,NY
      IF(YL(IY-1).LT.(YA(I)-EPS)) YL(IY)=AMIN1(YL(IY),YA(I))
17    CONTINUE
C-----
C-----CONTROL OF SUBREGION LIMITS
C-----
      IF(NR.EQ.1) GOTO 19
      DO 18  I=2,NR
      IF(YL(IY-1).LT.(SRDIAC(2,I)-EPS)) YL(IY)=AMIN1(YL(IY),SRDIAC(2,I))
      IF(YL(IY-1).LT.(SRDIAC(4,I)-EPS)) YL(IY)=AMIN1(YL(IY),SRDIAC(4,I))
18    CONTINUE
19    CONTINUE
C-----
C-----CONTROL OF YMAX
C-----
      YL(IY)=AMIN1(YL(IY),YMAX)
      IF(ABS(YL(IY)-YMAX).LT.EPS) GOTO 21
20    CONTINUE
21    CONTINUE
C-----PRINT COORDINATES OF X - AND Y - LINES
      PRINT 310,IX,(XL(I),I=1,IX)
      PRINT 320,IY,(YL(I),I=1,IY)
C-----
      NE=(IX-1)*(IY-1)
      NN=IX*IY
      PRINT 330,NN,NE
      MAX=IY+2
100   FORMAT()
110   FORMAT(20A4)
199   FORMAT(1H1)
200   FORMAT(//' TEMPERATURE ANALYSIS OF STRUCTURES EXPOSED TO FIRE'/
     1' SOLVES NON LINEAR TRANSIENT FIELD PROBLEMS'
     2 /' ***  TWO DIMENSIONAL VERSION  ***'
     3 /' PROGRAMMED BY ULF WICKSTROM'/' LUND FEB 1979'//1X,80(1H*)///)
205   FORMAT(///' TITLE OF RUN : ',20A4)
210   FORMAT(//' GEOMETRY'/1X,8(1H*)//
     1 ' MAXIMUM COORDINATES',13X,'XMAX=',G10.3,5X,'YMAX=',G10.3)
```

```
220    FORMAT(' MAXIMUM ELEMENT LENGTH',10X,'XBOX=',G10.3,5X,'YBOX='
      1        ,G10.3)
230    FORMAT(/' SUBREGIONS')
240    FORMAT(' NUMBER OF SUBREGIONS',I4//' SUBREGION DIAGONAL LIMITS'/
      1  /4X,'XMIN',6X,'YMIN',6X,'XMAX',6X,'YMAX',6X,'FICTITIOUS AREA'//
      2      (1X,4G10.3,10X,L1))
250    FORMAT(//' COORDINATES OF SPECIFIED X - LINES')
260    FORMAT(/6G10.3)
270    FORMAT(//' COORDINATES OF SPECIFIED Y - LINES')
280    FORMAT(/6G10.3)
300    FORMAT(///' THIS IS AN AXISYMMETRIC PROBLEM'//6X,'INNER RADIUS ',
      1     'XMIN = ',G10.3)
310    FORMAT(//' NUMBER AND COORDINATES OF X - LINES'//I3,' - ',7G10.3/
      1        (6X,7G10.3))
320    FORMAT(/' NUMBER AND COORDINATES OF Y-LINES'//I3,' - ',7G10.3/
      1        (6X,7G10.3))
330    FORMAT(/' NUMBER OF NODES=',I4,10X,'NUMBER OF ELEMENTS=',I4)
       RETURN
       END

       SUBROUTINE NGROUP(X,Y)
C-----THIS ROUTINE READS AND FORMS NODE GROUP DATA
       DIMENSION X(1),Y(1)
       PARAMETER NB=10,NNB=30,NNB2=2*NNB
       COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),
      1  EPSG(NB),BETA(NB),CPG(NB),FA(NB)
       COMMON/ENRAD/E(1000)
       COMMON/ENCON/H(50),TAIR(2)
       LOGICAL FA
C-----
       PRINT 200
       READ 100,NGROUP
C-----
       DO 10 I=1,NGROUP
C-----
       READ 100,NCHECK,NUMB(I),EPSG(I),BETA(1),CPG(I)
C-----
       IF(I.NE.NCHECK) GO TO 1000
       NUMI=NUMB(I)
C-----
       READ 100,(NBOUND(I,J),J=1,NUMI)
C-----
       NOD1=NBOUND(I,1)
       DO 10 J=2,NUMI
       NOD2=NBOUND(I,J)
       BAREA(I,J)=SQRT((X(NOD1)-X(NOD2))**2+(Y(NOD1)-Y(NOD2))**2)
       NOD1=NOD2
10     CONTINUE
C
C-----PRINT INPUT DATA
C
       DO 15 I=1,NGROUP
       NUMI=NUMB(I)
       PRINT 210,I
       IF (EPSG(I).EG.0..AND.BETA(I).EQ.0.) GOTO 20
       PRINT 220,EPSG(I),BETA(I),CPG(I)
20     PRINT 230,(NBOUND(I,J),J=1,NUMI)
200    FORMAT(//' NODE GROUPS'/1X,11(1H*))
210    FORMAT(/' NODE GROUP',I3)
220    FORMAT(' EMISSIVITY=',G9.3/' CONVECTION FACTOR=',G9.3/
      1      ' CONVECTION POWER=',G9.3)
230    FORMAT(' NODES',10I5/6X,10I5)
15     CONTINUE
       GOTO 1001
```

```
1000  PRINT 240
      STOP
1001  RETURN
100   FORMAT()
240   FORMAT(//' WRONG INPUT OF NODE GROUPS')
      END

      SUBROUTINE OUTMA2(IX,IY,NN,NE,X,Y,TIME,KTIME,T,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),T(NN),TT(NN),TMAX(NN),FLOW(NN)
      PRINT 200,TITFIR,X(NN),Y(NN)
      IDUM1=1-IY
      DO 10 I=1,IX
      IDUM1=IDUM1+IY
      IDUM2=IDUM1+IY-1
      IF(IY.LE.7) PRINT 210,(J,TT(J),J=IDUM1,IDUM2)
      IF(IY.GT.7) PRINT 230,(TT(J),J=IDUM1,IDUM2)
10    CONTINUE
      PRINT 220,TIME,KTIME
200   FORMAT(/////1X,75(1HF)/2H,F/' F    MAXIMAL TEMPERATURES'/' F    ',
     1 18A4/' F    XMAX=',F8.3,10X,'YMAX=',F8.3/' F')
210   FORMAT(' F',13(I5,F5.0))
220   FORMAT(2H F/2H F/' F  MAX-TIME',F7.2,10X,'NUMBER OF '
     1        'TIME INCREMENTS',I5/2H F/2H F/2H F,75(1HF))
230   FORMAT(' F',18F7.0)
      RETURN
      END

      SUBROUTINE OUT2(IX,IY,NN,NE,X,Y,TIME,KTIME,DELTI,T,TT,TMAX,FLOW,
     1    TFIRE,NODT,AXIAL)
C-----THIS ROUTINE PRINTS NODAL TEMPERATURES AND VOID AIR TEMPERATURES
      DIMENSION X(NN),Y(NN),T(NN),TT(NN),TMAX(NN),FLOW(NN)
      LOGICAL TMAX,AXIAL,LDUM,LEN
      COMMON/ENCON/H(50),TAIR(2)
      COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100),
     1   XSYM(2),YSYM(2)
      COMMON/TOUT/II,TOUT(100),TIMMAX,DTMAX,TIMFAC,KTMAX,KUPDA
      TIME1=TIME-DELTI
      DO 5 IJ=1,NN
      IF(TMAX(IJ)) GOTO 5
C-----IF THE NODAL TEMPERATURE DECREASES SET TMAX=.TRUE. AND PRINT
C-----MAX TEMPERATURE TT
      IF(TT(IJ).GT.1.001*T(IJ))
     1   PRINT 200,IJ,TT(IJ),TIME1,DELTI
      IF(TT(IJ).GT.1.001*T(IJ)) TMAX(IJ)=.TRUE.
      TT(IJ)=AMAX1(TT(IJ),T(IJ))
5     CONTINUE
C-----IF TIME=TOUT PRINT ALL TEMPERATURES
      IF((TIME-TOUT(II)).LT.-1.E-4) GOTO 70
      PRINT 100,TIME,KTIME,TFIRE,NODT
      IF(.NOT.LEN) GOTO 30
      PRINT 300
      DO 20 I=1,NENC
20    PRINT 310,I,TAIR(I)
30    CONTINUE
      II=II+1
      IDUM1=1-IY
      LDUM=IY.LT.7
      DO 10 I=1,IX
      IDUM1=IDUM1+IY
      IDUM2=IDUM1+IY-1
```

```
      IF(LDUM) PRINT 210,(J,T(J),J=IDUM1,IDUM2)
      IF(.NOT.LDUM) PRINT 220,(T(J),J=IDUM1,IDUM2)
10    CONTINUE
70    CONTINUE
100   FORMAT(////1X,6(1H*),6H  TIME,F8.3,2X,6(1H*),'  INCREMENT',
     1 ' NUMBER',I6,
     22X,17(1H*)//'  FIRE TEMPERATURE',F7.0,2X,6(1H*)
     3,2X,'  TIME INCREMENT LIMITING NODE',I5/)
200   FORMAT(///4H NOD,I4,5X,8HMAX TEMP,F6.0,5X,'TIME',G10.4,5X,
     1 5HDELTI,G10.4/1X,70(1HM))
210   FORMAT(13(I5,F5.0))
220   FORMAT(18F7.0)
300   FORMAT(' ENCLOSURE AIR TEMPERATURE')
310   FORMAT(' VOID NUMBER',I2,'   TAIR=',F5.0/)
      RETURN
      END

      SUBROUTINE PROG2(IX,IY,NN,NE,NR,N,KTOP,NODEL,MNODEL,X,Y,T,TT,TMAX,
     1   ELA,EV4,A,MAX,P,W,EN,F,FLOW,AXIAL,NODCPL,NODINT,DTA)
C-----------------------------------------------------------------------
C-----
C-----            ***  T A S E F  ***
C-----
C-----TEMPERATURE ANALYSIS OF STRUCTURES EXPOSED TO FIRE
C-----
C-----FINITE ELEMENT PROGRAM FOR ANALYSIS OF TRANSIENT NONLINEAR
C-----HEAT TRANSFER PROBLEMS
C-----
C-----PROGRAMMED BY
C-----ULF WICKSTROM
C-----LUND INSTITUTE OF TECHNOLOGY
C-----MARCH 1979
C-----
C----------------------------------------------------------------------
C-----THIS IS THE MAIN CONTROL ROUTINE
C-----
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-----NR           NUMBER OF REGIONS
C-----N            VECTOR OF REGION NUMBERS
C-----KTOP         NODES ADJACENT TO EACH ELEMENT
C-----NODEL        ELEMENTS ADJACENT TO EACH NODE
C-----MNODEL       NUMBER ELEMENTS ADJACENT TO EACH NODE
C-----X,Y          NODE COORDINATES
C-----T            CURRENT NODAL TEMPERATURES
C-----TT           MAXIMUM NODAL TEMPERATURES
C-----TMAX         TRUE IF MAXIMUM NODAL TEMPERATURE OBTAINED
C-----ELA,EV4      DUMMY GEOMETRICAL CONSTANTS
C-----A            HEAT CONDUCTION MATRIX
C-----P            HEAT CAPACITY VECTOR
C-----W            NODAL VOLUME VECTOR
C-----EN           NODAL ENTHALPY VECTOR
C-----F            INTERNAL NODAL HEAT FLOW VECTOR
C-----FLOW         EXTERNAL NODAL HEAT FLOW VECTOR
C-----AXIAL        TRUE IF AXISYMMETRIC PROBLEM
C-----NODCPL       INDICATES COUPLED NODES
C-----NODINT       INDICATES INTERFACE NODES
C-----DTA          DUMMY VECTOR FOR CRITICAL TIME INCREMENT CALCULATION
C-----
C-----PARAMETER CONSTANTS
C-----
C-----NB           MAXIMUM NUMBER OF NODE GROUPS
```

```
C-----NNB            MAXIMUM NUMBER OF NODES IN ONE NODE GROUP
C-----NCP            MAXIMUM NUMBER OF COUPLED GROUPS OF NODES
C-----MNR            MAXIMUM NUMBER OF REGIONS
C-----MNV            MAXIMUM NUMBER OF VALUE PAIRS
C-----
C-----COMMON FIELDS
C-----
C-----COUPLE          DATA ON COUPLED NODES
C-----        NCPLG       NUMBER OF COUPLED GROUPS OF NODES
C-----        NCOUPL      MATRIX OF COUPLED NODES
C-----DIM             DIMENSIONS OF CERTAIN ARRAYS
C-----        MAXNG       MAXIMUM NUMBER OF NODE GROUPS DEFINING ONE ENCLOSURE
C-----        MAXNOD      MAXIMUM NUMBER OF NODES AROUND ONE ENCLOSURE
C-----DUMMY           DUMMY MATRICES
C-----ENCLOS          ENCLOSURE DATA
              LEN         TRUE IF STRUCTURE CONTAINS VOID OR ENCLOSURE
C-----        NENC        NUMBER OF ENCLOSURES
C-----        NENCNG      VECTOR OF NUMBER OF NODE GROUPS
C-----        IGREN       MATRIX OF NODES
C-----        NNODEN      NUMBER OF NODES SURROUNDING AVOID
C-----        XSYM        TRUE IF VOID SYMMETRICAL ARROUND X-AXIS
C-----        YSYM        TRUE IF VOID SYMMETRICAL ARROUND Y-AXIS
C-----ENCON           ENCLOSURE CONVECTION DATA
C-----        H           ARRAY OF ENCLOSURE CONVECTION VECTORS
C-----        TAIR        ENCLOSURE AIR TEMPERATURE
C-----ENRAD           ENCLOSURE RADIATION DATA
C-----        E           ARRAY OF ENCLOSURE RADIATION MATRICES
C-----FIRE            FIRE TEMPERATURE DATA
C-----        TIM,TB      TIME - FIRE TEMPERATURE PAIRS
C-----        TITFIR      FIRE IDENTIFIER
C-----FQB             PRESRIBED HEAT FLOW DATA
C-----        NFQNG       NUMBER OF NODE GROUPS DEFINING PRESCRIBED FLOW BOUNDARIES
C-----        NFQG        VECTOR OF NODE GROUPS DEFINING PRESCRIBED FLOW
C-----        TR,TC       VECTORS OF MODIFIED TEMPERATURE
C-----        BR,BC       RADIATION AND CONVECTION BOUNDARY MATRICES
C-----BNOD            DATA ON NODE GROUPS
C-----        NUMB        VECTOR OF NUMBER OF NODES IN THE NODE GROUPS
C-----        NBOUND      MATRIX OF NODE NUMBERS IN THE NODE GROUPS
C-----        BAREA       MATRIX OF DISTANCES BETWEEN NODES
C-----        EPSG        VECTOR OF EMISSIVITY OF NODE GROUPS
C-----        BETA        VECTOR OF CONVECTION FACTORS OF NODE GROUPS
C-----        CPG         VECTOR OF CONVECTION POWERS OF NODE GROUPS
C-----        FA          TRUE FOR FIRE BOUNDARY NODE GROUPS
C-----PTB             PRESCRIBED TEMPERATURE
C-----        NPTNG       NUMBER OF NODE GROUPS DEFINING PRESCRIBED TEMPERATURE
C-----        NPTG        VECTOR OF NODE GROUPS DEFINING PRESCRIBED TEMPERATURES
C-----RGEO            GEOMETRIC DATA
C-----        ELFICT      TRUE FOR FICTITIOUS ELEMENTS
C-----        ET          ELEMENT THICKNESS
C-----        SRIDIAC     SUBREGION DIAGONAL DATA
C-----RMAT            MATERIAL DATA
C-----        CCC         TRUE IF CONDUCTIVITY IS FUNCTION MAXIMUM TEMPERATURE
C-----        TC,C        TEMPERATURE - CONDUCTIVITY PAIRS
C-----        TE,ENT      TEMPERATURE - SPECIFIC VOLUMETRIC ENTHALPY PAIRS
C-----        CR          NOMINAL SPECIFIC VOLUMETRIC HEAT
C-----        TQ,QE       TEMPERATURE - INTERNALLY GENERATED HEAT PAIRS
C-----        LQ          TRUE IF INTERNAL HEAT IS GENERATED
C-----TOUT            TIME DATA
C-----        II          COUNTER
C-----        TOUT        VECTOR OF PRINT OUT TIMES
C-----        TIMMAX      MAXIMUM TIME
C-----        DTMAX       MAXIMUM TIME INCREMENT
C-----        TIMFAC      TIME INCREMENT FACTOR
C-----        KTMAX       MAXIMUM NUMBER TIME INCREMENTS
```

```
C-----          KUPDA      NUMBER OF TIME STEPS BETWEEN UPDATING CONDUCTION MATRIX
C-----UNIT           UNIT DEPENDENT CONSTANTS
C-----          SIGMA      STEFAN-BOLTZMANN CONSTANT
C-----          TABS       ABSOLUTE TEMPERATURE SHIFT
C-----          TINIT      INITIAL TEMPERATURE
C-----
      DIMENSION N(NE),KTOP(4,NE),X(NN),Y(NN),T(NN),TT(NN),TMAX(NN),
     1   ELA(4,NE),EV4(NE),A(NN,MAX),P(NN),EN(NN),F(NN),FLOW(NN),
     2   NODCPL(NN),NODINT(NN),W(NN),NODEL(4,NN),MNODEL(NN),DTA(NN)
      LOGICAL TMAX,AXIAL
C-----FORM THE VECTOR N
      CALL REG2(NN,NE,NR,N,KTOP,X,Y,NODEL,MNODEL)
C-----DEFINE COUPLED NODES
      CALL COUPLA(NODCPL,NN,NODINT)
C-----DEFINE INTERFACE NODES
      CALL INTERF(NN,NE,NR,IX,IY,KTOP,N,NODINT,NODCPL)
C-----INPUT MATERIAL DATA
      CALL MAT(NR)
C-----FORM GEOMETRICAL DUMMY CONSTANTS
      CALL GEOCO2(NN,NE,N,KTOP,X,Y,AXIAL,ELA,EV4)
C-----INPUT INITIAL DATA
      CALL AMB
C-----FORM NODE GROUPS
      CALL NGROUP(X,Y)
C-----DEFINE FIRE PRESCRIBED HEAT FLOW BOUNDARIES
      CALL FQBNDA
C-----DEFINE FIRE PRESCRIBED TEMPERATURE BOUNDARIES
      CALL PTBNDA(NODCPL)
C-----DEFINE ENCLOSURE BOUNDARIES
      CALL ENCLO1(X,Y)
C-----INPUT TIME DATA
      CALL TIME
C-----FORM NODE VOLUME VECTOR
      CALL ASSW2(NN,NE,N,KTOP,X,Y,EV4,AXIAL,W)
C-----SUMMERIZE APPROPRIATE NODE VOLUMES
      CALL COUPLB(W)
C-----CALL TIME INTEGRATION CONTROL ROUTINE
      CALL FEM2(IX,IY,NN,NE,NP,N,KTOP,X,Y,T,TT,TMAX,ELA,EV4,A,MAX,P,W,
     1          EN,F,FLOW,AXIAL,NODCPL,NODINT,NODEL,MNODEL,DTA)
      RETURN
      END

      SUBROUTINE PTBNDA(NODCPL)
C-----INPUT NODE GROUPS OF PRESCRIBED TEMPERATURE
      DIMENSION NODCPL(1)
      PARAMETER NB=10,NNB=30,NNB2=2*NNB
      COMMON/PTB/NPTNG,NPTG(NB)
      COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),TH(NB),
     1  EPSG(NB),BETA(NB),CPG(NB),FA(NB)
      LOGICAL FA,FA1
C-----READ NUMBER OF BOUNDARY NODES GROUPS
      READ 100,NPTNG
      IF(NPTNG.EQ.0) RETURN
      PRINT 200
C-----EACH PRESCRIBED TEMPERATURE BOUNDARY NODE GROUP
      DO 20 IB=1,NPTNG
C-----
      READ 100,FA1,ING1
C-----FA1 = TRUE FIRE BOUNDARY ELSE AMBIENT TEMPERATURE
C-----ING1 = NODE GROUP NUMBER
      FA(ING1)=FA1
      NUMI=NUMB(ING1)
      NPTG(IB)=ING1
      DO 10 J=1,NUMI
```

```
      NOD=NBOUND(ING1,J)
      IF (NODCPL(NOD).EQ.0) PRINT 300,NOD
      IF (NODCPL(NOD).EQ.0) STOP
10    NODCPL(NOD)=0
      IF(FA1) PRINT 210,ING1
      IF(.NOT.FA1) PRINT 220,ING1
20    CONTINUE
100   FORMAT()
210   FORMAT(' NODE GROUP',I3,'  FIRE BOUNDARY')
220   FORMAT(' NODE GROUP',I3,'  AMBIENT BOUNDARY')
200   FORMAT(//' PRESCRIBED TEMPERATURE BOUNDARY'/1X,31(1H*)//
     1 ' NODE GROUPS AND TYPES OF BOUNDARIES'/)
300   FORMAT(/' NODE',I4,' IS A SLAVE NODE'/
     1   ' SLAVE NODES CANNOT HAVE PRESCRIBED TEMPERATURE')
      RETURN
      END

      SUBROUTINE PTBNDB(T,TFIRE)
C-----SET PRESCRIBED NODAL BOUNDARY TEMPERATURE
      DIMENSION T(1)
      PARAMETER NB=10,NNB=30,NNB2=2*NNB
      COMMON/PTB/NPTNG,NPTG(NB)
      COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),TH(NB),
     1 EPSG(NB),BETA(NB),CPG(NB),FA(NB)
      COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
      LOGICAL FA
C-----
      IF(NPTNG.EQ.0) RETURN
C-----EACH PRESCRIBED TEMPERATURE BOUNDARY NODE GROUP
      DO 10 IB=1,NPTNG
      TG=TAMB
      ING1=NPTG(IB)
      IF(FA(ING1)) TG=TFIRE
      NUMI=NUMB(ING1)
      DO 10 I=1,NUMI
      NODE=NBOUND(ING1,I)
      T(NODE)=TG
10    CONTINUE
      RETURN
      END

      SUBROUTINE RADVEC(F,ETA,N,Q)
C-----THIS ROUTINE FORMS THE LOCAL ENCLOSURE SURFACE RADIATION HEAT
C-----EXCHANGE VECTOR Q=E*ETA
      DIMENSION Q(1),ETA(1),E(N,N)
      QTOT=0.
      DO 20 I=1,N
      QT=0.
      DO 10 J=1,N
10    QT=QT+E(I,J)*ETA(J)
      QTOT=QTOT+QT
20    Q(I)=QT
      RETURN
220   FORMAT(/' TOTAL RADIATION HEAT EXCHANGE',E11.3)
      END

      SUBROUTINE REG2(NN,NE,NR,N,KTOP,X,Y,NODEL,MNODEL)
C-----THIS SUBROUTINE FORMS VECTOR OF REGION NUMBERS N OF EACH ELEMENT
      DIMENSION X(NN),Y(NN),N(NE),KTOP(4,NE),NODEL(4,NN),MNODEL(NN)
      PARAMETER MNR=10
      COMMON/RGEO/ELFICT(MNR),ET(MNR),SRDIAC(4,MNR)
      LOGICAL ELFICT
      EPS=1.E-7
      DO 10 I=1,NE
```

```
      N(I)=1
      IF(NR.EQ.1) GOTO 10
      ND1=KTOP(1,I)
      ND2=KTOP(4,I)
      DO 5 J=2,NR
      IF((X(ND1)-SRDIAC(3,J)).GT.-EPS) GOTO 5
      IF((Y(ND1)-SRDIAC(4,J)).GT.-EPS) GOTO 5
      IF((X(ND2)-SRDIAC(1,J)).LT.EPS) GOTO 5
      IF((Y(ND2)-SRDIAC(2,J)).LT.EPS) GOTO 5
      N(I)=J
5     CONTINUE
10    CONTINUE
      DO 40 I=1,NN
      II=0
      DO 30 IE=1,NE
      N1=N(IE)
      IF(ELFICT(N1)) GOTO 30
      DO 20 J=1,4
      IF(KTOP(J,IE).NE.I) GOTO 20
      II=II+1
      NODEL(II,I)=IE
      IF(II.EQ.4) GOTO 30
20    CONTINUE
30    CONTINUE
      MNODEL(I)=II
40    CONTINUE
      RETURN
      END

      SUBROUTINE TIME
C-----READ TIME INTEGRATION CONTROL DATA
      COMMON/TOUT/II,TOUT(100),TIMMAX,DTMAX,TIMFAC,KTMAX,KUPDA
      PRINT 200
C-----
      READ 100,NT,TIMMAX,DTMAX,TIMFAC,KTMAX,KUPDA
C-----
      IF(DTMAX.EQ.0) DTMAX=TIMMAX
      IF(TIMFAC.EQ.0) TIMFAC=.8
      IF(KTMAX.EQ.0) KTMAX=1000
      IF(KUPDA.EQ.0) KUPDA=1
C-----
      READ 100,(TOUT(I),I=1,NT)
C-----
      PRINT 210,TIMMAX,DTMAX,TIMFAC,KTMAX,KUPDA
      PRINT 220,(TOUT(I),I=1,NT)
100   FORMAT()
220   FORMAT(' PRINT OUT TIMES',3X,8G7.2/(19X,8G7.2))
200   FORMAT(//' TIME'/' ****'/)
210   FORMAT(' MAXIMUM TIME=',G8.3/' MAXIMUM TIME INCREMENT=',G8.3/
     1   ' CRITICAL TIME INCREMENT FACTOR=',G8.3/
     2   ' MAXIMUM NUMBER OF TIME INCREMENTS=',I5/
     3   ' NUMBER OF STEPS BETWEEN UPDATING OF CONDUCTION MATRIX=',I5)
      RETURN
      END

      SUBROUTINE VIEWFC(X,Y,D,EN,VIEW,MAXNOD)
C-----THIS ROUTINE CALCULATES VIEW-FACTORS AND ENCLOSURE ZONE AREAS
C-----SYMMETRY ARROUND ANY OR BOTH AXIS ARE TAKEN INTO ACCOUNT
      DIMENSION X(1),Y(1),D(1),VIEW(MAXNOD,MAXNOD)
      PARAMETER NB=10,NNB=30,NNB2=2*NNB
      COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),TH(NB),
     1  EPSG(NB),BETA(NB),CPG(NB),FA(NB)
      COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100),
     1   XSYM(2),YSYM(2)
```

```
      LOGICAL LEN
      LOGICAL LDUM,XSYM,YSYM,DSYM
      INTEGER EN
      DO 15 I=1,MAXNOD
      DO 15 J=1,MAXNOD
15    VIEW(I,J)=0.
C-----COMPUTE VIEW-FACTORS USING HOTTEL'S CROSSED-STRING METHOD
      NENG=NENCNG(EN)
      SIGNX2=1.
      SIGNY2=1.
      IN=0
C-----EACH NODE GROUP
      DO 100 IG=1,NENG
      DSYM=.FALSE.
      IF(XSYM(EN).AND.YSYM(EN)) DSYM=.TRUE.
      I1=IGREN(EN,IG)
      NUMI=NUMB(I1)
C-----EACH ZONE
      DO 100 I=2,NUMI
      IN=IN+1
      NOD1=NBOUND(I1,I-1)
      NOD2=NBOUND(I1,I)
      X1=X(NOD1)
      X2=X(NOD2)
      Y1=Y(NOD1)
      Y2=Y(NOD2)
      D1=BAREA(I1,I)
C-----FORM THE ZONE AREA VECTOR D
      D(IN)=D1
      JN=0
C-----EACH NODE GROUP
      DO 100 JG=1,NENG
      J1=IGREN(EN,JG)
      NUMJ=NUMB(J1)
C-----EACH ZONE
      DO 100 J=2,NUMJ
      JN=JN+1
      D2=BAREA(J1,J)
      NOD3=NBOUND(J1,J-1)
      NOD4=NBOUND(J1,J)
      IF(XSYM(EN))SIGNY2=-1.
      IF(YSYM(EN))SIGNX2=-1.
50    CONTINUE
      LDUM=SIGNX2.EQ.1..AND.SIGNY2.EQ.1.
      IF(.NOT.LDUM) GOTO 80
      IF(IN.GE.JN) GOTO 100
80    CONTINUE
      X3=SIGNX2*X(NOD3)
      X4=SIGNX2*X(NOD4)
      Y3=SIGNY2*Y(NOD3)
      Y4=SIGNY2*Y(NOD4)
      D3=SQRT((X1-X3)**2+(Y1-Y3)**2)
      D4=SQRT((X2-X4)**2+(Y2-Y4)**2)
      D5=SQRT((X1-X4)**2+(Y1-Y4)**2)
      D6=SQRT((X2-X3)**2+(Y2-Y3)**2)
      DUM=ABS(D5+D6-D4-D3)/2.
C-----HOTTEL'S CROSSED-STRING METHOD
      VIEW(IN,JN)=DUM/D1+VIEW(IN,JN)
C-----TAKE ADVANTAGE OF RECIPROCITY
      IF(LDUM) VIEW(JN,IN)=DUM/D2
C-----IF SYMMETRY ARROUND AXIS GO BACK WITH CHANGED SIGNS OF COORDINATES
      LDUM=(SIGNX2+SIGNY2).LT.-1.99
      IF(LDUM) SIGNY2=1.
      IF(LDUM) GOTO 50
```

```
C-----
      LDUM=DSYM.AND.SIGNX2.LT.0.
      IF(LDUM) SIGNX2=1.
      IF(LDUM) SIGNY2=-1.
      IF(LDUM) GOTO 50
C-----
      LDUM=(SIGNX2*SIGNY2).LT.0.
      IF(LDUM) SIGNX2=1.
      IF(LDUM) SIGNY2=1.
      IF(LDUM) GOTO 50
100   CONTINUE
      RETURN
      END

      SUBROUTINE XVERSY(X,Y,N,M,XS,YS)
C-----FIND YS AS FUNCTION OF XS BY LINEAR INTERPLLATION
C-----IN TABEL 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
      PRINT 1
      PRINT 2,XS,(X(I,M),Y(I,M),I=1,5)
      PRINT 3,M
      STOP
1     FORMAT(///1X,'INPUT VALUE TO XVERSY OUT OF RANGE'/)
2     FORMAT(1X,'INPUT X',E10.4/1X'X-Y VALUE PAIRS'//1X,5(2G8.3,5X))
3     FORMAT(/1X,'CURVE NUMBER',I4)
11    RETURN
      END
```

APPENDIX C - Example input

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

Example I

```
SQUARE PLATE
F,1,1,.125,.125,,,,
0
UNIT MATERIAL DATA
F,2,2,,,,
,1,10000,1,
,,10000,10000,
.00001,1000.,,,
1
1,17,,1,1,
9 18 27 36 45 54 63 72 81 80 79 78 77 76 75 74 73
1
F,1
0,
NOVOID
20,1,1,1,,,,
.05 .10 .15 .20 .25 .30 .35 .40 .45 .50 .55 .60 .65 .70 .75 .80 .85 .90 .95 1.
DUMMY TEMPERATURE
2
,,1000,,
```

Example II

```
I BEAM EMBEDDED IN CONCRETE
F,.14,.1,.05,.03,6,4,4
F,,,.05,.01
F,,.01,.003,.09
F,,.09,.05,.1
T,.05,,.14,.01
T,.05,.09,.14,.1
.015,.03,.07,.1,
.02,.035,.05,.08
12
33 34,,,,,,
25 26,,,,,,
17 18,,,,,,
1 2 9 10,,,,,,
3 11,,,,,,
4 12,,,,,,
5 13,,,,,,
6 14,,,,,,
7 15 8 16,,,,,,
23 24,,,,,,
31 32,,,,,,
39 40,,,,,,
BETONG
T,7,7,,
24.5,1.78,115,1.28,243,1.17,401,1.17,643,.92,895,.85,1500,.85,
,,100,55600,115,91000,200,129400,600,397200,1000,696700,1500,1000000.,
STAL
F,3,7,,
,60,800,27,2000,27
,,200.,217000,400,466000,600,758000,700,927000,800,1192000,1200,1766000
STAL
F,3,7,,
,60,800,27,2000,27
,,200.,217000,400,466000,600,758000,700,927000,800,1192000,1200,1766000
STAL
F,3,7,,
,60,800,27,2000,27
,,200.,217000,400,466000,600,758000,700,927000,800,1192000,1200,1766000
25.0001,25.,,,,
4
1,6,.6,.99,1.33
1,9,17,25,33,34,
2,4,.8,.99,1.33
34 42 50 58
3 6 .6 2.2,1.25
8 16 24 32 40 39
4 4 .8 2.2,1.25
39 47 55 63
4
T,1
T,2
F,3
F,4
0
NOVOID
15,1.5,,,,,,
.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0,1.1,1.2,1.3,1.4,1.5,
HE100B≠FIRE
9
,25,.05,525,.1,620,.3,725,.6,940,1,980.,1.025,600,1.2,475,1.5001,360,,,,
```

Example III

```
BOX GIRDER EMBEDDED IN CONCRETE
F,.14,.15,1.,1.,5,4,6
F,,,.0875,.004
F,.0345,.004,.0375,.112
F,,.112,.0375,.12
T,,.004,.0345,.112
.02,.05,.07,.11
.015,.029,.044,.056,.085,.13
14
1 2,,,,,,,,,
12 13,,,,,,,,,
23 24 34 35,,,,,,,,,
45 46,,,,,,,,,
56 57,,,,,,,,,
68 67,,,,,,,,,
25 36,,,,,,,,,
26 37,,,,,,,,,
27 38,,,,,,,,,
28 39,,,,,,,,,
29 40,,,,,,,,,
30 31 41 42,,,,,,,,,
19 20,,,,,,,,,
8 9,,,,,,,,,
BETONG
T,7,7,,
24.5,1.78,115,1.28,243,1.17,401,1.17,643,.92,895,.85,1500,.85,
,,100,55600,115,91000,200,129400,600,397200,1000,696700,1500,1000000.,
STAL
F,3,7,,
,60,800,27,2000,27
,,200.,217000,400,466000,600,758000,700,927000,800,1192000,1200,1766000
STAL
F,3,7,,
,60,800,27,2000,27
,,200.,217000,400,466000,600,758000,700,927000,800,1192000,1200,1766000
STAL
F,3,7,,
,60,800,27,2000,27
,,200.,217000,400,466000,600,758000,700,927000,800,1192000,1200,1766000
25,25.0001,,
4
1,7,.6,.99,1.33
1,12,23,34,45,56,67,
2,3,.8,.99,1.33
67,78,89
3,9,.8,2.20,1.25
11,22,33,44,55,66,77,88,99,
4,11,.6,1.6,1.33
2,13,24,25,26,27,28,29,30,19,8
3
T,1
T,2
F,3
0,     PTBNDA
VOID
1
F,T,4,,,,,,
15,1.5,,,,,,,,
.1,.2,.3,.4,.5,.6,.7,.8,.9,1.,1.1,1.2,1.3,1.4,1.5,
HSQ-FIRE
12
,25,.05,525,.1,625,.2,765,.3,900,.5,925,.7,980,1,1015,1.05,675,1.1,600,1.2,500.
1.501,350.,
```

PERSTORPS TRYCKERI