

LUND UNIVERSITY

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

Wickström, Ulf

1979

Link to publication

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

General rights

Unless other specific re-use rights are stated the following general rights apply: Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights. • Users may download and print one copy of any publication from the public portal for the purpose of private study

or research.

- You may not further distribute the material or use it for any profit-making activity or commercial gain
 You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

Take down policy

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

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

ACKNOWLEDGEMENT

.

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

.

7.	REFERENCES	56
APPENDIX	A - USER'S MANUAL	59
APPENDIX	B - PROGRAM LISTING OF TASEF-2	76
APPENDIX	C - EXAMPLE INPUT	110

•

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 <u>et al</u>. [2,3] the finite element programs FIRES-T [4] and later FIRES-T3 [5] were developed for analyzing thermal response of structures exposed to fire. An implicit backward difference time integration scheme is used in these programs. Computation therefore often becomes unnecessarily expensive, and materials with latent heat for instance humid concrete - cannot be analyzed accurately.

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

2. HEAT TRANSFER ANALYSIS

2.1 Basic Equations

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

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

and the Fourier law

k

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

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

$$= k I$$
 (

2.3

where k is thermal conductivity, and I the identity matrix. The gradient operator ⊽ is defined as



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

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

Specific volumetric enthalpy is by definition

$$e = \int_{T_{O}}^{T} c_{\rho} dT + \sum_{i} l_{i}$$
(2.6)

where T_0 is a reference temperature, usually zero, c specific heat, ρ density, and ℓ_1 latent volumetric heat due to phase changes at various temperature levels. The time derivative of

 $\dot{\mathbf{e}} = \mathbf{c}\rho\dot{\mathbf{T}} \tag{2.7}$

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

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

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

$$\mathbf{e} = \overline{\mathbf{c}\rho}\mathbf{T} \tag{2.9}$$

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_{ρ} 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_{ρ} 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-





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

$$\partial V = \partial V_{m} + \partial V_{m} \tag{2.10}$$

Temperature on the boundary ∂V_{τ} of a body is specified as

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

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

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

where <u>n</u> is the outward normal to the surface. Specified heat flow on ∂V_{α} therefore is

$$\hat{q}_{n} = -\underline{n}^{T} \underline{k} \underline{\nabla} T \qquad (2.13)$$

where $\hat{\boldsymbol{q}}_n$ 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

$$\hat{q}_{n}^{c} = \beta \left(T_{s} - T_{g}\right)^{\gamma}$$
(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

$$\hat{\mathbf{q}}_{n}^{r} = \varepsilon_{r} \sigma \left(\bar{\mathbf{T}}_{s}^{4} - \bar{\mathbf{T}}_{g}^{4} \right)$$
(2.15)

where σ is the Stefan-Boltzmann constant, and \overline{T}_{g} and \overline{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,

where $\boldsymbol{\varepsilon}_{q}$ is appropriate gas or flame emissivity.

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

$$\hat{q}_{n} = \hat{q}_{n}^{c} + \hat{q}_{n}^{r}$$
 (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

$$T = \sum_{i} N_{i}(x, y, z) T_{i}(t) = \underline{N} \underline{T}$$
(3.1)

The time differentiation of the temperature is

 $\dot{T} = \underline{N} \ \dot{\underline{T}}$ (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]:

$$\int_{V} \mathbf{v} \left(-\underline{\nabla}^{\mathrm{T}} \underline{\mathbf{k}} \underline{\nabla} \mathbf{T} + \frac{\partial}{\partial t} (\overline{\mathbf{c}\rho} \mathbf{T}) - \mathbf{Q} \right) d\mathbf{V} = 0$$
(3.3)

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

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

where <u>n</u> 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.

 $v_{i} = N_{i} \tag{3.5}$

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

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

$$\underline{\mathbf{F}}_{\mathrm{T}} + \frac{\partial}{\partial t}(\underline{\mathbf{E}}) = \underline{\mathbf{F}}_{\mathrm{Q}} + \underline{\mathbf{F}}_{\mathrm{q}}$$
(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

$$\underline{F}_{\mathrm{T}} = \underline{K} \ \underline{\mathrm{T}} \tag{3.8}$$

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

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

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

where

$$\underline{F} = \underline{F}_{O} + \underline{F}_{\alpha} \tag{3.10}$$

The nodal enthalpy vector is

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

where <u>C</u> is the nominal heat capacity matrix. This expression is substituted into Equation (3.9)

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

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

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

`

-14-

or

where

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

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

$$\underline{K} = \sum_{m} \underline{K}^{m}$$
(3.15)

$$\underline{C} = \sum_{m} \underline{C}^{m}$$
(3.16)

$$\underline{F} = \sum_{m} \underline{F}_{Q}^{m} + \sum_{\partial m} \underline{F}_{Q}^{\partial m}$$
(3.17)

where

$$K_{ij}^{m} = \int_{V} (\underline{v} \ N_{i})^{T} \underline{k} (\underline{v} \ N_{j}) dV$$
(3.18)

$$C_{ij}^{m} = \int_{V_{ij}} N_{ij} \overline{c\rho} N_{j} dV$$
(3.19)

$$F_{Qi}^{m} = \int_{V} N_{i} Q dV$$
(3.20)

and

$$F_{qi}^{\partial m} = \int_{\partial V} N_{i} \underline{n}^{T} \underline{k} \nabla T dS \qquad (3.21)$$

 \boldsymbol{v}^m and $\vartheta\boldsymbol{v}^m$ 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

$$\xi = (x - x_0)/a$$
 (3.22)

anđ

$$\eta = (y - y_0)/b$$
 (3.23)

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

$$N_{i} = (1 + \xi\xi_{i})(1 + \eta\eta_{i})/4 \qquad (3.24)$$

where i takes values from 1 to 4, and

$$\underline{\nabla} \mathbf{N}_{i} = \begin{pmatrix} \frac{\partial \mathbf{N}_{i}}{\partial \mathbf{x}} \\ \frac{\partial \mathbf{N}_{i}}{\partial \mathbf{y}} \end{bmatrix} = \begin{pmatrix} \frac{1}{a} \frac{\partial \mathbf{N}_{i}}{\partial \xi} \\ \frac{1}{b} \frac{\partial \mathbf{N}_{i}}{\partial \eta} \end{bmatrix} = \frac{1}{4} \begin{pmatrix} \frac{\xi_{i}}{a} (1+\eta\eta_{i}) \\ \frac{\eta_{i}}{b} (1+\xi\xi_{i}) \end{pmatrix} \quad (3.25)$$

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 $\underline{K}^{\mathrm{m}}$ is, after evaluation of simple integrals,

$$\underline{K}^{m} = \frac{1}{3} \frac{kd}{ab} \begin{bmatrix} a^{2}+b^{2} & -b^{2}+\frac{a^{2}}{2} & -\frac{b^{2}}{2} - \frac{a^{2}}{2} & \frac{b^{2}}{2} - a^{2} \\ a^{2}+b^{2} & \frac{b^{2}}{2} - a^{2} & -\frac{b^{2}}{2} - \frac{a^{2}}{2} \\ a^{2}+b^{2} & -b^{2}+\frac{a^{2}}{2} \\ a^{2}+b^{2} & -b^{2}+\frac{a^{2}}{2} \\ sym. & a^{2}+b^{2} \end{bmatrix}$$
(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:

$$C_{\underline{i}\underline{i}}^{m} = \overline{c\rho}^{m}(T_{\underline{i}}) W_{\underline{i}\underline{i}}^{m}$$
(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:

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

where

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

defines the global diagonal volume matrix.

3.3.3 Internally Generated Heat

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

$$F_{Qi} = \sum_{m} Q_{i}^{m} W_{ii}^{m}$$
(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):

$$F_{qi}^{\partial m} = -\int_{V_{qi}} N_{i} \hat{q}_{n} dS \qquad (3.31)$$

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

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



Figure 3.2. Heat flow to a boundary element Equations (2.14 and 2.15) are then used to yield

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

where \overline{T}_{gi} and \overline{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

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

where i and j are nodes adjacent to a boundary element $\Im m$ and

$$B_{rii}^{\partial m} = \frac{1}{3} sd\epsilon_r \sigma \qquad (3.35)$$

$$B_{rij}^{\partial m} = \frac{1}{2} B_{rii}^{\partial m}$$
(3.36)

$$B_{cii}^{\partial m} = \frac{1}{3} sd\beta$$
 (3.37)

$$B_{cij}^{\partial m} = \frac{1}{2} B_{cii}^{\partial m}$$
(3.38)

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

and

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

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

$$\underline{\mathbf{F}}_{\mathbf{q}} = \underline{\mathbf{B}}_{\mathbf{r}} \ \underline{\mathbf{T}}_{\mathbf{r}} + \underline{\mathbf{B}}_{\mathbf{c}} \ \underline{\mathbf{T}}_{\mathbf{c}} \tag{3.41}$$

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

$$B_{rij} = \sum_{\partial m} B_{rij}^{\partial m}$$
(3.42)

and

$$B_{cij} = \sum_{\partial m} B_{cij}^{\partial m}$$
(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

$$\Theta \underbrace{\mathbf{E}}_{t+\Delta t}^{*} \underbrace{\mathbf{E}}_{t+\Delta t} + (1-\Theta) \underbrace{\mathbf{E}}_{t}^{*} \underbrace{\mathbf{E}}_{t} + (\underbrace{\mathbf{E}}_{t+\Delta t} - \underbrace{\mathbf{E}}_{t}) / \Delta t =$$

$$= \Theta \underbrace{\mathbf{E}}_{t+\Delta t} + (1-\Theta) \underbrace{\mathbf{E}}_{t}$$
(3.44)

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

 $0 \le \Theta \le 1 \tag{3.45}$

If different values are assigned to 0 various time integration schemes are defined. Thus for 0 = 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 $\Delta t_{\rm cr}$. The value of this critical time increment depends on element size, material properties, and boundary conditions. If <u>C</u> 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

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

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

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

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

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

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



Temperature T

Figure 3.3. Translation of specific volumetric enthalpy into temperature

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

$$\frac{T_{i,t+\Delta t}^{j+1} - T_{i,t+\Delta t}^{j}}{T_{i,t+\Delta t}^{j+1} + T_{i,t+\Delta t}^{j}} < \delta/2 \qquad (3.51)$$

 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 <u>C</u> is assumed to be time independent, Equation (3.12) is

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

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

$$K_{\text{Fij}} = \frac{dF_{i}}{dT_{i}}$$
(3.53)

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

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

where

$$\overline{\underline{K}} = \underline{K} - \underline{K}_{\mathrm{F}} \tag{3.55}$$

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

$$\underline{\mathbf{T}} = \mathbf{e}^{-\lambda t} \underline{\mathbf{\phi}} \tag{3.56}$$

where ϕ is a vector independent of time t. Multiply by the inverse of the diagonal matrix <u>C</u>:

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

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

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

arises. Equation (3.58) is an n:th order equation where n is the number of temperature degrees of freedom in the system. There are n solutions of eigenvalues (thermal frequences) λ_1 , $\lambda_2 \dots \lambda_n$ with corresponding eigenvectors (thermal modes) ϕ_1 , $\phi_2 \dots \phi_n$.

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

$$\Delta t_{\rm cr} = \frac{2}{\lambda_{\rm max}}$$
(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

$$\lambda \leq \max(a_{ii} + \sum |a_{ij}|) \quad j \neq i \quad (3.60)$$

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.

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

Thus the maximum eigenvalue of Equation (3.58) is

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

and an upper limit to the critical time increment is

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

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

In Equation (3.63) it is implicit that for nodes for which the ratio of heat capacity to thermal conductance to adjacent nodes is small, the critical time increment will be very small. When possible without jeopardizing accuracy, thermal resistance between such nodes can then be neglected; the temperature of these nodes is set to the same value. All terms for these coupled nodes are combined. The resulting denominator in Equation (3.63) is reduced while C_{ii} is increased; the resulting critical time step for this region is thus substantially increased. When calculating temperature in fire-exposed steel structures, for example, the difference in temperature between opposite sides of steel sheets will in most cases be negligible. Corresponding nodes can therefore be coupled without 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 (<u>Temperature Analysis of</u> <u>Structures Exposed to Fire - Two Dimensional Version</u>) 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 timedependent fire temperature. A fire temperature history is specified by a number of points on a time-temperature curve. Temperature between these points is obtained by linear interpolation. If the time-temperature relation specified for the ISO 834 standard fire resistance test [11] is assumed, the fire temperature T_f may instead be calculated as

$$T_f = T_0 + 345 \log_{10}(430t + 1)$$
 (4.1a)

-28-

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:

$$\begin{array}{ccccccc} 625 & {}^{\rm OC/h} & {\rm if } t_{\rm u} \leq 0.5 & {\rm h} \\ 250 (3-t_{\rm u}) & {}^{\rm O}C/h & {\rm if } 0.5 \leq t_{\rm u} \leq 2 \, {\rm h} \\ 250 & {}^{\rm O}C/h & {\rm if } t_{\rm u} > 2 & {\rm h} \end{array} \tag{4.1b}$$

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_0 is suddenly subjected to an environment of uniform gas temperature T_g . Heat transfer q at the boundary to the body surface is

 $q = h(T_q - T_s)$

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/ ℓ^2
Bi = h ℓ/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/l^2 and 0.0200 a/l^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^OC 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 nume-rically calculated dimensionless temperature θ at center of square plate exposed to heat transfer from surrounding gas, Bi=1

Dimension- less time Fo	Exact solu- tion	Coarse mesh 4 elements ∆t=0.0667 at	/ ²	Fine mesh 16 elements ∆t=0.0200 at	t/l 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	



(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 :

$$e^{W}(T) = c^{W}T$$
 for $T < T_{1}$
 $e^{W}(T_{2}) = e^{W}(T_{1}) + \frac{1}{2}c^{W}(T_{2} - T_{1}) + a^{W}$ for $T = T_{2}$
 $e^{W}(T) = e^{W}(T_{2})$ 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

-38-

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]

$$Nu = C(Pr Gr)^{m}$$
 (a)

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

$$Nu = \frac{q^{C}d}{k(T_{s}-T_{q})}$$
(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

$$Gr = \frac{g \frac{1}{T_{b}} (T_{s} - T_{g}) d^{3}}{v^{2}}$$
(c)

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

$$d = \frac{A}{F}$$

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]:

$$k = 13.75 \cdot 10^{-5} T_{b}^{0.92} [W/mK]$$
(d)
$$v = 1.13 \cdot 10^{-9} T_{b}^{5/3} [m^{2}/s]$$
(e)

-40-

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

is

$$q^{c} = 13.75 \cdot 10^{-5} (5.48 \cdot 10^{18})^{m} C \frac{d^{3m-1}}{(\frac{13}{3}m - 0.92)} (T_{s} - T_{g})^{m+1}$$
 (f)
derived.

The characteristic length d of the beams considered is approximately 0.15 m. Substitution of Equation (e) into Equation (c) then gives $Gr < 8 \cdot 10^6$ for expected levels of temperature. Laminar convection is therefore expected on the cool side, and C and m are 0.54 and 0.25, respectively [21]. Equation (f) then yields

$$q^{c} = 3.59 d^{-0.25} T_{b}^{-0.16} (T_{s} - T_{q})^{1.25} [W/m^{2}]$$

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/m²K^{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

$$q^{c} = 36 T_{b}^{-0.52} (T_{s} - T_{q})^{1.33} [W/m^{2}]$$

and if T_b is assumed to be 1000K, β and γ are identified as 1.0 W/m²K^{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 timetemperature 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



Time h

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





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/m² 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 dre 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 threedimensional 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", <u>Acta Polytechnica Scandinavica</u>,
 Ci 48, Stockholm, 1968.
- [2] Wilson, E.L. and Nickell, R.E., "Applications of the Finite Element Method to Heat Conduction Analysis", <u>Nuclear Engineering and Design</u>, Vol. 4, 1966, pp. 276-286.
- [3] Wilson, E.L. and Farhoomand, I., "Non-Linear Heat Transfer Analysis of Axisymmetric Solids", Structural Engineering Laboratory, <u>Report No. 71-6</u>, 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", <u>Report No. UCB FRG 74-1</u>, 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)", <u>Report No.</u> <u>UCB FRG 77-15</u>, 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., <u>Handbook of Heat</u> Transfer, McGraw-Hill, New York, 1973.
- [7] Pettersson, O., Magnusson, S.E. and Thor, J., "Fire Engineering Design of Steel Structures", <u>Bulletin 52</u>, Division of Structural Mechanics and Concrete Construction, Lund Institute of Technology, Lund, 1976.

- [8] Zienkiewicz, O.C., <u>The Finite Element Method in</u> <u>Structural and Continuum Mechanics</u>, 3rd edition, McGraw-Hill, London, 1977.
- [9] Strang, G. and Fix, J.F., <u>An Analysis of the Finite Element Method</u>, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.
- [10] Wickström, U., "A Numerical Procedure for Calculating Temperature in Hollow Structures Exposed to Fire", <u>Report No. UCB FRG 77-9</u>, 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", <u>Report No. 79:2</u>, 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., <u>Egenspänningar i varmvalsade stålpro-</u> <u>filer (Residual Stresses in Hot-rolled Steel Sections</u>), Institutionen för Brobyggnad, Kungliga Tekniska Högskolan, Stockholm, 1967 (In Swedish).
- [14] Anderberg, Y., "Fire-Exposed Hyperstatic Concrete Structures - An Experimental and Theoretical Study", <u>Bulletin 55</u>, 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)", <u>Cement och</u> <u>Betong 1972:1</u>, Stockholm, 1972 (in Swedish).

- [16] Ahlgren, L., "Fuktfixering i porösa byggnadsmaterial (Moisture Fixation in Porous Building Materials)", <u>Report 36</u>, Division of Building Technology, Lund Institute of Technology, Lund, 1972 (Swedish).
- [17] Schmidt, E. and Eckert, E., "Über die Richtungsverteilung der Wärmestrahlung von Oberflächen", <u>Forsch.Ing.-Wes. 6</u>, 1935.
- [18] Thor, J., "Strålningspåverkan på oisolerade eller undertaksisolerade stålkonstruktioner vid brand", <u>Bulletin 29</u>, Division of Structural Mechanics and Concrete Construction, Lund Institute of Technology, Lund, 1972 (in Swedish).
- [19] Holman, J.P., <u>Heat Transfer</u>, 4th edition, McGraw-Hill, 1976.
- [20] Hadvig, S., <u>Stökiometri og Stofvaerdier (Stochiometry</u> 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", <u>International Journal of Heat and Mass Transfer</u>, Vol. 15, pp. 755, 1972.
- [22] Paulsen, O., <u>Om varmeovergang i Brandprøveovne</u> (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.
- Define boundary node groups for prescribed heat transfer, and form heat transfer matrices BR and BC.
- 7. Define boundary node groups for prescribed temperature.
- 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 Tⁱ⁺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
18	ENCON2	ENCLO2			ENCON ENCLOS DUMMY
19	ENRAD1	ENCLO1	VIEWFC, INVER, MULT,		BNOD, ENCLOS, ENRAD.
			ETRANS		
20	ENRAD2	ENCLO2	RADVEC		ENCLOS ENRAD UNTT.
					DUMMY
21	ETRANS	ENRAD1			
22	FEM2	PROG2	INIT, ASSP2, XVERSY, MINTP, ASSA2, MPACKV, FQBNDB, FQGEN, ENCLO2, COUPLB, ACOUPL, CTEMP,		RMAT,FIRE
			HTEMP, PTBNDB, COUPLC, OUT2, MAXCO, DTIME, OUTMA2		
23	FQBNDA	PROG2	BRBCA	NFQNG;FA1,ING1	UNIT, BNOD
24	FQBNDB	FEM2	BRBCB		FQB, BNOD, UNIT

-62-

TABLE A:2. (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,NOE,	RMAT, RGEO
				ET;TC,C;TE,ENT;TO,OE	
35	MAXCO	FEM2			TOUT
36	MINTP	FEM2			COUPLE
37	MESH2	MAIN2			
38	MPACKV	FEM2			
39	MULT	ENRAD1			DUMMY
40	NET 2	MAIN2		TITLE; AXIAL, XMAX, YMAX, XBOX, YBOX, NR, NX, NY; ELFICT_SEDIAC: XA_YA	RGEO
41	NGROUP	PROG2		NGROUP; NCHECK, NUMB, EPSG, BETA, GAMMA; NBOUND	BNOD, ENRAD, ENCON
42	OUT 2	FEM2			TOUT
43	OUTMA2	FEM2			FIRE
44	PROG2	MAIN 2	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 51	VIEWFC XVERSY	ENRAD1 ASSP2,COND2, FEM2,HTEMP, INTP			BNOD, ENCLOS



-64-

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 ycoordinates, 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≥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≤10)
- NX number of specified x-lines for refining of element mesh
- NY number of specified y-lines for refining of element mesh
- 2. Subregion specifications

(NR-1) cards

ELFICT, SRDIAC(4)

- ELFICT .TRUE. if the subregion is a void or cut out of the base structure. Otherwise .FALSE.
- SRDIAC(4) minimum x- and y-coordinates, and maximum x- and y-coordinates of subregion
- 3. Specified x-lines

If NX=0 omit this card

XA(NX)

XA(NX) coordinates of specified x-lines

TABLE A:3 (Cont.)

4. Specified y-lines

If NY=0 omit this card

YA(NY)

YA(NY) coordinates of specified y-lines

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

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

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

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

4.5 (XA(1))

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

C. COUPLED NODES

1. Number of groups of coupled nodes

NCPLG

2. Each group

NCOUPL(8)

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





(a) Structure to be analyzed



(b) Generated mesh with node numbers

Figure A.2. Example of mesh generation

у 🛔

6

D. MATERIAL DATA

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

1. Each material

a. Identification card

MAT

MAT arbitrary test to be written on output list

b. Material description

CCC,NTC,NTE,NQE,ET

CCC .TRUE. if conductivity is constant in cooling phase

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

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

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

ET thickness of elements; default 1
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 AMEIENT 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=5.67.10⁻⁸
- TABS shift for absolute temperature; default 273

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

NGROUP

 $(NGROUP \le 10)$

If NGROUP equal zero omit next card

2. Each node group

a. Properties

NCHECK, NUMB, EPSG, BETA, GAMMA

- NCHECK node group number in sequential order starting from 1
- NUMB number of nodes of a group (NUMB≤30)

EPSG emissivity

- BETA convection factor
- GAMMA convection power
- b. Node numbers

NBOUND (NUMB)

G. PRESCRIBED HEAT FLUX BOUNDARY

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

 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

2. Each boundary node group

FA1, ING1

Same as G.2

I. VOIDS

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

1. Control card

CONTRO

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

2. Number of voids

NENC

 $(NENC \le 2)$

3. For each void

a. XSYN, YSYM, IGREN(4)

XSYM .TRUE. if void is symmetric around the x-axis

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

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

J. TIME

1. Time control card

NT, TIMMAX, DTMAX, TIMFAC, KTMAX, KUPDA

- NT number of specified times for printing out of temperature
- TIMMAX maximum time of analysis
- DTMAX maximum time increment; default TIMMAX
- TIMFAC time increment factor; default 0.8
- KTMAX maximum number of time steps; default 1000
- KUPDA number of time steps between updating of heat conduction matrix; default 1
- 2. Specified times for temperature output

TOUT (NT)

K. FIRE TEMPERATURE HISTORY

1. Control card

TITFIR

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

2. Fire temperature

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

a. Number of points

NFP

 $(NFP \le 50)$

b. Data card

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

3. Go back to K1 and begin analysis with new fire, or terminate program by inserting blank card.

APPENDIX B - Listing of TASEF-2

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

((SUBROUTINE ACOUPL(A.DTA.NN,MAX) ADD SLAVE NODE QUANTITIES TO MASTER NODE QUANTITIES FOR EACH COUPLING NODE GROUP DIMENSION A(NN.MAX),DTA(NN) PARAMETER NCP=50 COMMON (COUPLE/ NCOUPL(NCP.8),NCPLG
<i>c</i>	
(
	IF (NCPLG.EQ.0) RETURN
	DO 50 TETINCELO
	AMX=U.
	DO 40 J=1/8
	TE (NOD.), EQ. 0.) GOTO 50
	NODK=NCOUPL(1,K)
	IF(NODK.EQ.0) GOTO 40
	NDUM=NODJ-NODK
	IF(NDUM) 30+30+60
60	CONTINUE
00	
	AMX=AMX+2,*A(NODJ+MAX=NDUM)
30	CONTINUE
C	
40	(XAM+LGON)A+XKA=XMA
5 N	DTA(MNOD) = DTA(MNOD) + AMX + A(MNOD + MAY)
20	
	END
C	-READ UNIT DEPENDENT CONSTANTS COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4 DATA SIGM,TAP/5.77F-8,273./ PRINT 200
C	
	READ IUU+IINIT+IAME+SIGMA+IAES
Ç	-
	IF(SIGMA.LT.1.E-20) SIGMA=SIGM
	IF(TABS.LT.1.E-20) TABS=TAB
	PRINT 210, TINIT, TAMB, SIGMA, TARS
	TAMB4∓(TAMB+TABS)**4
	RETURN
- 00	
100	
200	FORMAT(/// INTIAL DATA*/IX+12(1H*))
210	FORMAT(/* INITIAL TEMPERATURE=**G9+3/
	1 ' AMBIENT TEMPERATURE=',G9.3/
	2 ' STEFAN-BOLTZMANN CONSTANT='+G9.3/
	3 + ARSOLUTE TEMPERATURE SHIFT=+.69.3)
	CURPONITING ACCADINING N.KTOP.Y.Y.FLA.T.TT.TMAY.AYTAL MAY.AY
C	- JUDAOU LINE ADDAE NUMMERINGA THE ALADAL LEAT FAMPATALALIMMATATA -
L	-THIS SUBROUTINE COMPUTES THE GENERAL MEAT SOWHOUTION MATRIX A
C	- NN NUMBER OF NODES
C	- NE NUMBER OF FLEMENTS
c	- N ELEMENT REGION NUMBER
C	- ELA GEOMETRIC DUMMY VECTOR
~ ~	
c	TT TEMPERATURE HISTORY VECTOR
	THE THE ANT AND A THE AND
C	~
C	
с	- A GLOBAL HEAT CONDUCTION MATPLY
с с	- A GLOBAL HEAT CONDUCTION MATPLY PARAMETER MNP=10

DIMENSION N(NE),X(NN),Y(NN),T(NN),TT(NN),TMAX(NN),KTOP(4,NE), 54 LOGICAL TMAX,ELFICT,AXIAL 65 1 66 DO 5 I=1.WN DO 5 J=1.MAX 67 68 A(I,J)=ū. 5 69 70 71 DO 10 I=1,NE N1=N(1)IF(ELFICT(N1)) GOTO 10 72 73 K1=KTOP(1,1) KZ=KTOP(2,1) 74 K3=KTOP(3,1) 75 K4=KTOP(4,1) CALL COND2(T(K1),T(K2),T(K3),T(K4),TT(K1),TT(K2),TT(K3),TT(K4), 76 77 (K1) CMAX(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) 78 1 79 8 D A(K3, MAX)=A(K3, MAX)+C*ELA(1,1) 81 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) 82 83 84 A(K3, MAX-K3+K2)=A(K3, MAX-K3+K2)+C*ELA(4,1) 85 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) 86 27 38 89 10 CONTINUE 90 91 RETURN END 92 93 SUBROUTINE ASSPE(NN, N, X, Y, T, TT, TMAX, EV4, NODEL, MNODEL, P, W, 1 NODINT, AXIAL) C----FORM HEAT CAPACITY MATRIX 94 95 -FORM HENI CAPACITY MAINIX DIMENSION N(1),X(1),T(1),TT(1),TMAX(1),EV4(4,1),NODEL(4,NN), 1 MNODEL(1),P(1),K(1),NODINT(1) PARAMETER MAV=20,MNR=10 96 97 1 98 COMMON/EMAT/CCC(MAR),TC(MAV,MAR),C(MAV,MAR),TE(MAV,MAR), ENT(MAV,MAR),CR(MAV,MAR),TQ(MAV,MAR),QE(MAV,MAR),LQ(MAR) 99 100 1 LOGICAL CCC,LQ LOGICAL AXIAL,TMAX 101 102 103 С DO 30 I=1.NN 104 NODINI=NODINT(I) 105 IF(NODINI.LT.C) GOTO 30 106 TI =T(I) IF(NODINI.GT.J) GOTO ZC 127 108 P(I)=Û. 109 110 CALL INTP(MNODEL(I),NODEL(1,I),N,EV4,TI,P(I)) GOTO 30 111 2.0 112 CONTINUE CALL XVERSY(TE,ENT,MNV,NODINI,TI,ENI) 113 114 CRA=ENI/TI 115 P(I)=+(I)*CRA CONTINUE 30 116 117 RETURN 118 119 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=10 123. 121 122 123 COMMON/RGEO/ELFICT(MNR) 124 LOGICAL AXIAL, ELFICT DO 5 I=1,NN w(I)=0. 125 126 127 5

```
06 10 1=1,ac
12:
129
                    N1=N(1)
170
131
                    IF(EEFICT(N1)) SCTO 15
                    K1=KTUP(1,1)
K2=KTUP(2,1)
174
133
                    KS=KTOP(3,I)
                    κ4≈κτορ(4,I)
ψ(κ1)=ψ(κ1)+Εν4(I)
174
136
                    ₩(K_)=#(K2)+EV4(1)
                    w(K3)=w(K3)+EV4(I)
w(K4)=w(K4)+Ev4(I)
137
130
135
           ۴J
                    CUNTINUL
140
                    PETURA
141
                    E D
14
                    SUBROUTINE DEIFE(FIN)
14 5
           C----THIS ROUTINE FORMS VECTORS FOR FIRE BOUNDARY
C----THIS REPERATORS RELATION
CUMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
144
145
146
                    CUMMGR/FIRE/TIM(SU),TU(SC),TITFIR
147
                    LUGICAL FIN
INTEGER TITFIR(15), BLANK(15), 150(2)
145
145
150
                    DATA LLANK/19*4H /, ISO/4HISO ,4Ho34 /
151
152
                     FIN=.FALSE.
            C----IF - BLANK - TERMINATE RUN
C----IF - BLANK - TERMINATE RUN
C----IF - 1SU 034 - STANDARD FIRE ASSUMED
155
154
            C----FLEL INPUT TIME TLMPERATURE PAIKS
155
                    REAU 110,TITELE
DU 10 I=1,12
155
157
15 3
                    IF(TITFIR(I).NE.BLANK(I)) GOTO 20
            1υ
159
                    FINH.IRUE.
160
                    FLTURIE
141
            2...
                    CONTINUE
                    PRINT 200
PRINT 210,TITFIK
D0 30 1=1,7
162
145
164
                    IF(TITE,P(I).NE.ISO(I)) GUTO 50
            36
165
           C----ISO STANDARD CURVE
160
                                                                                  ,
167
                    DT=.05
                    DI=+CJ
TI=∪+
DU 40 I=1+2∪
162
164
                    UC -1. 1-1,20
TIM(1)=T1
TUC1)=T1NLT+1325.-450.*EXP(-.2*TI)-270.*EAP(-1.7*TI)-
170
171
                   1 625.*LXP(-14.*T1)
172
173
                    T1=TI+DT
            ن 4
174
                    T1=T1-DT
                    DT=5./27.
                     71=11+07
170
                    DU 45 1=21,50
TIM(1)=TI
177
172
                    T6(1)=TINIT+1321.-430.*EXP(-.2*TI)-270.*EXP(-1.7*TI)-
179
                   1 645.*cXP(+19.*TI)
TI=TI+DI
180
191
           45
182
                    PETURN
183
            5 J
                    CONTINUL
            C-----ARSITKARY FIRES
C----INPUT NUMBER OF TIME TEMPEPATURE PAIRS
PEAD TEU,NEP
154
155
180
                    IF(NEP.EQ.C) FIN =. FRUE.
197
            IF (FLA) RETURY
C----INPUT TIME TEMPERATURE PAIFS
REAU 100,(T1*(I),TB(I),I=1,NFP)
198
159
196
                    PRINT 220, (TIN(1), TU(1), I=1, NEP)
191
```

160 FORMATO 192 193 110 FORMAT(20A4) FOPMAT(//' FIRE BOUNDARY TEMPERATURE'/1X,25(1H*)) 200 210 220 194 FORMAT(/1x,2CA4) FORMAT(/' FIRE BOUNDARY TIME - TEMPERATURE INPUT PAIRS'// 1 ' TIME',5x,'TEMPERATURE'//(G1J.3,G1D.3)) 195 196 197 RETURN 198 199 END 200 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) 201 202 203 204 BR(1,1)=C. BR(1,2)=.33333333*EAR(ING1,2) NUM1=NUMI-1 IF(NUM1.EQ.1) GOTO 2C 205 206 207 DO 10 I=2,NUM1 BR(I,1)=.16666667*BAR(ING1,I) 208 209 BR(1,2)=.3333333*(BAR(ING1,1)+BAR(ING1,1+1)) 210 10 20 CONTINUE 211 CONTINUE 212 BR(NUMI,1)=.16666667*BAR(ING1,NUMI) 213 BR(NUMI,2)=.33333333*BAR(ING1,NUMI) 214 215 С D0 3C I=1,NUMI D0 3C J=1,2 216 217 218 BC(I,J)=BET*BR(I,J) BR(I,J)=EPSIG*BR(I,J) 219 35 CONTINUE 22ĉ 221 RETURN 222 END 223 SUBROUTINE BRBCB(BR, BC, TR, TC, TRD, TCD, NUMI, DTA, NN, MAX, FLOW, TG, 224 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=1C,NNE=3G,NNE=2*NNB
COMMON/BNOD/NUME(NB),NBOUND(NB,NNB),BAREA(NB,NNB),
1 EPS6(NR).HETA(NB).CPG(NE).FA(NB) 1 225 T,ING1) 226 227 228 229 **2**30 231 232 233 1 EPSG(NB), BETA(NB), CPG(NB), FA(NB) LOGICAL FA 234 2,35 236 Ć C----FIRST NODE 237 ¢ 238 NODE=NBOUND(ING1,1) TR2=TR(1) 239 240 TC2=TC(1) 241 TR3=TR(2) 242 TC3=TC(2) TRD2=TRD(1) 243 244 TCD2=TCD(1) 245 TRD3=TRD(2)246 247 TCD 3=TCD (2) BRZ=BR(1,2) 248 BC2=BC(1,2) 249 BR3=BR(2,1) BC3=BC(2,1) FLW=BR2*TR2+BC2*TC2 250 251 FLW=FLW+BR3*TR3 252 FLW=FLW+BC3*TC3 DA=BR2*TRD2+BC2*TCD2 253 254 DA=DA+9R3+TRD3 255

255	DA=DA+BC3+TCD3
257	DTA(NODE) = DTA(NODE) + .5 + 0 A
2.27	
258	FLUW(NUDE)=FLUW(NUDE)+FLW
259	C
260	CINTERMEDIATE NODES
261	
201	с.
202	
263	IF(NUM1.EQ.1)GOTO 20
254	DO 10 I=2.NUM1
265	NODERNBOUND (ING1 I)
200	
200	1 K 1 = 1 K 2
257	TR 2 = TR 5
265	TP3=TR(I+1)
240	TC1=TC2
275	102-102
270	
271	TCS=TC(I+1)
272	TRD 1=TRD 2
273	TKD2=TRD3
37/	TEN3=TON(1+1)
274	
215	
276	TCD2=TCD3
277	TCD3=TCD(I+1)
278	9R1=BR3
210	
614	
280	RK2=RK(T+1*1)
281	BC1=BC3
282	B(2=B((I_2))
283	P(J = P(J + 1, 1))
200	
284	
285	FLW=FLW+ER1*TR1+BR3*TR3
286	FLW=FLW+BC1*TC1+BC3*TC3
287	
201	
	USTORIORIORUTIORUTIORUTIORUTIORUTI
268	
288	DA=DA+BC1+TCD1+BC3+TCD3
288 289 290	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA
288 289 290 291	DA = DA + BC 1 + TCD 1 + BC 3 + TCD 3 DTA (NODE) = DTA (NODE) + • 5 * DA 1C FLOW (NODE) = FLOW (NODE) + FLW
288 289 290 291	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 20 CONTINUE
2 8 8 2 8 9 2 9 C 2 9 1 2 9 2 2 9 2	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOw(NODE)=FLOW(NODE)+FLW 2D CONTINUE
288 289 290 291 292 293	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 10 FLOW(NODE)=FLOW(NODE)+FLW 20 CONTINUE C
288 289 290 291 292 293 293 294	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C C+-LAST NODE
288 289 290 291 292 293 294 295	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2D CONTINUE C CLAST NODE C
288 289 291 292 293 294 295 295	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 10 FLOW(NODE)=FLOW(NODE)+FLW 20 CONTINUE C C C C NODE=NBOUND(ING1,NUMI)
288 289 291 292 294 295 295 296 296	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5+DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2
288 289 291 292 293 294 295 296 297	DA=DA+BC1+TCD1+BE3+TCD3 DTA(NODE)=DTA(NODE)+.5+DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C CLAST NODE C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR3=TR3
288 289 291 292 293 294 295 295 295 298	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TR1=TR2 TR2=TR3
288 289 291 292 293 295 295 295 297 297 299	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2.
289 289 291 292 293 294 295 296 297 298 298 298 298 298	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C CLAST NODE C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2- TC2=TC3
289 289 291 292 294 295 295 295 295 295 295 295 295 295 295	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2- TC2=TC3 TRD1=TRD2
289 289 291 295 295 295 295 295 295 299 299 299 300 299 300 201 201	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C++-LAST NODE C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2- TC2=TC3 TRD1=TRD2 TRD2=TRD3
289 289 291 292 293 295 295 295 295 295 295 295 295 295 295	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2 TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2
2889 2991 2992 2995 2995 2995 2995 2995 299	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2- TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2
2890 2991 2993 2995 2995 2995 2996 2996 2996 3003 3003 3003	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C 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
289 2991 2992 2995 2995 2995 2995 2995 299	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2 TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2 TCD2=TCD3 BK1=ER3
2890 2991 2993 2995 2995 2995 2995 2995 2990 3001 2990 3004 3004 3004 3004 3004	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2 TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2 TCD2=TCD3 BK1=BR3 BK1=BR3 BK1=BR3 BK2=BR(NUMI.2)
2890 2991 2993 2995 2995 2995 2995 2995 3001 2990 3005 3005 3005 3005	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2 TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2 TCD2=TCD3 BR1=BR3 BR1=BR3 BR1=BR3 BR1=BR3 BR1=BR3 BR1=BR3
288 290 291 292 293 294 295 299 299 299 299 302 299 302 302 302 302 302 302 302 302	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5+DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2- TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2 TCD2=TCD3 BK1=BR3 BK1=BR3 BK1=BC3 BC1=
2890 2991 2995 2995 2995 2995 2995 2996 2996 2996	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5+DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2 TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2 TCD2=TCD3 BR1=BC3 BC2=BC(NUMI,2) BC1=BC3 BC2=BC(NUMI,2)
2890 2992 2992 2995 2995 2995 2995 2995 29	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2 TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2 TCD2=TCD3 BK1=CR3 BK1=CR3 BK1=CR3 BK2=BR(NUMI,2) BC1=BC3 BC2=BC(NUMI,2) FLW=BR2+TR2+BC2+TC2
28901 2991 2995 2995 2995 2995 2995 2995 299	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2- TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2 TCD2=TCD3 BK1=BR3 BK1=BR3 BK2=BR(NUMI,2) BC1=BC3 BC2=BC(NUMI,2) FLW=C2+TC2 FLW=C2+TC2 BC1=C2+TC2+TC2 BC1=C2+TC2+TC2 BC1=C2+TC2+TC2+TC2 BC1=C2+TC2+TC2+TC2+TC2 FLW=C2+TC2+TC2+TC2+TC2 FLW=C2+TC2+TC2+TC2+TC2+TC2+TC2+TC2+TC2+TC2+T
28901 2991 2993 2995 2995 2996 2996 2996 2996 2996 3003 0056 30056 30056 3005 3005 3005 3	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2 TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2 TCD2=TCD3 BK1=RR3 BK1=RR3 BK1=RR3 BK2=BR(NUMI,2) BC1=BC3 BC2=BC(NUMI,2) FLW=BR2+TR2+BC2+TC2 FLW=FR+TRD2+BC2+TC2 FLW=FR+TRD2+BC2+TC2
289 2991 2992 2995 2995 2995 2995 2995 299	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5+DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2 TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2 TCD2=TCD3 BK1=ER3 BK1=ER3 BK1=ER3 BK2=BR(NUMI,2) BC1=BC3 BC2=BC(NUMI,2) FLW=ER2+TR2+BC2+TC2 FLW=FLW+BK1+TR1+BC1+TC1 DA=BR2+TRD2+BC2+TCD2 ACDA+BC1+TRD1+BC1+TC1
2890 2991 2995 2995 2995 2995 2995 2995 2995	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C CLAST NODE C NODE=N3OUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2- TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2 TCD2=TCD3 BF1=BR3 BF1=BR3 BF1=BR3 BC2=BC(NUMI,2) FLW=EC*TR2+BC2*TC2 FLW=FLW+BR1*TR1+BC1*TC1 DA=BR2*TRD2+BC2*TC2 DA=DA+BR1*TRD1+BC1*TC1 DA=DA+BR1*TRD1+BC1*TC1
2890 2991 2992 2995 2995 2995 2995 2995 2995	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C CLAST NODE C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2 TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2 TCD2=TCD3 BK1=BR3 BK1=BR3 BK1=BR3 BK2=BR(NUMI,2) BC1=BC3 BC2=BC(NUMI,2) FLW=BR2*TR2+BC2*TC2 FLW=FLW+BR1*TR1+BC1*TC1 DA=BR2*TRD2+BC2*TC2 DA=DA+BR1*TR01+BC1*TC1 DTA(NODE)=DTA(NODE)+.5*DA
28901 2991 2993 2995 2995 2995 2995 2995 2995 2995	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2- TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2 TCD2=TCD3 BK1=BR3 BK1=BR3 BK2=BR(NUMI,2) BC1=BC3 BC2=BC(NUMI,2) FLW=BR2*TR2+BC2*TC2 FLW=FLW+BR1*TR1+BC1*TC1 DA=BR2*TRD2+BC2*TC2 DA=DA+BR1*TRD1+BC1*TC1 DA=BR2*TRD2+BC2*TC2 DA=DA+BR1*TRD1+BC1*TC1 DA=BR2*TRD2+BC2*TC2 DA=DA+BR1*TRD1+BC1*TC1 DA=DA+BR1*TC1+BC1*TC1 DA=DA+BR1*TC1+BC1*TC1 DA=DA+BR1*TC1+BC1*TC1 DA=DA+BR1*TC1+BC1*TC1 DA=DA+BR1*TC1+BC1*TC1 DA=DA+BR1*TC1+BC1*TC1 DA=DA+BR1*TC1+BC1*TC1 DA=DA+BR1*TC1+BC1*TC1 DA=DA+BR1*TC1+BC1*TC1+BC1*TC1+BC1*TC1 DA=DA+BR1*TC1+BC1*TC1+BC1*TC1+BC1
2890 2991 2992 2995 2995 2995 2995 2995 2995	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C NODE=NBOUND(ING1,NUM1) TR1=TR2 TR2=TR3 TC1=TC2 TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2 TCD2=TCD3 BK1=BR3 BK1=BR3 BK2=BR(NUM1,2) BC1=BC3 BC2=BC(NUM1,2) FLW=BR2*TR2+BC2+TC2 FLW=FLW+BR1*TR1+BC1*TC1 DA=BR2*TR2+BC2+TC2 DA=DA+BR1*TR01+BC1*TC01 DTA(NODE)=FLOW(NODE)+FLW RETURN
2890 2991 2992 2995 2995 2995 2995 2995 2995	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5+DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C CLAST NODE C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2: TC2=TC3 TRD1=TRD2 TRD2=TRD3 TCD1=TCD2 TCD2=TCD3 BK1=ER3 BK1=ER3 BK2=BR(NUMI,2) BC1=BC3 BC2=BC(NUMI,2) FLW=ER2+TR2+BC2+TC2 FLW=FLW+BK1+TR1+BC1+TC1 DA=BR2+TRD2+BC2+TC2 DA=DA+BR1+TRD1+BC1+TCD1 DTA(NODE)=FLOW(NODE)+.5+DA FLOW (NODE)=FLOW(NODE)+FLW RETURN END
2890122995 299122995 299522995 299522995 299522995 299522995 3000567896 300567896 311234567 311234567	DA=DA+BC1+TCD1+BC3+TCD3 DTA(NODE)=DTA(NODE)+.5*DA 1C 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 BK1=BR3 BK1=BR3 BK1=BC3 BC2=BC(NUMI,2) FLW=BR2+TR2+BC2+TC2 FLW=FLW+BR1+TR1+BC1*TC1 DA=BR2+TRD2+BC2+TC2 DA=DA+BR1+TRD1+BC1*TCD1 DTA(NODE)=FLOW(NODE)+FLW RETURN END
28901 2991 2992 2995 2995 2995 2995 2995 299	DA = DA + BC + TCD 1 + BC + TCD 3 DTA (NODE) = DTA (NODE) + .5 + DA 1C FLOW (NODE) = FLOW (NODE) + FLW 2C CONTINUE C C C LAST NODE C NODE = NBOUND (ING 1, NUMI) TR 1 = TR 2 TR 2 = TR 3 TC 1 = TC 2 TC 2 = TC 3 TR 0 = TR D 3 TC 0 = TC D 2 TC 0 2 = TC D 3 BK 1 = BR 3 BK 2 = BC (NUMI, 2) BC 1 = BC 3 SC 2 = BC (NUMI, 2) FLW = BR 2 + TR 2 + BC 2 + TC 2 FLW = FLW + BK 1 + TR 1 + BC 1 + TC 1 DA = BR 2 + TR D 1 + BC 1 + TC 1 DTA (NODE) = DTA (NODE) + .5 + DA FLOW (NODE) = FLOW (NODE) + FLW RETURN END
2890122995 299122995 299522995 299500000000000000000000	DA = DA + BC1 + TCD1 + BC3 + TCD3 DTA (NODE) = DTA (NODE) + .5 + DA 1C FLOW (NODE) = FLOW (NODE) + FLW 2C CONTINUE C C LAST NODE C NODE = NSOUND (ING1, NUMI) TR 1 = TR2 TR2 = TR3 TC1 = TC2 TC2 = TC3 TR01 = TC2 TC2 = TC3 TR01 = TC2 TC2 = TC3 BC1 = BC3 BC2 = BC (NUMI, 2) BC1 = BC3 BC2 = BC (NUMI, 2) FLW = BR2 + TR2 + BC2 + TC2 FLW = FLW + BC1 + TC1 + TC1 DA = BR2 + TR2 + BC2 + TC2 FLW = FLW + BC1 + TC1 + TC1 DA = BR2 + TR02 + BC2 + TC2 DA = DA + BR1 + TR01 + BC1 + TC1 DTA (NODE) = FLOW (NODE) + FLW RETURN END SUBROUTINE CONDECT1, T2, T3, T4, TT1, TT2, TT3, TT4, TMAX1, TMAX2, TMAX3,
28901229956789001234567890012333333311234567890012333333333333112345578900112331153115575	DA=DA+BC1+TCD1+BC2+TCD3 DTA(N0DE)=DTA(N0DE)+.5*DA 1C FLOW(N0DE)=FLOW(N0DE)+FLW 2C CONTINUE C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2- TC2=TC3 TRD1=TRD2 TRD2=TR03 ICD1=TCD2 TC0=TCD3 BK1=BR3 SK2=BR(NUMI,2) ELW=BR2+TR2+BC2+TC2 FLW=ER2*TR2+BC2+TC2 FLW=FLW+BR1+TR1+BC1+TC1 DA=DR2+TRD2+BC2+TC2 DA=DA+BR1+TR0+BC2+TC2 DA=DA+R1+TR0+BC2+TC2 DA=DA+R1+TR0+BC2+TC2 DA=DA+R1+TR0+BC2+TC2 DA=DA+R1+TR0+BC2+TC2 DA=DA+R1+TR0+BC2+TC2 DA=DA+R1+TR0+BC2+TC2 DA=DA+R1+TR0+BC2+TC2 DA=DA+R1+TR0+BC2+TC2 DA=DA+R1+TR0+BC2+TC2 DA=DA+R1+TR0+BC2+TC2 DA=DA+R1+TR0+BC2+TC2 DA=DA+R1+TR0+BC2+TC2+TC2 DA=DA+R1+TR0+BC2+TC2+TC2 DA=DA+R1+TR0+BC2+TC2+TC2 DA=DA+R1+TR0+BC2+TC2+TC2 DA=DA+R1+TR0+BC2+TC2+TC2+TC2+TC2+TC2+TC2+TC2+TC2+TC2+T
28901 2992 2992 2995 2995 2995 2995 2995 299	DA=DA+BC1+TCD1+BC3+TCD3 DTA(N0DE)=DTA(N0DE)+.5*DA 1C FLOW(NODE)=FLOW(NODE)+FLW 2C CONTINUE C C ONTINUE C NODE=NBOUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2 TC2=TC3 TRD1=TCD2 TCD2=TCD3 BK1=BR3 SK2=BR(NUMI,2) BC1=BC3 SC2=BC(NUMI,2) FLW=FLW+BK1*TR1+BC1*TC1 DA=BR2*TRD2+BC2*TC2 DA=DA+BR1*TR01+BC1*TC1 DA=BR2*TRD2+BC2*TC2 DA=DA+BR1*TR01+BC1*TC1 DA=BR2*TRD2+BC2*TC2 DA=DA+BR1*TR01+BC1*TC1 DA=BR2*TRD2+BC2*TC2 DA=DA+BR1*TR01+BC1*TC1 DTA(NODE)=TLOW(NODE)+FLW RETURN END SUBROUTINE COND2(T1,T2,T3,T4,TT1,TT2,TT3,TT4,TMAX1,TMAX2,TMAX3, 1
2890122995 299122995 299522995 299522995 29950000000000	DA=DA+BC1+TCD1+BC2+TCD3 DTA(N0DE)=DTA(N0DE)+.5*DA 1C FLOW(N0DE)=FLOW(N0DE)+FLW 2C CONTINUE C C NODE=N3OUND(ING1,NUMI) TR1=TR2 TR2=TR3 TC1=TC2: TC2=TC3 TRD1=TRD2 TCD2=TCD3 BK1=RR3 BK1=RR3 BK2=BR(NUMI,2) EC D = BR2*TR24BC2*TC2 FLW=FLW+BR1+TR1+BC1*TC1 DA=BR2*TR24BC2*TC2 FLW=FLW+BR1+TR1+BC1*TC1 DTA(N0DE)=DTA(N0DE)+.5*DA FLOW(N0DE)=FLOW(N0DE)+FLW RETURN END SUBROUTINE COND2(T1,T2,T3,T4,TT1,T12,TT3,TT4,TMAX1,TMAX2,TMAX3, 1

```
320
             C----GET FLEMENT CONDUCTIVITY
                      LOGICAL TMAX1, TMAX2, TMAX3, TMAX4, AXIAL
321
                      PARAMETER M.V=20,MNR=10
COMMON/RMAT/CCC(MNR),TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR),
ENT(MNV,MNR),CR(MNV,MNR),TQ(MNV,MNR),QE(MNV,MNR),LQ(MNR)
322
323
374
                     1
                      LOGICAL CCC,LQ
IF(CCC(N1)) TM=(TT1+TT2+TT3+TT4)/4
325
32£
                      IF(.NOT.CCC(N1)) TM=(T1+T2+T15+T14)/4.
IF(.NOT.CCC(N1)) TM=(T1+T2+T3+T4)/4.
CALL XVERSY(TC,C,MNV,N1,TM,CE)
327
325
329
                      RETURN
336
                      END
331
332
                      SUBROUTINE COUPLA (NODCPL, NN, NODINT)
             C----READ COUPLED NODES AND FORM CONTROL VECTOR NCOUPL
PARAMETER NCP=50
333
334
             PARAMETER NUMESO

COMMON/COUPLE/ NCOUPL(NCP,E),NCPLG

DIMENSION NODCPL(NN),NODINT(NN)

C-----NODCPL = -1 NODE UNCOUPLED

C----NODCPL = 0 SLAVE NODE

C----NCDCPL = NCOUPL(1,J) MASTER NODE
335
336
337
338
339
340
             .
c -----
                      00 5 I=1,NN
NODCPL(I)=-1
341
             5
342
                      PRINT 200
343
             с----
344
                     READ 100,NCPLG
345
             (-----
346
347
                    IF(NCPLG.EQ.0) GOTO 35
                     PRINT 205
DO 20 I≈1,NCPLG
348
349
35C
             c -----
                     READ 100, (NEOUPL(I,J), J=1,8)
351
             C----
352
                     II=NCOUPL(1,1)
353
354
             C----COUPLED NODES ARE ALWAYS INTERFACE NODES
                      NODCPL(II)=I
D0 10 J=2,3
355
356
357
                      JI=NCOUPL(1,J)
358
                       IF(II.EQ.G) GOTO 10
                      NODINT(II)=0
359
                      NODCPL(II)=C
350
                      լ ։ = ၂
361
                      PRINT 210, (NCOUPL (I,J), J=1,JJ)
CONTINUE
362
             10
363
             20
364
                      RETURN
365
366
                      CONTINUE
PRINT 220
             36
367
             160
                      FORMAT()
368
                      FORMAT(//* COUPLED NODES'/1X,13(1H*)/)
FORMAT(/* MASTER SLAVES'/)
FORMAT(I3,6X,7I3)
369
             200
376
             205
371
372
             220
                      FORMAT(* NO COUPLED NODES*)
373
                      RETURN
                      END
374
375
             SUBROUTINE COUPLB(V)
C----ADD SLAVE NODE QUANTITIES TO MASTER NODE QUANTITIES
C----FCR EACH COUPLING NODE GROUP
375
377
378
                      DIMENSION V(1)
PARAMETER NCP=50
379
380
                     COMMON /COUPLE/ NCOUPL(NCP,8),NCPLG
381
             c -----
382
                      IF(NCPLG.EQ.0)RETURN
383
```

```
384
                   DO 3C I=1,NCPLG
                   MNOD=NCOUPL(I,1)
385
                   VYN=V(MNOD)
386
387
                   00 10 J=2,8
388
                   NOD=NCOUPL(I,J)
                  IF(NOD.EQ.D) GOTO 20
VMN=VMN+V(NOD)
389
390
391
                  CONTINUE
           10
392
           c -----
393
           2 ټ
                  CONTINUE
                   V(MNOD)=VMN
396
           <mark>ئ 3</mark>
395
                   CONTINUE
396
                   RETURN
397
                  END
398
399
                  SUBROUTINE COUPLC(T)
400
           C-----UPDATE SLAVE NODE TEMPERATURE
DIMENSION T(1)
                   PARAMETER NCP=50
4.92
                   COMMON/COUPLE/NCOUPL(NCP,8),NCPLG
403
                   IF(NCFLG.EG.D)RETURN
DO 2C I=1,NCPLG
MNOD=NCOUPL(I,1)
404
405
407
                   TMNOD=T(MNOD)
                   DO 1C J=2,8
NOD=NCOUPL(I,J)
IF(NOD.EQ.C) GOTO 20
408
429
410
411
           10
                   T(NOD)=TMNOD
412
           20
                   CONTINUE
                   RETURN
413
414
                   END
415
           SUBROUTINE CTEMP(NODE,T,P,EN,FLOW,F,DELTI,NODEL,MNODEL,N,EV4,NDC)
C----CALCULATE TEMPERATURE OF INTERFACE NODES
PARAMETER NCP=50
416
417
418
                   COMMON/COUPLE/NCOUPL(NCP,8),NCPLG
419
                   DIMENSION P(1), EN(1), FLOW(1), F(1), NODEL(4,1), MNODEL(1), N(1), EV4(1)
420
                   DATA EPS/.CC5/
421
                   EN(NODE)=EN(NODE)+(FLOW(NODE)-F(NODE))*DELTI
422
423
                   ENI=EN(NODE)
                   PI=P(NODE)
424
                   DO 40 J=1,5
425
426
                   T=ENI/PI
                   P1=0.
CALL INTP(MNODEL(NODE),NODEL(1,NODE),N,EV4,T,P1)
427
428
                   IF (NDC.LT.C) GOTO 30
429
                   DO 20 I=2,8
430
                   NOD=NCOUPL(NDC,I)
IF (NGD.EG.O) GOTO 30
CALL INTP(MNODEL(NOD),NODEL(1,NOD),N,EV4,T,P1)
431
432
433
           2 C
434
           30
                   ERR=(PI-P1)/(PI+P1)
                   IF(ABS(ERR).LT.EPS) GOTO 50
435
                   PI=(P1+PI)/2.
436
437
                   CONTINUE
           4 č
                  CONTINUE
PPINT 20C,NODE,T,ERR
FORMAT(/' CONVERGENCE NOT ACHIEVED FOR NODE',IS,' TEMP=',G9.3,
43ĉ
639
           200
                       .
                          ERR= , 69.2)
440
                  1
                   T=ENI/P1
441
           50
                  P(NODE)=P1
442
                   RETURN
443
444
                  ΕND
445
                  SUBROUTINE DTIME(NN, P, DTA, MAX, NODINT, NODCPL, TIME, DELTI, NODT)
445
           C----THIS ROUTINE CALCULATES TIME INCREMENT
447
```

```
443
                       DIMENSION P(NN), NODINT(NN), NODCPL(NN), DTA(NN)
449
                        CCMMON/TOUT/II, TOUT(100), TIMMAX, DTMAX, TIMFAC, KTMAX, KUPDA
                       DELTI=DIMAX
DG 1C I=1,NN
IF (NODCPL(I).EG.D) GOTO 10
IF (NODINT(I).LT.D) GOTO 10
450
451
452
453
454
                        DUM=TIMFAC*P(I)/DTA(I)
                       DELTI=AMIN1(DELTI,DUM)
455
                        IF (DELTI.EQ.DUM) NODT=I
456
457
              12
                        CONTINUE
453
                        DUM=TOUT(II)-TIME
                        IF(DUM.GT.C) DELTI=AMIN1(DELTI,DUM)
459
                        RETURN
460
461
                       END
462
                       SUBROUTINE ENCLOT(X,Y)
463
             C----THIS IS THE FIRST OF A SET OF ROUTINES FOR CALCULATION OF
C----THIS IS THE FIRST OF A SET OF ROUTINES FOR CALCULATION OF
C-----INBEDDED IN SOLIDS. THE SAME SURFACE NODES AS FOR THE SOLID STATE
C-----FINITE ELEMENT ANALYSIS ARE EMPLOYED.
464
465
466
447
              C----
468
              C----PROGRAMMED BY
469
              C-----ULF WICKSTROM
C----JUNE 1977
470
471
472
              C----REVICED FEB 1979
              с----
473
              C----MAJOR VARIABLES,
474
                                          → NUMBER OF NODES IN THE NODE GROUPS
                                NUMB
              C---+
475
                               NUMBE - NUMBER OF NUMBERS IN THE NODE GROUPS

BAREA - AREA DETWEEN NODES. THIRD DIMENSION ASSUMED UNITY

NENC - NUMBER OF VOIDS

NENCING - NUMBER OF NODE GROUPS SURROUNDING EACH VOID
476
              (----
477
              Č----
              c----
47ĉ
              ç-----
479
                               IGREN - NODE GROUP NUMBERS SURROUNDING EACH VOID
NODEN - NUMBER OF NODES SURROUNDING EACH VOID
INODEN - ALL NODE NUMBERS IN ALL VOIDS
480
              .
.----
              c----
481
              c----
48z
              c-----
                                           - NODE RADIATION MATRICES
423
                                Ε
                                          - EMISSIVITY OF NODE GROUP ZONES
- CONVECTION VECTORS
- CONVECTION FACTORS OF THE NODE GROUP ZONES
              C----
                                EPSG
484
              485
                                н
              C----
                                BETA
486
                                           - VOID AIR TEMPERATURE
              c -----
487
                                TAIR
488
              C----
                       PARAMETER NB=1C,NNB=3C,NNB2=2*NNB
COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(N3,NNB),
489
490
                           EPSG(NB), BETA(NB), CPG(NB), FA(NB)
491
                      1
                        COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100),

xsym(2),ysym(2)

common/enrad/e(1900)
49ž
493
                      1
494
                       COMMON/ENCON/H(5),TAIR(2)
COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
COMMON/DIM/MAXNG,MAXNOD
495
496
697
                       DATA MAXNG, MAXNOD/4,25/
498
                        LOGICAL LEN
499
                       LOGICAL XSYM, YSYM, SYM, LDUM
INTEGER EN
500
5.01
             INIEGER EN

PRINT 18G

C----READ CONTRÓL CARD

C----IF NO VOIDS IN STRUCTURE RETURN

READ 90, CONTRO

IF (CONTRO. EQ. (HVOID) GOTO 10

00101 100
502
503
504
5 3 5
$06
507
                       PRINT 190
508
                       RETURN
509
                       CONTINUE
              10
510
                        TAIR=TINIT
511
                        LEN= TRUE.
```

,

ş

C----READ AND ESTABLISH NODE GROUP DATA C----READ NUMBER OF VOIDS READ 100,NENC PRINT 200,NENC IND=C C----EACH VOID DO 152 EN=1.NENC C-----READ SYMMETRI PROPERTIES AND NODE GROUPS THAT DEFINES THE C----VOID READ 10:.XSYM(EN),YSYM(EN),(IGREN(EN,J),J=1,MAXNG) SYM=XSYM(EN).DR.YSYM(EN) IND=IND+1 I1=IGREN(EN,1) INODEN(IND)=NBOUND(I1,1) NODE1=INODEN(IND) 527 C---+EACH NODE GROUPE DO 20 IG=1,MAXNG I1=IGREN(EN,IG) IF(I1.EG.C) GOTO 30 K1=16 NUMI=NUMB(I1) LDUM=INODEN(IND).NE.NBOUND(11,1) IF(LDUM) PRINT SOD,EN,IND,I1 TF(LDUM) STOP C----EACH ZONE DO 2C I=2,NUMI IND=IND+1 INCOEN(IND)=NBOUND(11,1) INCOENTINUE PRINT 21C,EN,(IGREN(EN,J),J=1,K1) IF(XSYM(EN)) PRINT 250,EN IF(YSYM(EN)) PRINT 260,EN 54ú NENCNG(EN)=K1 IF(SYM) GOTO 40 LDUM=INODEN(IND) .NE .NODE1 IF(LDUM) PRINT 51G, EN, NODE1, INODEN(IND) IF(LDUM) STOP IND=IND-1 CONTINUE 4 ũ NNODEN(EN)=IND CONTINUE CALL ENRADI(X,Y) CALL ENCONT RETURN FORMAT(A4) 557 FORMAT() FORMAT(//' VOIDS'/' *****') 190 200 FORMAT(/* THIS STRUCTURE HAS NO VOIDS //) FORMAT(/' THIS STRUCTURE HAS NO VOIDS'') FORMAT(/' NUMBER OF VOIDS=',12/) FORMAT(' VOID NUMBER',12,' IS SURROUNDED BY THE FOLLOWING ' 1 ,*NODE GROUP(S)',413) FORMAT(' VOID NUMBER',12,' IS SYMMETRICAL AROUND THE X-AXIS 5.63 •) FORMAT(' VOID NUMBER', 12, ' IS SYMMETRICAL AROUND THE Y-AXIS •) ////// SURKOUNDING NODEGROUPS NOT COMPATIBLE /// EN= ,I
/ IND=',I3,' I1=',I3)
FORMAT(///' FIRST AND LAST NODE ARE NOT IDENTICAL FOR '
/ ,'VOID NUMBER',I3//' FIRST NODE=',I4/' SECOND NODE=',I4)
END FORMAT(///* SURROUNDING NODEGROUPS NOT COMPATIBLE*//* EN=*,13, SUBROUTINE ENCLO?(T,FLOW) C----THIS ROUTINE IS CALLED FROM THE BASIC FINITE ELEMENT PROGRAM C----TO CALCULATE THE RATE OF HEAT EXCHANGE BETWEEN ENCLOSURE SURFACES

576 577 C----AS AFUNCTION OF CURRENT TEMPERATURE DIMENSION T(1), FLOW(1) 578 CO*MON/ENCLOS/LEN, NENC, NENCNG(2), IGREN(2,4), NNODEN(2), INODEN(100), 579 1 XSYM(2),YSYM(2) LOGICAL LEN IF(.NOT.LEN) RETURN 580 581 C----CALCULATE RATE OF RADIATION HEAT EXCHANGE CALL ENRAD2(T,FLOW) C----CALCULATE RATE OF CONVECTION HEAT EXCHANGE 582 583 584 585 CALL ENCONZ(T, FLOW) 586 RETURN 587 END 588 589 SUBROUTINE ENCON1 C----THIS ROUTINE FORMS CONVECTION ARRAY H PARAMETER NS=10,NNB=30,NNB2=2*NNB 59ú 591 592 COMMON/BNOD/NUMB(NB), NBOUND(NB, NNB), BAREA(NB, NNB), TH(NB), Common/Enconnerg(ND), FA(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(53), TAIR(2) 593 1 594 595 1 596 577 COMMON/DUMMY/HZ(25),DUM2(25) LOGICAL LEN LOGICAL XSYM,YSYM,SYM INTEGER EN 598 599 500 501 I N D = 1C----FORM ZONE CONVECTION ARRAY 502 C----EACH VOID DO 150 EN=1,NENC 603 504 605 SYM=XSYM(EN).OR.YSYM(EN) **Ι N =** 0 506 507 NENG=NENCNG (EN) C----EACH NODE GROUP DO 10 IG=1,NENG I1=IGREN(EN,IG) 608 509 510 611 NUMI=NUMB(11) BE=BETA(I1) 512 C----EACH ZONE DO 10 I=2,NUMI IN=IN+1 613 614 615 516 16 HZ(IN)=BE*BAREA(I1,I) C----FORM NODE CONVECTION ARRAY 617 CALL HTRANS(HZ, H(IND), IN, SYM) 518 N = I N 519 IF(SYM) N=N+1 62,0 IND=IND+N 521 622 150 CONTINUE 523 RETURN 524 END 625 526 SUBROUTINE ENCON2(T,FLOW) C----THIS ROUTINE CALCULATES THE AIR TEMPERATURE AND CONVECTIVE HEAT C-----FLOW IN EACH ENCLOSURE 627 528 DIMENSION T(1), FLOW(1) 629 COMMON/ENCON/H(50), TAIR(2) 630 COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100), 1 XSYM(2),YSYM(2) COMMON/DUMMY/HBAR(25),TEN(25) 631 632 633 1 LOGICAL LEN INTEGER EN 534 635 DATA PE/.ECO1/ 636 637 IND=C 538 C----EACH VOID DO 150 EN=1, NENC 539

540	N=NNODEN(EN)
541	CSTORE APPROPRIATE NODAL TEMPERATURES IN DUMMY VECTOR TEN
	NO 10 7-1 N
344	
543	NODE TNODER (IND +1)
544	1C TEN(I)=T(NODE)
645	CCALCULATE THE AIR TEMPERATURE TA BY ITERATION
212	CHARTER STARTING WALKE FORM FORMED TIME STEP
540	
547	TATINI KENJ
54 5	DO 50 ITER=1,10
449	SHBAR=]_
320	
651	CEACH NODE
55Z	DO 20 I=1,N
453	$T \cap T F = a \cap S (T F \cap (T) - T a)$
151	
524	
655	IF(TDIF+LT++C001) 6010 20
556	H5AR(I)=H(I)+T0IF***•33
657	SHBAR=SHBAR+HBAR(I)
525	376/-378/778/77/5/5/1/
559	20 CONTINUE
550	IF(SHBAR) 25,90,25
461	25 TANEW=SHRT/SHBAR
60Z	CCONVERGENCE CHECK OF AIR TEWPERATORE
563	IF(ABS((TANEW-TA)/(TANEW+TA))+LT+PE} GOTO 65
554	IF(ITER.GT.1) GOTO 30
665	$\mathbf{T} \mathbf{y} = \mathbf{T} \mathbf{A}$
000	
556	IT-IANCH
667	TA=(TANEw+TA)*•5
558	6010 50
440	30
207	
570	DY = (Y - iANEW)
671	$\mathbf{D} = \mathbf{D} \mathbf{Y} - \mathbf{D} \mathbf{X}$
472	τε(b)40.70.40
277	
575	
574	1 X = 1 A
675	TYTANEW
676	CUSE LINEAR INTERPOLLATION TO SPEED UP CONVERGENCE
477	TA = DN / D
611	
575	20 CONTRACE
679	PRINT 200,EN
680	STOP
4.8.1	6. CONTINUE
201	
50Z	
583	DY=TY-TANEW
684	D = D Y - D X
485	IF(0)65.70.65
	C. HIGAD TAITEDDOLLATION TO IMPROVE THE CALCULATED TEMPERATURE
550	LINEAR INTERPOLATION TO INTROT THE EXCLUTION TO THE DECOUNTED TENTER TRADE
687	65 TA=(TX*DY-TY*DX)/D
588	70 TAIR(EN)=TA
480	CCALCULATE CONVECTION HEAT FLOW AND ADD TO THE GLOBAL HEAT FLOW
100	
570	
591	9 T O T = C •
592	DO 30 I=1,N
A Q R	NODE = INODEN(IND + I)
40/	
079	9 - 10 A A A A A A A A A A A A A A A A A A
575	0101=0101+0
595	8C FLOW(NODE)=FLOW(NODE)+Q
597	90 CONTINUE
200	
270	
699	1 N D = 1 N D + N
700	15C CONTINUE
7.01	RETURN
772	200 FORMAT(/// FORVERGENCE NOT ACHIEVED FOR THE AIR TEMPERATURE".
102	200 FURDENTIAL CONTRACTOR NON-COLUMN FOR THE STATE OF AN AND A STATE OF A STA
703	1 - IN ENCENDER NOWBER (13)

704 220 705 706 1 END 707 SUBROUTINE ENRADI(X,Y) 708 C-----FORM RADIATION MATRICES FOR EACH VOID AND STORE THEM IN C-----THE VECTOR E. 709 715 C-----THE VELTOR 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/ENCLOSILEN NEW (NENCNG(2) ISPEN(2.4),NNODEN(2),IND COMMON/ENCLOSILEN NEW (NENCNG(2)) ISPEN(2.4),NNODEN(2),IND 711 712 713 714 715 COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100), 1 xSYM(2),YSYM(2) COMMON/ENRAD/E(1)CO) 715 717 1 718 719 COMMON/UNIT/SIGMA, TABS COMMON/DIM/MAXNG,MAXNOD COMMON/DUMMY/D(25),DUM2(25) . 72C 721 722 DIMENSION VIEw(25,25) EGUIVALENCE (A(1),VIEw(1)) DATA IND,IE/D,1/ LOGICAL LEN LOGICAL XSYM,YSYM,SYM 723 724 725 726 727 INTEGER EN C----EACH VOID DO 150 EN=1,NENC 728 729 730 CALL VIEWF(X,Y,D,EN,VIEW,MAXNOD) C----FORM THE MATRICES A AND B 731 732 733 NENG=NENCNG(EN) IN≖C C----EACH NODE GROUP 734 DO 120 IG=1,NENG I1=IGREN(EN,IG) 735 735 NUMI=NUMB(11) 737 C----EACH ZONE DO 12C I=2,NUMI IN=IN+1 738 739 742 741 JN=C JN=0 DO 120 JG=1, NENG J1=IGREN(EN, JG) NUMJ=NUMB(J1) 742 743 744 745 EPSJ=EPSG(J1) 746 DO 12J J=2,NUMJ JN=JN+1 747 B(IN,JN)=VIEW(IN,JN)+SIGMA 748 A(IN, JN) = -VIEW(IN, JN) * (1, -EPSJ)/EPSJ/D(JN)IF(IN.NE.JN) GOTO 12C B(IN,JN) = -SIGMA+B(IN,JN) 749 755 751 752 753 A(IN, JN) = 1./EPSJ/D(JN) + A(IN, JN)120 CONTINUE 754 N≖IN C----- INVERT A AND STORE RESULT IN A 755 756 CALL INVER(A, N, MAXNOD) C----MULTIPLY A AND B AND STORE RESULT IN A CALL MULT(A, B, N, MAXNOD) SYM=+FALSE+ 757 758 759 760 IF(XSYM(EN).OR.YSYM(EN)) SYM=.TRUE. 761 N Z = N IF(SYM) N=N+1 762 C-----TRANSFORM THE LOCAL RADIATION MATRICE A AND STORE THE RESULT IN 763 764 C----VECTOR E C-----B IS EMPLOYED AS A DUMMY MATRIX CALL ETRANS(A, B, E(IE), N, NZ, SYM, MAXNOD) 765 756 767 IE=IE+N*N

```
764
                      TND=130+N
759
             150
                      CONTINUE
77.
                      RETURN
771
                      END
772
                      SUBROUTINE ENRADZ(T,FLOW)
773
774
             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
775
             C----VECTOR FLOW
DIMENSION T(1),FLOW(1)
776
778
                      COMMON/ENCLOS/LEN, NENC, NENCNG(2), IGREN(7,4), NNODEN(2), INODEN(100),
                    1 XSYM(2),YSYM(2)
COMMON/ENRAD/E(1000)
779
780
781
                      COMMON/UNIT/SIGMA, TAES
782
                      COMMON/DUMMY/ETA(25),Q(25)
783
                      LOGICAL LEN
INTEGER EN
784
                     IL=1
IND=C
785
726
            C----EACH VOID
DO 15J EN=1,NENC
N=NNODEN(EN)
787
788
789
            C-----CALCULATE ABSULUTE TEMPERATURES TO THE FOURTH POWER FOR FOR THE
C-----NODES OF THE ENCLOSURE SURFACE
DO 10 I=1,N
790
791
792
                      NODE=1NODEN(IND+I)
.
793
794
                      DUM=T(NODE)+TABS
                      DOW=DOW*DOW
795
             10
                     ETA(I)=DUM*DUM
795
             C-----CALCULATE ENCLOSURE SURFACE RADIATION HEAT EXCHANGE VECTOR Q=E*ETA
797
             CALL RADVEC(E(IE), ETA,N,Q)
C----ADD TO GLOBAL HEAT FLOW VECTOR FLOW
DC 2C I=1,N
798
799
800
801
                      NODE=INODEN(IND+I)
                      FLOW (NODE) = FLOW (NODE) +Q(I)
802
            2 Û
                      IE=IE+N*N
8.03
804
                      IND=IND+N
             150
                      CONTINUE
805
806
                      RETURN
                      END
807
828
            SUBROUTINE ETRANS(A,B,E,N,NZ,SYM,MAX)
C-----THIS ROUTINE TRANSFORMS THE ZONE RADIATION MATPIX 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)
809
810
811
812
813
814
                      LOGICAL SYM
             C----E=SAT*A*SA
815
            C-----B=SAT*A
DO 1C I=2,NZ
DO 1C J=1,NZ
1C B(I,J)=A(I-1,J)+A(I,J)
816
817
818
819
                      DO 30 J=1,NZ
IF(SYM) GOTO 20
B(1,J)=A(1,J)+A(NZ,J)
8 Z C
821
822
823
                      GOTO 30
                      B(1,J)=A(1,J)
B(N,J)=A(NZ,J)
824
            2 L
825
                      CONTINUE
             3 Ú
826
                     -E=B*SA
827
             C ----
                     L=========

D0 50 I==1,N

E(I,J)===25*(R(I,J)+B(I,J=1))

D0 7C I==1,N
828
829
830
            56
831
```

832 IF(SYM) GOTO 52 E(I,1)=.25*(B(I,1)+B(I,N)) GOTO 70 833 874 \$35 E(I,1) = .25 + B(I,1)6 C 836 E(I,N)=.25*8(I,NZ) 837 70 CONTINUE 538 RETUPN 839 END 840 SUBROUTINE FEM2(IX,IY,NN,NE,NR,N,KTOP,X,Y,T,TT,TMAX,ELA,EV4,A,MAX, 1 P,W,EN,F,FŁOW,AXIAL,NODCPL,NODINT,NODEL,MNODEL,DTA) C----THIS ROUTINE INITIALIZES SYSTEM ARRAYS AND 341 842 343 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),T9(50),TITFIR COMMON/FIRE/TIM(50),T9(50),TITFIR COMMON/FIRE/TIM(50),T9(50),TITFIR COMMON/FIRE/TIM(50),T9(50),TITFIR 344 845 846 847 348 849 85 Û 851 COMMON/TOUT/II, TOUT(100), TIMMAX, DTMAX, TIMFAC, KTMAX, KUPDA 852 LOGICAL TMAX, AXIAL LOGICAL FIN, CON, NODI, UPDA 853 854 DATA FIN/.FALSE./ 855 C ----856 5 CONTINUE 857 KTIME=0 CON=.TRUE. 858 359 C----INITIALIZE NODAL TEMPERATURES CALL INIT(NN,T,TT,TMAX,NODINT) C----INPUT FIRE BOUNDARY TEMPERATURE 860 551 CALL SFIRE(FIN) 862 C----FIRST TIME INCREMENT FOR CALCULATING INCREMENT LENGTH ONLY 353 864 DELTI≃0. C----IF FIN= TRUE. ANALIZE NEW FIRE C----IF FIN=.TRUE. TERMINATE RUN IF(FIN)GOTO 1300 345 856 357 CALL ASSP2(NN,N,X,Y,T,TT,TMAX,EV4,NODEL,MNODEL,P,W,NODINT, AXIAL) 848 369 1 C----INITIALIZE NODAL ENTHALPY VECTOR EN 870 $\label{eq:c----homogeneous nodes - en = enthalpy(heat) per unit volume c----interface nodes - en = enthalpy(heat)$ 871 872 373 DO 10 I=1, NN NDI=NODINT(I) 374 IF(NODCPL(I).EQ.J.OR.NDI.LT.C) GOTO 10 875 IF(NDI.GT.O) CALL XVERSY(TE,ENT,MNV,NDI,T(I),EN(I)) C----MASTER NODES AND INTERFACE NODES IF(NODCPL(I).GT.J.AND.NDI.EG.D) CALL MINTP(I,NODCPL(I),P) 875 877 878 IF(NDI.EG.C) EN(I)=P(I) *T(I) 879 880 CONTINUE 10 c -----381 C----START TIME INTEGRATION LOOP 882 č----9 R Z 700 CONTINUE 884 DUM1=FLOAT(KTIME)/FLOAT(KUPDA) 885 885 DUM2=AINT(DUM1) UPDA=DUM1.EQ.DUM2.OR.KTIME.EQ.1 887 888 KTIME=KTIME+1 C----CALCULATE INTERNAL HEAT FLOW BY CONDUCTION 889 89D IF (UPDA) CALL ASSA2(NN,NE,N,KTOP,X,Y,ELA,T,TT,TMAX,AXIAL,MAX,A) 891 1 DO 20 I=1,NN 892 20 DTA(I)=A(I,MAX) CALL MPACKV(A,T,F,MAX,NN) C----GET FIRE TEMPRATURE 393 894 895

CALL XVERSY(TIM,TP,50,1,TIME,TFIRE) C----CALCULATE BOUNDARY HEAT FLOW 896 897 898 CALL FRENDB(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 899 900 901 CALL ENCLOZ(T, FLOW) 902 903 904 C-----CALCULATE HEAT CAPACITY MATRIX AT CURRENT TEMPERATURE CALL ASSP2(NN,N,X,Y,T,TT,TMAX,EV4,NODEL,MNODEL,P,W,NODINT, 905 AXIAL) 1 C----SUM APPROPIATE QUANTITIES OF COUPLED NODES 906 907 CALL COUPLE(F) CALL COUPLE(FLOW) CALL ACOUPL(A,DTA,NN,MAX) C----CALCULATE NEW NODAL TEMPERATURES 908 909 910 DO 5C I=1,NN NDC=NODCPL(I) 911 912 913 NDI=NODINT(I) 914 IF(NDI+LT+0+OR+NDC+E0+0) GOTO 50 C----HOMOGENEOUS NODES ONLY 915 NODI=NDI.GT.O 915 IF(NODI)CALL HTEMP(T(I), W(I), EN(I), FLOW(I), F(I), NDI, DELTI) 917 IF(NODI) GOTO 50 C----INTERFACE NODES 918 919 IF(NODCPL(I).GT.D) CALL MINTP(I,NODCPL(I),P) 920 CALL CTEMP(I,T(I),P,EN,FLOW,F,DELTI,NODEL,MNODEL,N,EV4,NDC) 921 922 5 C CONTINUE C----SET PRESCRIBED NODAL TEMPERATURES CALL PTBNDB(T,TFIRE) 923 924 CALL COUPLE(T) 925 C----PRINT CURRENT NODAL AND VOID AIR TEMPERATURES 926 CALL OUT2(IX,IY,NN,NE,X,Y,TIME,KTIME,DELTI,T,TT,TMAX,FLOW,TFIRE, 1 NODT,AXIAL) 927 928 1 C----SET CON=.FALSE. TO TERMINATE TIME INTEGRATION CALL MAXCO(NN,TMAX,TT,T,TIME,KTIME,CON) C----CALCULATE NEW TIME INCREMENT DELTI 929 930 931 CALL DTIME (NN, P, DTA, MAX, NODINT, NODCPL, TIME, DELTI, NODT) 932 933 TIME≑TIME+DELTI IF(CON) GOTO 700 934 935 C ----936 C----END TIME INTEGRATION LOOP C - - - - -937 720 CONTINUE 93R TIME=TIME-DELTI 939 C-----PRINT MAXIMUM TEMPERATURE OBTAINED DURING AVALYSIS CALL OUTMAZ(IX,IY,NN,NE,X,Y,TIME,KTIME,T,TT,TMAX,FLOW,AXIAL) GOTO 5. 946 941 942 943 1000 RETURN 944 END 945 946 SUBROUTINE FOBNDA C----THIS ROUTINE FORMS RADIATION AND CONVECTION MATRICES BR AND BC COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4 PARAMETER NB=10,NNB=32,NNB2=2*NNB 947 948 249 COMMON/FQB/NFQNG,NFQG(NB),TR(NNB),TC(NNB) 95C 1 ,BR(NNB2),BC(NNB2) COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB), 951 1 952 1 EPSG(NB), BETA(NB), CPG(NB), FA(NB) 953 LOGICAL FA,FA1 C----READ NUMBER OF BOUNDARY NODES GROUPS READ 100,NFGNG 954 955 -956 957 C----958 IF(NFQNG.EQ.C) RETURN PRINT 200 959

I N D = 1 C----EACH BOUNDARY FLOW NODE GROUP DO 10 IB=1,NFQNG READ 100, FA1, ING1 C----IF FA1=.TRUE. FIRE BOUNDARY ELSE AMBIENT TEMPERATURE C----ING1 \pm NODE GROUP NUMBER FA(ING1)=FA1 NFQG(IB)=ING1 970 971 NUMI=NUMB(ING1) BET=BETA(ING1) EPSIG=EPSG(ING1)*SIGMA IF(EPSIG.EQ.O.AND.BET.EQ.C) PRINT 300 IF(EPSIG.EQ.O.AND.BET.EQ.C) STOP CALL BRECA(BR(IND), BC(IND), EPSIG, BET, BAREA, NUMI, NB, ING1) INDEIND+Z*NUMI IF(FA1) PRINT 212,ING1 IF(.NOT.FA1) PRINT 222,ING1 977 CONTINUE 1 J C FORMAT() FORMAT()' PRESCRIBED FLOW BOUNDARY'/1X,24(1H*)/ 1 /' NODE GROUPS AND TYPES OF BOUNDARIES'/) FORMAT(' NODE GROUP',I3,' FIRE BOUNDARY') FORMAT(' NODE GROUP',I3,' AMBIENT BOUNDARY') FORMAT(/' BOTH EMISSIVITY AND CONVECTION FACTOR ZERO') 98C 98Ż 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/FQE/NFQNG,NFQG(NB), TR(NNB), TC(NNB) 1 , JR(NNB2), BC(NNB2), TRD(NNB), TCD(NNB) 992 COMMON/BNOD/NUMB(NB), NEOUND(NB,NNB), BAREA(NB,NNB), EPSG(NB), BETA(NB), CPG(NB), FA(NB) COMMON/UNIT/SIGMA, TARS, TINIT, TAMB, TAMB4 LOGICAL FA LOGICAL FA C----NULL FLOW VECTOR D0 777 I=1,NN 777 FLOW(I)=0. C-----RETURN IF NO PRESCRIBED BOUNDARY FLOW IF(NFQNG.EG.O) RETURN 1003 TF4=(TFIRE+TABS)**4 IND = 1C----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 1024 TC(1)=+(-DUM)**CP GOTO 20 1025 5 1026 1027 1028 1 C TC(I)=DUM**CP CONTINUE żΰ 1029 c -----CALL BRECE(BR(IND), BC(IND), TR, TC, TRD, TCD, NUMI, DTA, NN, 1030 1031 MAX, FLOW, TG, T, ING1) 1 1032 IND#IND+2*NUMI 1033 30 CONTINUE 1034 RETURN 1035 END 1036 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=2G,MNR=1C COMMON/RMAT/CCC(MNR),TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR), 1037 1038 1039 1040 1041 ENT(MNV,MNR), CR(MNV,MNR), TQ(MNV,MNR), QE(MNV,MNR), LQ(MNR) 1042 1 1043 LOGICAL CCC,LQ DO 20 I=1,NE N1=N(I) 1044 1045 IF(.NOT.LQ(NT)) SOTO 20 1040 1047 00 10 K=1,4 NOD=KTOP(K,I) 1048 1049 CALL XVERSY(TQ,GE,MNV,N1,T(NOD),FGEN) FLOW(NOD)=FLOW(NOD)+EV4(1)*FGEN 10 1051 ЗC CONTINUE 1052 1053 RETURN ΞNĐ 1054 SUBROUTINE GEOCO2(NN,NE,N,KTOP,X,Y,AXIAL,ELA,EV4) C----THIS SUBROUTINE COMPUTES ELEMENT GEOMETRICAL CONSTANTS 1055 1056 1057 C -----NUMBER OF NODES NUMBER OF ELEMENTS Element region number c----ΝN 1058 C-----1059 NE -----N 106Û NODE COORDINATES TRUE IF AXIAL SYMMETPIC PROBLEM ELEMENT GEOMERTIC CONSTANTS 1061 c----Х,Ү 1062 c----AXIAL (----ELA 1063 ELEMENT VOLUME/4 E V 4 1064 C -----DIMENSION X(NN), Y(NN), EV4(NE), ELA(4, NE), KTOP(4, NE), N(NE) 1065 PARAMETER MNR=10 COMMON/RGEO/ELFICT(MNR),ET(MNR),SRDIAC(4,MNR) 1366 1067 LOGICAL AXIAL, ELFICT 1068 D0 5 I=1,NE 1369 1070 $v1 = v(\tau)$ K1=KTOP(1,1) 1071 1072 K4=KTOP(4,I) 1073 A=X(K4)-X(K1) 3=Y(K4)-Y(K1) 1074 1075 ETT=ET(N1) 1376 ELA(1,1)=ET1*(A*A+B*B)/3./A/B ELA(2,1)=ET1*(-2*A*A+B*B)/6./A/B ELA(3,1)=ET1*(A*A-2.*B*B)/6./A/B 1077 1578 ELA(4,I)=-ET1*(A*A+B*B)/6./A/B 1079 1080 EV4(1)=ET1+A+B/4. 1381 5 CONTINUE 1082 1083 RETURN END 1584 SUBROUTINE HTEMP(T,W,EN,FLOW,F,N1,DELTI) C----CALCULATE TEMPERATURE FOR HOMOGENEOUS NODES 1085 1386 1087 PARAMETER MNV=20,MNR=10

```
COMMON/RMAT/TC(MNV,MNR),C(MNV,MNR),TE(MNV,MVR),ENT(MNV,MNR),
1088
                                           CR (MNV, MNR)
1089
1590
                        1
                          D=(FLOW-F)*DELTI
1091
                          EN=EN+D/W
1092
                          CALL XVERSY(ENT, TE, MNV, N1, EN, T)
1093
1094
                          RETURN
                          END
1095
                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
1295
1097
                C----IF SYMMETRY IS PRESENT EXPAND CONVECTION VECTOR
DIMENSION HZ(1),H(1)
1399
1100
                          DO 10 I=2,N
H(I)=.5*(HZ(I-1)+HZ(I))
1101
1102
1103
1124
                10
                          CONTINUE
                          IF(SYM) GOTO 20
H(1)=.5*(HZ(N)+HZ(1))
1105
1106
                          GOTO 30
1107
                          CONTINUE
H(1)=.5*HZ(1)
1108
                2 ũ
1109
1110
                          H(N+1) = .5 + HZ(N)
1111
                36
                          RETURN
1112
                          END
1113
1114
                SUBROUTINE INIT(NN,T,TT,TMAX,NODINT)
C----SET INITIAL NODAL TEMPERATURE
DIMENSION T(NN),TT(NN),TMAX(NN),NODINT(NN)
1115
1115
1117
1118
                          LOGICAL TMAX
                          COMMON/TOUT/II,TOUT(100),TIMMAX
COMMON/UNIT/SIGMA,TAES,TINIT,TAMB,TAMB4
1119
                          II=1
DC 1 I=1,NN
1120
1121
                           IF(NODINT(I).LT.D) GOTO 1
1122
                          T(I)=TINIT
TT(I)=T(I)
1123
1124
                           TMAX(I)=.FALSE.
1125
                1
1126
                          RETURN
1127
                          END
1125
1129
                SUBROUTINE INTERF(NN,NE,NR,NX,NY,KTOP,N,NODINT,NODCPL)

C----THIS ROUTINE FORMS VECTOR FOR IDENTIFICATION OF

C-----NODINT=-1 + FICTITIOUS NODES

C-----NODINT=-1 - FICTITIOUS NODE

C-----NODINT= 1 - HOMOGENEOUS NODE

PARAMETER MNR=10,NCP=50

COMMON/REGO(ELETT(MED)
1130
1131
1132
1133
1134
1135
                          COMMON/RGEO/ELFICT(MNR)
1136
1137
                          COMMON/COUPLE/NCOUPL(NCP,8),NCPLG
DIMENSION KTOP(4,NE),N(NE),NODINT(NN),NODCPL(NN)
1138
                          LOGICAL ELFICT
PRINT 200
1139
1140
                          DO 5 I=1,NN
NODINT(I)=-1
1141
1142
1143
                5
                           IF(NR.EU.1) GOTO 50
1144
                           N X 1 = N X - 1
                          NY1 = NY - 1
1145
                          IF(NY.EQ.2)GOTO 25
1146
1147
                C----
                          DO 20 I=1,NX1
INY=(I-1)*NY
INY1=(I-1)*NY1
1148
1149
1150
                          IE1=INY1+1
1151
```

1152		N1=N(I51)
1153	C	
1154		D0 20 J=2,NY1
1155		IF(ELFICT(N1)) GOTO 15
1156		DO 10 IDUM=1.3
1157		IE2=IE1+(IDUM-2)*NY1+1
1155		IF(IE2.LE.C.OR.IE2.GT.NE)GOTO 10
1159		N2=N(TE2)
1160		IF(FLFICT(N2)) GOTO 10
1161		16(N1.EQ.N2) 50TO 10
1162		TE(TDHM_NE_2)GOTO 1G
9143		NOD=TNY+3
1164		
1165		NOD = NOD + NY
1166		
1167	(NUD2N1(((UD)=0
1448	1	CONTINIE
4460	15	CONTINUE
1107		TC1-TC1+1
4474		121-12111
117	c	
4477	2.	 CONTTUDE
1173	26	
1774	20	CONTINUE
1175	L	
11/6		IF(NX.EQ.236010.56
1177		DU 4L I=1,NY1
1178		INX=I+1
1179		IE1=I
1180		VI=N(IEI)
1181	63	
1182		DO 40 J=2,NX1
1183	C	
1184		IE2=IET+NYT
1185		NZ=N(IEZ)
1186		IF(ELFICT(N1).OR.ELFICT(N2)) GOTO 30
1187		IF(N1.EQ.N2) GOTO 30
1188		NOD≈(J−1)*N¥+I
1189		NODINT(NOD)=C
1190		NOD=NOD+1
1191		NODINT(NOD)=0
1192	C	
1193	30	CONTINUE
1194		IE4=IE5
1195		¥1=N2
1196	C	
1197	4 0	CONTINUE
1198	5 Û	CONTINUE
1199	C	·
1290		DO 7C I=1,NE
1201		N1=N(I)
1202		IF(ELFICT(N1)) GOTO 70
1203		D0 6C J=1,4
1204		NOD=KTOP(J,I)
1205	60	IF(NODINT(NOD).EQ1) NODINT(NOD)=N1
1206	70	CONTINUE
1207	C	IF ONE NODE IN A COUPLED GROUP IS AN INTERFACE NODE
1208	C	ALL NODES IN THE GROUP ARE CONSIDERED INTERFACE NODES
1209		IF (NCPLG.EQ.Û) GOTO 120
1210		DO 110 I=1,NCPLG
1211		DO 85 J=1,8
1212		NOD=NCOUPL(I,J)
1213		IF(NOD.EQ.D) GOTO 110
1214		IF(NODINT(NOD).EQ.O) GOTO 90
1215	85	CONTINUE

1216	90	CONTINUE
1217		DO 103 J=1,8
1218		NOD = NCOUPL(I,J)
1219		TE(NOD-E9.0) GOTO 110
1220		NODINT (NOD) = 0
1221	100	
1221	440	
1222	110	
1225	120	
1224		PRINI 200
1225		DO SE I=1,NX
1226		II=(I-1)*NY
1227	8C	PRINT 210, (NODINT(11+J), J#1, NY)
1228	205	FORMAT(/* -1 - FICTITIOUS NODE*/* J - INTERFACE NODE*/
1229	1	1 - HOMOGENEOUS NODE")
1230	200	FORMAT(//* INTERFACE NODES*/16(1H*))
1231	210	FORMAT(/1CX,5GI2)
1232		RETURN
1233		END
1234		
1235		SUBROUTINE INTP(MND,NODEL,N,EV4,TI,PI)
1236	c	CALCULATE HEAT CAPACITY OF INTERFACE NODES
1237	-	DIMENSION NODEL (4).N(1).EV4(1)
1238		
1230		COMMON/PMAT/TC(MNV_MNR).C(MNV_MNR).TE(MNV_MNR).ENT(MNV_MNR)
1237		COMPONENTIAL CENTRAL CONTRAL CO
1240	-	
1241		
7242		
1243		EV4IE=EV4(IE)
1244		CALL XVERSY(TE,ENT,MNV,NT,TI,ENI)
1245		CRA=ENI/TI
1246		PI=FI+EV4IE*CRA
1247	10	CONTINUE
1248		RETURN
1249		E N D
1256		
1251		SUBROUTINE INVER(A,M,MAX)
1252	C	THIS ROUTINE INVERTS THE MATRIX A AND STORES THE RESULT IN THE
1253	C	SAME MATRIX
1254		DIMENSION A(MAX.1)
1255		DG 200 N=1.#
1256		$D = \Delta (N \cdot N)$
1257		50 1C3 J=1.M
1258	160	$\Delta(N, L) = -\Delta(N, L)/D$
1250	, 20	
1237		
1200		
1201		00 140 J*130 Trai to No coto 140
1404		IT (JEE AND BUTE 1445 ALT DEALT DEALT DEAL AND A
1600	4.0	А Ц 1 J J -
1254	140	CONTINUE
1265	150	A(I,N)=A(I,N)/D
1266	500	A (N, N) = 1 + C / D
1267		RETURN
1258		END
1269		
1270	COMMEN	IT MAIN PROGRAM CODED IN NUALGOL FOR DYNAMIC ALLOCATION OF ARRAYS
1271		FOR INFORMATION ABOUT ARRAYS SEE SUBROUTINE PROGZ;
1272	BEGIN	
1273	INTEGE	R NN,NE,NR,IX,IY,MAX;
1274	REAL A	RRAY XE, YL, XA, YA(1:100);
1275	BOOLEA	N AXIAL:
1276	FXTERN	AL FORTRAN PROCEDURE NET2, DIM2;
1277	NETZCY	I YI TX TY NR AXTAL XA YA NN NE MAX);
1278		a FG TN
1270		BODIEAN ARRAY TMAX(1:NN):
16(7		DEVENUE TRUESTERISE

INTEGER ARRAY KTOP(1:4,1:NE),N(1:NE),NODEL(1:NN,1:4),MNODEL, 1280 1281 NODINT(1:NN); 1282 REAL ARRAY X, Y, T, TT, P, W, EN, F, FLOW, NODCPL, DTA(1:NN), EV4(1:NE), 1283 ELA(1:4,1:NE),A(1:NN,1:MAX); EXTERNAL FORTRAN PROCEDURE MESH2,PROG2; 1284 1285 MESHZ(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); 1236 1287 1285 END 1289 ΕΛĐ 1290 1291 SUBROUTINE MAT(NR) 1292 C----THIS ROUTINE READS MATERIAL INPUT NUMBER OF REGIONS REGIONAL MATERIAL DATA (----1293 NR C----COMMON RMAT 1294 TRUE IF CONDUCTIVITY IS TAKEN AT MAXIMUM TEMPERATURE 1295 {-----000 c -----1296 MAXIMUM TEMPERATURE TC,C TEMPERATURE VERSUS CONDUCTIVITY PAIRS TE,ENT,CR TEMPERATURE VERSUS SPECIFIC ENTHALPY AND SPECIFIC ENTHALPY / TEMPERATURE TQ,QE TEMPERATURE VERSUS INTERNALLY 1297 C-+--c----1298 C - - - - -1299 1300 C----GENERATED HEAT 1301 C -----LG TRUE IF HEAT IS GENERATED INTERNALLY REGIONAL GEOMETRICAL DATA 1302 C-----1303 C----COMMON RGEO c -----FLEMENT THICKNESS 1394 cŤ SUBREGION LIMMITS 1305 1306 1307 XR C-----DIAGONAL COORDINATES OF SUBREGIONS SRDIAC C-----C----PARAMETER MAX NUMBER OF VALUE PAIRS MAX NUMBER OF REGIONS (-----1308 MNV C - - - - -1309 MNR 1310 DIMENSION MAT(20) 1311 PARAMETER MNV=20, MNR ± 10 COMMON/RGED/ELFICT(MNR),ET(MNR),SRDIAC(4,MNR) COMMON/RMAT/CCC(MNR),TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR), ENT(MNV,MNR),CR(MNV,MNP),TG(MNV,MNR),GE(MNV,MNR),LG(MNR) 1312 1313 1314 1 LOGICAL CCC.LQ LOGICAL ELFICT 1315 1316 1317 DATA CCC, LQ/MNR*.FALSE., MNR*.FALSE./ 1318 PRINT 95 NRR=NR 1319 DO 2 1=1,NR 1320 1321 IF(ELFICT(I)) NRR=NRR-1 IF(ELFICT(1)) GOTO 2 PRINT 100,1 C----INPUT NAME OF REGION FOR IDENTIFICATION 1322 1323 1324 READ 200,MAT 1325 C-----INPUT MATERIAL AND ELEMENT PROPERTIES FOR EACH REGION 132,6 READ 1, CCC(I), NTC, NTE, NGE, ET(I) 1327 IF(NTC.LE.1.OR.NTE.LE.1.OR.NGE.EQ.1) GOTO 1300 1328 IF(NQE.GE.2) LG(I)=.TRUE. MAXNTE=MAXD(MAXNTE,NTE) 1329 1330 1331 IF(eT(I).EQ..0) ET(I)=1. IF(.NOT.CCC(I)) PRINT 101,ET(I),MAT 1332 IF(CCC(I)) PRINT 102, ET(I), MAT NT=MAXD(NTC,NTE) 1333 1334 1335 C----INPUT TEMPERETURE CONDUCTIVITY PAIRS READ 1, (TC(K,I),C(K,I),K=1,NTC) C----INPUT TEMPERATURE SPECIFIC VOLUMETRIC HEAT PAIRS 1336 1337 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,NGE) 1338 1339 134C DO 38 K=1,NTE 1341 1342 IF(K.NE.1) CR(K,I)=(ENT(K,I)-ENT(1,I))/(TE(K,I)-TE(1,I)) 38 1343 CR(1,1)=CR(2,1)

PRINT 110 D0 39 K=1,NT PRINT 115 IF(C(K,I).GT..CCC01) PRINT 12G,TC(K,I),C(K,I) IF(CR(K,I).GT..CODC1) PRINT 13C,TE(K,I),ENT(K,I),CR(K,I) CONTINUE IF (NGE.EQ.0) GOTO 45 PRINT 140, (TQ(K,I),QE(K,I),K=1,NQE) CONTINUE CONTINUE RETURN FORMAT() FORMAT() FORMAT(/1X, 'MATERIAL DATA'/1X,13(1H*)) FORMAT(/1X, 'REGION NUMBER',I3) FORMAT(1H+,30X,'THICKNESS',F9.3/1X,20A4) FORMAT(1H+,30X,'THICKNESS',F9.3/1X,20A4/ 1 * CONDUCTIVITY IS KEPT CONSTANT AFTER REACHING MAXIMUM* 2 ,' TEMPERATURE') FORMAT(/1X, 'TEMP',6X,'CONDUCTIVITY',13X,'TEMP',8X,'ENTALPHY',6X, 1 *ENT/TEMP'/) FORMAT(11) = E6 0 E15 () FORMAT(1H+, F6.0,E15.4) FORMAT(1H+,33X,F6.0,E15.4,E15.4) FORMAT(/' INTERNALLY GENERATED HEAT'/' TEMP',&X,"HEAT' 1 //(1x,F6.0,E15.4)) FORMAT(20A4) PRINT 12CU,NTC,NTE,NQE STOP STOP 1200 FORMAT(///* PROGRAM TERMINATED WHEN READING MATERIAL INPUT* 1 //*NTC=*,I3,* NTE=*,I3,* NQE=*,I3. 1373 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 LOGICAL CON COMMON/TOUT/II,TOUT(10C),TIMMAX,DTMAX,TIMFAC,KTMAX IF(TIME.GE..9999*TIMMAX) CON=.FALSE. IF (KTIME.GT.KTMAX) PRINT 2GC,KTIME IF (KTIME.GT.KTMAX) CON=.FALSE. FORMAT(//* TERMINATED AT MAXIMUM NUMBER OF TIME * 1, 'INCREMENTS KTIME=*IS) 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) $x(\kappa\kappa) = x \lfloor (I)$ Y (KK)=YE(J) CONTINUE C----COMPUTE THE TOPOLOGY MATRIX KTOP IX1 = IX - 1111=11-1 1403 D0 10 I=1,IX1 D0 10 J=1,IX1 IE=IY1*(I-1)*J KTOP(1,IE)=IY*(I+1)+J kTOP(2,IE)=KTOP(1,IE)+1

-97-

1438	KTOP	°(3,1E)=KTO	OP(1,IE)+IY
1429	ктон	°(4,IE)=K↑C	0P(3,1E)+1
1410	10 CONT	TINUE	
1411	RETU	JRN	
1412	END		
1413			
1414	SUBF	ROUTINE MIN	NTP(I,NODCPL,P)
1415	CSUM	HEAT CAPAC	CITY OF COUPLED NODES
1416	PAR	AMETER NCP=	=50
1417	COM	MON/COUPLE/	/RCOUPL(NCP,8),NCPLG
1418	DIME	ENSION P(1))
1419	DO 3	30 J=2,8	
1420	¥00*	=NCOUPL(NOD	DCPL,J)
1421	11()	NOD∗EC.C) R	RETURN
1422	36 P(I))=P(I)+P(NC	00)
1423	RETU	JRN	
1424	END		
1425			
1426	SUBF	ROUTINE MPA	ACKV(A,X,R,MI,NN)
1427	CTHIS	S ROUTINE M	MULTIPLIES BANDED AND PACKED SYMMETRIC MATRIX
1428	C4IT1	H VECTOR	A + X = R
1429	CA -	- MATRIX W	WITH DIAGONAL ELEMENTS IN RIGHT HAND SIDE COLUMN
1430	DIME	ENSION A (NN	N,MI),R(NN),X(NN)
1431	DO .	3 I=1,NN	
1432	R(I)	9=0	
1432	DU	2 J=1,8I	-
1434	2 IF((I+J-MI)+GT	T.0)
1435	*R(I)	= R(I) + A(I)	, J) * X { [+ j - M] }
1436	IFC	I.EQ.NN) GO	0 TO 3
1437	I 1 =1	MINC(MI-1),	,(NN+I))
1438	DO	1]=1,11	
1439	1 9(1)=R(I)+A(I+	+J,MI-J)*X(I+J)
1440	3 CON	TINUE	
1441	RETI	JRN	
1442	END		
1443			
1444	SUBE	ROUTINE MUL	LILA, B, N, MAX)
1445	CTHIS	S ROUTINE *	FULTIPLIES THE MATRILES A AND 3 AND STORE THE
1446	CRESU	JLT IN A WI	ITH CHANGED SIGNS
1447	DIME	ENSION ACMA	AX,MAX),B(MAX,MAX)
1448	COMM	10N/DUMMY/E	ETA(25),DUM2(25)
1449	DO	20 I=7,N	
1450	DD	10 J=1,N	
1451	ETA	(J)=0.	
1452	DO	10 K=1,N	
1453	TO ETA	(J)=EIA(J)*	*8(1,K)*8(K,J)
1454	D0 4	20 J=1 N	
1455	20 ACI,	JJ=EIA(JJ	
1455	REIL	JEN	
1457	END		
1458			TOYUL UL TY TU AND AVTAL VA VA UN NE MAY
1459	5031	COULTNE NET	I Z VAL JIE JIA JII JUR JAALAEJA HJI AJIAJUE JURAA Taali jaata ayo achucoate ither dadalleli uttu ayis
1460	LINPL	FI GEUMEIRI CALCULATE	NUMBED OF GENERATED NODES AND FIFMENTS
1451	LAND	CALLULATE	NUNDER UT GENERALEN NUVES ANN GEENERIS
3464	L	VI.	COORDINATES OF Y-IINES
1455			COODSTNATES OF ATLANES
1404	L	1	COORDINATES OF TELES YELLINGS
1455	(X A V X	COORDINATES OF SPECIFIED VEINES
1400	(т A T V	VORDING VE AFLINES FOORDING LES OF SECTIFED LAFTGED
1407 -	(1 4	NUMBED OF Y-FINES
1400	C	1 T N D	NUMBER OF FEGIONS
1407	(N (5 A 1 A 1	TRHEL TE AXTHSYMMETRIE PROALEM
1475	(M A L A L N L	NHWAED NODES
1471		THE IN	NO SER NOVED

.

```
NUMBER OF ELEMENTS
1472
             C----
                           ΝF
                                        NUMBER OF SPECIFIED X-LINES
NUMBER OF SPECIFIED Y-LINES
             C-----
                           ΝX
1473
1474
                           ΝY
1475
             C----
1475
                     PARAMETER MNR=10
                     COMMON/RGEO/ELFICT(MNR),ET(MNR),SRDIAC(4,MNR)
1477
                     LOGICAL AXIAL, ELFICT
DIMENSION XL(1), YL(1), XA(1), HEAD(20)
1475
1479
1430
                     PRINT 199
             PRINT 199
C*****MACHINE DEPENDENT STATEMENT * IN LDC*LIB.SEG * REQUIRED IN MAP ELEMENT
CALL SEG("TASEF")
PRINT 200
C----INPUT TITLE OF RUN
1481
1482
1483
1484
                                                                                     .
                    READ 110, HEAD
PRINT 205, HEAD
1485
1486
1427
             c-----
1488
             C----INPUT MAIN GEOMETRICAL DATA
             C----
1489
                    READ 100, AXIAL, XMAX, YMAX, XBOX, YBOX, NR, NX, NY
1621
                     PRINT 210,XMAX,YMAX
PRINT 220,XBOX,YBOX
IF(NR.EQ.C) NR=1
1491
1492
1493
            EPS= XMAX/10000.
                     NR1=NR-1
1494
1495
1496
             C-----INPUT SUBREGION LIMITS
C----READ THE DIAGONAL COORDINATES FOR EACH SUBREGION
1497
1498
1499
             C----
1500
                     IF(NR.EQ.1) GOTO 5
                     PRINT 23C
READ 100, (ELFICT(J), (SRDIAC(I,J),I=1,4),J=2,NR)
PRINT 24G,NR1, ((SRDIAC(I,J),I=1,4),ELFICT(J),J=2,NR)
1501
15.52
1503
                     CONTINUE
1504
             5
                    IF(NX.EQ.C) GOTO 6
1505
1526
1527
             C----
             C----INPUT SPECIFIED X - LINES
             c----
1508
                     PRINT 250
1509
1510
                     READ 100,(XA(I),I=1,NX)
PRINT 260,(XA(I),I=1,NX)
1511
                     CONTINUE
1512
             6
                     IF(NY.EQ.0) GOTO 7
1513
             C-----INPUT SPECIFIED Y - LINES
PRINT 270
READ 196, (YA(I),I=1,NY)
1514
1515
1516
                     PRINT 280, (YA(I), I=1, NY)
1517
1518
             7
                     CONTINUE
             C-----
C----IF AN AXI-SYMMETRIC PROBLEM INPUT INVER RADIUS
C-----
1519
1520
1521
1522
                    NX≖NX+1
                     IF(.NOT.AXIAL) 60T0 8
1523
                     READ 100,XA(1)
PRINT 300,XA(1)
1524
1525
1526
             8
                   CONTINUE
             c----
1527
             C----GENERATE X-LINES
1528
1529
                    x = (1) = x = (nx)
1530
                     DO 15 IX=2,100
1531 -
1532
                     XL(IX)=XL(IX-1)+X80X
1533
             C-----
C----CONTROL OF SPECIFIED X-LINES
1534
             C----
1535
```

```
00 10 I=1,NX
1536
                      IF(XL(IX-1).LT.(XA(I)-EPS)) XL(IX)=AMIN1(XL(IX),XA(I))
1537
1538
1539
              10
                     CONTINUE
              C----
1540
              C----CONTROL OF SUBREGION LIMITS
1541
              C-----
                      IF(NR.EQ.1) GOTO 12
1542
                      II = 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))
1543
1544
1545
1546
              11
                      CONTINUE
1547
              12
                      CONTINUE
1548
              C----
              C----CONTROL OF XMAX
1549
              C----
1550
1551
                      XL(IX)=AMIN1(XL(IX),XMAX)
                      IF(ABS(XL(IX)-XMAX).LT.EPS) GOTO 16
1552
              15
                      CONTINUE
1553
1554
                     CONTINUE
              16
              C----GENERATE Y-LINES
1555
1556
              C-----
1558
                     DO 20 IY=2,100
                      YL(IY)=YL(IY-1)+YBOX
1559
1560
              c-----
              C-----CONTROL OF SPECIFIED Y-LINES
1561
1562
              (-----
                      DO 17 I≂1,NY
IF(YL(IY-1).LT.(YA(I)-EPS)) YL(IY)=AMIN1(YL(IY),YA(I))
1563
1564
              17
                     CONTINUE
1565
1566
              C----
1567
              C----CONTROL OF SUBREGION LIMITS
              C----- IF(NR.EQ.1) GOTO 19
1568
1569
                      D0 18 I=2,NR
IF(YL(IY-1).LT.(SRDIAC(2,I)-EPS)) YL(IY)=AMIN1(YL(IY),SRDIAC(2,I))
1570
1571
                      IF(YL(IY-1).LT.(SRDIAC(4,1)-EPS)) YL(IY)=AMIN1(YL(IY),SRDIAC(4,1))
1572
1573
              18
                      CONTINUE
1574
              19
                      CONTINUE
1575
              C-----
C-----CONTROL OF YMAX
1576
1577
              C-----
                      YL(IY)=AMIN1(YL(IY),YMAX)
IF(AES(YL(IY)-YMAX).LT.EPS) GOTO 21
1578
1579
1580
              ΖĈ
                      CONTINUE
1581
              21
                      CONTINUE
              C----PRINT COORDINATES OF X - AND Y - LINES
PRINT 31C,IX,(XL(I),I=1,IX)
1582
1583
1584
                     PRINT 320, IY, (YL(I), I=1, IY)
1585
              C----
                      NE = (IX - 1) + (IY - 1)
1586
1587
                      NN=IX*IY
1588
                      PRINT 330,NN,NE
1589
                      MAX = IY + Z
              100
                      FORMAT()
1590
              115
199
1591
                      FORMAT(20A4)
                      FORMAT(1H1)
FORMAT(//' TEMPERATURE ANALYSIS OF STRUCTURES EXPOSED TO FIRE'/
1592
1593
              200
                    FORMAT(//* TEMPERATURE ANALYSIS OF SIRUCTORES EAPOSED TO FIRE*/
1* SOLVES NON LINEAR TRANSIENT FIELD PROBLEMS*
2 /* *** TWO DIMENSIONAL VERSION ****
3 /* PROGRAMMED BY ULF WICKSTROM*/* LUND FEB 1979*//1X,8C(1H*)///)
FORMAT(///* TITLE OF RUN : ",2OA4)
FORMAT(//* GEOMETRY*/1X,8(1H*)//
1 * MAXIMUM COORDINATES*,13X,*XMAX=*,G10+3,5X,*YMAX=*,G10+3)
1594
1595
1596
1597
              265
1598
              210
1599
```

```
FORMATC' MAXIMUM ELEMENT LENGTH',10X, "XBOX=",G10.3,5X, "YBOX="
1600
                             220
                                            FORMAT(' MAXIMUM ELEMENT LENGTH',1CX,'XBUX=',GTG.3,5X,'YBUX='
format(' SUBREGIONS')
FORMAT(' SUBREGIONS',I4//' SUBREGION DIAGONAL LIMITS'/
1 /4X,'XMIN',6X,'YMIN',6X,'XMAX',6X,'YMAX',6X,'FICTITIOUS AREA'//
2 (1X,4610.3,10X,L1))
FORMAT(//' COORDINATES OF SPECIFIED X - LINES')
FORMAT(//G1L.3)
FORMAT(/G1L.3)
F
1601
1502
                              230
1603
                             240
1504
1605
1606
1607
                             250
                             260
                             270
                                               FORMATC//* COORDINATES OF SPECIFIED Y - LINES*)
1608
                                              FORMAT(//* COURDINATES OF SPECIFIED T = LINES /
FORMAT(//*G1C.3)
FORMAT(///* THIS IS AN AXISYMMETRIC PROBLEM*//6X,*INNER RADIUS *,
*XMIN = *,G1C.3)
FORMAT(//* NUMBER AND COORDINATES OF X + LINES*//I3,* - *,7610.3/
1509
                              280
1510
1611
                             300
1512
                              310
                                              1 (6X,7610.3))
FORMAT(/' NUMBER AND COORDINATES OF Y-LINES'//I3,' - ',7610.3/
1613
1614
                                             1
                             320
                                                                (6x,7610.3))
1515
                                             1
                                               FORMAT(/' NUMBER OF NODES=', 14, 1Dx, 'NUMBER OF ELEMENTS=', 14)
1516
                              330
1617
                                               RETURN
                                               END
1618
1519
                             SUBROUTINE NGROUP(X,Y)
C----THIS ROUTINE READS AND FORMS NODE GROUP DATA
DIMENSION X(1),Y(1)
PARAMETER NB=1C,NNB=3C,NNB2=2*NNB
1620
1621
1522
 1623
                                            COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),
1 EPSG(NB),BETA(NB),CPG(NB),FA(NB)
1624
1625
                                               COMMON/ENRAD/E(10CO)
1525
1627
                                               COMMON/ENCON/H(50), TAIR(2)
1628
                                             LOGICAL FA
                             PRINT 200
READ 190,NGROUP
1629
1630
1631
1632
1633
                                             DO 10 I=1,NGROUP
                              C-----
1634
                                             READ 100,NCHECK,NUMB(I),EPSG(I),BETA(I),CPG(I)
1635
1636
                              C -----
1637
                                               IF(I.NE.NCHECK) GO TO 1000
1638
                                             NUMI=NUM9(I)
                             c----
1639
1540
                                             READ 100, (NBOUND(I,J), J=1, NUMI)
1541
                              (+---
                                              NOD1=NBOUND(I,1)
1542
1643
                                               DO 10 J≈2,NUMI
1644
                                               NOD2=NBOUND(I,J)
                                               BAREA(1, J)=SORT((X(NOD1)-X(NOD2))**2+(Y(NOD1)-Y(NOD2))**2)
1645
                                               NOD1=N0D2
1646
                             1C
                                              CONTINUE
1547
1548
                              С
                             C----PRINT INPUT DATA
1549
1650
                             c
                                               DO 15 I=1,NGROUP
1651
                                              NUMI=NUMB(I)
PRINT 210,I
1652
1653
                                               IF (EPSG(I).EG.C..AND.BETA(I).EQ.J.) GOTO 20
1654
                                             IF (EPSG(I),EU.C.AND.BETA(I),CPG(I)

PRINT 220,EPSG(I),BETA(I),CPG(I)

PRINT 230,(NBOUND(I,J),J=1,NUMI)

FORMAT(/' NODE GROUP'/IX,11(1+*))

FORMAT(/' NODE GROUP',I3)

FORMAT(' NODE GROUP',I3)

1 CONVECTION POWER=',G9.3/'

CONVECTION POWER=',G9.3)

FORMAT(' NODES',10I5/6X,10I5)
1655
                             20
200
1556
1657
1658
                             210
 1559
                             220
1660
                                            1
                             230
1661
                              15
                                               CONTINUE
1552
1663
                                               GOTO 1001
```

1000 PRINT 240 STOP RETURN 1558 EORMAT() 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 1676 DIMENSION X (NN), Y (NN), T (NN), TT (NN), TMAX (NN), FLOW (NN) PRINT 200, TITFIR, X(NN), Y(NN) TOUM1=1-IY 1679 DO 10 I=1, IX IDUM1=IDUM1+IY IDU#2=IDUM1+IY+1 IF(IY.6T.7) PRINT 210,(J,TT(J),J=IDUM1,IDUM2) IF(IY.6T.7) PRINT 230,(TT(J),J=IDUM1,IDUM2) 1 C CONTINUE FORMAT(* F , 18 F7.C) RETURN ENÐ 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 THIS RUBILINE PRINTS RUBAL TEMPERATURES AND VOLD 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(10G), XSYM(2), YSYM(2) COMMON/TOUT/II, TOUT(166), 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 1709 IF(TT(IJ).GT.1.001*T(IJ)) PRINT 200,IJ,TT(IJ),TIME1,DELTI IF(TT(IJ).GT.1.G01+T(IJ)) TMAX(IJ)=.TRUE. TT(IJ)=AMAX1(TT(IJ),T(IJ)) 1713 CONTINUE C----IF TIME=TOUT PRINT ALL TEMPERATURES IF((TIME-TOUT(II)).LT.-1.E-4) GOTO 70 PRINT 10C,TIME,KTIME,TFIRE,NODT 1717 IF(.NOT.LEN) GOTO 30 PRINT 30C DO 20 I=1,NENC PRINT 310,I,TAIR(I) 30 CONTINUE 11=11+1 TDUM1=1-IY LDUM=IY.LT.7 DO 10 I=1,IX IDUM1=IDUM1+IY TDUM2=TDUM1+IY-1

IF(LDUM) PRINT 210.(J.T(J).J=IDUM1.IDUM2) IF(.NOT.LDUM) PRINT 220.(T(J).J=IDUM1.IDUM2) 1728 1729 1730 1731 10 70 CONTINUE CONTINUE FORMAT(////1X,6(1H*),6H TIME,F8.3,2X,6(1H*),* INCREMENT*, 1 * NUMBEP*,I6, 22X,17(1H*)//* FIRE TEMPERATURE*,F7.0,2X,6(1H*) 1732 1733 100 1734 1735 1736 1737 Si2X; TIME INCREMENT LIMITING NODE'+15/)
FORMAT(///4H NOD:14:5X:8HMAX TEMP:F6:0:5X:*TIME':G10.4:5X;
1 5HDELTI:G10.4/1X:70(1HM)) 200 1738 1739 FORMAT(13(15,F5.0)) 210 FORMAT(18F7.0) FORMAT(' FNCLOSUPE AIR TEMPERATURE') FORMAT(' VOID NUMBER',12,' TAIR=',F5.0/) 220 1740 300 1741 310 1742 1743 RETURN END 1744 1745 SUBROUTINE 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) 1746 1 1747 C+==--C-----1748 1749 1750 *** TASEF *** č-----1751 1752 1753 C----TEMPERATURE ANALYSIS OF STRUCTURES EXPOSED TO FIRE č____ C----FINITE ELEMENT PROGRAM FOR ANALYSIS OF TRANSIENT NONLINFAR 1754 1755 C----HEAT TRANSFER PROBLEMS č-----C----PROGRAMMED BY 1756 C-----ULF WICKSTROM C-----LUND INSTITUTE OF TECHNOLOGY 1757 1758 1759 C----MARCH 1979 C----1760 C-----THIS IS THE MAIN CONTROL ROUTINE 1761 1762 1763 C----C----DEFINITIONS OF VARIABLES C----IX,IY NUMBER OF X 1764 NUMBER OF X- AND Y-LINES NUMBER OF NODES IN BASE STRUCTURE NUMBER OF ELEMENTS IN BASE STRUCTURF NUMBER OF REGIONS 1765 1766 C----NN C----NE C----NR 1767 1768 1769 C----N VECTOR OF REGION NUMBERS C-----N C-----NODEL C----NODEL C-----X,Y C-----T C-----T NODES ADJACENT TO EACH ELEMENT ELEMENTS ADJACENT TO EACH NOPF NUMBER ELEMENTS ADJACENT TO EACH NODE 1770 1771 1772 NODE COORDINATES 1773 CURRENT NODAL TEMPERATURES 1774 1775 1776 C----TMAX TRUE IF MAXIMUM NODAL TEMPERATURE OBTAINED HEAT CONDUCTION MATRIX HEAT CONDUCTION MATRIX HEAT CAPACITY VECTOR NODAL VOLUME VECTOR C-----ELA+EV4 C-----A 1777 1778 С----Р С----W 1779 1780 NODAL VOLUME VECTOR NODAL ENTHALPY VECTOR INTERNAL NODAL HEAT FLOW VECTOR EXTERNAL NODAL HEAT FLOW VECTOR TRUE IF AXISYMMETRIC PROBLEM INDICATES COUPLED NODES INDICATES INTERFACE NODES DUMMY VECTOR FOR CRITICAL TIME INCREMENT CALCULATION C----EN C----F C----FLOW C----AXIAL C----NODCPL 1781 1782 1783 1784 1785 1786 C----NODINT Č----DTA 1787 C----1788 1789 C----PARAMETER CONSTANTS c----1790 C----NB MAXIMUM NUMBER OF NODE GROUPS 1791

1792	CNNB	MAXIMU	M NUMBER OF NODES IN ONE NODE GROUP
1700		MAXIMI	IM NUMBER OF COUPLED GROUPS OF NODES
1705	CMN4K	MAXIMU	IN NUMBER OF REGIONS
1795	C=====MIN V	MAXIMU	M NUMBER OF VALUE PAIRS
1707	C	CIELDC	
1709		FIELUS	
1700	C	D474 0	
1800	(UATA (.	NUMBER OF CONDUCT OF NOTICE
1801	C	NCELO	MATRIX OF COUPLED BRODES
1602	CDTM		MATRIX OF CHUMLED NUDES
1803	(MAYNG	NAVIMUM NUMPER OF NORE CROUPS RECTAINS OVE ENGLASHOE
1804	(MAYNOD	MAXIMUM NUMBER OF NODES AROURS DEFINING UNE ENCLUSURY
1805	Change DUMMY	DEMMY	MATRICES
1806		ENCLOS	
1807		LEN	TRUE TE STRUCTURE CONTAINS VOID OR ENGLACING
1808	C	NENC	NUMBER OF ENCLOSURES
1809	Č	NENCNG	VECTOR OF NUMBER OF NODE GROUPS
1810	C	TGREN	MATRIX OF NOBEC
1811	Č	NNODEN	NUMBER OF NODES SURROUNDING AVOID
1812	Č	XSYM	TRUE TE VOID SYMMETRICAL ARROUND Y-AVIS
1813		YSYM	TRUE TE VOID SYMMETRICAL ARROUND Y-AYIS
1814	CENCON	FNCLOS	URE CONVECTION DATA
1815	C	н	ARRAY OF ENCLOSURE CONVECTION VECTORS
1816	č	TAIR	ENCLOSURE ATR TEMPERATURE
1817	CENRAD	ENCLOS	URE RADIATION DATA
1818	C	ε	ARRAY OF ENCLOSURE RADIATION MATRICES
1819	CFIRE	FIRE T	EMPERATURE DATA
1820	C	TIM, TP	TIME - FIRE TEMPERATURE PAIRS
1821	(TITFIR	FIRE IDENTIFIER
1822	CF08	PRESRI	BED HEAT FLOW DATA
1823	C	NFQNG	NUMBER OF NODE GROUPS DEFINING PRESCRIBED FLOW BOUNDABIES
1824	C	NFQG	VECTOR OF NODE GROUPS OFFINING PRESCRIBED FLOW
1825	C	TRATC	VECTORS OF MODIFIED TEMPERATURE
1826	C	BR+BC	RADIATION AND CONVECTION BOUNDARY MATRICES
1827	CBNOD	DATA O	N NODE GROUPS
1828	C	NUMB	VECTOR OF NUMBER OF NODES IN THE NODE GROUPS
1829	C	NBOUND	MATRIX OF NODE NUMBERS IN THE NODE GROUPS
1830	C	BAREA	MATRIX OF DISTANCES BETWEEN NODES
1831	C	EPSG	VECTOR OF EMISSIVITY OF NODE GROUPS
1832	C	BETA	VECTOR OF CONVECTION FACTORS OF NODE GROUPS
1835	Ç	CPG	VECTOR OF CONVECTION POWERS OF NODE GROUPS
1834	C	FA SECTO	TRUE FOR FIRE BOUNDARY NODE GPOUPS
1835	C18	PRESCR.	IBED TEMPERATURE
1030	(NUMBER OF NODE GROUPS DEFINING PRESCRIPED TEMPERATURE
1037		NPIG CONCT	VECTOR OF NODE GROUPS DEFINING PRESCRIBED TEMPERATORES
1830	C	FUETOT	RIC DATP TOUE FOR FICTITIOUS FLEMENTS
1840	C		TRUE FOR FICHTINGS ELEMENTS
1841	(ELEMENT TRANSFOR
1842	CessesRMAT	MATERT	AF DATA
1843	Caseses	000	TRUE IE CONDUCTIVITY IS FUNCTION MAXIMUM TEMPERATURE
1844	č	TC+C	TEMPERATURE - CONDUCTIVITY PATRS
1845	Č	TEFENT	TEMPERATURE - SPECIFIC VOLUMETRIC ENTHALPY PAIRS
1846	<u> </u>	CR N	IOMINAL SPECIFIC VOLUMETRIC HEAT
1847	č	TQ,QE	EMPERATURE - INTERNALLY GENERATED HEAT PATRS
1848	C	LQ	TRUE IF INTERNAL HEAT IS GENERATED
1849	CTOUT	TIME DA	ATA
1850	C	II	COUNTER
1851	C	TOUT N	ECTOR OF PRINT OUT TIMES
1852	C	TIMMOX N	AXIMUM TIME
1853	C	DTMAX N	AAXIMUM TIME INCREMENT
1854	C	TIMFAC 1	IME INCREMENT FACTOR
1855	C	KTMAX N	MAXIMUM NUMPER TIME INCREMENTS

NUMBER OF TIME STEPS BETWEEN UPDATING CONDUCTION MATRIX 1856 KUPD^ (-----UNIT DEPENDENT CONSTANTS 1857 C----UNIT 1858 STEFAN-BOLTZMANN CONSTANT č----STGMA Č-----ABSOLUTE TEMPERATURE SHIFT INITIAL TEMPERATURE 1859 TABS Č----TINIT 1860 c----1861 DIMENSION N(NE), KTOP(4, NE), X(NN), Y(NN), T(NN), TT(NN), TMAX(NN), 1862 ELA(4+NE)+EV4(NF)+4(NN+MAX)+P(NN)+FN(NN)+F(NN)+FLOW(NN)+ 1863 NODCPL (NN) + NODINT (NN) + W (NN) + NODEL (4+NN) + MNODEL (NN) + DTA (NN) 1864 2 LOGICAL TMAX, AXIAL C----FORM THE VECTOR N 1865 1866 1867 CALL REG2(NN+NE+NR+N+KTOP+X+Y+NODEL+MNODEL) ---DEFINE COUPLED NODES CALL COUPLA(NODCPL+NN+NODINT) 1868 1869 --- DEFINE INTERFACE NODES 1670 1871 CALL INTEPF(NN:NE:NR:IX,IY,KTOP:N:NODINT:NODCPL) 1872 CALL MAT(NR) --FORM GEOMETRICAL DUMMY CONSTANTS 1873 1874 CALL GEOCO2(NN,NE,N,KTOP,X,Y,AXIAL,ELA,FV4) --INPUT INITIAL DATA 1875 1876 1877 CALL AMB ---FORM NODE GROUPS CALL NGROUP(X+Y) 1878 1879 ---DEFINE FIRE PRESCRIBED HEAT FLOW BOUNDARIES 1880 c-CALL FOBNDA 1881 ---DEFINE FIRE PRESCRIBED TEMPERATURE BOUNDARIES 1882 CALL PTBNDA(NODCPL) -DEFINE ENCLOSURE BOUNDARIES 1883 1884 с-CALL ENCLOI(X,Y) 1885 ---INPUT TIME DATA 1886 1887 CALL TIME ---FORM NODE VOLUME VECTOR 1888 C-CALL ASSW2 (NN, NE, N, KTOP, X, Y, EV4, AXJAL, W) 1889 ---SUMMERIZE APPROPRIATE NODE VOLUMES 1890 -SUMMERIZE AFFRANCES HELE CALL COUPLB(W) -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) 1891 1892 C---1893 1894 1 RETURN 1895 1896 END 1897 SUBROUTINE PTBNDA(NODCPL) C----INPUT NODE GROUPS OF PRESCRIBED TEMPERATURE 1898 1899 DIMENSION NODCPL(1) 1900 PARAMETER NB=10,NNB=30,NNB2=2*NNB 1901 1902 COMMON/PTB/NPTNG+NPTG(NB) COMMON/BNOD/NUMB(NB), NBOUND(NB, NNB), BAREA(NB, NNB), TH(NP), EPSG(NB), BETA(NB), CPG(NB), FA(NB) 1903 1904 1 LOGICAL FAIFA1 C----READ NUMBER OF BOUNDARY NODES GROUPS 1905 1906 READ 100+MPTMG IF(NPTNG.EQ.0) RETURN 1907 1908 1909 PRINT 200 1910 ---EACH PRESCRIBED TEMPERATURE BOUNDARY NODE GROUP C 1911 DO 20 IB=1.NPTNG ---1912 C-READ 100, FA1, ING1 1913 C-----FA1 = TRUE FILE BOUNDARY ELSE AMBIENT TEMPERATURE C-----ING1 = NODE GROUP NUMBER 1914 1915 1916 FA(ING1)=FA1 NUMI=NUMB(ING1) 1917 NPTG(I8)=ING1 1918 1919 DO 10 J=1,NUMI
1920 1921 1922 1923 1924 1925	10	NOD=NBOUND(ING1,J) IF (NODCPL(NOD).E0.0) PRINT 300,NOD IF (NODCPL(NOD).E0.0) STOP NODCPL(NOD)=0 IF(FA1) PPINT 210,ING1 IF(.NOT.FA1) PRINT 220,ING1
1926	20	CONTINUE
1927	103	FORMAT()
1928	210	FORMAT(* NODE GROUP*+I3+* FIRE BOUNDARY*)
1929	220	FORMAT(NODE GROUP', I3, * AMBIENT BOUNDAPY')
1930	200	FORMAT(//* PRESCRIBED TEMPERATURE BOUNDARY*/1X+31(1H*)//
1931	-	1 ' NODE GROUPS AND TYPES OF BOUNDARIES'/)
1932	300	FORMAT(/ NODE' + 14 + 15 & SLAVE NODE'/
1933		1 ' SLAVE NODES CANNOT HAVE PRESCRIBED TEMPERATURE!)
1934		RETURN
1035		END
1036		
1930		
1937	~	
1938	Ç	-SET PRESCRIBED NUMAE BOUNDART TEMPERATORE
1939		DIMENSION I(1)
1940		PARAMETER NB=10+NNH=30+NNB2=2*NNB
1941		COMMON/PTB/NPTNG,NPTG(NB)
1942		COMMON/BNOD/NUM8(NB)+NBOUND(NB+NNB)+BAREA(NB+NNB)+TH(NB)+
1943		1 EPSG(NB)+BETA(NB)+CPG(NB)+FA(NB)
1944		COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
1945		LOGICAL FA
1946	C	-
1947		IF (NPTNG, EQ. 0) RETURN
1948	C	-EACH PRESCRIBED TEMPERATURE BOUNDARY NODE GROUP
1949		DO 10 IB=1,NPTNG
1950		TG=TAMB
1951		ING1=NPTG(IB)
1952		IF(FA(ING1)) TG=TFIRE
1953		NUMI=NUMB(INGI)
1954		DO 10 I=1+NUMI
1955		NODE=NBOUND(ING1+I)
1956		T(NODE)=TG
1957	10	CONTINUE
1958		RETURN
Ĩ959		END
1960		
1961		SUBROUTINE RADVEC(F,ETA+N+Q)
1962	C	-THIS ROUTINE FORMS THE LOCAL ENCLOSURE SURFACE RADIATION HEAT
1963	C	-EXCHANGE VECTOR G=E*ETA
1964		DIMENSION Q(1)+ETA(1)+E(N+N)
1965		QTOT=0.
1966		DO 20 I=1+N
1967		QT=0.
1968		D0 10 J=1+N
1969	10	QT=QT+E(I,J)*ETA(J)
1970		QTOT=QTOT+QT
1971	20	(I)=9T
1972		RETURN
1973	220	FORMAT(/' TOTAL RADIATION HEAT EXCHANGE'+E11.3)
1974		ÉND
1975		
1976		SUBROUTINE REG2(NN, NE, NR, N, KTOP, X, Y, NODEL, MNODEL)
1977	C	-THIS SUBROUTINE FORMS VECTOR OF REGION NUMBERS N OF EACH ELEMENT
1978		DIMENSION X(NN)+Y(NN)+N(NE)+KTOP(4+NE)+NODEL(4+NN)+MNODEL(NN)
1979		PARAMETER MNR=10
1980		COMMON/RGEO/ELFICT(MNR)+ET(MNR)+SRDIAC(4+MNR)
1981		LOGICAL ELFICT
1982		EPS=1.E-7
1983		DO 10 I=1+NE

```
N(I)=1
1984
                            IF(NR.EQ.1) GOTO 10
1985
                            NO1=KTOP(1+I)
1986
                            ND2=KTOP(4+1)
1987
                           D0 5 J=2.NR
IF((X(ND1)-SRDIAC(3,J)).AT.-EPS) GOTO 5
1988
1989
                           IF((Y(ND1)-SRDIAC(4,J)).GT.-EPS) GOTO 5
IF((X(ND2)-SRDIAC(1,J)).LT.EPS) GOTO 5
1990
1991
                            IF((Y(ND2)-SPDIAC(2,J)).LT.EPS) GOTO 5
1992
1993
                            N(I)=J
1994
1995
                           CONTINUE
                 5
10
1996
                            00 40 I=1,NN
1997
                            II=0
                           DO 30 IE=1.NE
N1=N(IE)
1998
1999
2000
                            IF(ELFICT(N1)) GOTO 30
2001
                            DO 20 J=1+4
                            IF(KTOP(J,IE).NE.I) GOTO 20
2002
2003
                            II=II+1
                            NODEL(II,I)=IE
2004
                            IF(II.EQ.4) GOTO 30
2005
2006
                            CONTINUE
                 50
2007
                 30
                            CONTINUE
                            MNODEL(I)=II
2009
                 40
                            CONTINUE
2010
2011
                            RETURN
                            END
2012
                 SUBROUTINE TIME
C----READ TIME INTEGRATION CONTROL DATA
COMMON/TOUT/II.TOUT(100).TIMMAX.DIMAX.TIMFAC.KTMAX.KUPDA
2013 2014
2015
                           PRINT 200
2016
2017
                 C-----
                           READ 100+NT+TIMMAX+DTMAX+TIMFAC+KTMAX+KUPDA
2018
2019
                 C-----
                            IF(DTMAX.EQ.0) DTMAX=TIMMAX
2020
                           IF(11MFAC.EQ.0) TIMFAC=.8
IF(KTMAX.EQ.0) KTMAX=1000
IF(KUPDA.EQ.0) KUPDA=1
2021
2022
2024
                 C----
                           READ 100, (TOUT(I), I=1,NT)
2025
2026
                 C----
                           PRINT 210.TIMMAX.DTMAX.TIMFAC.KTMAX.KUPDA
PRINT 220.(TOUT(I).I=1.NT)
2027
2028
                           FORMAT()
FORMAT()
FORMAT(' PRINT OUT TIMES',3X,8G7.2/(19X,8G7.2))
FORMAT(//' TIME'/' ****'/)
FORMAT(//' TIME'/' ****'/)
2029
                 100
2030
                 220
                 200
210
                           FURMAT(//: fIME''' ****')
FORMAT(' MAXIMUM TIME='',G8.3/' MAXIMUM TIME INCREMENT='',G8.3/
L ' CRITICAL TIME INCREMENT FACTOR='',G8.3/
2 ' MAXIMUM NUMBER OF TIME INCREMENTS='',I5/
3 ' NUMBER OF STEPS BETWEEN UPDATING OF CONDUCTION MATRIX=',I5)
2031
2032
2033
2034
2035
                          2
                          3
2036
2037
                           RETURN
                            END
2038
                 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,NNR=30,NNB2=2*NNB
COUNDER (ND), NDD(ND), PAREA (ND, NND), TH(ND),
2039
2040
2041
2042
2043
                          COMMON/ENCD/NUMB(NR), NBOUND(NB, NNR), BAREA(NB, NNB), TH(NB),

1 EPSG(NB), FETA(NB), CPG(NB), FA(NB)

COMMON/ENCLOS/LEN, NENC, NENCNG(2), TGREN(2,4), NNODEN(2), TNODEN(100),
2044
2045
2046
                                XSYM(2) + YSYM(2)
2047
                          1
```

2054 CCORPUTE VIEW-FACTOPS USING HOTTEL+5 CROSSED-STRING METHOD 2055 NEMGENMENCROIEN 2056 SIGNX221. 2057 SIGNX221. 2058 IN-D0 2059 CCORPUTE VIEW-FACTOPS USING HOTTEL+5 CROSSED-STRING METHOD 2057 SIGNX221. 2058 IN-D0 2059 CEACH NODE GROUP 2062 IF(XSYM(EN).AND.YSYM(EN)) DSYM=.TRUE. 2063 Il=IGREN(FN.IG) 2064 NUMI-NUMB(I1) 2065 CEACH ZONE 2066 NODI-TE2.NUMI 2067 IN-TIN+1 2068 NOD2=NBOUND(11.F1) 2070 X12X(NOD1) 2071 X22X(NOD2) 2072 Y12Y(NOD1) 2073 Y22Y(NOD2) 2074 D1=BAREA(11.J) 2075 CEACH NODE GROUP 2076 D(10 JG-TI-NEM6 2080 J1-TGREN(EN.JG) 2081 NUMJ-NUMR(J1) 2082 CEACH ZONE 2084 JNUJ-NUME 2085 D2-BAREA(11.J) 2086<	2048 2049 2050 2051 2052		LOGICAL LEN LOGICAL LDUM,XSYM,YSYM,DSYM INTEGER EN DO 15 I=1,MAXNOD DO 15 J=1,MAXNOD
2056 SIGNX2=1. 2057 SIGNX2=1. 2058 IN=0 2059 CEACH NODE GROUP 2060 D0 100 IG=1.NENG 2061 D5YM=.FALSE. 2062 IF(XSYM(EN).AND.YSYM(EN)) DSYM=.TRUE. 2063 II=IGRENT(FN.IG) 2064 NUMI=NUMB(II) 2065 CEACH ZONE 2066 NODI=NBOUND(II.I=1) 2066 NODI=NBOUND(II.I=1) 2070 X1=X(NODI) 2071 X2=X(NODI) 2072 Y1=Y(NODI) 2073 Y2=Y(NOD2) 2074 D1=SAREA(II.I) 2075 CFORM THE ZONE AFEA VECTOR D 2076 OIN)=D1 2076 OIN)=D1 2077 JN=0 2078 CEACH NODE GROUP 2079 D0 100 JG=1.NENG 2080 011=GRENT(EN.JG) 2081 NUMI=NUMB(I) 2082 CFACH XODE GROUP 2083 D0 100 JG=1.NENG 2084 UX=JM+1 2085 D2=BAREA(JI.J) 2084 011=GRENT(EN.JG) 2084 0258AREA(JI.J) 2084 0258AREA(JI.J) 2085 D2=BAREA(JI.J) 2086 NOD3=NBOUND(JI.J) 2086 NOD3=NBOUND(JI.J) 2086 NOD3=NBOUND(JI.J) 2086 NOD3=NBOUND(JI.J) 2086 NOD3=NBOUND(JI.J) 2086 00 JI=GRENT(EN.JG) 2087 00 JZ=1.NENG 2080 01 JZ=GRENT(EN.JG) 2084 0258AREA(JI.J) 2085 02 CFACH 20NE 2085 02 CFACH 20NE 2086 NOD3=NBOUND(JI.J) 2086 NOD3=NBOUND(JI.J) 2086 NOD3=NBOUND(JI.J) 2086 NOD3=NBOUND(JI.J) 2087 NOD4=NBOUND(JI.J) 2088 IF(XSYM(EN))SIGNX2=-1. 2090 50 CONTINUE 2091 JDUMESIGNX2=F0.1.ANN.SIGNY2.E0.1. 2092 IF(I.NGT.LDUM) GOTO 80 2093 B0 CONTINUE 2093 JSGT(XI.Y=X)+=2+(YZ-Y3)+=2) 2094 B0 CONTINUE 2095 XI=SIGNY2*Y(NOD3) 2094 B0 CONTINUE 2095 XI=SIGNY2*Y(NOD3) 2095 Y=SIGNY2*Y(NOD3) 2096 D3=SRT((XZ-X)+=2+(YZ-Y3)+=2) 2010 D4=SIGNY2*Y(NOD3) 2097 Y=SIGNY2*Y(NOD3) 2097 JSGT(XI.Y=X)+=2+(YZ-Y3)+=2) 2010 D4=SIGNY2*Y(NOD3) 2097 JSGT(XI.Y=X)+=2+(YZ-Y3)+=2) 2010 D4=SGRT(XZ-X)+=2+(YZ-Y3)+=2) 2010 D4=SIGNY2*Y(NOD3) 2037 Y=SIGNY2*Y(NOD3) 2037 Y=SIGNY2*Y(NOD3) 2038 C	2053 2054 2055	15 C	VIEW(I;J)=9. •COMPUTE VIEW-FACTORS USING HOTTEL'S CROSSED-STRING METHOD NENGENENCNG(EN)
<pre>2059 C====EAC: NODE C=0.00 D0 10 IG=L:NENG 2061 D5YME.FALSE. 2062 IF (X5YM(EN).AND.Y5YM(EN)) D5YM=.TRUE. 2063 II=IGREN(EW.IG) 2064 NUMI=NUMG(I) 2066 D0 10 IZ=NUMI 2066 NODI=NBOUND(I].I=1) 2066 NODE=NBOUND(I].I=1) 2066 NODE=NBOUND(I].I=1) 2070 X1=X(NODE) 2071 X2=X(NODE) 2072 Y1=Y(AODE) 2073 Y2=Y(AODE) 2074 D1=BAREA(I].I] 2075 C=====FORM THE ZONE AFEA VECTOR D 2076 C====FORM THE ZONE AFEA VECTOR D 2077 JN=0 2078 C=====EACH XONE GROUP 2079 D= 100 JG=I.NENG 2080 J=IEGREN(EN.JG) 2081 NUMJ=NUMG(J) 2082 C====EACH XONE GROUP 2083 D= 100 JG=I.NENG 2084 JN=JM+1 2085 D=2=BAREA(J).J] 2085 D=2=BAREA(J).J] 2086 NOD3=NBOUND(J].J] 2086 NOD3=NBOUND(J].J] 2087 NOD=NENGUND(J].J] 2088 IF (X5YM(EN))SIGNY2==1. 2099 JF (TSYM(EN))SIGNY2==1. 2090 S0 CONTINUE 2093 IF (IN.GE.JN) GOTO 100 2094 B0 CONTINUE 2093 IF (IN.GE.JN) GOTO 100 2094 B0 CONTINUE 2095 X3=SIGNY2*X(NOD3) 2096 Y4=SIGNY2*Y(NOD4) 2096 Y4=SIGNY2*Y(NOD4) 2096 Y4=SIGNY2*Y(NOD4) 2096 Q=====SAT(X2)************************************</pre>	2056 2057 2058	c	SIGNX2=1. SIGNY2=1. IN=0
2662 IF (XSYM(EN).AND.YSYM(EN)) PSYME.TRUE. 2663 ILIGREN(EN).G0 2664 NUMI=NUMB(I1) 2665 CECAC ZONE 2666 D0 100 IE2.NUMI 2667 INSIN-1 2668 NODI=NBOWND(I1.I-1) 2669 NODE=NBOWND(I1.I-1) 2669 NODE=NBOWND(I1.I-1) 2670 X12X(NOD2) 2071 X22X(NOD2) 2072 Y12Y(NOD1) 2073 Y22Y(NOD2) 2074 D13BAEA(II.I) 2075 CFORM THE ZONE AFEA VECTOR D 2076 D(10.JOET.NENG 2080 J101GFEN(EN).JG) 2079 D0 100 JGE.NEWG 2081 NUMJ=NUMM(J1) 2082 CFORM THE ZONE AFEA VECTOR D 2084 J0 100 JE2.NUMJ 2085 D2-BAPEA(J1.J.J) 2086 NOD=NEOUND (J1.J.J) 2086 NOD=NEOUND (J1.J.J) 2087 NOD=NEOUND (J1.J.J) 2088 D2-BAPEA(JJ1.G) SIGNX2=-1. 2089 IF (YSYM(EN) SIGNY2	2059 2060 2061	L	DO 100 IG=1.NENG DSYM=.FALSE.
2005 CEACH ZONE 2066 D0 100 IT=2*NUMI 2067 IN=IN+1 2068 NODI=NBOUND(II:I) 2070 X1=X(NODI) 2071 X2=X(NOD2) 2072 Y1=Y(NOD1) 2073 Y2=Y(NOD2) 2074 DI=SAREA(I:I) 2075 CFORM THE ZONE AREA VECTOR D 2076 D(IN)=D1 2077 JN=0 2078 CFORM THE ZONE AREA VECTOR D 2079 D0 100 JG=1.NENG 2080 JI=EREN(EN.JG) 2081 NUMJ=NUMB(J1) 2082 CEACH ZONE 2083 D0 100 JG=2*NUMJ 2084 JN=JN+1 2085 D2=BAREA(J1.J) 2086 D0 CONTINUE 2087 CONTINUE 2091 LDUM=SIGNY2*EG.1AND*SIGNY2*EG.1. 2092 IF (NOT*LOUM) GOTO 100 2093 IF (IN*6E-JM) GOTO 100 2094 CONTINUE 2095 X3=SIGNY2*Y (NOD3) 2096 GaSGRT(2062 2063 2064		IF(XSYM(EN).AND.YSYM(EN)) DSYM=.TRUE. I1=IGREN(EN.IG) NEMI=NIMM(I1)
2067 IN-IN+1 2068 NOD1=NBOUND(I1,I-1) 2069 NOD2=NBOUND(I1,I) 2071 X1=X(NOD1) 2072 Y1=Y(NOD2) 2072 Y1=Y(NOD2) 2073 V2=Y(NOD2) 2074 D1=BAREA(I1.I) 2075 CFORM THE ZONE AREA VECTOR D 2076 D(IN)=D1 2077 JN=0 2078 CEACH MODE GROUP 2079 D0 100 JG=1,NENG 2080 J1=IGREN(EN,JG) 2081 NUMJ=NUMM(J1) 2082 CEACH ZONE 2083 D0 100 J=2;NUMJ 2084 JN=JN+1 2085 D2=BAREA(J1,J) 2086 NOD3=NBOUND(J1,J) 2086 NOD3=NBOUND(J1,J) 2086 NOD3=NBOUND(J1,J) 2086 IF(XSTM(EN))SIGNY2=-1. 2090 50 CONTINUE 2091 L0UM=SIGNY2,E0,1,AND.SIGNY2,E0,1. 2092 IF(N,NCT.LDUM) GOTO 80 2094 B0 CONTINUE 2095 X3=SIGNY2*Y(NOD3) 2096 X3=SIGNY2*Y(NOD3) 2096 Z3=SIGNY2*Y(NOD4) 2097 Y3=SIGNY2*Y(NOD4) 2096 J3=SGRT((X1=X3)**2+(Y1=Y3)**2) 2096 D3=SGRT((X1=X3)**2+(Y1=Y3)**2) 2096 C	2065 2066	c	DO 100 I=2+NUMI
2030 X1=X(NOD1) 2071 X2=X(NOD2) 2072 Y1=Y(NOD1) 2073 Y2=Y(NOD2) 2074 D1=SAREA(IL.1) 2075 CFORM THE ZONE AREA VECTOR D 2076 D(IN)=D1 2077 JN=0 2078 CEACH NODE GROUP 2079 D0 100 JG=1,NEN6 2081 NUMU=NUMR(J1) 2082 CEACH ZONE 2084 JN=INFENKEN.JG) 2084 JN=INFENKEN.JG) 2084 JN=INFENKEN.JG) 2085 D2=BAREA(J1.J) 2086 NOD=X=NEUMOI(J1.J=1) 2087 NOD=NBOUND(J1.J=1) 2088 IF (YSM(EN)) SIGNY2=-1. 2099 IF (SYM(EN)) SIGNY2=-1. 2091 DUM=SIGNX2*EG+1AND.SIGNY2-EG+1. 2092 IF (IN.0EC.JN) @ OTO 80 2093 IF (IN.0EC.JN) @ OTO 100 2094 B0 CONTINUE 2095 X3=SIGNX2*X(NOD3) 2096 Y4=SIGNY2*Y(NOD3) 2097 Y3=SIGNY2*Y(NOD3) 2098 Y4=SIGNY2*Y(NOD3)	2067 2068 2069		IN=IN+1 NODI=NBOUND(II,I-1) NOD2=NBOUND(II,I)
2072 Y1=Y(NOD1) 2073 Y2=Y(NOD2) 2074 D1=SAREA(I1.I) 2075 CFORM THE ZONE APEA VECTOR D 2076 D(IN)=D1 2077 JN=0 2078 CEACH NODE GROUP 2080 J1=IGREN(EN.JG) 2080 J1=IGREN(EN.JG) 2080 D0 J00 J=2.NUMJ 2082 CEACH ZONE 2083 D0 100 J=2.NUMJ 2084 JN=JN+1 2085 D2=BAREA(J1.J) 2086 NOD3=NBOUND(J1.J-1) 2086 IF (XSYM(EN))SIGNY2=-1. 2089 S0 CONTINUE 2091 L0UM=SIGNX2.E0.1.AND.SIGNY2.E0.1. 2092 IF (NOT.LDUM) GOTO 80 2093 OCONTINUE 2094 S0 CONTINUE 2095 X3=SIGNY2*X(NOD4) 2096 Y4=SIGNY2*X(NOD4) 2096 Y4=SIGNY2*Y(NOD3) 2096 Y4=SIGNY2*Y(NOD4) 2097 Y3=SIGNY2*Y(NOD4) 2097 Y3=SIGNY2*Y(NOD4) 2096 Z4=SGRT((X2-X3)**2+(Y2-Y4)**2) 2100 D4=SGRT((X2-X4)**2+(Y2-Y4)**2) 2101 D5=SGRT((X2-X4)**2+(Y2-Y4)**2) 2102 D6=SGRT((X2-X3)**2+(Y2-Y3)**2) 2103 DUM=ABS(D5+D6-D4-D3)/2. 2104 CTAKE ADVANTAGE OF RECIPROCITY 2105 VIEW(IN.JN)=DUM/D1+VIEW(IN.JN) 2106 CTAKE ADVANTAGE OF RECIPROCITY 2108 CTAKE ADVANTAGE OF RECIPROCITY 2109 L0UM=SIGNY2+1. 2100 D4=SGRT((X2-X3)**2+(Y2-Y3)**2) 2103 CUM=ABS(D5+D6-D4-D3)/2. 2104 CTAKE ADVANTAGE OF RECIPROCITY 2105 VIEW(IN.JN)=DUM/D1+VIEW(IN.JN) 2106 CTAKE ADVANTAGE OF RECIPROCITY 2108 CTAKE ADVANTAGE OF RECIPROCITY 2109 L0UM=(SIGNY2=1. 2100 IF(L0UM) SIGNY2=1.	2070 2071		x1=x(NOD1) x2=x(NOD2)
2075 CFORM THE ZONE AREA VECTOR D 2076 D(IN)=D1 2077 J.H=0 2078 CFORM THE ZONE AREA VECTOR D 2079 D0 100 JG=1,NENG 2080 J1=GREN(EN.JG) 2081 NUMJ=NUMB(J1) 2082 CEACH ZONE 2083 D0 100 J2=,NUMJ 2084 JN=JN+1 2085 D2=BAREA(J1,J) 2086 NOD3=NBOUND(J1,J-1) 2086 NOD3=NBOUND(J1,J) 2086 NOD3=NBOUND(J1,J) 2088 IF (YSYM(EN))SIGNY2=-1. 2089 IF (YSYM(EN))SIGNY2=-1. 2090 50 CONTINUE 2091 LOUM=SIGNY2.EQ.1.AND.SIGNY2.EQ.1. 2092 IF (.N.OT.LOUM) GOTO 80 2094 80 CONTINUE 2095 X3=SIGNY2*Y(NOD3) 2096 Y4=SIGNY2*Y(NOD4) 2097 Y3=SIGNY2*Y(NOD4) 2097 Q3=SGRT((X1-X3)**2+(Y1-Y3)**2) 2100 D4=SGRT((X1-X4)**2+(Y1-Y4)**2) 2100 D4=SGRT((X1-X4)**2+(Y1-Y4)**2) 2101 D5=SGRT((X1-X4)**2+(Y1-Y4)**2) 2102 D6=SGRT((X1-X4)**2+(Y1-Y4)**2) 2103 DUM=ABS(D5+D6-D4-D3)/2. 2104 CHOTTEL'S CROSSED=STRING METHOD 2105 VIEW(IN.JN)=DDM/D1+VIEW(IN.JN) 2106 CTAKE ADVAITAGE OF RECIPROCITY 2107 IF (LDUM) VIEW(JN,IN)=DUM/D2 2108 CIF SYMMETRY ARROUND AXIS GO RACK WITH CHANGED SIGNS OF COORDINATES 2109 IF (LDUM) SIGNY2=1. 2100 IF (LDUM) SIGNY2=1.	2072 2073 2074		Y1=Y(NOD1) Y2=Y(NOD2) D1=BARF4(11.1)
2077 JNED 2078 CEACH NODE GROUP 2079 D0 100 JGE1.NENG 2080 J1EIGREN(EN.JG) 2081 NUMJENUMB(J1) 2082 CEACH ZONE 2083 D0 100 JE2.NUMJ 2084 JNEJN+1 2085 D2=BAREA(J1.J) 2086 NOD3=NBOUND(J1.J-1) 2086 NOD3=NBOUND(J1.J) 2088 IF(XSYM(EN))SIGNY2=-1. 2090 50 CONTINUE 2091 LDUM=SIGNX2.EG.1AND.SIGNY2.EG.1. 2092 IF(N.NGE.JN) GOTO 100 2094 80 CONTINUE 2095 X3=SIGNX2*X(NOD3) 2096 X4=SIGNX2*X(NOD4) 2097 Y3=SIGNY2*Y(NOD3) 2098 Y4=SIGNY2*Y(NOD4) 2099 D3=SQRT((X1-X3)**2+(Y1-Y3)**2) 2100 D4=SQRT((X1-X3)**2+(Y2-Y4)**2) 2101 D5=SQRT((X1-X4)**2+(Y2-Y4)**2) 2102 D6=SQRT((X1-X4)**2+(Y2-Y3)**2) 2103 DUM=ASIGN54D=Oh-D3)/2. 2104 CHOTTEL'S CROSSED=STRING METHOD 2105 VIEW(IN.JN)=DUM/D1+VIEW(IN.JN) 2106 CTAKE ADVANTAGE OF RECIPROCITY 2107 IF(LDUM) VIEW(JN.IN)=DUM/D2 2108 CFS YMMETRY ARROUND AXIS GO RACK WITH CHANGED SIGNS OF COORDINATES 2109 IF(LDUM) SIGNY2=1. 2100 IF(LDUM) SIGNY2=1. 2101 IF(LDUM) SIGNY2=1. 2102 D45-SQRT ARROUND AXIS GO RACK WITH CHANGED SIGNS OF COORDINATES 2109 IF(LDUM) SIGNY2=1. 2100 IF(LDUM) SIGNY2=1. 2101 IF(LDUM) SIGNY2=1. 2102 IF(LDUM) SIGNY2=1. 2103 IF(LDUM) SIGNY2=1. 2104 IF(LDUM) SIGNY2=1. 2105 IF(LDUM) SIGNY2=1. 2105 IF(LDUM) SIGNY2=1. 2106 IF(LDUM) SIGNY2=1. 2107 IF(LDUM) SIGNY2=1. 2109 IF(LDUM) SIGNY2=1. 2100 IF(LDUM) SIGNY2=1. 2101 IF(LDUM) SIGNY2=1. 2102 IF(LDUM) SIGNY2=1. 2103 IF(LDUM) SIGNY2=1. 2104 IF(LDUM) SIGNY2=1. 2105 IF(LDUM) SIGNY2=1. 2105 IF(LDUM) SIGNY2=1. 2106 IF(LDUM) SIGNY2=1. 2107 IF(LDUM) SIGNY2=1. 21	2075 2076	c	FORM THE ZONE AREA VECTOR D D(IN)=D1
2080 J1=IGREN(EN,JG) 2081 NUMJ=NUMB(J1) 2082 CEACH ZONE 2083 D0 100 J=2,NUMJ 2084 JN=JN+1 2085 D2=D8REA(J1,J) 2086 NOD3=NBOUND(J1,J-1) 2087 NOD4=NBOUND(J1,J) 2088 IF(XSYM(EN))SIGNY2=-1. 2089 IF(YSYM(EN))SIGNY2=-1. 2091 LDUM=SIGNY2.E0.1AND.SIGNY2.E0.1. 2092 IF(IN.GE_JN) GOTO 80 2093 IF(IN.GE_JN) GOTO 100 2094 S0 2095 X3=SIGNY2*X(NOD3) 2096 X4=SIGNY2*X(NOD4) 2097 Y3=SIGNY2*Y(NOD4) 2098 Y4=SIGNY2*Y(NOD4) 2099 D3=SGRT((X1-X3)**2+(Y1-Y3)**2) 2100 D4=SGRT((X2-X3)**2+(Y1-Y4)**2) 2101 D5=SGRT((X2-X3)**2+(Y2-Y4)**2) 2102 D6=SGRT((X2-X3)**2+(Y1-Y4)**2) 2103 DUM=ABS(D5+D6-D4-D3)/2. 2104 CHOTTEL'S CROSSED=STRING METHOD 2105 VIEW(IN,JN)=DUM/D2 2104 CHOTTEL'S CROSSED=STRING METHOD 2105 VIEW(IN,JN)=DW/D2	2077 2078 2079	C	JN=U •EACH NODE GROUP DO 100 JG=1,NENG
2082 CLEMPLEACH ZUNE 2083 D0 100 J=2*NUMJ 2084 JN=JN+1 2085 D2=BAREA(J)*J) 2086 NOD3=NBOUND(J)*J=1) 2087 NOD4=NBOUND(J)*J=1) 2088 IF(XSYM(EN))SIGNY2==1. 2090 50 CONTINUE 2091 LDUM=SIGNX2*EQ*1.*AND*SIGNY2*EQ*1. 2092 IF(N*KEN))SIGNX2==1. 2093 IF(IN*GE*JN) GOTO 100 2094 80 CONTINUE 2095 X3=SIGNX2*X(NOD3) 2096 X4=SIGNX2*X(NOD4) 2097 Y3=SIGNY2*Y(NOD3) 2098 Y4=SIGNY2*Y(NOD3) 2099 D3=SQRT((X1-X3)**2+(Y1-Y3)**2) 2100 D4=SQRT((X2-X4)**2+(Y2-Y4)**2) 2101 D5=SQRT((X1-X4)**2+(Y2-Y4)**2) 2102 D6=SQRT((X2-X3)**2+(Y2-Y4)**2) 2103 DUM=ABS(05+D6-D4-D3)/2. 2104 C=====+TAKE ADVANTAGE OF RECIPROCITY 2105 VIEW(IN*JN)=DUM/D1* 2106 C=====+TAKE ADVANTAGE OF RECIPROCITY 2107 IF(LDUM) YIEW(JN*IN)=DUM/D2* 2108 C======+FSYMETRY ARROUND AXIS GO RACK WITH CHANGED S	2080 2081	c	J1=IGREN(EN,JG) NUMJ=NUMB(J1)
2085 D2=BAREA(J1,J) 2086 N003=NBOUND(J1,J) 2087 N0D4=NROUND(J1,J) 2088 IF(XSYM(EN))SIGNY2=-1. 2090 50 CONTINUE 2091 LDUM=SIGNX2.EQ.1AND.SIGNY2.EQ.1. 2092 IF(N.OT.LDUM) GOTO 80 2093 IF(IN.GE.JN) GOTO 100 2094 80 CONTINUE 2095 X3=SIGNX2*X(NOD3) 2096 X4=SIGNX2*X(NOD4) 2097 Y3=SIGNY2*Y(NOD4) 2098 Y4=SIGYY2*Y(NOD4) 2099 D3=SQRT((X1-X4)**2+(Y1-Y3)**2) 2100 D4=SQRT((X2=X4)**2+(Y2=Y4)**2) 2101 D5=SQRT((X1=X4)**2+(Y1=Y3)**2) 2102 D6=SQRT((X2=X3)**2+(Y2=Y3)**2) 2103 DUM=ABS(D5+D6=D4=D3)/2. 2104 C=====HOTEL'S CROSED=STRING METHOD 2105 VIEW(IN,JN)=DUM/D1+VIEW(IN,JN) 2106 C=====TAKE ADVANTAGE OF RECIPROCITY 2107 IF(LDUM) VIEW(UN,IN)=DUM/D2 2108 C=====IF SYMETRY ARROUND AXIS GO RACK WITH CHANGED SIGNS OF COORDINATES 2109 LDUM=(SIGNX2+SIGNY2).LT.=1.99 2110 IF(LDUM) SIGNY2=1. <td>2083 2084</td> <td>·</td> <td>DO 100 J=2+NUMJ JN=JN+1</td>	2083 2084	·	DO 100 J=2+NUMJ JN=JN+1
2088 IF (XSYM(EN))SIGN[2=1]. 2089 IF (YSYM(EN))SIGN[2=1]. 2090 50 2091 LDUM=SIGN[2,E]]. 2092 IF (.NOT.LDUM) GOTO 80 2093 IF (IN.6E.JN) GOTO 100 2094 80 CONTINUE 2095 X3=SIGN[2*X](NOD3) 2096 X4=SIGN[2*X](NOD4) 2097 Y3=SIGN[2*Y](NOD4) 2098 Y4=SIGN[2*Y](NOD4) 2099 D3=SQRT((X1-X3)**2+(Y1-Y3)**2) 2100 D4=SQRT((X1-X4)**2+(Y1-Y4)**2) 2101 D5=SQRT((X1-X4)**2+(Y2-Y4)**2) 2102 D6=SQRT((X2-X3)**2+(Y2-Y3)**2) 2103 DUM=ABS(D5+D6-D4=D3)/2. 2104 C===+HOTEL'S CROSED=STRING METHOD 2105 VIEW(IN,JN)=DUM/D1+VIEW(IN,JN) 2106 C===+TAKE ADVANTAGE OF RECIPROCITY 2107 IF (LDUM) VIEW(JN, IN)=DUM/D2 2108 C====-IF SYMETRY ARROUND AXIS GO RACK WITH CHANGED SIGNS OF COORDINATES 2109 LDUM=(SIGN2+SIGN2).LT.=1.99 2110 IF (LDUM) SIGN2=1.	2085 2086 2087		D2=BAREA(J1,J) NOD3=NBOUND(J1,J-1) NOD4=NBOUND(J1,J)
2091 LDUM=SIGNX2.EQ.1AND.SIGNY2.EQ.1. 2092 IF(.NOT.LDUM) GOTO 80 2093 IF(IN.GE.JN) GOTO 100 2094 80 CONTINUE 2095 X3=SIGNX2*X(NOD3) 2096 X4=SIGNY2*Y(NOD3) 2097 Y3=SIGNY2*Y(NOD4) 2098 Y4=SIGNY2*Y(NOD4) 2099 D3=SQRT((X1-X3)**2+(Y1-Y3)**2) 2100 D4=SQRT((X1-X4)**2+(Y2-Y3)**2) 2101 D5=SQRT((X1-X4)**2+(Y2-Y3)**2) 2102 D6=SQRT((X2-X3)**2+(Y2-Y3)**2) 2103 DUM=ABS(D5+D6-D4=D3)/2. 2104 C===+HOTEL'S CROSED=STRING METHOD 2105 VIEW(IN,JN)=DUM/D1+VIEW(IN,JN) 2106 C====-TAKE ADVANTAGE OF RECIPROCITY 2107 IF(LDUM) VIEW(JN,IN)=DUM/D2 2108 C====-IF SYMETRY ARROUND AXIS GO RACK WITH CHANGED SIGNS OF COORDINATES 2109 LDUM=(SIGNX2+SIGNY2).LT.=1.99 2110 IF(LDUM) SIGNY2=1. 2110 IF(LDUM) SIGNY2=1.	2088 2089 2090	50	IF (XSYM(EN))SIGN(2=-1. IF (YSYM(EN))SIGN(2=-1. CONTINUE
2094 80 CONTINUE 2095 X3=SIGNX2*X(NOD3) 2096 X4=SIGNX2*X(NOD4) 2097 Y3=SIGNY2*Y(NOD3) 2098 Y4=SIGNY2*Y(NOD4) 2099 D3=SQRT((X1-X3)**2+(Y1-Y3)**2) 2100 D4=SQRT((X1-X4)**2+(Y2-Y4)**2) 2101 D5=SQRT((X1-X4)**2+(Y2-Y4)**2) 2102 D6=SQRT((X2-X3)**2+(Y2-Y3)**2) 2103 DUM=ABS(D5+D6=D4=D3)/2* 2104 C====+HOTTEL'S CROSED=STRING METHOD 2105 VIEW(IN,JN)=DUM/D1+VIEW(IN,JN) 2106 C====-TAKE ADVANTAGE OF RECIPROCITY 2107 IF(LDUM) VIEW(JN,IN)=DUM/D2 2108 C====-IF SYMMETRY ARROUND AXIS GO RACK WITH CHANGED SIGNS OF COORDINATES 2109 LDUM=(SIGNX2+SIGNY2).LT.=1.99 2110 IF(LDUM) SIGNY2=1.	2091 2092 2093		LDUM=SIGNX2.EQ.1AND.SIGNY2.EQ.1. IF(.NOT.LDUM) GOTO 80 IF(IN.GE.JN) GOTO 100
2097 Y3=SIGNY2*Y(NOD3) 2098 Y4=SIGNY2*Y(NOD4) 2099 D3=SQRT((X1-X3)**2+(Y1-Y3)**2) 2100 D4=SQRT((X2-X4)**2+(Y2-Y4)**2) 2101 D5=SQRT((X1-X4)**2+(Y2-Y4)**2) 2102 D6=SQRT((X2-X3)**2+(Y2-Y3)**2) 2103 DUM=ABS(D5+D6-D4-D3)/2. 2104 CHOTTEL'S CROSSED-STRING METHOD 2105 VIEW(IN,JN)=DUM/D1+VIEW(IN,JN) 2106 CTAKE ADVANTAGE OF RECIPROCITY 2107 IF(LDUM) VIEW(JN,IN)=DUM/D2 2108 CIF SYMMETRY ARROUND AXIS GO RACK WITH CHANGED SIGNS OF COORDINATES 2109 LDUM=(SIGNX2+SIGNY2).LT1.99 2110 IF(LDUM) SIGNY2=1. 2110 IF(LDUM) SIGNY2=1.	2094 2095 2096	80	CONTINUE X3=SIGNX2*X(NOD3) X4=SIGNX2*X(NOD4)
2007 D4=SQRT1((X2-X4)**2+(Y2-Y4)**2) 2101 D5=SQRT1((X2-X4)**2+(Y2-Y4)**2) 2102 D6=SQRT1((X2-X3)**2+(Y2-Y3)**2) 2103 DUM=ABS(05+D6-D4-D3)/2. 2104 CHOTTEL'S CROSSED-STRING METHOD 2105 VIEW(IN,JN)=DUM/D1+VIEW(IN,JN) 2106 CTAKE ADVANTAGE OF RECIPROCITY 2107 IF(LDUM) VIEW(JN,IN)=DUM/D2 2108 CIF SYMMETRY ARROUND AXIS GO RACK WITH CHANGED SIGNS OF COORDINATES 2109 LDUM=(SIGNX2+SIGNY2).LT1.99 2110 IF(LDUM) SIGNY2=1. 2110 IF(LDUM) SIGNY2=1.	2097 2098		Y3=SIGNY2*Y(NOD3) Y4=SIGNY2*Y(NOD4) D3=SOPY(V1=Y3)**2+(Y1=Y3)**2)
2102 DUB_SUR(((12-A)TAPE(12))(TAPE)) 2103 DUM_ABS(D5+D6-D4-D3)/2. 2104 CHOTTEL'S CROSSED-STRING METHOD 2105 VIEW(IN,JN)=DUM/D1+VIEW(IN,JN) 2106 CTAKE ADVANTAGE OF RECIPROCITY 2107 IF(LDUM) VIEW(JN,IN)=DUM/D2 2108 CIF SYMMETRY ARROUND AXIS GO RACK WITH CHANGED SIGNS OF COORDINATES 2109 LDUM=(SIGNX2+SIGNY2).LT1.99 2110 IF(LDUM) SIGNY2=1.	2100		D4=SQRT((X2-X4)*+2+(Y2-Y4)*+2) D5=SQRT((X1-X4)*+2+(Y1-Y4)*+2) C4=C4C7(X2-X4)*+2+(Y1-Y4)*+2)
2105 VIEW(IN,JN)=DUM/D1+VIEW(IN,JN) 2106 CTAKE ADVANTAGE OF RECIPROCITY 2107 IF(LDUM) VIEW(JN,IN)=DUM/D2 2108 CIF SYMMETRY ARROUND AXIS GO BACK WITH CHANGED SIGNS OF COORDINATES 2109 LDUM=(SIGNX2+SIGNY2).LT1.99 2110 IF(LDUM) SIGNY2=1. 2111 IF(LDUM) SIGNY2=1.	2102 2103 2104	c	DUMAEAS(105+D6-D4-D3)/2. HOTTEL'S CROSSED-STRING METHOD
2108 CIF SYMMETRY ARROUND AXIS GO BACK WITH CHANGED SIGNS OF COORDINATES 2109 LDUM=(SIGNX2+SIGNY2).LT1.99 2110 IF(LDUM) SIGNY2=1. 2110 IF(LDUM) SIGNY2=1.	2105 2106 2107	C	VIEW(IN-JN)=DUM/D1+VIEW(IN-JN) •TAKE ADVANTAGE OF RECIPROCITY IF(LDUM) VIEW(JN-IN)=DUM/D2
	2108 2109 2110	C	IF SYMMETRY ARROUND AXIS GO BACK WITH CHANGED SIGNS OF COORDINATES LDUM=(SIGNX2+SIGNY2).LT1.99 IF(LDUM) SIGNY2=1. IF(LDUM) SOUV2=1.

2112	c	-
2113	•	LDUM=DSYM.AND.SIGNX2.LT.0.
2114		IF(LDUM) SIGNX2=1.
2115		IF(LDUM) SIGNY2=-1.
2116		IF(LDUM) GOTO 50
2117	C	-
2118		LDUM=(SIGNX2*SIGNY2).LT.0.
2119		IF(LDUM) SIGNX2=1.
2120		IF(LDUM) SIGNY2=1.
2121		IF(LDUM) GOTO 50
2122	100	CONTINUE
2123		RETURN
2124		END
2125		
2126		SUBROUTINE XVERSY(X+Y+N+M+XS+YS)
2127	C	-FIND YS AS FUNCTION OF XS BY LINEAR INTERPLLATION
2128	C	-IN TABEL OF X- AND Y-VALUES
2129		DIMENSION $Y(N,1) \cdot X(N,1)$
2130		DO 10 I=2,N
2131		IF(XS.GE.X(I.M))GOTO 10
2132		YS=Y(I+1+M)+(XS+X(I=1+M))*(Y(I+M)-Y(J+1+M))/(X(I+M)-X(I=1+M))
2133		GOTO 11
2134	10	CONTINUE
2135		PRINT 1
2136		PRINT 2+X5+(X(I+M)+Y(I+M)+I=1+5)
2137		PRINT 3+M
2138		STOP
2139	1	FORMAT(///1X, INPUT VALUE TO XVERSY OUT OF RANGE //)
2140	2	FORMAL(1X, INPUT X', EIU.4/1X'X-Y VALUE PAIRS'//1X,5(268.3,5X))
2141	\$	FORMATCZIX/ICURVE NUMBERI/14)
2142	11	KETUKN
2140		LND

.

APPENDIX C - Example input

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

Example I

1 2 3 4 5 6 7 8 9 10 11 12	SQUARE PLATE F.1.1125125 O UNIT MATERIAL DATA F.2.2 .1.10000.1. .00001.10000. .00001.10000. 1 1 1.171.1. 9 18 27 36 45 54 63 72 81 80 79 78 77 76 75 74 73 1
13	F•1
14	0 •
15	NOVOID
16	20,1,1,1,,,,
17	.05 .10 .15 .20 .25 .30 .35 .40 .45 .50 .55 .60 .65 .70 .75 .80 .85 .90 .95 1.
18	DUMMY TEMPERATURE
19	2
20	**1000**
21	

Example II

1 I BEAM EMBEDDED IN CONCRETE	
2 F1.141.11.051.03161414	
3 F++++05++01	
4 F,,01,,003,09	
9 .025.055.05.08	
11 33 34+++++	
12 25 26,,,,,	
13 17 18,,,,,	
14 1 2 9 10,,,,,	
19 7 15 8 16,	
20 23 24	
21 31 32/////	
22 39 40	
23 BETONG	
25 24.511.78111511.281243111714011117145773210251.85716101604	000.,
28 F + 3 + 7 + +	
29 ,60,800,27,2000,27	
30 ,,200,,217000,400,466000,600,758000,700,927000,800,119200,1200,1	766000
31 STAL	
32 F,3,7,,	
33 +60+800+27+2000+27 34	766000
36 F137777	
37 ,60,800,27,2000,27	
38 ,,200,,217000,400,466000,600,758000,700,927000,800,1192000,1200,1	766000
39 25.0001/25	
40 4	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	
42 1/9/1//2010/01/1/33	
45 3 6 6 2 2 1 2 5	
46 8 16 24 32 40 39	
47 4 4 .8 2 .2 .1 . 25	
48 39 47 55 63	
52 F+3	
53 F+4	
54 0	
55 NOVOID	
56 15+1.5+++++	
57 .11.21.31.44.51.61.71.81.911.011.11.211.311.411.51	
58 HE10087F1RE	
60 ,25+.05+525+.1+620+.3+725+.6+940+1+980++1+925+500+1+2+475+1+5001+	360,+++

Example III

1	BOX GIRDER EMBEDDED IN CONCRETE
2	Fy,14y,15y1,y1,y5y4y6
3	Frrr•0875++004
4	F,.0345,.004,.0375,.112
5	F,,,112,,0375,,12
6	T,,,004,.0345,.112
7	024.054.074.11
ģ	015.029.044.056.055.13
ě	
	1.9
10	1 277777777
11	12 13:::::::
12	23 24 34 35,,,,,,,
13	45 46,,,,,,,,,
14	56 57+++++++
15	68 67
16	25 36++++++
17	26 37+++++++
18	27 38
10	
20	
20	27 407/177777
<1 2 1	
22	19 20111111
23	8 9 • • • • • • • •
24	BETONG
25	7,7,7,,
26	24,5,1,78,115,1,28,243,1,17,401,1,17,643,92,895,85,1500,85,
27	,,100,55600,115,91000,200,129400,600,397200,1000,696700,1500,1000000.r
28	STAL
29	F-3-7
30	•60•800•27•2000•27
31	200217000.400.466000.600.758000.700.927000.800.1192000.1200.1766000
32	
33	
30	60, 900, 97, 9000, 97
34	100000121720001000 UK6000.600.758000.700.927000.800.1192000.1200.1766000
35	1200-121/0001400148800018001/280001/001-2/00048001142008110011/2000
36	
37	
38	,60,800,27,2000,27
39	**500**511000*460*466600*200*228000*100*351000*800*1135600*15500*118800
40	25,25,0001,,
4 <u>1</u>	4
42	1,7,.6,.99,1.33
43	1,12,23,34,45,56,67,
44	2,3,8,8,99,1,33
45	67,78,89
46	3.9.8.2.20.1.25
47	11,22,33,44,55,66,77,88,99,
49	1 + 1 + -6 + 133
40	0.13.00.15.06.07.08.09.30.19.8
47	271372472372072772072770
50	
21	
52	
ວວ =	
54	UI PIBNDA
55	VOID
56	1
57	Fo1,4,,,,,
58	15+1.5+++++++
59	.1,.2,.3,.4,.5,.6,.7,.8,.9,1.,1,1,1,2,1.3,1,.4,1,5,
60	HSQ-FIRE
61	12
62.	,25,.05,525,.1,625,.2,765,.3,900,.5,925,.7,980,1,1015,1.05,675,1.1,600,1.2,500.
63	1,501,350.,

· · · · · ·

.

.

PERSTORPS TRYCKERI

.