UARI Research Report No. 126

NTIS HC $6.00

THERMAL ELASTOPLASTIC STRUCTURAL ANALYSIS OF NON-METALLIC THERMAL PROTECTION SYSTEMS

by

T. J. Chung and G. Yagawa

D RA

Final Technical Report

(NASA-CR-124005) THERMAL ELASTOPLASTIC STRUCTURAL ANALYSIS OF NON-METALLIC THERMAL PROTECTION SYSTEMS Final Technical Report (Alabama Univ., Huntsville.) 78 p HC $6.00 CSCL 20M G3/33

N73-15948

Unclas 16354

This research work was supported by
the National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Contract NAS8-27792

The University of Alabama in Huntsville
School of Graduate Studies and Research
Huntsville, Alabama 35807

August 1972

78

UARI Research Report No. 126

THERMAL ELASTOPLASTIC STRUCTURAL ANALYSIS OF NON-METALLIC THERMAL PROTECTION SYSTEMS

by

T. J. Chung and G. Yagawa

Final Technical Report

This research work was supported by
the National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Contract NAS8-27792

The University of Alabama in Huntsville
School of Graduate Studies and Research
Huntsville, Alabama 35807

August 1972

TABLE OF CONTENTS

PREFACE

This report presents the results of studies conducted during the period September 1, 1971 - August 31, 1972, under NASA Research Contract NAS8-27792, "Thermal Elastoplastic Structural Analysis of Non-metallic Thermal Protection Systems." This study was monitored by Mr. C. R. Zimmerman, Analytical Mechanics Division, Asttronautics Laboratory of NASA's Marshall Space Flight Center.

LIST OF FIGURES

NOMENCLATURE

B^{ij}	= Tensor of thermoelastic constants
$\overset{*}{B}{}^{ij}$	= Tensor of thermoplastic constants
c	= Specific heat
$d\lambda$	= Positive constant
D	= Dissipation
$E^{ijk\ell}$	= Tensor of elastic moduli
$\overset{*}{E}{}^{ijk\ell}$	= Tensor of plastic moduli
$E_{(p)}$	= Plastic modulus
F^{j}	= Components of mechanical force
h	= heat supply
q	= heat flux
T_o, T	= Reference temperature and temperature change
u^j, $\dot{u}^j$, $\ddot{u}^j$	= Components of displacements, velocity, and acceleration
$\bar{\gamma}$	= Equivalent yield strain
γ_{ij}	= Strain tensor
Δt	= Time interval
ϵ	= Strain energy density
η	= Entropy
θ	= Absolute temperature
ρ_o, ρ	= Density at initial and deformed configuration
$\bar{\sigma}$	= Equivalent yield stress
σ^{ij}	= Stress tensor

ϖ = Free energy

ψ_{1N} = Normalized displacement interpolation function

Ω_R = Normalized temperature interpolation function

THERMAL ELASTOPLASTIC ANALYSIS OF NON-METALLIC THERMAL PROTECTION SYSTEMS

T. J. Chung and C. Yagawa

ABSTRACT

This report presents an incremental theory and numerical procedure to analyze a three-dimensional thermoelastoplastic structure subjected to high temperature, surface heat flux, and volume heat supply as well as mechanical loadings. Heat conduction equations and equilibrium equations are derived by assuming a specific form of incremental free energy, entropy, stresses and heat flux together with the first and second laws of thermodynamics, von Mises yield criteria and Prandtl-Reuss flow rule. The finite element discretization using the linear isotropic three-dimensional element for the space domain and a difference operator corresponding to a linear variation of temperature within a small time increment for the time domain lead to systematic solutions of temperature distribution and displacement and stress fields. Various boundary conditions such as insulated surfaces and convection through uninsulated surface can be easily treated. To demonstrate effectiveness of the present formulation a number of example problems are presented.

1. INTRODUCTION

The mechanics of thermoelastoplastic solids has attracted the attention of many investigators in recent years. The development of the theories of thermoelastoplasticity began with an attempt to consider the plastic deformation as a thermodynamic state variable and with the controversial treatment of finite deformations. Crucial difficulties lie in the proper choice of free energy functional and methods of numerical computation.

Biot [1] and Coleman and Gurtin [2] used the concept of state variable in dealing with thermodynamics of viscoelastic materials. Perzyna and Wojno [3], Kratochvil and Dillon [4], subsequently, employed the similar concept by decomposing the total strains into elastic and plastic strain components. The recent work by Valanis [5] uses also the concept of hidden variables, but introducing no yield surfaces. Finite elastoplastic deformations with thermodynamic considerations were also discussed by Green and Naghdi [7]. Schapery [8] studied a thermodynamic constitutive theory with history effects represented by single integrals. Recently, Oden and Bhandari [9] proposed a general functional theory of thermodynamics of viscoelastoplastic solids and showed that various theories of viscoelasticity, rate independent viscoplasticity and plasticity can be obtained as special cases by imposing suitable constraints on the material parameters.

All of these theories mentioned above, however, lead to considerable mathematical and computational difficulties in dealing with problems of geometrical complexity. The present paper is an attempt to establish an alternate approach. We derive incremental governing equations for three-dimensional thermoelastoplastic solids and provide convenient solution techniques by finite elements in space domain and the difference operator in time domain. Although an extension to materials with memory presents no special difficulty, the present study is not intended for viscous behavior. In this study, the Lagrangian coordinates are used disregarding the finite strains, but large deformations are included in the formulation.

Finally, example problems are presented to demonstrate the merits of the present formulation. Effects of thermomechanical coupling, thermoplastic deformations and stress, and developments of plastic regions as a function of time are shown and evaluated. Various features not included in this study such as temperature dependent material properties or dynamically coupled inertia effects may be treated with minor modifications.

2. THERMOMECHANICAL PRELIMINARIES

It is assumed that the behavior of the body under thermomechanical environments obeys the laws of conservation of mass, balance of linear and angular momentum, conservation of energy, and the Clausius-Duhem inequality,

$$\int_{v_0} \rho_0 \, dv_0 = \int_v \rho \, dv \tag{1a}$$

$$\sigma^{ij}\big|_i + \rho F^j - \rho \ddot{u}^j = 0, \quad \sigma^{ij} = \sigma^{ji} \tag{1b}$$

$$\rho \dot{\varepsilon} = \sigma^{ij}\dot{\gamma}_{ij} + q^j\big|_j + \rho h \tag{1c}$$

$$D + \frac{1}{\theta} q^i \theta_{,i} \geq 0 \tag{1d}$$

$$D = \rho\theta\dot{\eta} - q^{j}\big|_{j} - \rho h \tag{1e}$$

Here ρ is the mass density with the subscript o indicating undeformed configuration. σ^{ij} is the second Piola-Kirchhoff stress tensor; superposed dots represent time rates; strokes and commas are covariant and ordinary differentiations; F^j and u^j are the body forces and displacements; γ_{ij} is the strain tensor; ϵ, h and η are the internal energy, heat supply and entropy per unit mass; q^j is the heat flux per unit area; θ is the absolute temperature; D is the internal dissipation. It is understood that for small strains $\rho_o = \rho$ and for rectangular cartesian coordinates covariant and ordinary differentiations are the same.

The free energy φ is defined as

$$\varphi = \epsilon - \theta\eta \tag{2}$$

which leads to

$$\rho\dot{\varphi} = \sigma^{ij}\dot{\gamma}_{ij} - D - \rho\eta\dot{\theta} \tag{3}$$

3. THERMOELASTOPLASTIC BEHAVIOR

As discussed in Section 1, our approach is to avoid the functional theory or the state variable concept. Instead of expressing the free energy as functionals of all the histories of strains and temperature and as a function of temperature gradient, we postulate an existence of φ in incremental quantity so that for a small time interval Δt

$$\varphi(\Delta t) = \hat{\Phi}\{\gamma_{ij}^{(e)}(\Delta t),\ \gamma_{ij}^{(p)}(\Delta t),\ \theta(\Delta t)\} \tag{4a}$$

Similarly,

$$\sigma^{ij}(\Delta t) = \hat{\Sigma}\{\gamma_{ij}^{(e)}(\Delta t),\ \gamma_{ij}^{(p)}(\Delta t),\ \theta(\Delta t)\} \tag{4b}$$

$$q^i(\Delta t) = \hat{Q}\{\gamma_{ij}^{(e)}(\Delta t),\ \gamma_{ij}^{(p)}(\Delta t),\ \theta(\Delta t)\} \tag{4c}$$

$$\eta(\Delta t) = \hat{H}\{\gamma_{ij}^{(e)}(\Delta t),\ \gamma_{ij}^{(p)}(\Delta t),\ \theta(\Delta t)\} \tag{4d}$$

Here it is assumed that the total strain is the sum of the elastic and plastic components with (e) and (p) representing, respectively, "elastic" and "plastic".

The implication of (4a) through (4d) is that the free energy, stresses, heat flux, and entropy are functions of elastic strain, plastic strain and temperature only within the small time interval. It is then a simple matter to derive the governing heat conduction equations and equations of equilibrium in "incremental form". All histories may be carried over from one increment to another through the step-by-step numerical time integration.

Although the free energy can be shown in various forms depending on the material properties or the purpose of analysis, the present study is limited to a particular case including only quadratic terms as follows:

$$\begin{aligned}\rho\varphi(\Delta t) = {} & \tfrac{1}{2}E^{ijk\ell}\gamma_{ij}^{(e)}(\Delta t)\gamma_{k\ell}^{(e)}(\Delta t) + \tfrac{1}{2}\overset{*}{E}{}^{ijk\ell}\gamma_{ij}^{(p)}(\Delta t)\gamma_{k\ell}^{(p)}(\Delta t)\\ & -B^{ij}T(\Delta t)\gamma_{ij}^{(e)}(\Delta t) - \overset{*}{B}{}^{ij}T(\Delta t)\gamma_{ij}^{(p)}(\Delta t)\\ & - \frac{c\ T^2(\Delta t)}{2T_o}\end{aligned} \tag{5}$$

in which $E^{ijk\ell}$ and $\overset{*}{E}{}^{ijk\ell}$ are tensors of elastic and plastic moduli; B^{ij} and $\overset{*}{B}{}^{ij}$ are tensors of thermoelastic and thermoplastic moduli; c is the specific heat; T_o and T are the reference temperature and temperature change related by $\theta = T_o + T$. Here $\overset{*}{E}{}^{ijk\ell}$ and $\overset{*}{B}{}^{ij}$ are the functions of current state of stress in the inelastic range. It should be noted that such inelastic material properties cannot be admitted had the form of free energy been expressed in entire history domain rather than in a small time interval. These arrays of plastic moduli remain constant only during the small time interval and must be updated as the state of stress changes.

Likewise, the expression (3) in a small time increment is given by

$$\rho\dot{\omega}(\Delta t) = \sigma^{ij}(\Delta t)\dot{\gamma}_{ij}(\Delta t) - D(\Delta t) - \rho\eta(\Delta t)\dot{\theta}(\Delta t)$$

$$= \sigma^{ij}(\Delta t)(\dot{\gamma}_{ij}^{(e)}(\Delta t) + \dot{\gamma}_{ij}^{(p)}(\Delta t)) - D(\Delta t) - \rho\eta(\Delta t)\dot{T}(\Delta t) \qquad (6)$$

and the time rate of (5) becomes

$$\rho\dot{\phi}(\Delta t) = E^{ijk\ell}\gamma_{k\ell}^{(e)}(\Delta t)\dot{\gamma}_{k\ell}^{(e)}(\Delta t) + \overset{*}{E}{}^{ijk\ell}\gamma_{k\ell}^{(p)}(\Delta t)\dot{\gamma}_{k\ell}^{(p)}(\Delta t)$$

$$- B^{ij}\dot{T}(\Delta t)\gamma_{ij}^{(e)}(\Delta t) - B^{ij}T(\Delta t)\dot{\gamma}_{ij}^{(e)}(\Delta t)$$

$$- \overset{*}{B}{}^{ij}\dot{T}(\Delta t)\gamma_{ij}^{(p)}(\Delta t) - \overset{*}{B}{}^{ij}T(\Delta t)\dot{\gamma}_{ij}^{(p)} - \frac{c\ T(\Delta t)\dot{T}(\Delta t)}{T_o} \qquad (7)$$

In view of (6) and (7) and dropping (Δt) for simpler notation, we obtain

$$(E^{ijk\ell}\gamma_{k\ell}^{(e)} - B^{ij}T - \sigma^{ij})\dot{\gamma}_{ij}^{(e)} + (-B^{ij}\gamma_{ij}^{(e)} - \overset{*}{B}{}^{ij}\gamma_{ij}^{(p)}$$

$$- \frac{c\ T}{T_o} + \rho\eta)\ \dot{T} + D + \overset{*}{E}{}^{ijk\ell}\gamma_{ij}^{(p)}\dot{\gamma}_{k\ell}^{(p)} - \overset{*}{B}{}^{ij}T\dot{\gamma}_{ij}^{(p)} - \sigma^{ij}\dot{\gamma}_{ij}^{(p)} = 0 \qquad (8)$$

For all arbitrary values of $\dot{\gamma}_{ij}^{(e)}$ and $\dot{T}$ the following relationships must be true from (8):

$$\sigma^{ij} = E^{ijk\ell}\gamma_{k\ell}^{(e)} - B^{ij}T \qquad (9)$$

$$\rho\eta = B^{ij}\gamma_{ij}^{(e)} + \overset{*}{B}{}^{ij}\gamma_{ij}^{(p)} + \frac{c\ T}{T_o} \qquad (10)$$

$$D = -\overset{*}{E}{}^{ijk\ell}\gamma_{ij}^{(p)}\dot{\gamma}_{k\ell}^{(p)} + \overset{*}{B}{}^{ij}\dot{T}\gamma_{ij}^{(p)} + \sigma^{ij}\dot{\gamma}_{ij}^{(p)} \qquad (11)$$

It is interesting to note that the internal dissipation consists of terms associated with thermoelastoplastic strains or energy dissipated by plastic deformations.

To evaluate $\overset{*}{E}{}^{ijk\ell}$ and $\overset{*}{B}{}^{ij}$ the von Mises yield criteria and associated flow rule may be used as discussed by earlier investigators [10, 11, 12]. Writing (9) in differential form,

$$d\sigma^{ij} = E^{ijk\ell}(d\gamma_{k\ell} - d\gamma_{k\ell}^{(p)}) - B^{ij}dT \tag{12}$$

where

$$d\gamma_{k\ell}^{(p)} = \frac{\partial F}{\partial \sigma^{k\ell}} d\lambda \tag{13}$$

The plastic potential F is related by the equivalant yield stress $\bar{\sigma}$ and the second deviatoric stress invariant J,

$$F = \bar{\sigma}^2 = 3J \tag{14}$$

from which we derive the incremental yield stress in the form

$$d\bar{\sigma} = \frac{3}{2\bar{\sigma}} \frac{\partial J}{\partial \sigma^{ij}} d\sigma^{ij} = Z_{ij} d\sigma^{ij} \tag{15}$$

and $d\lambda$ in (13), a positive constant, can be shown to be

$$d\lambda = d\bar{\gamma}^{(p)}/2\bar{\sigma} \tag{16}$$

with the incremental equivalent yield strain $d\bar{\gamma}^{(p)}$ related by the bilinear plastic modulus $E_{(p)}$ as

$$d\bar{\gamma}^{(p)} = d\bar{\sigma}/E_{(p)} \tag{17}$$

Using (13) through (17) in (12) we obtain

$$d\sigma^{ij} = E^{ijk\ell}(d\gamma_{k\ell} - Z_{k\ell} d\bar{\gamma}^{(p)}) - B^{ij}dT \tag{18}$$

In view of (15), (17), and (18) $d\bar{\gamma}^{(p)}$ assumes the form

$$d\bar{\gamma}^{(p)} = H^{-1}(Z_{ij}E^{ijk\ell}d\gamma_{k\ell} - Z_{ij}B^{ij}dT) \tag{19}$$

where

$$H^{-1} = E_{(p)} + Z_{rs}Z_{tu}E^{rstu} \tag{20}$$

Substituting (20) in (18) yields

$$d\sigma^{ij} = E^{ijk\ell}d\gamma_{k\ell} + \overset{*}{E}{}^{ijk\ell}d\gamma_{k\ell} - B^{ij}dT - \overset{*}{B}{}^{ij}dT \tag{21}$$

in which

$$\overset{*}{E}{}^{ijk\ell} = -H^{-1}E^{ijmn}Z_{pq}Z_{mn}E^{k\ell pq} \tag{22}$$

$$\overset{*}{B}{}^{ij} = -H^{-1}B^{mn}Z_{mn}Z_{k\ell}E^{ijk\ell} \tag{23}$$

We now return to (1e), a representation of irreversible thermodynamic processes and express it for a small time increment in the form,

$$\rho\theta(\Delta t)\dot{\eta}(\Delta t) - q(\Delta t)^{j}\Big|_{j} - \rho h(\Delta t) - D(\Delta t) = 0 \tag{24}$$

Once again dropping (Δt) and substituting from (10) and (11), it is possible to write (24) in the form

$$(T_0 + T)(B^{ij}\dot{\gamma}^{(e)}_{ij} + \overset{*}{B}{}^{ij}\dot{\gamma}^{(p)}_{ij} + \frac{c\dot{T}}{T_0})$$
$$- q^{j}\Big|_{j} - \rho h + \overset{*}{E}{}^{ijk\ell}\gamma^{(p)}_{k\ell}\dot{\gamma}^{(p)}_{ij} - \overset{*}{B}{}^{ij}T\dot{\gamma}^{(p)}_{ij} - \sigma^{ij}\dot{\gamma}^{(p)}_{ij} = 0 \tag{25}$$

where

$$\gamma^{(p)}_{ij} = \gamma_{ij} - \gamma^{(e)}_{ij} \tag{26}$$

and

$$\dot{\gamma}^{(p)}_{ij} = Z_{ij}\dot{\bar{\gamma}}^{(p)} = Z_{ij}H^{-1}(Z_{mn}E^{mnk\ell}\dot{\gamma}_{k\ell} - Z_{mn}B^{mn}\dot{T}) \tag{27}$$

Substituting (26) and (27) into (25) yields

$$(T_0 + T)\{(B^{ij} + \overset{*}{B}{}^{ij} + \overset{*}{\beta}{}^{ij})\dot{\gamma}_{ij} + (\tilde{B} + \tilde{\beta})\,\dot{T}$$
$$+ \frac{c\dot{T}}{T_0}\} - q^{j}\Big|_{j} - \rho h + \overset{*}{E}{}^{ijmn}\,\overset{*}{G}\,\gamma_{ij}\,\dot{\gamma}_{mn} - \overset{*}{E}{}^{ijmn}\,\overset{*}{G}\,\dot{\gamma}_{ij}\,\gamma^{(e)}_{mn}$$
$$-\overset{*}{E}{}^{ijmn}W_{mn}\,\gamma_{ij}\,\dot{T} + \overset{*}{E}{}^{ijmn}W_{mn}\,\dot{T}\,\gamma^{(e)}_{ij} - \overset{*}{\beta}{}^{ij}\dot{\gamma}_{ij} - \tilde{\beta}\,T$$
$$- \sigma^{ij}(\overset{*}{G}\,\dot{\gamma}_{ij} + W_{ij}\,\dot{T}) = 0 \tag{28}$$

where

$$\overset{*}{\beta}{}^{ij} = H^{-1}\overset{*}{B}{}^{ij}Z_{pq}Z_{k\ell}E^{pqk\ell} \tag{29a}$$
$$\tilde{B} = H^{-1}B^{ij}Z_{ij}Z_{pq}B^{pq} \tag{29b}$$
$$\tilde{\beta} = -H^{-1}\overset{*}{B}{}^{ij}Z_{ij}Z_{pq}B^{pq} \tag{29c}$$
$$\overset{*}{G} = H^{-1}Z_{rs}Z_{pq}E^{rspq} \tag{29d}$$
$$W_{mn} = H^{-1}Z_{mn}Z_{pq}B^{pq} \tag{29e}$$

B

Note that all quantities in (28) are now expressed in terms of the total strain and elastic strain but not plastic strain. The plastic behavior is exhibited by $\overset{*}{E}^{ijk\ell}$, $\overset{*}{B}^{ij}$, $\overset{*}{\beta}^{ij}$, $\tilde{B}$, $\tilde{\beta}$, $\overset{*}{G}$, and W_{mn}. It can easily be shown that for isotropic solids $\overset{*}{B}$, $\overset{*}{\beta}$, $\tilde{B}$, $\tilde{\beta}$, and W_{mn} are always zero. They need be considered only for anisotropic solids. Here elastic strains follow simply from the elastic constitutive law.

The expression in (28) is the transient heat conduction equation with a complete elastoplastic coupling and internal dissipation. The formulation of incremental finite element equations from (28) will be shown in the following section.

4. HEAT CONDUCTION EQUATIONS-FINITE ELEMENT FORMULATION

To introduce the finite element application to (28) the element temperature T and element displacements u_i $(i = 1, 2, 3)$ are replaced by a linear combination of all nodal temperatures T^R and all nodal displacements u^N with suitable interpolation functions [9, 16], in the form

$$T = \Omega_R u^R \tag{30}$$

$$u_i = \psi_{iN} u^N \tag{31}$$

For the 8 node isoparametric element, we have $R = 8$ and $N = 24$. Ω_R and ψ_{iN} are the normalized interpolation functions for temperature and displacements, respectively.

The strain-displacement relationship is given by

$$\gamma_{ij} = \tfrac{1}{2}(u_{i,j} + u_{j,i} + u_{m,i}u_{m,j}) \tag{32}$$

On substituting (31) into (32), we have

$$\gamma_{ij} = A_{Nij}u^N + C_{NMij}u^N u^M \tag{33}$$

where A_{Nij} and C_{NMij} are the strain transformation operators,

$$A_{Nij} = \tfrac{1}{2}(\psi_{iN},_j + \psi_{jN},_i)$$

$$C_{NMij} = \tfrac{1}{2}\,\psi_{KN},_i\psi_{KM},_j$$

Now let it be required to solve the differential equation (28) rewritten in the form

$$L(\gamma_{ij},\ \dot{\gamma}_{ij},\ T,\ \dot{T}) = 0$$

or upon substitution of (30) into (33),

$$L(u^N,\ \dot{u}^N,\ T^R,\ \dot{T}^R) = 0 \qquad (34)$$

where L is the differential operator and $L(u^N, \dot{u}^N, T^R, \dot{T}^R)$ is considered as the local residual $L(u^N, \dot{u}^N, T^R, \dot{T}^R)$. Requiring this local residual to be orthogonal to the subspace spanned by the functions Ω_R for each finite element; i. e.,

$$\int_V L(u^N,\ \dot{u}^N,\ T^R,\ \dot{T}^R)\Omega_R dv = 0 \qquad (35)$$

which is essentially the Galerkin's method, we obtain the finite element model of (28),

$$\begin{aligned}\int_V &\{(T_O + \Omega_U T^U)\Omega_R[(B^{ij} + \overset{*}{B}{}^{ij} + \overset{*}{\beta}{}^{ij})(A_{Mij} + 2C_{MPij}u^P)\dot{u}^M \\ &+ (\tilde{B} + \tilde{B})\,\Omega_s\dot{T}^s + \frac{C}{T_O}\,\Omega_s\dot{T}^s] - q^j{}_{|j}\,\Omega_R - \rho h\Omega_R + \overset{*}{E}{}^{ijmn}\overset{*}{G}\Omega_R(A_{Nij}u^N \\ &+ C_{NMij}u^N u^M)(A_{Pmn} + 2C_{Pqmn}u^q)\dot{u}^P - \overset{*}{E}{}^{ijmn}\overset{*}{G}\Omega_R(A_{Pij} + 2C_{Pqij}u^q)\dot{u}^P\gamma^{(e)}_{mn} \\ &- \overset{*}{E}{}^{ijmn}W_{mn}\Omega_R(A_{Nij}u^N + C_{NMij}u^M u^N)\,\dot{T} + \overset{*}{E}{}^{ijmn}W_{mn}\Omega_R\dot{T}\gamma^{(e)}_{ij} \\ &- \overset{*}{\beta}{}^{ij}\Omega_R(A_{Nij}\dot{u}^N + 2C_{NMij}\dot{u}^M u^N)T + \tilde{\beta}\,T\Omega_R \\ &- \sigma^{ij}\Omega_R(A_{Nij}\dot{u}^N + 2C_{NMij}\dot{u}^N u^M)\overset{*}{G} - \sigma^{ij}\Omega_R W_{ij}\dot{T} = 0\end{aligned} \qquad (36)$$

Introducing the linear Fourier law,

$$q^i = \kappa^{ij}T,_j$$

where $\varkappa^{ij}$ is the thermal conductivity, using the Green-Gauss theorem, and after some algebra in (36), we finally arrive at the finite element heat conduction equations for a kth time increment,

$$N_{RS}\dot{T}^{S}_{(k)} + R_{RS}T^{S}_{(k)} = P^{(Q)}_{R(k)} + P^{(q)}_{R(k)} + P^{(c)}_{R(k)} + P^{(EP)}_{R(k)} + P^{(TP)}_{R(k)} \tag{37}$$

in which

heat capacity matrix

$$N_{RS} = \int_V c\Omega_R\Omega_S dv \tag{38a}$$

conductivity matrix

$$R_{RS} = \int_V \varkappa^{ij}\Omega_{R},_{i}\Omega_{S},_{j}dv \tag{38b}$$

volume heat supply vector

$$P^{(Q)}_{R(k)} = \int_V \rho h\Omega_R dv \tag{38c}$$

surface heat flux vector

$$P^{(q)}_{R(k)} = \int_A q^i n_i \Omega_R dA \tag{38d}$$

Here n_i is the unit normal to surface. If the surface is uninsulated and convection loss is to take place due to ambient temperature T' then the following boundary condition should be met:

$$q^i n_i + q + \bar{\alpha}(T - T') = 0$$

$$\text{or} \qquad q^i n_i = -q - \bar{\alpha}(T - T')$$

where $\bar{\alpha}$ is the film coefficient. This requires the surface heat flux vector (38d) to be replaced by

$$P^{(q)}_{R(k)} = -\int_A \Omega_R(q - \bar{\alpha}T')dA - \int_A \Omega_R\bar{\alpha}\Omega_S dAT^{S}_{(k)} \tag{38d-1}$$

The second term of the left-hand side of (38d-1) may be added to the conductivity matrix (38b) so that

$$R_{RS} = \int_V \kappa^{ij}\Omega_{R,i}\Omega_{S,j}dv + \int_A \bar{\alpha}\ \Omega_R\Omega_S dA \tag{38b-1}$$

and

$$P^{(q)}_{R(k)} = -\int_A \Omega_R(q - \bar{\alpha}T')dA \tag{38d-2}$$

Pseudo heat capacity vector

$$P^{(c)}_{R(k)} = \{\int_V \frac{C}{T_O}\Omega_R\Omega_S T_{(k-1)}dv\}\dot{T}_{(k-1)} \tag{38e}$$

Pseudo elastoplastic coupling vector

$$P^{(EP)}_{R(k)} = \{\int_V (T_O + T_{(k-1)})\Omega_R(B^{ij} + \overset{*}{B}{}^{ij} + \overset{*}{\beta}{}^{ij})(A_{Mij}$$

$$+2C_{MPi}\,u^P_{(k-1)})dv\}\dot{u}^M_{(k-1)} + \{\int_V (T_O + T_{(k-1)})\Omega_R$$

$$(\tilde{B} + \tilde{\beta})\Omega_S dv\}\dot{T}^S_{(k-1)} \tag{38f}$$

Pseudo thermoplastic dissipation vector,

$$P^{(TP)}_{R(k)} = -\{\int_V \overset{*}{E}{}^{ijmn}\overset{*}{G}\Omega_R(A_{Nij}u^N_{(k-1)} + C_{NMij}u^N_{(k-1)}u^M_{(k-1)})$$

$$(A_{pmn} + 2C_{pqmn}u^q_{(k-1)})dv\}\dot{u}^P_{(k-1)}$$

$$+ \{\int_V \overset{*}{E}{}^{ijmn}\overset{*}{G}\Omega_R(A_{pij} + 2C_{pqij}u^q_{(k-1)})dv\}\dot{u}^P_{(k-1)}\gamma^{(e)}_{mn(k-1)}$$

$$+ \{\int_V \overset{*}{E}^{ijmn} W_{mn} \Omega_R (A_{Nij} u^N_{(k-1)} + C_{NMij} u^M_{(k-1)} u^N_{(k-1)}) dv\} \dot{T}_{(k-1)}$$

$$- \{\int_V \overset{*}{E}^{ijmn} W_{mn} \Omega_R dv\} \dot{T}_{(k-1)} \gamma^{(e)}_{ij(k-1)}$$

$$+ \{\int_V \overset{*}{\beta}^{ij} \Omega_R (A_{Nij} \dot{u}^N_{(k-1)} + 2C_{NMij} u^M_{(k-1)} \dot{u}^N_{(k-1)}) dv\} T_{(k-1)}$$

$$- \{\int_V \tilde{\beta}\, \Omega_R dv\} T_{(k-1)} \qquad + \qquad \int_V \sigma^{ij} \Omega_R (A_{Nij} \dot{u}^N_{(k-1)}$$

$$+ 2C_{NMij} \dot{u}^N_{(k-1)} u^M_{(k-1)}) dv + \{\int_V W_{ij} \sigma^{ij} \Omega_R dv\} \dot{T}_{(k-1)} \tag{38g}$$

Here $(k - 1)$ refers to the previous time step; and if the pseudo elasto-plastic coupling vector and pseudo thermoplastic vector are removed the expression (37) is identical to the uncoupled transient heat conduction equations.

5. INCREMENTAL FINITE ELEMENT EQUATIONS OF EQUILIBRIUM

The standard finite element equations of equilibrium is given by

$$\int_V \sigma^{ij} \frac{\partial \gamma_{ij}}{\partial u^N} dv = F^{(a)}_N \tag{39}$$

in which $F^{(a)}_N$ is the nodal applied load vector. The incremental form of (39) is obtained by taking a variation of (39) in the form

$$\int_V d\sigma^{ij} \frac{\partial \gamma_{ij}}{\partial u^N} dv + \int_V \sigma^{ij} d \{\frac{\partial \gamma_{ij}}{\partial u^N}\} dv = dF^{(a)}_N \tag{40}$$

The first and second terms in the left-hand side of (40) correspond, respectively, to incremental changes of stresses and geometries. In view of (20), (21), (22), and (26) and performing appropriate differentiations, it is now possible to write (40) in incremental quantities,

$$\int_V [(E^{ijk\ell} + \overset{*}{E}{}^{ijk\ell})(A_{Nk\ell}du^N + 2C_{NMk\ell}u^M du^M)$$

$$- (B^{ij} + \overset{*}{B}{}^{ij})\Omega_R dT^R](A_{Nij} + 2C_{NMij}u^M)dv$$

$$+ \int_V 2\sigma^{ij}C_{NMij}du^M dv = dF_N^{(a)} \tag{41}$$

It should be noted that σ^{ij} in the last term of (41) implies the initial stresses or the residual stresses in the structure just prior to a new change in geometry.

After some algebra the final form of incremental thermoelastoplastic equation of equilibrium for the jth incremental step

$$(K_{NM}^{(e)} + K_{NM}^{(G)} + K_{NM}^{(P)})du^M_{(j)} = dF^{(a)}_{N(j)} + dF^{(T)}_{N(j)} + dF^{(N)}_{N(j)} \tag{42}$$

in which $K_{NM}^{(e)}$, $K_{NM}^{(G)}$, and $K_{NM}^{(P)}$ are the standard stiffness matrices representing linear elastic, geometrically nonlinear, and plastic behavior, respectively,

$$K_{NM}^{(e)} = \int_V E^{ijk\ell}A_{Nij}A_{Mk\ell}dv$$

$$K_{NM}^{(G)} = \int_V 2\,\sigma^{ij}C_{NMij}dv$$

$$K_{NM}^{(P)} = \int_V \overset{*}{E}{}^{ijk\ell}A_{Nij}A_{Mk\ell}dv$$

and the incremental thermoelastoplastic load vector $dF_N^{(T)}$ is

$$dF^{(T)}_{N(j)} = \{\int_V (B^{ij} + \overset{*}{B}{}^{ij})\Omega_R(A_{Nij} + 2C_{NMij}u^M_{(j-1)})dv\}dT^R_{(j-1)}$$

$$+ \{\int_V B^{ij}\Omega_R C_{NMij}dv\}du^M_{(j-1)}T^R_{(j-1)} \tag{43}$$

All the rest of the terms in (41) other than those mentioned above may be grouped in $F^{(N)}_{N(j)}$ called the pseudo nonlinear load vector but may be dropped because of their negligible effects.

6. SOLUTION PROCEDURE

We are now provided with incremental heat conduction equations and equilibrium equations. Either thermal loads or mechanical loads or both may be applied. Depending on loading and boundary conditions, we can either start from equilibrium equations or heat conduction equations, but both equations must be solved iteratively within a time increment. Any existing recursive formula for step-by-step integration or difference operator may be used to solve heat conduction equations. In the present study a difference operator for linear variation of temperature within a time increment is combined with iterative solution of plastic equilibrium equations.

The incremental temperature at any time step k is given by [14,15].

$$\underset{\sim}{T}_{(k)} = 2\left(\frac{2}{\Delta t}\underset{\sim}{N} + \underset{\sim}{R}\right)^{-1}\left\{\underset{\sim}{P}_{(k - \frac{\Delta t}{2})} + \frac{2}{\Delta t}\underset{\sim}{R}\,\underset{\sim}{T}_{(k-1)}\right\} - \underset{\sim}{T}_{(k-1)} \tag{44}$$

Here $\underset{\sim}{N}$, $\underset{\sim}{R}$, $\underset{\sim}{P}$, and $\underset{\sim}{T}$ are assembled forms of N_{NM}, R_{NM}, $P_R^{(Q)} + P_R^{(q)} + P_R^{(c)} + P_R^{(Ep)} + P_R^{(TP)}$, and T^R, respectively. The reference temperature and initial thermal input can easily be incorporated in (44) and the temperature change at the end of the first time increment calculated.

The results of this solution are used in the assembled incremental equations of equilibrium to determine the displacements and stresses. These stresses are checked, element by element, for yielding. If any element has yielded the plastic tangent stiffness matrix is constructed and the standard iterative cycles are repeated until convergence is achieved [10,11,12].

With the final values of displacements, it is possible to calculate the displacement rates by

$$\dot{u}_k = (u_k - u_{k-1})/\Delta t \tag{45}$$

for use in the heat conduction equations, Returning to the heat conduction equations for the second time increment, the process is repeated as before except that the elastoplastic coupling is now to be included. The elastoplastic and thermoplastic model as determined in the converged solution of the plastic equilibrium equations of (42) will be used in the heat conduction equations. The marching with time increments, thus, continues until the desired length of time has been reached. Details of the computer program are given in Appendix A – Capabilities and Limitations of the Program, Appendix C – Flow Chart, Appendix D – Subroutines Organization Chart, Appendix E – Description of Subroutines, Appendix F – Data Input Format, and Appendix G – Program Listing.

In the present study, we use temperature and displacement approximations based on a three-dimensional linear isoparametric function and the integration is performed by an 8 point Gaussian quadrature [16] (see Appendix B).

7. APPLICATIONS

In order to verify, first of all, correctness of the present approach a comparison study was made for an uncoupled heat conduction of a beam reported by Wah [17] who used a classical series solution and substantiated his results with Boley [19]. An excellent agreement was obtained as shown in Figure 1.

Next, a series of example problems were tested to determine effects of various terms in the governing equations; namely, behavior due to coupling and non-coupling, linear elastic and elastoplastic properties. Figure 2 shows the information on geometry, boundary conditions, material constants and temperature input. Only one-quarter of the symmetrical three-dimensional solid is shown. The temperature change of 200°C is applied at nodes of the center-left end element. The material properties given in Figure 2 represent a mild steel. The transient temperature distribution in the direction of x with $y = z = 0$ is shown in Figure 3. Effects of elastic and elastoplastic couplings are studied in this case. It is interesting to note that there exists little difference in temperature distribution between elastic of elastoplastic coupling for the time period examined. However, the displacement w at $y = 100$ mm and $z = 300$ mm becomes larger for elastoplastic coupling than that for elastic coupling after approximately one hour as shown in Figure 4. The changes of w displacements vs. time at point A are plotted in Figure 5. It is seen that uncoupled displacements are larger than the coupled displacements, a trend confirmed by Oden and Poe [18] in their study of thermoelastic one-dimensional problems. It should be noted that the elastoplastic displacements are larger than the elastic displacements in both cases. Figure 6 reveals an interesting fact that temperature is also lower for a coupled case than for an uncoupled case but little difference is noted for either elastic or elastoplastic behavior. Figure 7 shows the stress in z direction σ_z from that for elastic coupling, a fact well known in mechanics. Finally, plastic regions developing with elapse of time are shown in Figure 8, indicating little effects of coupling.

The effects of convection through uninsulated surfaces with the film coefficient $\bar{\alpha}$ = 1.0 Kg/mm hr ^{o}C, ambient temperature T' = 1000^{o}C and the heat flux q = -100 Kg/mm hr at z = 300 mm on the x-y plane are investigated and the results are shown in Figure 9. Large elastoplastic displacements (w) result due to surface heating together with ambient conditions. Variations of material properties from element to element are accommodated in the program and an example problem for such case is described in Figure 10. Once again the transient temperature distributions are almost identical for elastic coupling and elastoplastic coupling as shown in Figure 11. Significant deviations exist, however, for displacements (w) (Figure 12) between elastic and elastoplastic couplings as temperature rises as noted earlier for the uniform material. Because of ambient temperature and heat flux through uninsulated surfaces and variable material properties throughout the structure, the pattern of development of plastic regions (Figure 13) differs considerably from that of the previous example of uniform material and insulated surface. In the foregoing example problems, geometric nonlinearities are excluded for the interest of computing time.

It should be noted that for the case of an isotropic solid, the tensor of thermoplastic moduli (23) and expressions of (29a), (29b), (29c), and (29e) are zero but must be updated in the case of an anisotropic solid as plastic deformation progresses [9]. Although the temperature dependent material properties, finite strains, and the dynamic-coupled inertia effects can easily be handled, such applications are not included in the present study.

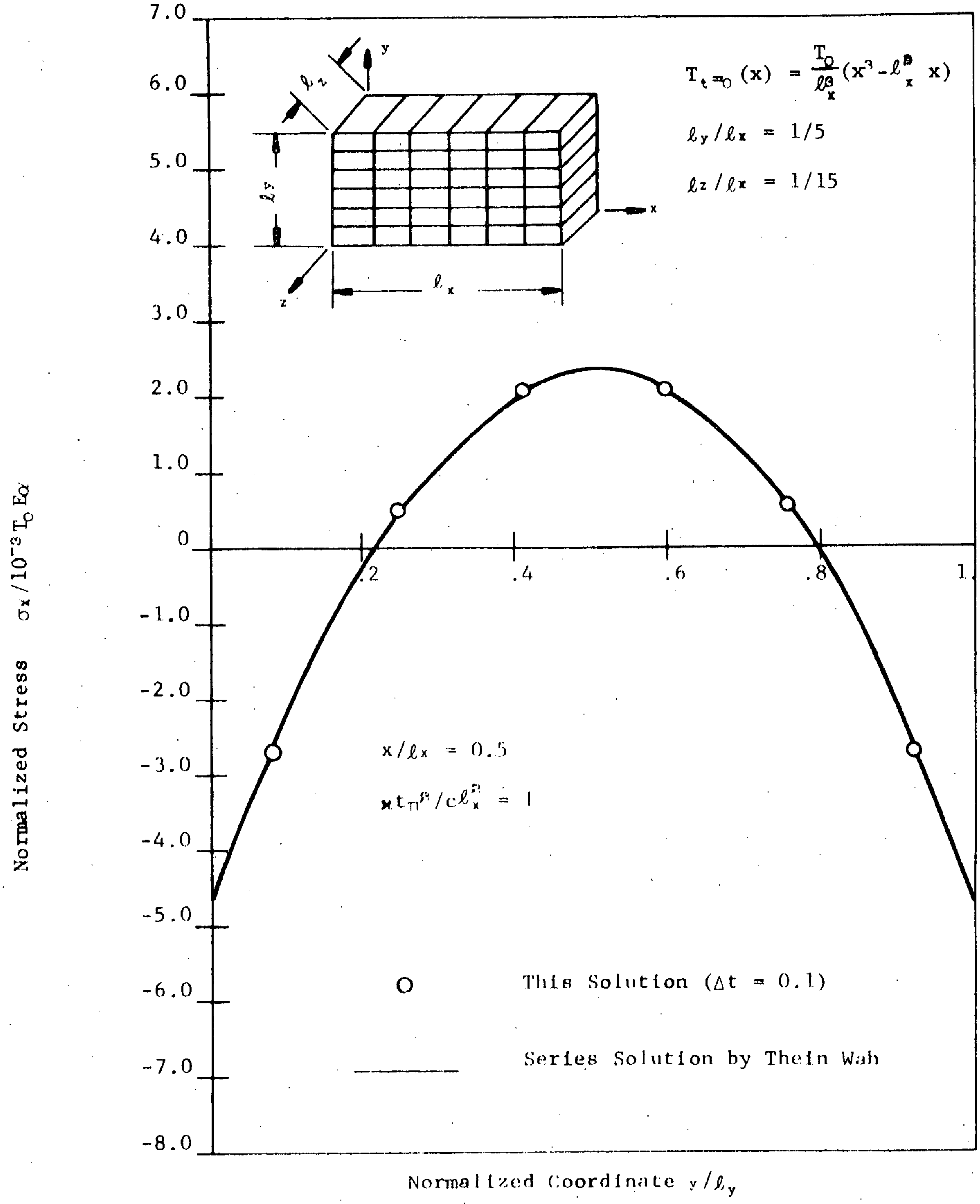
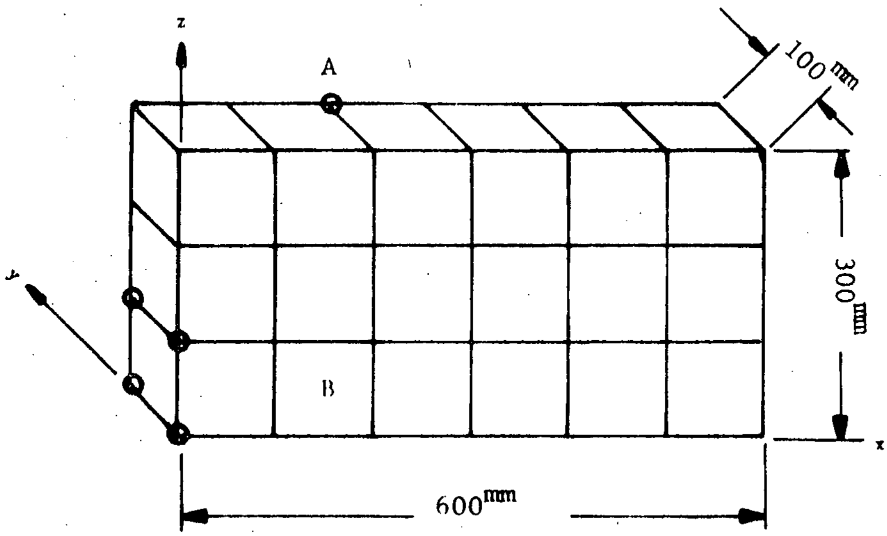


Figure 1 Transient thermal stress distribution in the free-free beam, uncoupled.



DISPLACEMENT BOUNDARY CONDITIONS:

u=v=w=0 At x=0 and x=600 mm

v=0 At y=0

w=0 At z=0

TEMPERATURE BOUNDARY CONDITIONS:

Insulated on all the surfaces except at the points (x,y,z) = (0,0,0), (0,100,0), (0,0,100) and (0,100,100) which are kept at 200°C.

CONSTANTS:

$E = 2.0 \times 10^4$ (kg/mm²), $E_{(p)} = 2.0 \times 10^3$ (kg/mm²), $\sigma_Y = 25.0$ (kg/mm²)

$\nu = 0.3$, $\alpha = 1.3 \times 10^{-5}$ (/°C) , $\kappa = 9.0 \times 10^3$ (kg/hr °C)

$c = 0.3$ (kg/mm² °C) , $T_o = 27$ (°C) , $\Delta t = 0.05$ (hrs.)

Figure 2 Discretized geometry of three dimensional solid and input data.

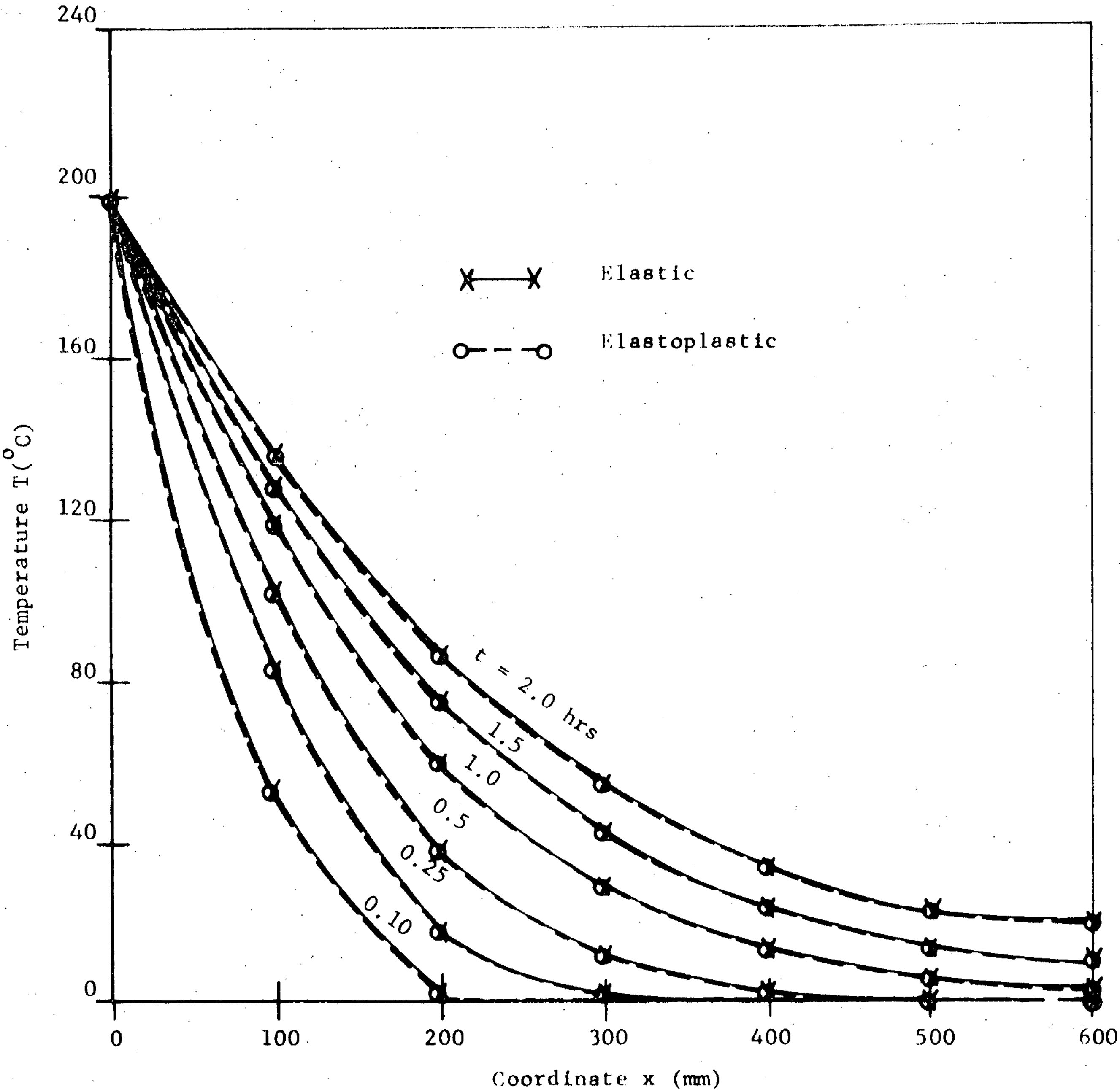


Figure 3 Temperature distribution of y = z = 0 in Fig. 2, coupled.

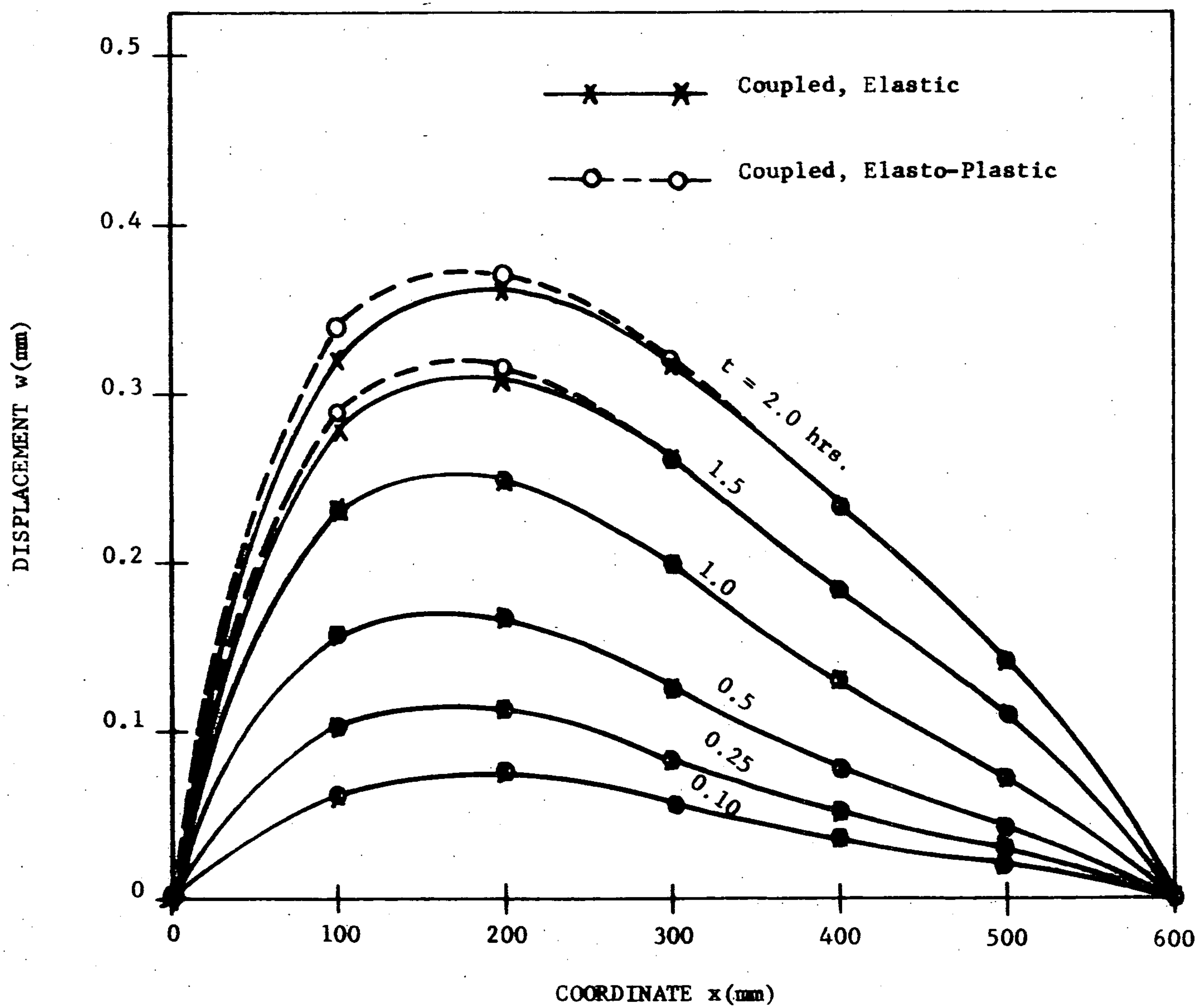


Figure 4 Displacement (w) at y = 100mm, z = 300mm in the x - direction.

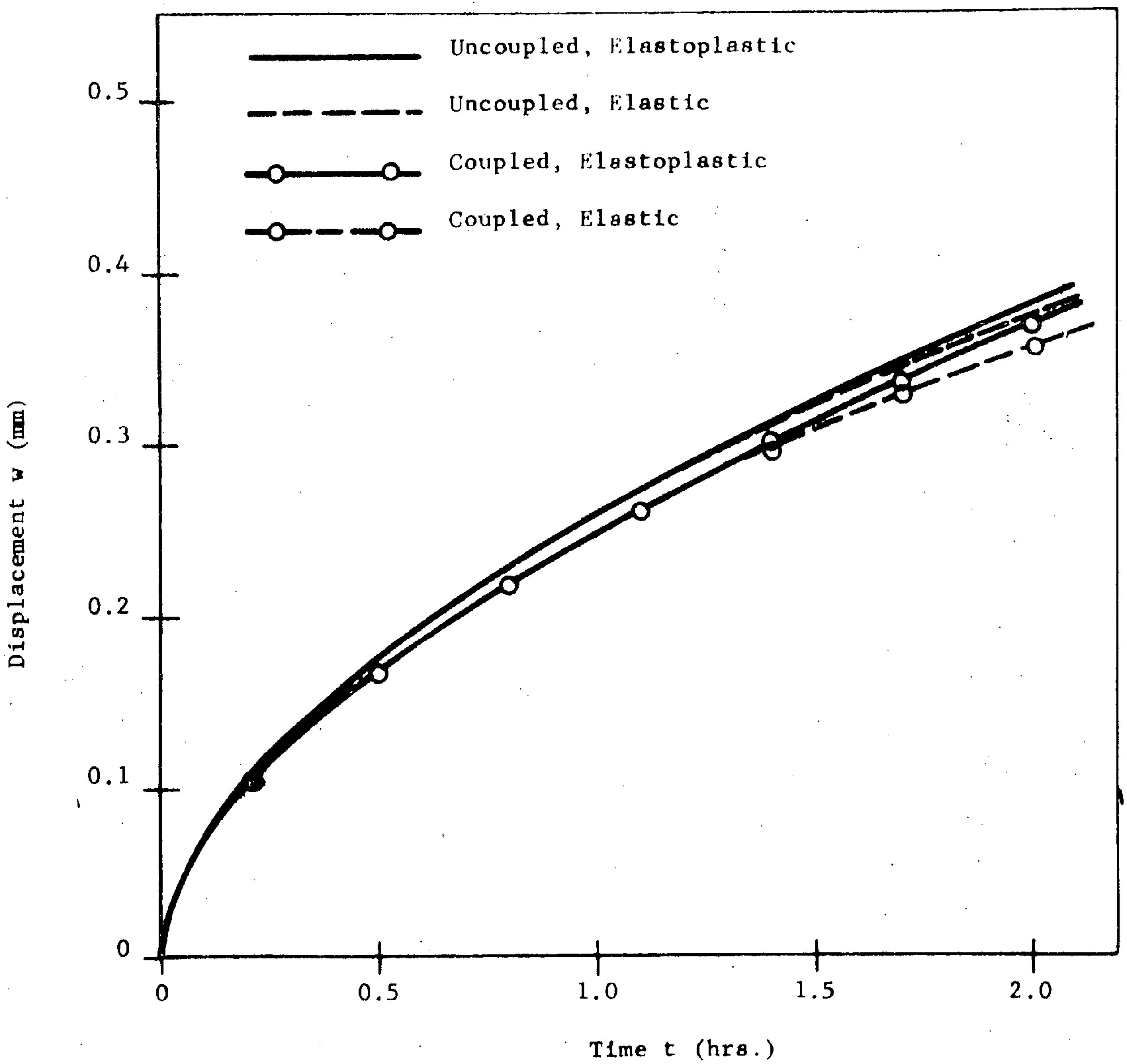


Figure 5 Transient Displacement w at point A of Fig. 2.

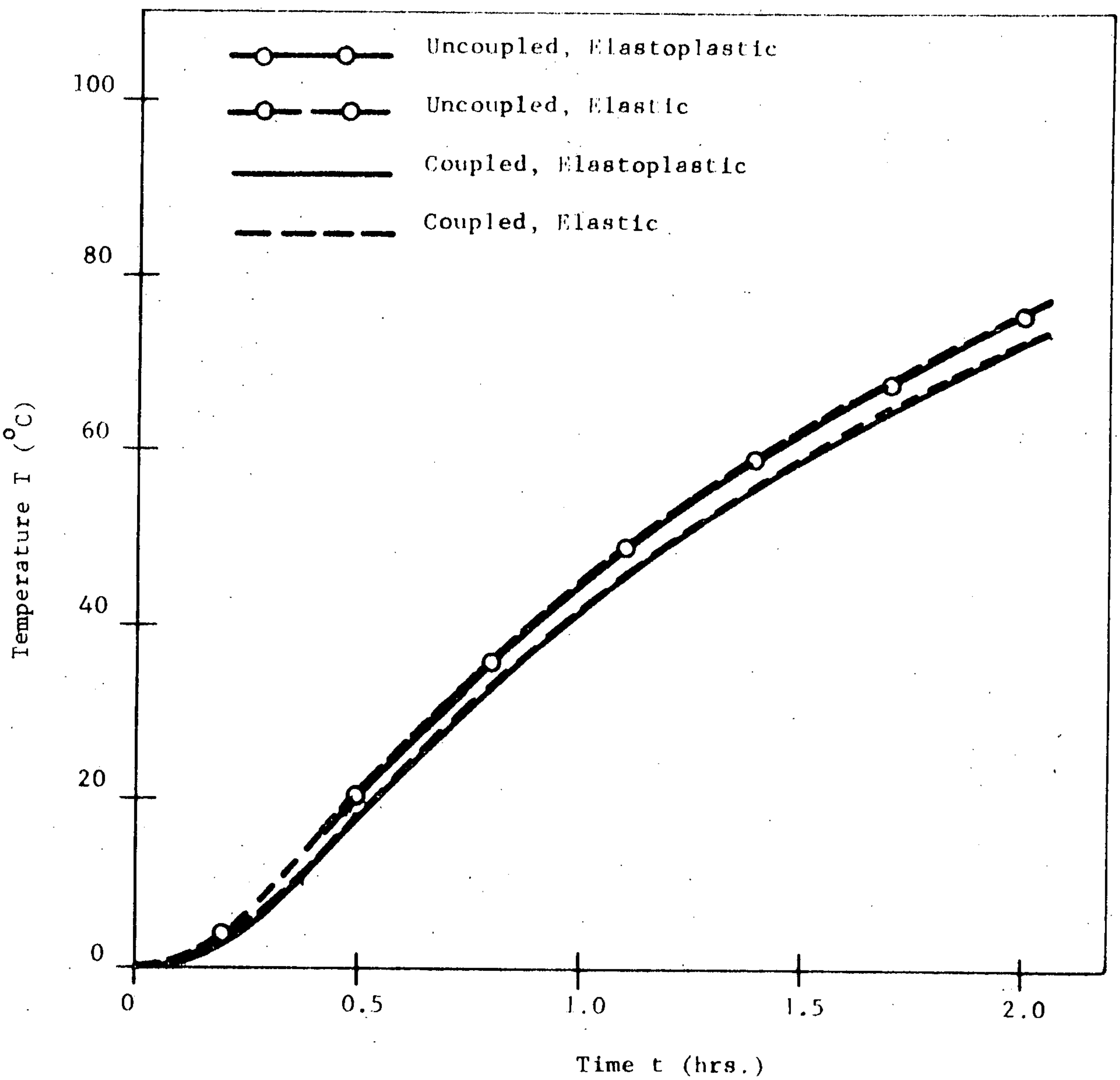


Figure 6 Transient temperature change at point A of Fig. 2.

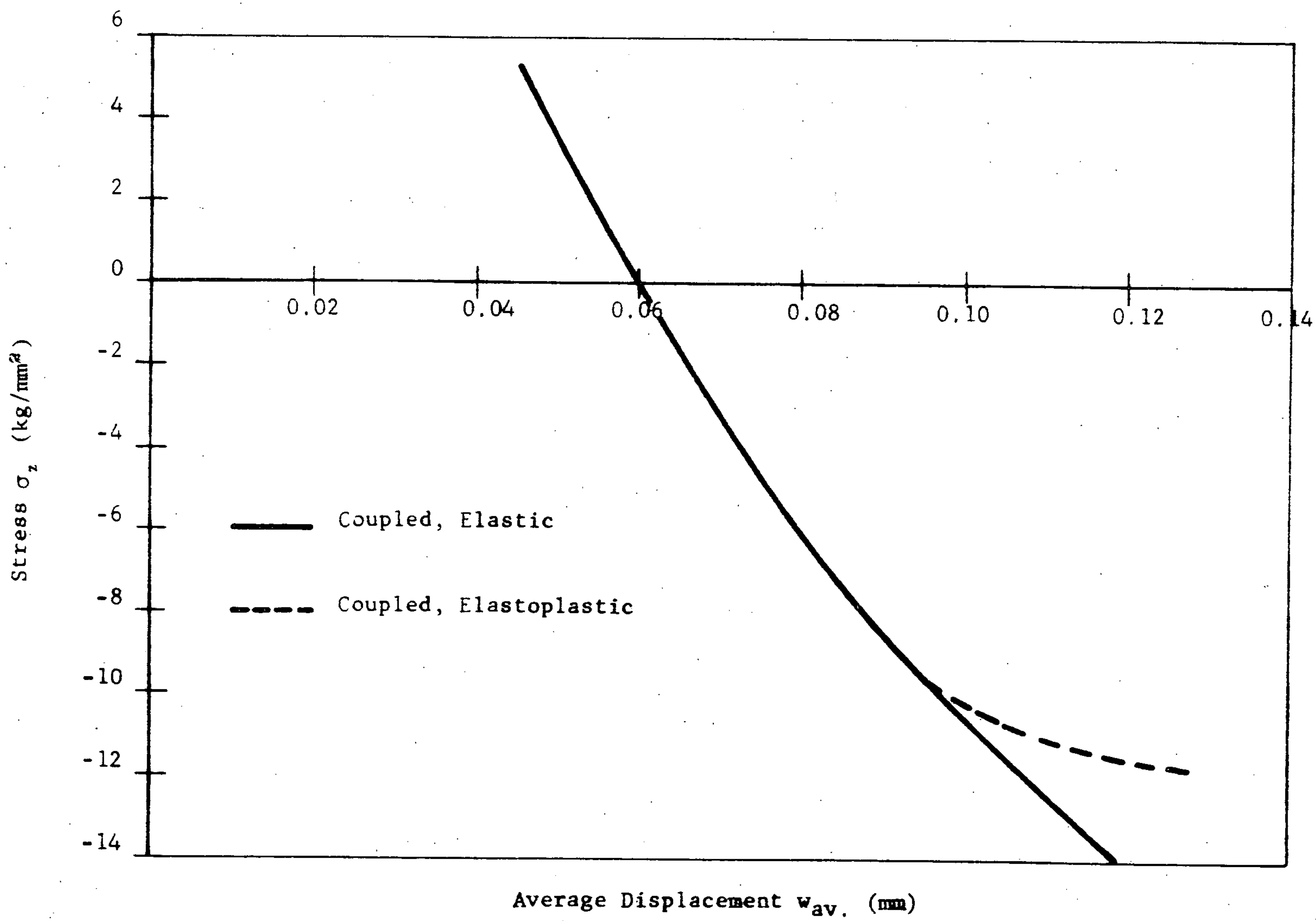


Figure 7 Stress σ_z - displacement w for element B of Fig. 2.

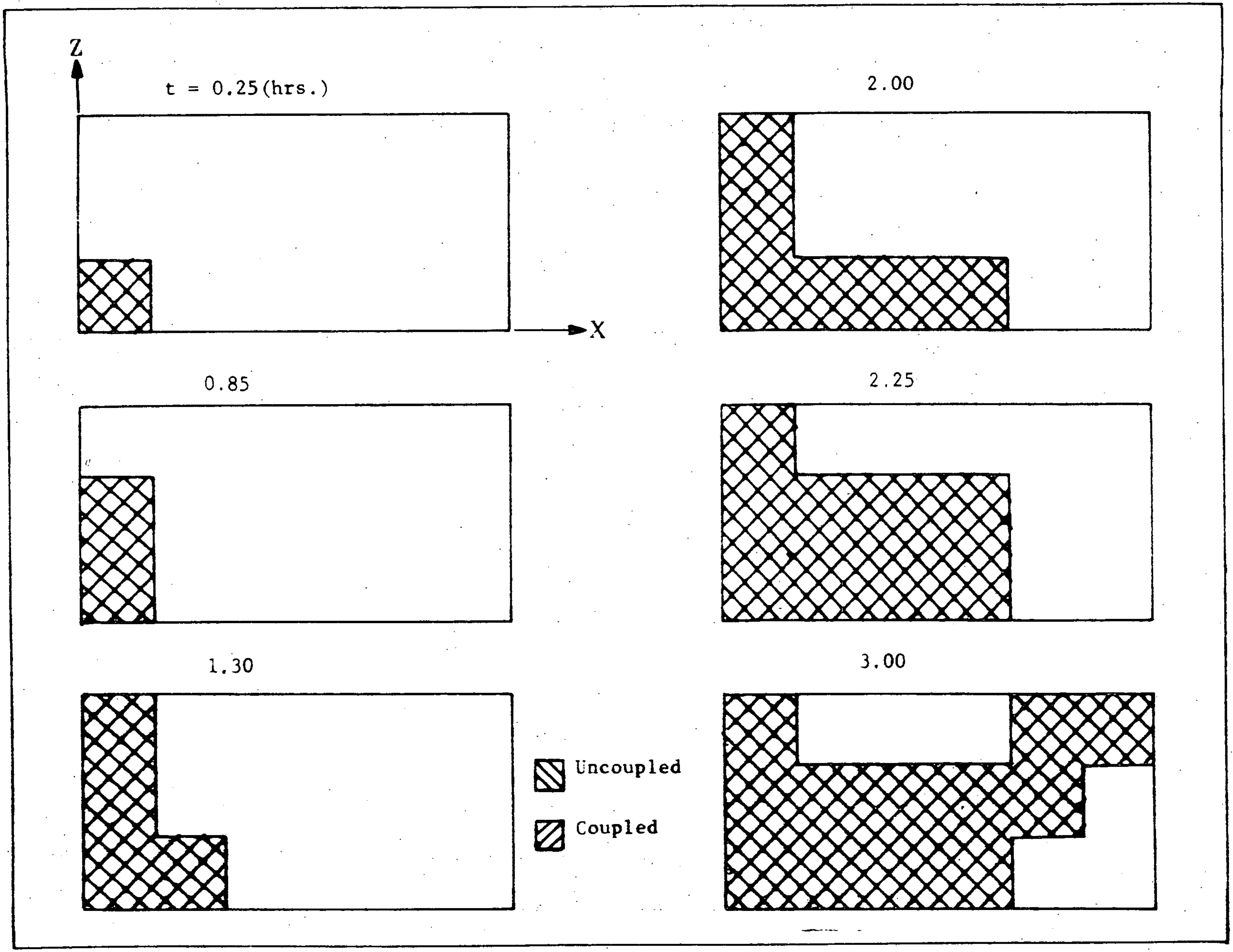


Figure 8 Development of plastic regions, coupled and uncoupled.

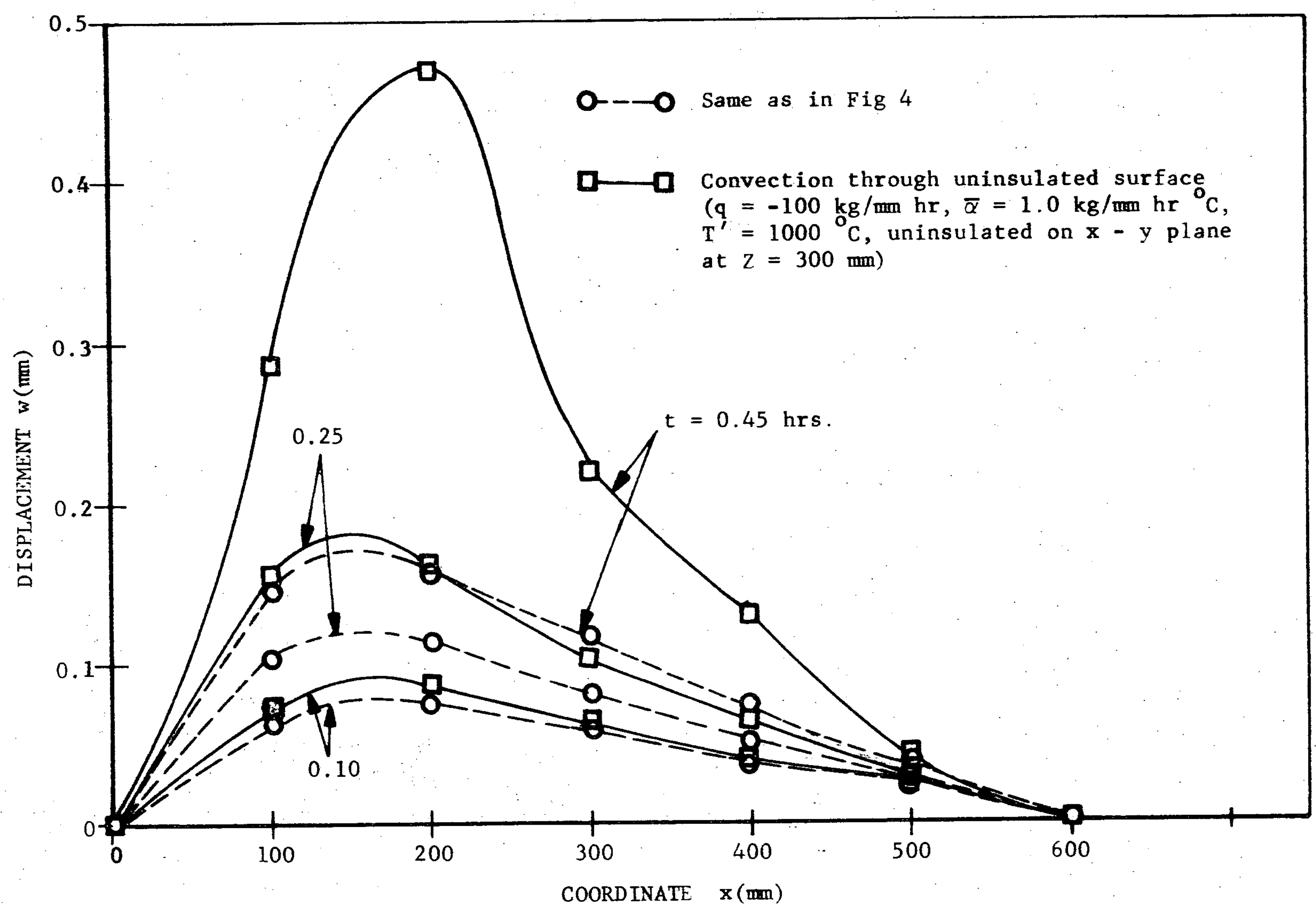
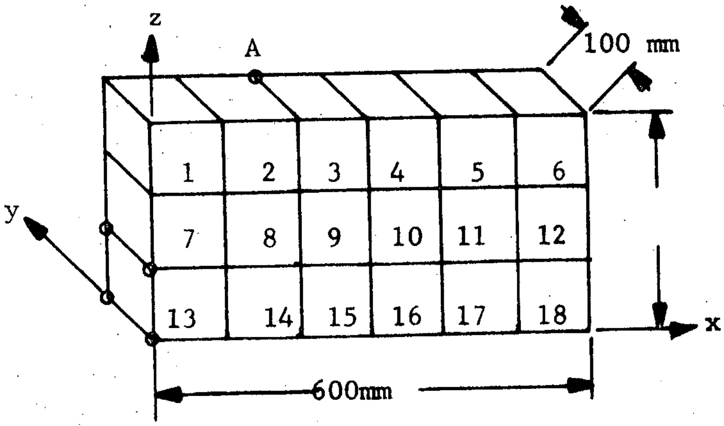


Figure 9: Comparison of Coupled Elastoplastic Displacements (w) With and Without Surface Insulation at y = 100 mm, z = 300 mm in the x-Direction.



DISPLACEMENT BOUNDARY CONDITIONS:

$u = v = w = 0$	At $x = 0$ and $x = 600$ mm
$v = 0$	At $y = 0$
$w = 0$	At $z = 0$

TEMPERATURE BOUNDARY CONDITIONS:

$T = 200^{\circ}C$ at the points $(x,y,z) = (0,0,0)$, $(0,100,0)$, $(0\ 0,100)$ and $(0,100,100)$.

$\bar{\alpha} = 1.0$(kg/mm.hr $^{\circ}$C), $T' = 1000.0$ ($^{\circ}$C) and $q = -100.0$(kg/mm.hr) on the surface $Z = 300$ mm.

Insulated on all other surfaces.

CONSTANTS:

$E = 2.0 \times 10^4$ (kg/mm^2)	for elements 1 ~ 6,
$E = 1.0 \times 10^4$ (kg/mm^2)	for elements 7 ~ 12,
$E = 0.7 \times 10^4$ (kg/mm^2)	for elements 13 ~ 18,

$E_{(p)} = 1.0 \times 10^3$ (kg/mm^2) , $\sigma_Y = 9.5$(kg/mm^2) , $\nu = 0.3$

$\alpha = 1.3 \times 10^{-5}$ (/$^{\circ}$C) , $x = 9.0 \times 10^3$ (kg/hr $^{\circ}$C)

$c = 0.3$ (kg/mm^2 $^{\circ}$C) , $T_o = 27(^{\circ}C)$., $\Delta t = 0.05$ (hrs.)

Figure 10: Discretized Geometry of Three-Dimensional Solid and Input Data.

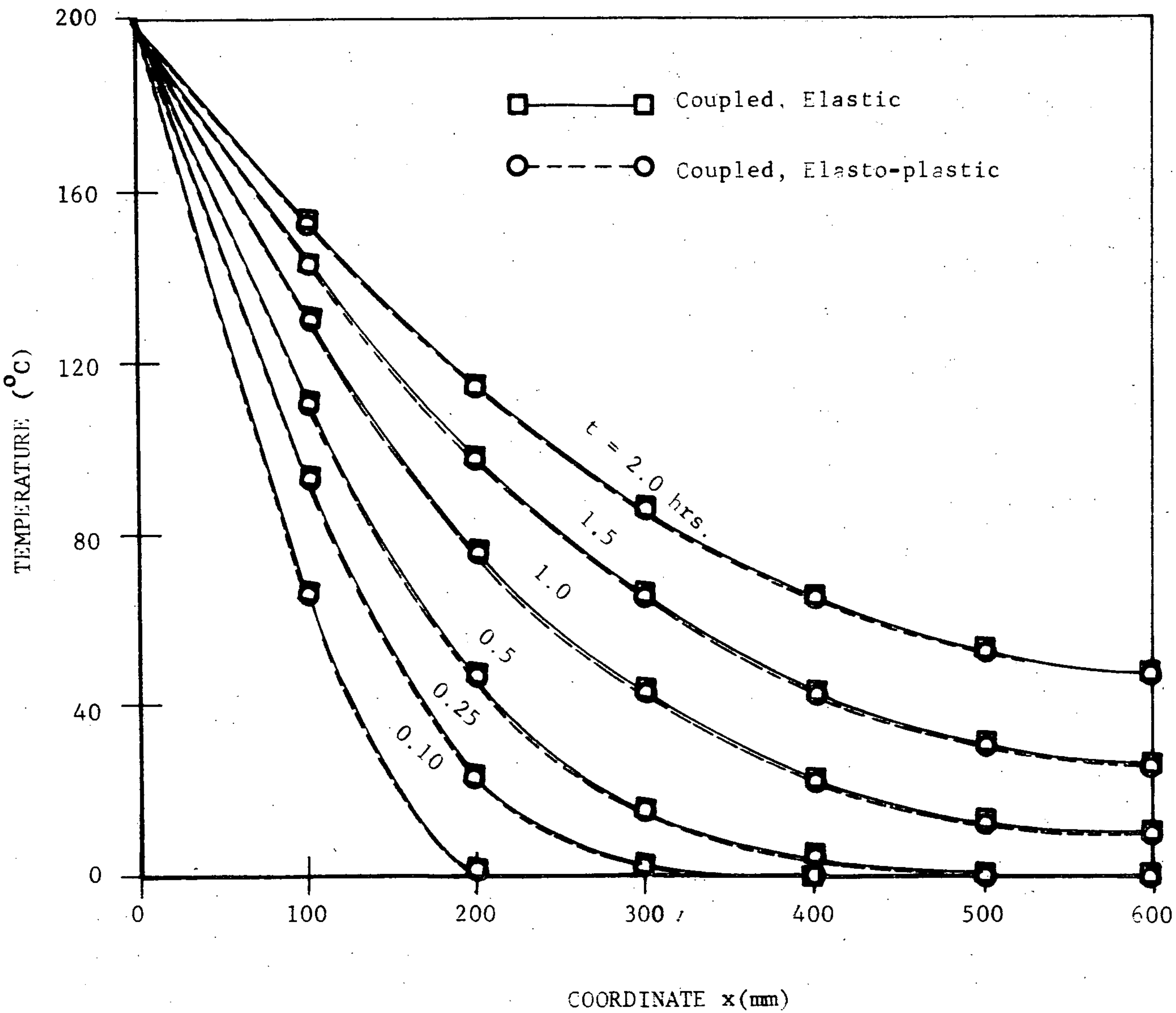


Figure 11: Temperature Distribution of y = z = 0 in Figure 10.

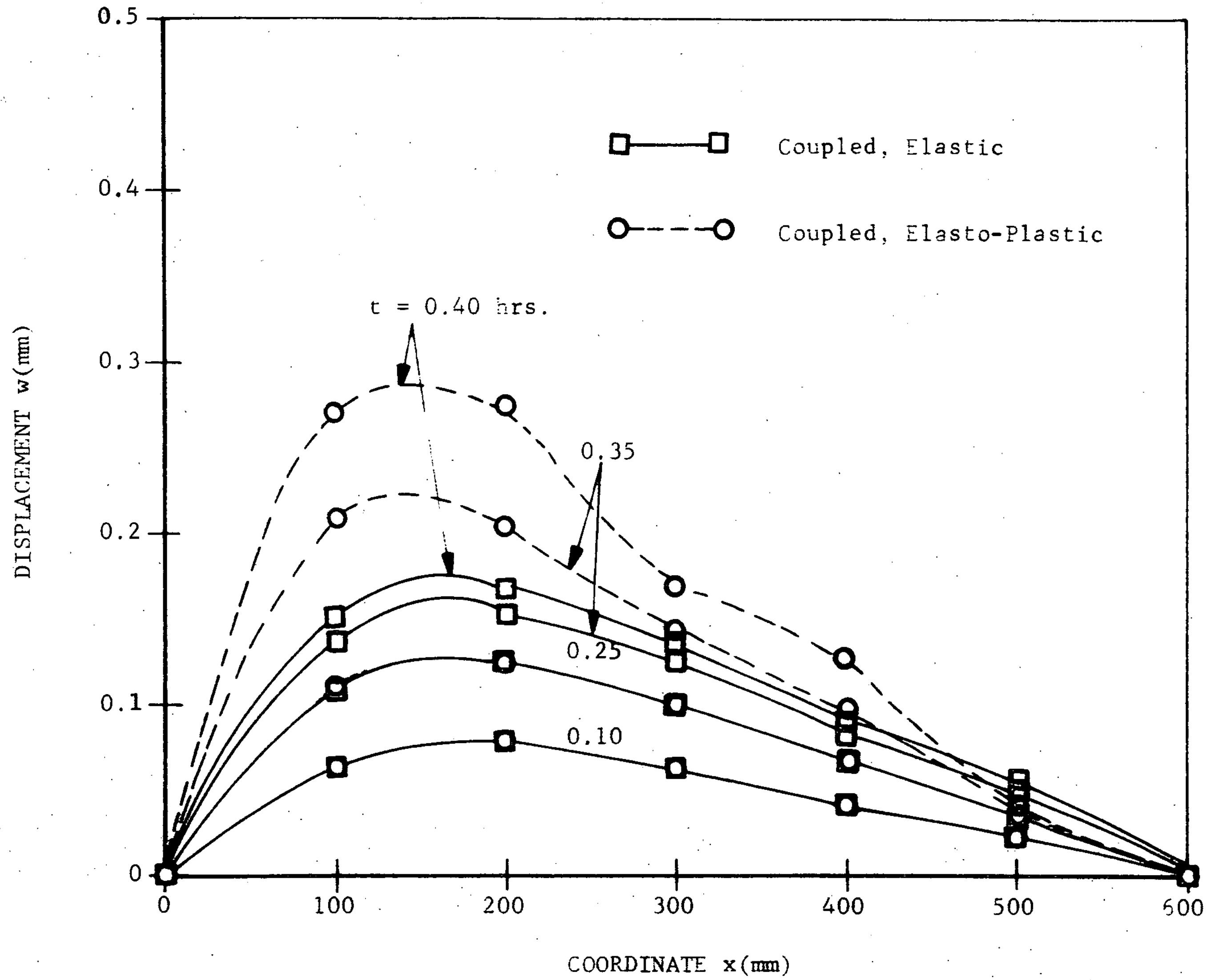


Figure 12: Displacement (w) at y = 100 mm, z = 300 mm in the x-Direction in Figure 12.

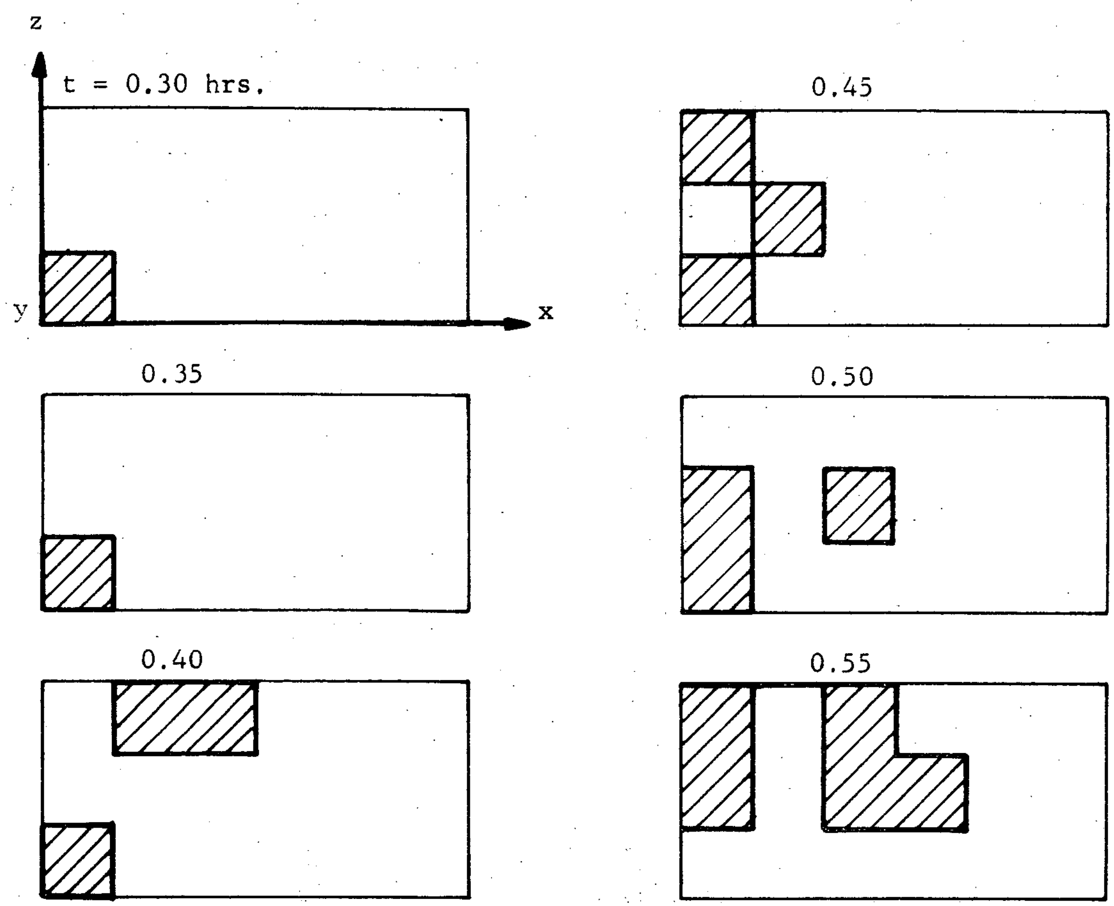


Figure 13: Development of Plastic Regions in Figure 10.

8. CONCLUDING REMARKS

A three-dimensional thermoelastoplastic analysis has been carried out using the incremental theory consistent with the first and second laws of thermodynamics. Complicated mathematical operations emanating from the functional theory or state variables are replaced by proposing an incremental free energy as a function of total and inelastic strain and temperature unique only within a small time increment. The similar incremental functional dependency is then valid for stresses, entropy, and heat flux. Such treatment lends itself to numerical techniques taking advantage of the finite element method and time integration by difference operators.

For the example problems and material properties considered in this study it appears that elastoplastic coupling is significant for displacements and stresses, but that neither elastic nor elastoplastic coupling has any effect on temperature distribution. It may be argued, however, that for certain types of material and geometry these conclusions would not necessarily be true. Exhaustive study on such effects is beyond the scope of the present report.

REFERENCES

1. Biot, M. A., "Theory of Stress-Strain Relations in Anisotropic Viscoelasticity and Relaxation Phenomena," J. of Applied Physics, 25, 1954, pp. 1385-1391.

2. Coleman, B. D. and Gurtin, M. E., "Thermodynamics with Internal State Variables," J. of Chemical Physics, Vol. 47, No. 2, July, 1967, pp. 597-613.

3. Perzyna, P., and Wojno, W., "Thermodynamics of a Rate Sensitive Plastic Material," Arch. Mechanics Stos., 1968, pp. 501-510.

4. Kratochvil, J. and Dillon, O. W., "Thermodynamics of Elastic-Plastic Materials as a Theory with Internal State Variables," J. of Applied Physics, Vol. 40, No. 8, July 1969, pp. 3207-3218.

5. Valanis, K. C., "Theory of Viscoplasticity without Yield Surface," Arch. of Mechanics Stos., 4, 1971, pp. 517-533.

6. Lee, E. H. and Liu, D. T., "Finite Strain Elastic-Plastic Theory Particularly for Plane Wave Analysis," J. of Applied Physics, 38, 1967, pp. 19-27.

7. Green, A. E. and Naghdi, P. M., "General Theory of Elastic-Plastic Continuum," Archives of Rational Mechanics and Analysis, 18, 1965, pp. 251-281.

8. Shapery, R. A., "On a Thermodynamic Constitutive Theory and its Application to Various Nonlinear Materials," IUTAM Symposium on Thermoinelasticity, Edited by B. A. Boley, Springer-Verlag Wien, New York, 1970, pp. 259-285.

9. Oden, J. T. and Bhandari, D., "A Theory of Thermo-Plastic Materials with Memory," UARI Research Report No. 118, AFOST-IR-0904, University of Alabama in Huntsville, April, 1972.

10. Marcal, P. V., "Finite Element Analysis with Material Nonlinearities - Theory and Practice," Proc. of the U.S.-Japan Seminar on Matrix Methods in Structural Analysis and Design, Tokyo, 1969.

11. Yamada, Y., Yoshimura, N., and Sakurai, T., "Plastic Stress-Strain Matrix and its Application for the Solution of Elastic-Plastic Problems by the Finite Element Method," Int. J. of Mech. Sciences, Vol. 10, 1968, pp. 343-354.

12. Chung, T. J. and Eidson, R. L., "Dynamic Analysis of Viscoelasto-plastic Anisotropic Shells," Int. J. of Computers and Structures, to appear.

13. Oden, J. T., Finite Elements of Nonlinear Continua, McGraw-Hill Book Co., 1972.

14. Wilson, E. L. and Nickell, R. E., "Application of the Finite Element Method to Heat Conduction Analysis," Nucl. Engineering Design, Vol. 4, No. 3, 1966, pp. 276-286.

15. Chung, T. J. and Bhutwala, J. N., "Steady State and Transient Heat Conduction and Thermal Stress Analysis of Two and Three Dimensional Structures," UARI Research Report No. 96, The University of Alabama in Huntsville, January, 1971.

16. Zienkiewicz, O. C., Finite Element Method in Engineering Science, McGraw-Hill Book Co., 1971.

17. Wah, T., "Thermal Stresses in Thin Beams," J. of Solids and Structures, Vol. 2, 1966, pp. 293-305.

18. Oden, J. T. and Poe, J., "On the Numerical Solution of a Class of Problems in Dynamic Coupled Thermoelasticity," Developments in Theoretical and Applied Mechanics, V, edited by D. Frederick, Pergamon Press, Oxford, 1970.

19. Boley, B. A., "Determination of Temperature, Stresses and Deflections in Two-Dimensional Thermoelastic Problems," J. of Aeronautical Science 22, 1956.

APPENDIX A

CAPABILITIES AND LIMITATION OF THE PROGRAM

General. This program analyzes a three-dimensional solid subjected to both thermal and mechanical loadings. The program takes into account the elastoplastic behavior coupled with transient heat conduction. The formulation is based on the first and second laws of thermodynamics, von Mises yield criteria, associated Prandtle-Reuss flow rule, and the linear Fourier law. The finite element discretization by means of a linear isoparametric interpolation for both temperature and displacement fields is utilized. Integration is performed by Gaussian quadrature. A step-by-step time integration assuming a linear variation of temperature within a time increment is used to solve heat conduction equations. Capabilities and limitations of this program are listed as follows:

(1) Capable of handling up to 60 nodes and 30 elements.

(2) Temperatures may be specified at nodes (50).

(3) Heat flux and heat supply may be specified on the element surface (50) and inside the entire element solid (30), respectively.

(4) Surfaces may be insulated or exposed to ambient temperatures.

(5) Capable of incorporating 100 restrained generalized coordinates.

(6) Geometric nonlinearities are not considered in the program.

(7) Capable of handling laminated structure with varying material properties from element to element.

APPENDIX B

VARIOUS INTEGRALS IN ISOPARAMETRIC ELEMENT

In the present study, linear hexahedral isoparametric elements are used to model the three dimensional solids and consequently constitute the basis for displacement and temperature fields. Although details of isoparametric elements may be found in Zienkiewicz (1972), some of the integrals essential in the program by Gaussian quadrature are shown in explicit form:

$$\int_v \psi_N dv = \frac{1}{8^2} \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} (1 + \xi\xi_N)(1 + \eta\eta_N)(1 + \xi\xi_N)\left|J\right| d\xi d\eta d\zeta$$

$$\int_v \psi_M \psi_M dv = \frac{1}{8^2} \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} (1 + \xi\xi_N)(1 + \eta\eta_N)(1 + \zeta\zeta_N)(1 + \xi\xi_M)(1 + \eta\eta_M)$$

$$(1 + \zeta\zeta_M)\left|J\right| d\xi d\eta d\zeta$$

$$\int_v \psi_N \psi_M \psi_R dv = \frac{1}{8^3} \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} (1 + \xi\xi_N)(1 + \eta\eta_N)(1 + \zeta\zeta_N)(1 + \xi\xi_M)$$

$$(1 + \eta\eta_M)(1 + \zeta\zeta_M)(1 + \xi\xi_R)(1 + \eta\eta_R)(1 + \zeta\zeta_R)$$

$$\times \left|J\right| d\xi d\eta d\zeta$$

$$\int_v \psi_N A_{Mij} dv = \frac{1}{8} \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} (1 + \xi\xi_N)(1 + \eta\eta_N)(1 + \zeta\zeta_N) A_{Mij} \left|J\right| d\xi d\eta d\zeta$$

$$\int_v \psi_N \psi_Q A_{M1j} dv = \frac{1}{8} \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} (1 + \xi\xi_N)(1 + \eta\eta_N)(1 + \zeta\zeta_N)$$

$$(1 + \xi\xi_Q)(1 + \eta\eta_Q)(1 + \zeta\zeta_Q) \times A_{M1j} \; |J| \; d\xi d\eta d\zeta$$

$$\int_v \psi_{N,1} \psi_{M,j} dv = \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} \psi_{N,1} \psi_{M,j} \; |J| \; d\xi d\eta d\zeta$$

$$\int_v A_{Mmn} A_{N1j} dv = \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} A_{Mmn} A_{N1j} \; |J| \; d\xi d\eta d\zeta$$

$$\int_A \psi_N dA = \frac{1}{8} \int_{-1}^{1} \int_{-1}^{1} (1 + \xi\xi_N)(1 + \eta\eta_N)(1 + \zeta\zeta_N) \; d\bar{A}$$

and

$$\int_A \psi_N \psi_M dA = \frac{1}{8} \int_{-1}^{1} \int_{-1}^{1} (1 + \xi\xi_N)(1 + \eta\eta_N)(1 + \zeta\zeta_N)$$

$$(1 + \xi\xi_M)(1 + \eta\eta_M)(1 + \zeta\zeta_M) \; d\bar{A}$$

where the differential area $d\bar{A}$ is expressed as

$$d\bar{A} = \left[\left(\frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \eta} - \frac{\partial z}{\partial \xi} \frac{\partial y}{\partial \eta} \right)^2 + \left(\frac{\partial z}{\partial \xi} \frac{\partial x}{\partial \eta} - \frac{\partial x}{\partial \xi} \frac{\partial z}{\partial \eta} \right)^2 + \left(\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi} \frac{\partial x}{\partial \eta} \right)^2 \right]^{\frac{1}{2}} d\xi d\eta$$

for $\zeta = \pm 1$ plane

$$= \left[\left(\frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \zeta} - \frac{\partial z}{\partial \eta} \frac{\partial y}{\partial \zeta} \right)^2 + \left(\frac{\partial z}{\partial \eta} \frac{\partial x}{\partial \zeta} - \frac{\partial x}{\partial \eta} \frac{\partial z}{\partial \zeta} \right)^2 + \left(\frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \zeta} - \frac{\partial y}{\partial \eta} \frac{\partial x}{\partial \zeta} \right)^2 \right]^{\frac{1}{2}} d\eta d\zeta$$

for $\xi = \pm 1$ plane

$$= \left[\left(\frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \xi} - \frac{\partial z}{\partial \zeta}\frac{\partial y}{\partial \xi} \right)^2 + \left(\frac{\partial z}{\partial \zeta}\frac{\partial x}{\partial \xi} - \frac{\partial x}{\partial \zeta}\frac{\partial z}{\partial \xi} \right)^2 + \left(\frac{\partial x}{\partial \zeta}\frac{\partial y}{\partial \xi} - \frac{\partial y}{\partial \zeta}\frac{\partial x}{\partial \xi} \right)^2 \right]^{\frac{1}{2}} d\zeta d\xi$$

$$\text{for } \eta = \pm 1 \text{ plane}$$

Here, displacement and temperature fields are related by

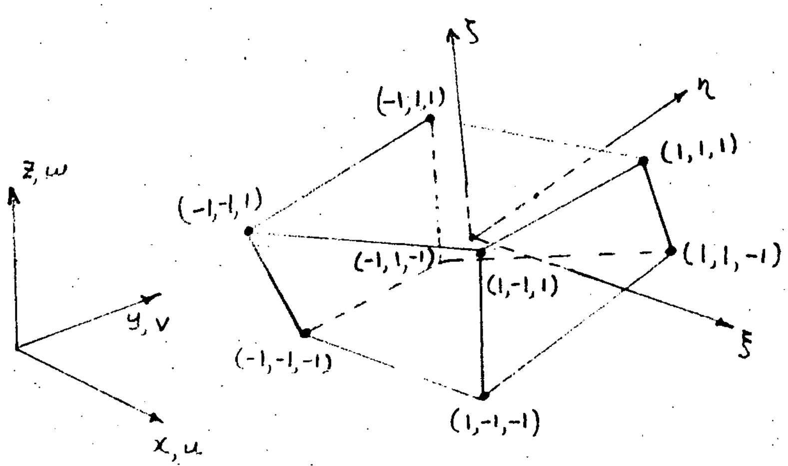
$$u = \sum_{i=1}^{8} \psi_i u_i, \text{ etc., and } T = \sum_{i=1}^{8} \psi_i T_i$$

where

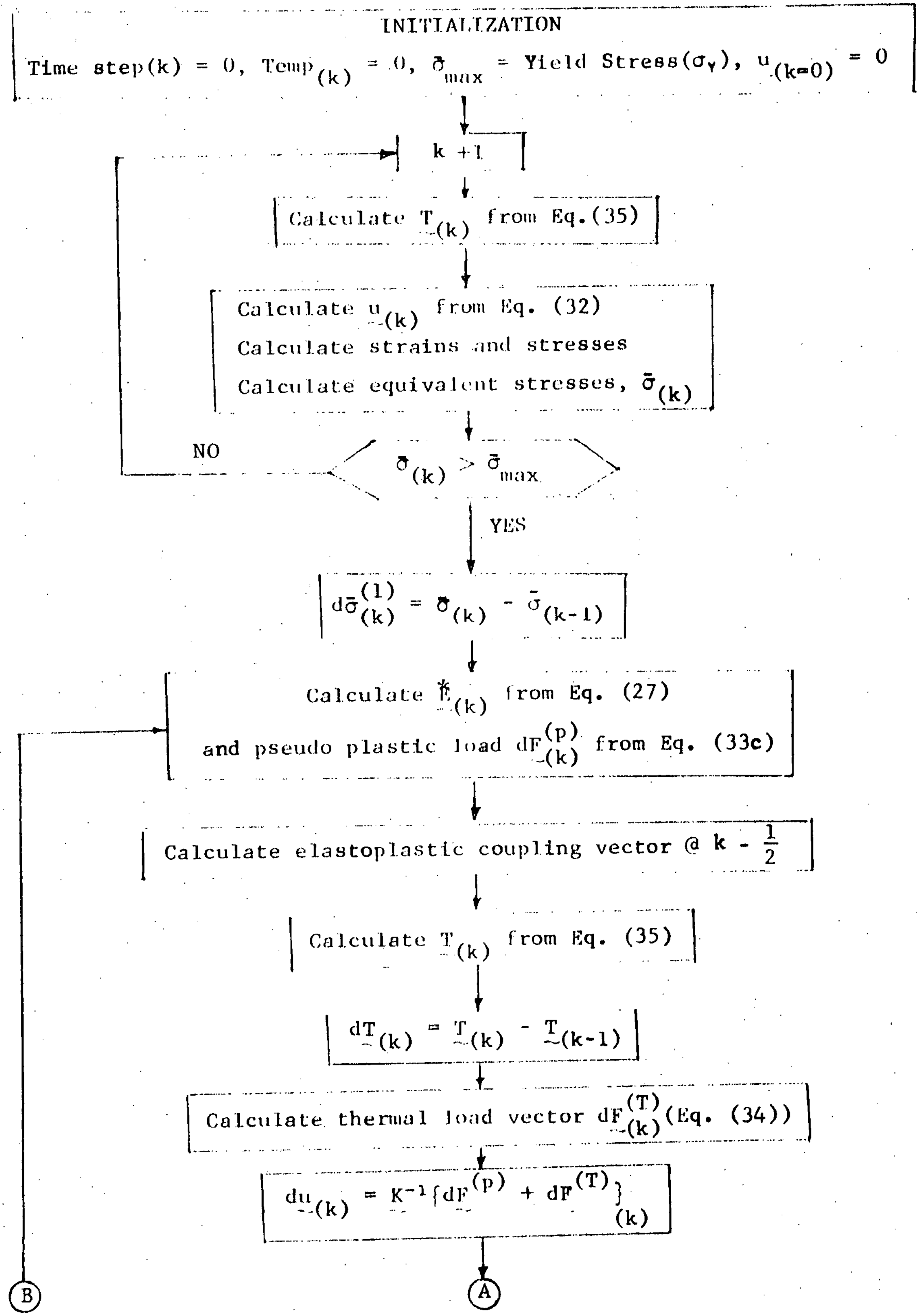
$$\psi_i = \frac{1}{8} (1 + \xi\xi_i)(1 + \eta\eta_i)(1 + \zeta\zeta_i)$$

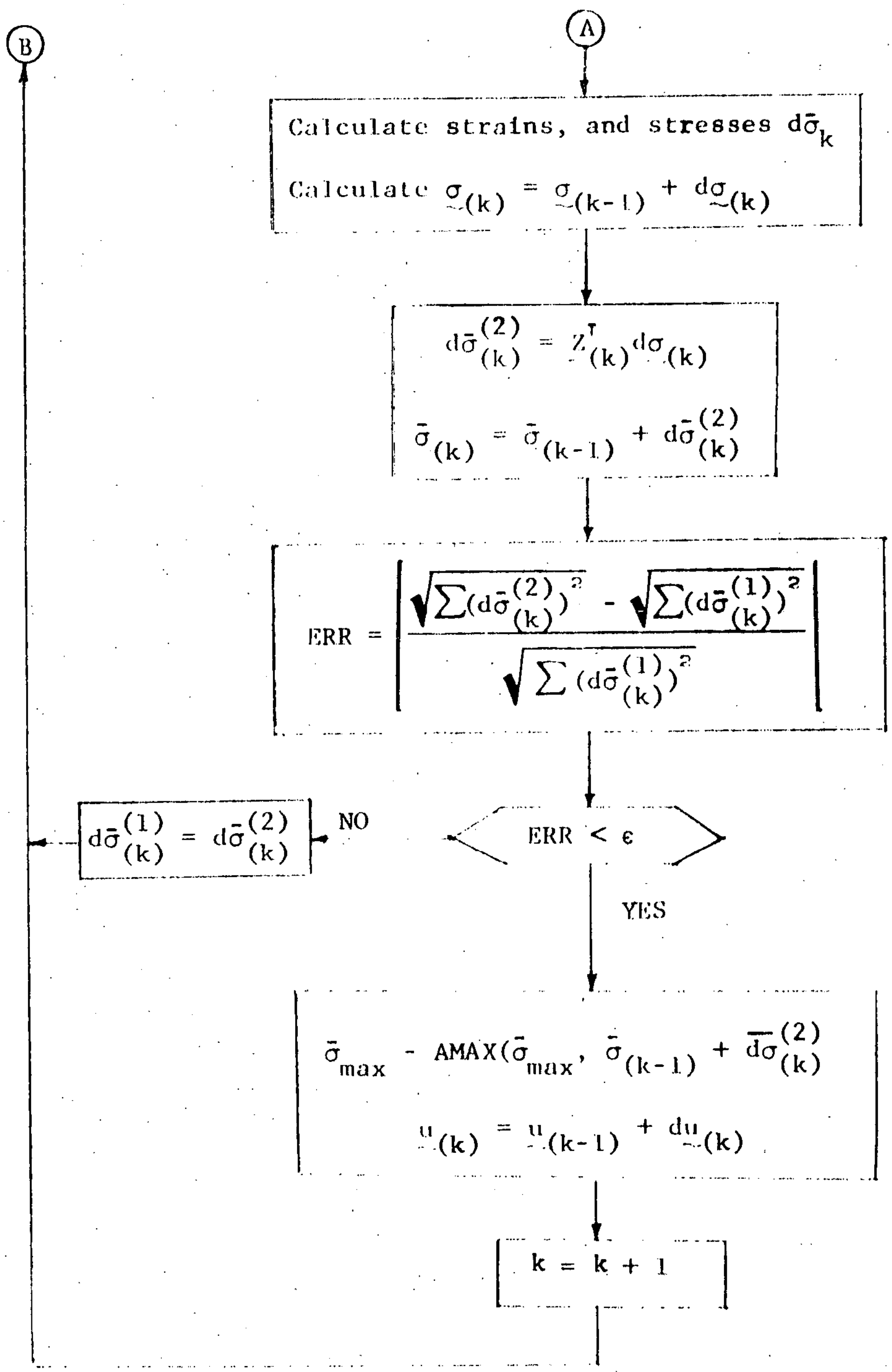
The determinant of Jacobian is given by

$$\begin{aligned} |J| = \frac{1}{8^3} \sum_{i=1}^{8} \sum_{j=1}^{8} \sum_{k=1}^{8} \Big[& \xi_i(1 + \eta\eta_i)(1 + \zeta\zeta_i)x_i \{\eta_j(1 + \xi\xi_j)(1 + \zeta\zeta_j) \\ & y_j\zeta_k(1 + \xi\xi_k)(1 + \eta\eta_k)Z_k \\ & - \eta_j(1 + \xi\xi_j)(1 + \zeta\zeta_j)Z_j\zeta_k(1 + \xi\xi_k)(1 + \eta\eta_k)y_k\} \\ & - \xi_i(1 + \eta\eta_i)(1 + \zeta\zeta_i)y_i\{\eta_j(1 + \xi\xi_j)(1 + \zeta\zeta_j)x_j\zeta_k(1 + \xi\xi_k)(1 + \eta\eta_k)Z_k \\ & - \eta_j(1 + \xi\xi_j)(1 + \zeta\zeta_j)Z_j\zeta_k(1 + \xi\xi_k)(1 + \eta\eta_k)x_k\} \\ & + \xi_i(1 + \eta\eta_i)(1 + \zeta\zeta_i)Z_i\{\eta_j(1 + \xi\xi_j)(1 + \zeta\zeta_j)x_j\zeta_k(1 + \xi\xi_k)(1 + \eta\eta_k)y_k \\ & - \eta_j(1 + \xi\xi_j)(1 + \zeta\zeta_j)y_j\zeta_k(1 + \xi\xi_k)(1 + \eta\eta_k)\, x_k\} \Big] \end{aligned}$$



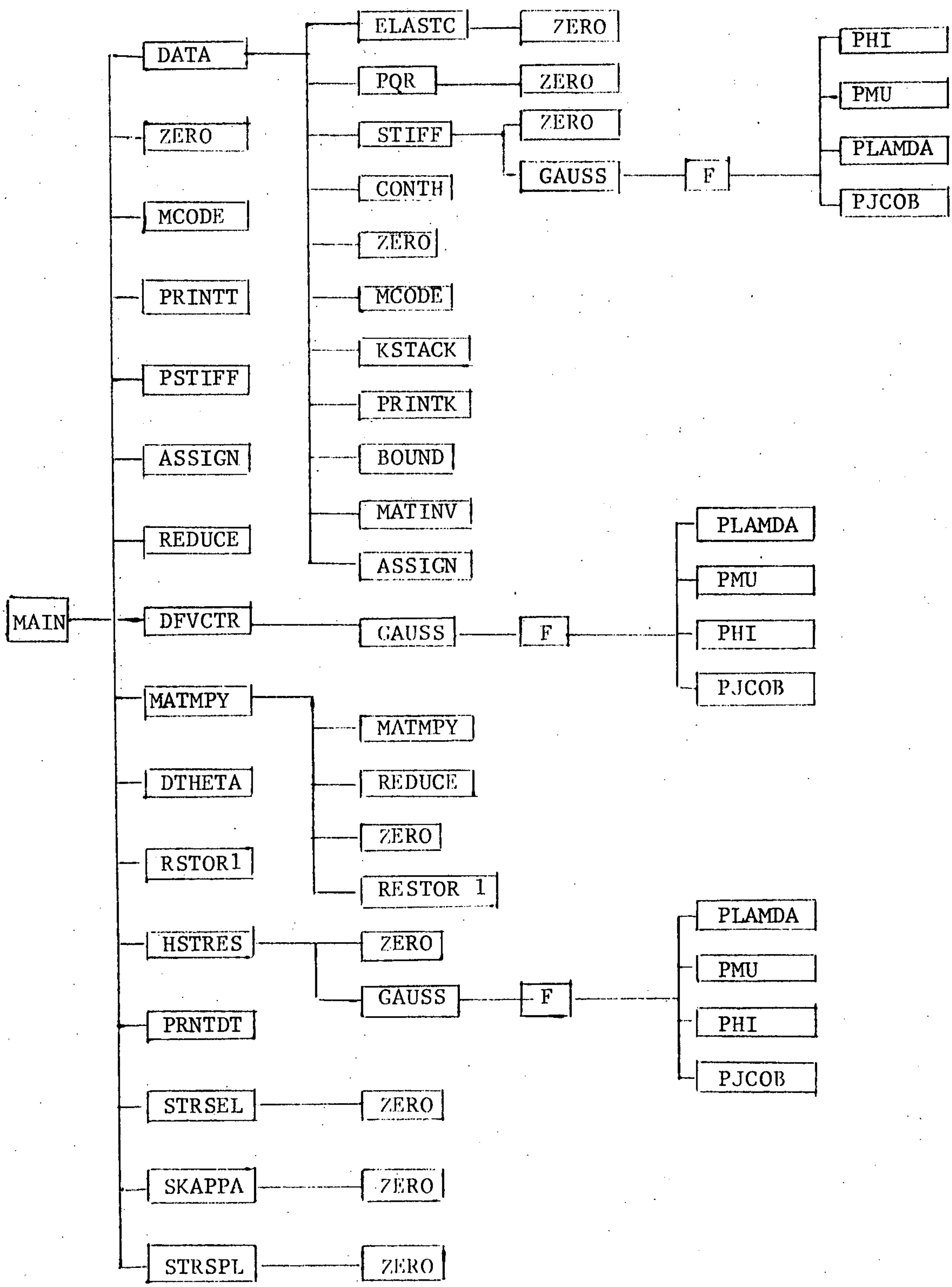
APPENDIX C: FLOW CHART




A
B
Calculate strains, and stresses $d\bar{\sigma}_k$
Calculate $\underset{\sim}{\sigma}_{(k)} = \underset{\sim}{\sigma}_{(k-1)} + d\underset{\sim}{\sigma}_{(k)}$
$d\bar{\sigma}^{(2)}_{(k)} = \underset{\sim}{Z}^{T}_{(k)} d\underset{\sim}{\sigma}_{(k)}$
$\bar{\sigma}_{(k)} = \bar{\sigma}_{(k-1)} + d\bar{\sigma}^{(2)}_{(k)}$
$ERR = \left| \dfrac{\sqrt{\sum (d\bar{\sigma}^{(2)}_{(k)})^2} - \sqrt{\sum (d\bar{\sigma}^{(1)}_{(k)})^2}}{\sqrt{\sum (d\bar{\sigma}^{(1)}_{(k)})^2}} \right|$
$ERR < \epsilon$
NO
$d\bar{\sigma}^{(1)}_{(k)} = d\bar{\sigma}^{(2)}_{(k)}$
YES
$\bar{\sigma}_{max} = AMAX(\bar{\sigma}_{max}, \bar{\sigma}_{(k-1)} + \overline{d\sigma}^{(2)}_{(k)}$
$\underset{\sim}{u}_{(k)} = \underset{\sim}{u}_{(k-1)} + d\underset{\sim}{u}_{(k)}$
$k = k + 1$

APPENDIX D

SUBROUTINE ORGANIZATION CHART



APPENDIX E

DESCRIPTIONS OF SUBROUTINES

Subroutine Name	Descriptions
ASSIGN	Rearranges the nodal displacement vector
BOUND	Applies the boundary conditions and reorders the matrices accordingly
CONTH	Defines various quantities for Guassian Quadrature integrations
DATA	Reads all input data
DFVCTR	Calculates all psuedo coupling vectors and heat input vectors
DTHETA	Solves heat conduction equations
ELASTC	Calculates the elasticity matrix
F	Function subroutine for Gaussian integration
GAUSS	Integration by Gaussian quadrature
HSTRESS	Calculates the thermal load vector in equilibrium equations
MATINV	Matrix inversion
MATMPY	Matrix multiplications

Subroutine Name	Descriptions
MCODE	Reassigns the global node number to the local element node number
KSTACK	Assembles the local element matrices into a global form
PHI	Function subroutine involved in Gaussian integrations
PJCOB	Calculates Jacobian in Gaussian integration
PLAMDA	Function subroutine involved in Gaussian integration
PMU	Function subroutine involved in Gaussian integration
PQR	Defines constants necessary for Gaussian integration
PRINTK	Print stiffness matrix, heat capacity matrix, and conductivity matrix
PRINTT	Print nodal temperatures
PRNTDT	Print nodal displacement and equivalent nodal forces
PSTIFF	Calculates the plastic tangent stiffness matrix
REDUCE	Modifies the equivalent nodal vectors in correspondence to BOUND
RESTOR 1	Restores the equivalent nodal vectors to the form prior to REDUCE

Subroutine Name	Descriptions
SKAPPA	Calculates the plasticity matrix
STIFF	Calculates the elastic stiffness matrix
STRSEL	Calculates stresses prior to yielding
STRSPL	Calculates incremental stresses after yielding
ZERO	Zeroes out all matrices

APPENDIX F

DATA INPUT FORMAT

Card 1: FORMAT (7 I5)

(1)	NELEMT	-	Number of elements
(2)	INODE	-	Number of nodes
(3)	NB	-	Number of constrained displacements
(4)	IPT	-	Number of integration points in the Gaussian quadrature
(5)	NBHC	-	Number of prescribed nodal temperatures
(6)	ITER	-	Number of time increments
(7)	NKE	-	Number of type of elements

Card 2: FORMAT (3 F 10.0)

(1)	RT	-	Reference temperature
(2)	EPSS	-	Percent error limit, ϵ in the elastoplastic analysis
(3)	DELT	-	Incremental time interval, Δt

Card 3: FORMAT (8 F 10.0)

(1)	E(I)	-	Young's modulus
(2)	EP(I)	-	Plastic modulus
(3)	SYIELD(I)	-	Yield stress
(4)	XNU(I)	-	Poisson's ratio
(5)	TKX(I)	-	Thermal conductivity
(6)	SH(I)	-	Specific heat
(7)	ALPHA(I)	-	Coefficient of thermal expansion
(8)	DENSTY(I)	-	Density

Cards 4: FORMAT (16I5)

(1) NY(I) - Type of elements I

Repeat cards 4 as required to complete all elements.

Cards 5: FORMAT (8I5)

(1) to (8)

MA(I), MB(I),MS(I)- Node numbers of element I

Repeat cards 5 NELEMT times.

Cards 6: FORMAT (6F10.0)

(1) to (3)

X(I), Y(I), Z(I) - x, y and z coordinates of node number I.

Repeat cards 6 as required to complete all nodes.

Card 7: FORMAT (2I5)

(1) LSTRES = 0 if stress anslysis is not desired (no elastic or elastoplastic coupling)

= 1 if stress analysis is to be included (elastic or elastoplastic coupling present)

(2) LHEAT = 0 if heat conduction analysis is not desired

= 1 otherwise

Card 8: FORMAT (2I5)

(1) KLOAD = 0 if mechanical loads are not applied

= 1 if mechanical loads are applied

(2) MLOAD = 0 if temperatures are not prescribed

= 1 if temperatures are prescribed

(An assumption is made LSTRES = LHEAT = MLOAD = 1 and KLOAD = 0 after card 9)

Cards 9: FORMAT (I5, F10.0)

(1) IHBND(I) - Node number at which temperature is prescribed

(2) TFR(I) - Prescribed temperature

Repeat cards 9 NBHC times.

Cards 10: FORMAT (4I5, 3F10.0)

(1)* NS(L,I) - Node numbers of the uninsulated surface (I changes from 1 to 4)

(2)* SC(L,I) SC(L,1) = $\bar{\alpha}$ (film coefficient)

SC(L,2) = q (heat flux)

SC(L,3) = T' (ambient temperature)

*L indicates the number of uninsulated surfaces subjected to convection.

Provide a blank card to signify end of data.

Cards 11: FORMAT (4I5)

(1) NODE - Node number at which displacements are constrained.

(2) IU = 0 if U is not constrained

= 1 if U is constrained

(3) IV = 0 if V is not constrained

= 1 if V is constrained

(4) IW = 0 if W is not constrained

= 1 if W is constrained

Provide a blank card to signify end of data.

Cards 12: FORMAT (8F10.0)

(1) DHV(I) - Heat supply of element I.

Repeat cards 12 as required to complete all elements.

APPENDIX G

COMPUTER PROGRAM LISTING

```
      COMMON /BLKA/NELEMT,INODE,NR,NBHC,LHEAT,LSTRES,NN,NM,NNH,NMH
      COMMON /BLKB/NOPRNT,NPAGE,LINES,MM,KODE,KTOTAL,MTOTAL,MLOAD,KLOAD
      COMMON /BLKC/ESK(24,24),HC(8,8),HSK(8,8),TYPEH(8,8),TYPEI(8,8),
     *TYPEJ(8,8),TYPEM(8,8),TYPEN(8,8),TYPES(8,8)
      COMMON/BLD/SK(20000),HK(2000),CK(2000),HCK(2000),IBND(100),
     *   IHBND(50)
      COMMON/BLK/MA(30),MB(30),MC(30),MD(30),MP(30),MQ(30),MR(30),MS(30)
      COMMON /BLK1/A(10),B(10),C(10),P(33),Q(33),R(33),W(6),H(6),IPT
      COMMON /BLK2/H1(8),H2(8),H3(8),H4(8),H5(8),H6(8),H7(8),H8(8),
     *H9(8),H10(8),G1(8),G2(8),G3(8),G4(8),G5(8),G6(8),G7(8),G8(8)
      COMMON/BLK4/T0(60),T1(60),T2(60),DT0(60),DT1(60),DT2(60)
      COMMON/BLK3/NS(50,4),SC(50,3),AREA4(50)
      COMMON/BLK5/IGAUS,ITER,EPSS,IGE
      COMMON/BLK6/D(6,6),E(10),EP(10),SYIELD(10),XNU(10),TKX(10),
     *              SH(10),ALPHA(10),DENSTY(10),NY(30),DELT,RT
      COMMON/BLK7/X(60),Y(60),Z(60),MNODE(8),MKODE(24)
      COMMON/BLK8/U0(180),U1(180),U2(180),DU0(180),DU1(180),DU2(180)
      COMMON /BLK9/DSIGMA(6),SIGMA(6,30),SIGMA1(6,30)
      COMMON/BLK10/SIGMAB(30),SIGM1B(30),DSIG1B(30),DSIG2B(30),
     *              SIGMAX(30)
      COMMON/BLK11/STRN(6,30),DSTRN(6,30),STRNP(6,30),ABC
      DIMENSION C1(180),BSK(180),DF(60),DFB(60),DHV(30),EDF(8)
      DIMENSION DP(24),FR(180),PK(24,24),AJ(6,6),FP(180),TP(60)
      IYIELD=0
      TIME=0.0
      III=0
      NNN=0
      CALL DATA(FP,BSK,C1)
      DO 15 I=1,NELEMT
      NYI=NY(I)
      SIGMAX(I)=SYIELD(NYI)
   15 CONTINUE
      NE=NELEMT
      DO 12 I=1,NNH
      TP(I)=T0(I)
      T2(I)=T0(I)
   12 T1(I)=T0(I)
C*****TIME INCREMENT
 1000 III=III+1
 2000 NNN=NNN+1
      TIME=TIME+DELT
      IF(III.GT.ITER) STOP
      WRITE(6,6) III,TIME
      READ(5,2) (DHV(I),I=1,NELEMT)
   13 CONTINUE
      WRITE(6,10) (DHV(I),I=1,NELEMT)
      DUN=0.
      CALL ZERO(DF,INODE,1)
      L=1
      NX=0
   20 CONTINUE
      CALL MCODE(MKODE,MNODE,L,3)
      DH=DHV(L)
      KAP=1.
      CALL DFVCTR(EDF,KAP,DH,NX,NNN,1,L)
      DO 30 I=1,8
      MNO=MNODE(I)
   30 DF(MNO)=EDF(I)+DF(MNO)
      IF(L.GE.NELEMT) GO TO 40
      L=L+1
      NX=1
      IF(NY(L).NE.NY(L-1)) NX=0
      GO TO 20
```

```
   40 CONTINUE
C*****SURFACE HEAT BOUNDARY
      L=1
   41 IF(NS(L,1).EQ.0) GO TO 42
      AI=AREA4(L)*(SC(L,2)-SC(L,1)*SC(L,3))
      DO 43 I=1,4
      NI=NS(L,I)
   43 DF(NI)=DF(NI)-AI
      L=L+1
      GO TO 41
   42 CONTINUE
      WRITE(6,5) (I,DF(I),I=1,INODE)
C*****DTHETA VECTOR
      CALL DTHETA(DFB,DF,MLOAD,T0,T1,TP)
      CALL PRINTT(DF,DFB,X,Y,Z,1,INODE,NPAGE,60)
C*****DT VECTOR
      NX=0
      CALL ZERO(FR,180,1)
      IF(NY(I).NE.NY(I-1)) NX=0
      NYI=NY(I)
      ALPHAI=ALPHA(NYI)*E(NYI)/(1.-2.*XNU(NYI))
      CALL MCODE(MKODE,MNODE,I,3)
      CALL MSTRES ( T0,FR,MNODE,MKODE,NX,ALPHAI,60,180)
      NX=1
  260 CONTINUE
      DO 265 I=1,NN
  265 FR(I)=FR(I)+FP(I)*FLOAT(III)
      IF(IGE.EQ.0) GO TO 269
C*****GEOMETRICALLY NONLINEAR VECTOR
      CALL ZERO(AJ,6,6)
      DO 266 I=1,NELEMT
      CALL MCODE(MKODE,MNODE,I,3)
      CALL ZERO(DP,24,1)
      CALL PSTIFF(PK,AJ,I,IGE)
      CALL ASSIGN(PK,MNODE,0,8,1,24)
      DO 267 J=1,24
      DO 267 K=1,24
      NK=MKODE(K)
      DU0I=U1(NK)
  267 DP(J)=DP(J)+PK(J,K)*DU0I
      DO 268 J=1,24
      NK=MKODE(J)
  268 FR(NK)=DP(J)+FR(NK)
  266 CONTINUE
  269 CONTINUE
C*****DU VECTOR
      CALL REDUCE(FR,IBND,NN,NB,180)
      CALL MATMPY(SK,NM,FR, U0,180)
      CALL RSTORI(FR,IBND,NN,NB,180)
      CALL RSTORI( U0,IBND,NN,NB,180)
      CALL PRNTOT(FR, U0,X,Y,Z,MM,INODE,NPAGE,180)
C*****STRAIN & STRESS
      WRITE(6,88)
      DO 50 I=1,NELEMT
      CALL MCODE(MKODE,MNODE,I,3)
      CALL STRSEL( U0, T0,IYIELD,I)
      WRITE(6,87) I,IYIELD,SIGMAB(I),(SIGMA(J,I),J=1,6) ,SIGMAX(I)
   50 CONTINUE
      IF(IYIELD.EQ.1) GO TO 3000
   99 CONTINUE
C*****PRINT OUT
      WRITE(6,7)
      DO 120 I=1,INODE
      J=3*I-2
      J1=J+1
      J2=J+2
  120 WRITE(6,66)I,U0(J),U0(J1),U0(J2),T0(I)
C*****SHIFT
      DO 125 I=1,INODE
      T2(I)=T1(I)
      T1(I)=T0(I)
      DO 125 J=1,3
      IJ=(I-1)*3+J
      U2(IJ)=U1(IJ)
      U1(IJ)=U0(IJ)
  125 CONTINUE
      DO 126 I=1,NELEMT
      SIGMIB(I)=SIGMAB(I)
      DO 126 J=1,6
      SIGMAI(J,I)=SIGMA(J,I)
  126 CONTINUE
      GO TO 1000

C*****FOR YIELDED
 3000 CONTINUE
      NNN=0
      DSIG2=0.
      DO 130 I=1,NELEMT
      DSIG2B(I)=-1.0
      IF(SIGMAB(I).GE.SIGMAX(I)) DSIG2B(I)=1.0
      DSIG1B(I)=SIGMAB(I)-SIGMIB(I)
  130 DSIG2=DSIG2+DSIG1B(I)**2
 3500 CONTINUE
      NNN=NNN+1
      IF(NNN.GT.10) STOP
      WRITE(6,9)III,NNN
      CALL SKAPPA
      REWIND 2
      KAP=1.0
 4000 CONTINUE
      CALL ZERO(DF,60,1)
      REWIND2
      L=1
      NX=0
   25 CONTINUE
      CALL MCODE(MKODE,MNODE,L,3)
      DH=DHV(L)
      CALL DFVCTR(EDF,KAP,DH,NX,NNN,0,L)
      DO 35 I=1,8
      MNO=MNODE(I)
   35 DF(MNO)=EDF(I)+DF(MNO)
      IF(L.GE.NELEMT) GO TO 45
      L=L+1
      IF(NY(L).NE.NY(L-1)) NX=0
```

```
      GO TO 25
   45 CONTINUE
      WRITE(6,5) (I,DF(I),I=1,INODE)
C*****SURFACE HEAT BOUNDARY
      L=1
   46 IF(NS(L,1).EQ.0) GO TO 47
      AI=AREA4(L)*(SC(L,2)-SC(L,1)*SC(L,3))
      DO 48 I=1,4
      NI=NS(L,I)
   48 DF(NI)=DF(NI)-AI
      L=L+1
      GO TO 46
   47 CONTINUE
C*****DTHETA VECTOR
      CALL DTHETA(DFB,DF,MLOAD,TO,T1,TP)
      CALL PRINTT(DF ,DFB,X,Y,Z,1,INODE,NPAGE,60)
C*****DT VECTOR
      DO 250 I=1,INODE
      DTO(I)=TO(I)-T1(I)
  250 CONTINUE
      CALL ZERO(FR,180,1)
      DO 261 I=1,NELEMT
      IF(NY(I).NE.NY(I-1)) NX=0
      NYI=NY(I)
      ALPHAI=ALPHA(NYI)*E(NYI)/(1.-2.*XNU(NYI))
      CALL MCODE(MKODE,MNODE,I,3)
      CALL HSTRES (DTO,FR,MNODE,MKODE,NX,ALPHAI,60,180)
  261 CONTINUE
      DO 270 I=1,NN
  270 FR(I)=FR(I)+FP(I)
C*****DP VECTOR
      REWIND 2
      REWIND 9
      DO 146 I=1,NELEMT
      READ(2) AJ
      CALL MCODE(MKODE,MNODE,I,3)
      CALL ZERO(DP,24,1)
      CALL PSTIFF(PK,AJ,I,IGE)
      CALL ASSIGN(PK,MNODE,0,8,I,24)
      DO 145 J=1,24
      DO 145 K=1,24
      NK=MKODE(K)
      DUO1=UO(NK)-U1(NK)
      IF(NNN.EQ.1)  DUO1=U1(NK)-U2(NK)
  145 DP(J)=DP(J)+PK(J,K)*DUO1
      DO 147 J=1,24
      NK=MKODE(J)
  147 FR(NK)=DP(J)+FR(NK)
  146 CONTINUE
C*****DU VECTOR
      CALL REDUCE(FR,IBND,NN,NB,180)
      CALL MATMPY(SK,NM,FR,DUO,180)
      CALL RSTOR1(FR,IBND,NN,NB,180)
      CALL RSTOR1(DUO,IBND,NN,NB,180)
      CALL PRNTDT(FR,DUO,X,Y,Z,MM,INODE,NPAGE,180)
C*****STRAIN & STRESS
      REWIND 2
      DO 140 I=1,NELEMT
      CALL MCODE(MKODE,MNODE,I,3)
      READ(2) AJ
      CALL STRSPL(DUO,DTO,AJ,I)
  140 CONTINUE
      DO 170 I=1,INODE
      DO 170 J=1,3
      IJ=3*(I-1)+J
  170 UO(IJ)=U1(IJ)+DUO(IJ)
C*****PRINT OUT
      WRITE(6,4) III,NNN
      WRITE(6,7)
      DO 180 I=1,INODE
      J=3*I-2
      J1=J+1
      J2=J+2
  180 WRITE(6,66)I,UO(J),UO(J1),UO(J2),TO(I)
      DSIG1=DSIG2
      DSIG2=0.
      WRITE(6,90)
      DO 150 I=1,NELEMT
      SIGMAB(I)=SIGM1B(I)+DSIG2B(I)
      WRITE(6,89) I,DSIG2B(I),(SIGMA(J,I),J=1,6) ,SIGMAX(I),SIGMAB(I)
  150 DSIG2=DSIG2+DSIG2B(I)**2
      ERROR=ABS(SQRT(DSIG2)-SQRT(DSIG1))/SQRT(DSIG1)
      WRITE(6,9) III,NNN,ERROR,DSIG2,DSIG1
      IF(ERROR.GT.EPSS) GO TO 3500
      GO TO 5000
 5000 CONTINUE
C*****CONVERGED
      DO 160 I=1,NELEMT
      SIGMAB(I)=SIGM1B(I)+DSIG2B(I)
  160 IF(SIGMAB(I).GT.SIGMAX(I)) SIGMAX(I)=SIGMAB(I)
C*****NEXT TIME INCREMENT
      III=III+1
      IF(III.GT.ITER) STOP
      NNN=0
      READ(5,2) (DHV(I),I=1,NELEMT)
   14 CONTINUE
      WRITE(6,10) (DHV(I),I=1,NELEMT)
C*****SHIFT
      DO 190 I=1,INODE
      T2(I)=T1(I)
      T1(I)=TO(I)
      DT2(I)=DT1(I)
      DT1(I)=DTO(I)
      DO 190 J=1,3
      IJ=(I-1)*3+J
      U2(IJ)=U1(IJ)
      U1(IJ)=UO(IJ)
      DU2(IJ)=DU1(IJ)
      DU1(IJ)=DUO(IJ)
  190 CONTINUE
      DO 200 I=1,NELEMT
      SIGM1B(I)=SIGMAB(I)
      DSIG1B(I)=DSIG2B(I)
      DO 200 J=1,6
```

```
            SIGMAI(J,I)=SIGMA(J,I)
        200 CONTINUF
            TIME=TIME+DELT
            WRITE(6,6) III,TIME
            GO TO 3500
          1 FORMAT(8I5)
          2 FORMAT(8F10.0)
          3 FORMAT(6F10.0)
          4 FORMAT(/9X,' TIME STEP =',I5,' ITERATION NO.',I5/)
          5 FORMAT(I5,E13.5)
          6 FORMAT(1H1,'***TIME INCREMENT ',I5,5X,'TOTAL TIME =',1PE13.5/)
         66 FORMAT(4X,I5,4E13.5)
          7 FORMAT(1X,'NODE NO.',5X,'U',14X,'V',14X,'W',8X,'TEMPERATURE'/)
          8 FORMAT(16I5)
          9 FORMAT(1H ,10X,' III = ',I5,10X,' NNN = ',I5,10X,' ERROR =',E13.5,
           *'DSIG2=',E13.5,'DSIG1=',E13.5)
         10 FORMAT(1H0,'DHV=',8E13.5)
         11 FORMAT(12E10.3)
         87 FORMAT(1H ,2I6,8E13.5)
         88 FORMAT(1H0,'ELEMENT IYIELD SIGMABAR      SIGMAX        SIGMAY
           *SIGMAZ         SIGMAXY        SIGMAYZ        SIGMAZX        SIGMAMAX')
         89 FORMAT(1H ,I3,9E12.4)
         90 FORMAT(1H0,'ELEMENT DSIGMABAR      SIGMAX          SIGMAY          SIGMAZ
           *       SIGMAXY        SIGMAYZ        SIGMAZX        SIGMA-MAX    SIGMB-1')
        500 FORMAT(3(1PE13.5))
        803 FORMAT(32I3)
        118 CONTINUE
        119 CONTINUE
            END

END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
            SUBROUTINE DATA(FR,BSK,CI)
            COMMON /BLKA/NELEMT,INODE,NB,NBHC,LHEAT,LSTRES,NN,NM,NNH,NMH
            COMMON /BLKB/NOPRNT,NPAGE,LINES,MM,KODE,KTOTAL,MTOTAL,MLOAD,KLOAD
            COMMON /BLKC/ESK(24,24),HC(8,8),HSK(8,8),TYPEH(8,8),TYPEI(8,8),
           *TYPEJ(8,8),TYPEM(8,8),TYPEN(8,8),TYPES(8,8)
            COMMON/BLD/SK(20000),HK(2000),CK(2000),HCK(2000),IBND(100),
           *   IHBND(50)
            COMMON/BLK/MA(30),MB(30),MC(30),MD(30),MP(30),MQ(30),MR(30),MS(30)
            COMMON /BLK1/A(10),B(10),C(10),P(33),Q(33),R(33),W(6),H(6),IPT
            COMMON /BLK2/H1(8),H2(8),H3(8),H4(8),H5(8),H6(8),H7(8),H8(8),
           *H9(8),H10(8),G1(8),G2(8),G3(8),G4(8),G5(8),G6(8),G7(8),G8(8)
            COMMON/BLK4/T0(60),T1(60),T2(60),DT0(60),DT1(60),DT2(60)
            COMMON/BLK3/NS(50,4),SC(50,3),AREA4(50)
            COMMON/BLK5/IGAUS,ITER,EPSS,IGE
            COMMON/BLK6/D(6,6),E(10),EP(10),SYIELD(10),XNU(10),TKX(10),
           *            SH(10),ALPHA(10),DENSTY(10),NY(30),DELT,RT
            COMMON/BLK7/X(60),Y(60),Z(60),MNODE(8),MKODE(24)
            DIMENSION FR(1),CI(1),BSK(1),TFR(50),JM(50),IM(50)


            READ(5,1) IGAUS
            IF(IGAUS.EQ.1) GO TO 1000
            READ(5,802)(P(I),Q(I),R(I),I=1,33)
            READ(5,802)(A(I),B(I),C(I),I=1,10)
            WRITE(6,900)(P(I),I=1,33)
            WRITE(6,900)(Q(I),I=1,33)
            WRITE(6,900)(R(I),I=1,33)
            WRITE(6,900)(A(I),I=1,10)
            WRITE(6,900)(B(I),I=1,10)
            WRITE(6,900)(C(I),I=1,10)
            READ (5,599)((TYPEH(I,J),I=1,8),J=1,8)
            READ (5,599)((TYPEI(I,J),I=1,8),J=1,8)
            READ (5,599)((TYPEJ(I,J),I=1,8),J=1,8)
            READ (5,599)((TYPEM(I,J),I=1,8),J=1,8)
            READ (5,599)((TYPEN(I,J),I=1,8),J=1,8)
            READ (5,599)((TYPES(I,J),I=1,8),J=1,8)
            READ(5,802)((ESK(I,J),I=1,24),J=1,24)
            WRITE(6,900)((ESK(I,J),I=1,24),J=1,24)
            READ (5,802)((HSK(I,J),I=1,8),J=1,8)
            WRITE(6,900)((HSK(I,J),I=1,8),J=1,8)
            READ (5,802)((HC(I,J),I=1,8),J=1,8)
            WRITE(6,900)((HC(I,J),I=1,8),J=1,8)
        599 FORMAT(6(1PE13.5))
        802 FORMAT(6(1PE13.5))
        900 FORMAT(8(1PE13.5))
       1000 CONTINUE
          1 FORMAT(16I5)
            READ(5,1) NELEMT,INODE,NB,NBHC,IPT,ITER,NKE
            READ(5,9) RT,EPSS,DELT
            READ(5,9) (E(I),EP(I),SYIELD(I),XNU(I),TKX(I),SH(I),ALPHA(I),
           *          DENSTY(I),I=1,NKE)
            READ(5,1)(NY(I),I=1,NELEMT)
            WRITE(6,1)NELEMT,INODE,NB,NBHC,IPT,ITER,NKE
            WRITE(6,500) RT,EPSS,DELT
            WRITE(6,500) (E(I),EP(I),SYIELD(I),XNU(I),TKX(I),SH(I),ALPHA(I),
           *             DENSTY(I),I=1,NKE)
            WRITE(6,1)   (NY(I),I=1,NELEMT)
```

```
  500 FORMAT(8(1PE13.5))
    9 FORMAT(8F10.0)
      NE=NELEMT
      READ (5,2)(MA(I),MB(I),MC(I),MD(I),MP(I),MQ(I),MR(I),MS(I),I=1,NE)
      WRITE(6,2)(MA(I),MB(I),MC(I),MD(I),MP(I),MQ(I),MR(I),MS(I),I=1,NE)
    2 FORMAT(8I5)
      READ (5,3)(X(I),Y(I),Z(I),I=1,INODE)
    3 FORMAT(6F10.0)
      WRITE(6,500)(X(I),Y(I),Z(I),I=1,INODE)
      CALL CONTH
      NOPRNT=1
      NOPRNT=0
      NPAGE=0
      KODE=8
      MM=3
      NPAGE=1
      LINES=1
      NNH=INODE
      NHM=NNH-NBHC
      NN=MM*INODE
      NM=NN-NB
      READ(5,1)LSTRES,LHEAT
      WRITE(6,1)LSTRES,LHEAT
      READ(5,1)KLOAD,MLOAD
      WRITE(6,1)KLOAD,MLOAD
      IF(LHEAT.NE.1) GO TO 250
      MTOTAL=(NNH*(NNH+1))/2
      CALL ZERO(HK,MTOTAL,1)
      CALL ZERO(CK,MTOTAL,1)
      IF(MLOAD.EQ.0) GO TO 550
   16 FORMAT(3(F10.0,I10))
  502 FORMAT(3(1PE13.5,I10))
      READ(5,720) (IHBND(I),TFR(I),I=1,NBHC)
  720 FORMAT(I5,F10.0)
      WRITE(6,502)(TFR(I),IHBND(I),I=1,NBHC)
      DO 901 I=1,NBHC
      KX=IHBND(I)
  901  TO(KX)=TFR(I)
      WRITE(6,601)
  601 FORMAT(2X,' NODAL INPUT TEMPERATURE VECTOR '/)
      WRITE(6,600)(TO(I),I=1,NNH)
  600 FORMAT(1PE13.5)
  550 CONTINUE
      REWIND 3
      REWIND 4
   40 L=1
      NX=0
   41 CONTINUE
      M=NY(L)
      CALL MCODE(MKODE,MNODE,L,MM)
      IF(IGAUS.EQ.0) GO TO 1010
      CALL PQR(X,Y,Z,MNODE,NX)
      CALL ELASTC (D,E(M),XNU(M))
      CALL STIFF(D,NX,NOPRNT)
      DO 345 I=1,8
      DO 345 J=1,8
      HC(I,J)=HC(I,J)*SH(M)*2./DELT
  345 HSK(I,J)=HSK(I,J)*TKX(M)
 1010 CONTINUE
      WRITE(3)ESK,MKODE,MNODE
      WRITE(4)TYPEH,TYPEI,TYPEJ,TYPEM,TYPEN,TYPES
      CALL KSTACK(CK,HC ,8,1,MNODE,8,NNH)
      CALL KSTACK(HK,HSK,8,1,MNODE,8,NNH)
      IF(L.GE.NELEMT) GO TO 42
      L=L+1
      NX=1
      IF(NY(L).NE.NY(L-1)) NX=0
      GO TO 41
   42 IF(NOPRNT .EQ. 0) GO TO 43
      CALL PRINTK(CK,BSK,NNH)
      CALL PRINTK(HK,BSK,NNH)
   43 CONTINUE
C*****SURFACE HEAT BOUNDARY
      L=1
C*****NS=COUNT.CLOCK,SC(L1)=ALPHAB,SC(L,2)=Q,SC(L,3)=T'
   50 READ(5,51) (NS(L,I),I=1,4),(SC(L,I),I=1,3)
      WRITE(6,52)(NS(L,I),I=1,4),(SC(L,I),I=1,3)
   51 FORMAT(4I5,6F10.0)
   52 FORMAT(1H ,4I5,3E15.7)
      IF(NS(L,1).EQ.0) GO TO 60
      I1=NS(L,1)
      I2=NS(L,2)
      I3=NS(L,3)
      I4=NS(L,4)
      AREA4(L)=0.25*ABS((Y(I1)-Y(I2))*(Z(I1)-Z(I4))
     *                  +(Y(I1)-Y(I4))*(Z(I1)-Z(I2))
     *     +(Z(I1)-Z(I2))*(X(I1)-X(I4))+(Z(I1)-Z(I4))*(X(I1)-X(I2))
     *     +(X(I1)-X(I2))*(Y(I1)-Y(I4))+(X(I1)-X(I4))*(Y(I1)-Y(I2)) )
      A1=AREA4(L)*SC(L,1)*4./9.
      A2=AREA4(L)*SC(L,1)*1./9.
      A3=AREA4(L)*SC(L,1)*2./9.
      DO 55 I=1,4
      DO 55 J=1,4
      NI=NS(L,I)
      NJ=NS(L,J)
      IJ=NNH*NJ-NJ*(NJ-1)/2-NNH+NI
      IF(NI.LT.NJ) GO TO 55
      IF (I.EQ.J) HK(IJ)=HK(IJ)+A1
      IF(IABS(I-J).EQ.2) HK(IJ)=HK(IJ)+A2
      IF(I.NE.J.AND.IABS(I-J).NE.2) HK(IJ)=HK(IJ)+A3
   55 CONTINUE
      L=L+1
      GO TO 50
   60 CONTINUE
      DO 346 I=1,MTOTAL
  346 HCK(I)=CK(I)+HK(I)
      DO 544 I=1,MTOTAL
  544 HK(I)=HCK(I)
      IF(MLOAD.EQ.0)NMH=NNH
      IF(MLOAD.EQ.0) GO TO 545
      CALL BOUND(HCK,IHBND,NNH,NBHC)
  545 CONTINUE
      CALL PRINTK(HCK,BSK,NMH)
      CALL MATINV(HCK,CI,NMH)
```

```
      CALL PRINTK(HCK,BSK,NMH)
      DO 305 I=1,MTOTAL
  305 HCK(I)=2.*HCK(I)
  250 CONTINUE
      IF(LSTRES.EQ.0) RETURN
      KTOTAL=(NN*(NN+1))/2
      I=1
  710 READ(5,705) NODE,IU,IV,IW
      IF(NODE.EQ.0) GO TO 700
      IF(IU.EQ.1) IBND(I)=3*NODE-2
      IF(IU.EQ.1) I=I+1
      IF(IV.EQ.1) IBND(I)=3*NODE-1
      IF(IV.EQ.1) I=I+1
      IF(IW.EQ.1) IBND(I)=3*NODE
      IF(IW.EQ.1) I=I+1
      GO TO 710
  700 CONTINUE
  705 FORMAT(4I5)
      WRITE(6,7)(IBND(I),I=1,NB)
    7 FORMAT(14I5)
      CALL ZERO(SK,KTOTAL,1)
      IF(KLOAD.EQ.0) GO TO 560
      READ(5,11)NH
      WRITE(6,11)NH
   11 FORMAT(10I5)
C IM(I) IS THE DIRECTION OF THE LOAD AT THAT JOINT, JM(I) IS JOINT NUMBER
  503 FORMAT(3(1PE13.5,2I5))
   17 FORMAT(3(F10.0,2I5))
      READ(5,17)(TFR(I),IM(I),JM(I),I=1,NH)
      WRITE(6,503)(TFR(I),IM(I),JM(I),I=1,NH)
      DO 902 I=1,NH
      KY=MM*JM(I)-MM+IM(I)
  902 FR(KY)=FR(KY)+TFR(I)
  560 CONTINUE
      REWIND 3
   19 L=1
      NX=0
   20 CONTINUE
      M=NY(L)
      IF (LHEAT.NE.1) GO TO 261
      READ(3) ESK,MKODE,MNODE
      GO TO 262
  261 CALL MCODE(MKODE,MNODE,L,MM)
      IF(IGAUS.EQ.0) GO TO 262
      CALL PQR(X,Y,Z,MNODE,NX)
      CALL ELASTC (D,E(M),XNU(M))
      CALL STIFF(D,NX,NOPRNT)
  262 CALL ASSIGN(ESK,MNODE,NX,KODE,L,24)
   10 CALL KSTACK(SK,ESK,8,3,MKODE,24,NN)
      IF(L.GE.NELEMT) GO TO 30
      L=L+1
      NX=1
      IF(NY(L).NE.NY(L-1)) NX=0
      GO TO 20
   30 IF(NOPRNT.EQ.0) GO TO 39
      CALL PRINTK(SK,BSK,NN)
   39 CONTINUE
      CALL BOUND(SK,IBND,NN,NB)
      CALL MATINV(SK,C1,NM)
      CONTINUE
      RETURN
      END

END OF COMPILATION:        NO  DIAGNOSTICS.

      SUBROUTINE DFVCTR(EDF,KAPPA,DH,NX,NNN,IE,NL)
      COMMON/BLK4/TO(60),T1(60),T2(60),DTO(60),DT1(60),DT2(60)
      COMMON/BLK5/IGAUS,ITER,EPSS,IGE
      COMMON/BLK6/D(6,6),E(10),EP(10),SYIELD(10),XNU(10),TKX(10),
     *              SH(10),ALPHA(10),DENSTY(10),NY(30),DELT,RT
      COMMON/BLK7/X(60),Y(60),Z(60),MNODE(8),MKODE(24)
      COMMON/BLK8/UO(180),U1(180),U2(180),DUO(180),DU1(180),DU2(180)
      COMMON/BLK11/STRN(6,30),DSTRN(6,30),STRNP(6,30),ABC
      DIMENSION S1(8),S2(8,8,8),S3(8,8,8,3),S4(8,8,3),EDF(8)
      NN=NY(NL)
      BIJ=-E(NN)*ALPHA(NN)/(1.-2.*XNU(NN))
      IF(NX.NE.0) GO TO 500
      IF(IGAUS.EQ.0) GO TO 1000
      DO 10 I=1,8
      CALL GAUSS(10,AA,I,1,1)
   10 S1(I)=AA
      DO 20 I=1,8
      DO 20 J=1,8
      DO 20 K=1,8
      CALL GAUSS(12,AA,I,J,K)
   20 S2(I,J,K)=AA
      DO 30 I=1,8
      DO 30 IK=1,3
      DO 30 J=1,8
      DO 30 K=1,8
      IF(IK.EQ.1) IT=16
      IF(IK.EQ.2)IT=17
      IF(IK.EQ.3) IT=18
      CALL GAUSS(IT,AA,I,J,K)
   30 S3(I,J,K,IK)=AA
      DO 40 I=1,8
      DO 40 IK=1,3
      DO 40 J=1,8
      IF(IK.EQ.1) IT=13
      IF(IK.EQ.2) IT=14
      IF(IK.EQ.3) IT=15
      CALL GAUSS(IT,AA,I,J,1)
   40 S4(I,J,IK)=AA
      GO TO 500
 1000 CONTINUE
      READ(5,606)S1
      READ(5,606)S2
      READ(5,606)S3
      READ(5,606)S4
```

```
  500 CONTINUE
      DO 90 I=1,8
      SUM=0.0
      SUM=SUM+S1(I)*DH
      SUM=SUM+S1(I)*ABC
      DO 100 IR=1,8
      IRA=MNODE(IR)
      DO 100 IM=1,8
      IMA=MNODE(IM)
      C=SH(NN)*S2(I,IR,IM)/RT/DELT
      IF(NNN.NE.1) GO TO 50
      TO1=T1(IRA)
      DTO1=T1(IMA)
      DT12=T2(IMA)
      T12=T2(IRA)
      GO TO 60
   50 TO1=TO(IRA)
      DTO1=TO(IMA)
      DT12=T1(IMA)
      T12=T1(IRA)
   60 CONTINUE
      S=C*(2.*TO1*DTO1-TO1*DT12-T12*DTO1)
                       S=C*T1(IRA)*(T1(IMA)-T2(IMA))
      IF(IE.EQ.0.AND.NNN.GT.1) S=C*TO(IRA)*(TO(IMA)-T1(IMA))
  100 SUM=SUM+S
      SIJ=0.
      DO 120 IK=1,3
      SQ=0.
      DO 130 IM=1,8
      IKM=IK+(IM-1)*3
      MK=MKODE(IKM)
      TEMP=RT
      DO 140 IQ=1,8
      IQA=MNODE(IQ)
      IF(NNN.NE.1) GO TO 150
      TO1=T1(IQA)
      DUO1=DU1(MK)
      DU12=DU2(MK)
      GO TO 160
  150 TO1=TO(IQA)
      DUO1=DU0(MK)
      DU12=DU1(MK)
  160 CONTINUE
                       DUO1=U1(MK)
                       DU12=U2(MK)
                       TO1=T1(IQA)
      IF(IE.EQ.0.AND.NNN.GT.1) DUO1=U0(MK)
      IF(IE.EQ.0.AND.NNN.GT.1) DU12=U1(MK)
      IF(IE.EQ.0.AND.NNN.GT.1) TO1=TO(IQA)
  140 TEMP=TEMP+TO1/8.0
  130 SQ=SQ+TEMP*S4(I,IM,IK)*(DUO1-DU12)
  120 SIJ=SIJ+BIJ*SQ
      SUM=SUM+SIJ*(KAPPA)/DELT
   90 EDF(I)=SUM

      NOPRNT=1
      NOPRNT=0
      IF(NOPRNT.EQ.0) GO TO 501
      IF(NX.NE.0) RETURN
      WRITE(6,701)
      WRITE(6,700)(MNODE(I),I=1,8)
      WRITE(6,605)
      WRITE(6,600)(EDF(I),I=1,8)
  701 FORMAT(/10X,' ELEMENT NODES '/)
  700 FORMAT(8I5)
  600 FORMAT(8(1PE13.5))
      WRITE(6,601)
  601 FORMAT(/20X,' MATRIX S1 '/)
      WRITE(6,600)(S1(I),I=1,8)
      WRITE(6,602)
  602 FORMAT(/20X,' MATRIX S2 '/)
      WRITE(6,600)(((S2(I,J,K),I=1,8),J=1,8),K=1,8)
      WRITE(6,603)
  603 FORMAT(/20X,' MATRIX S3 '/)
      WRITE(6,600)((((S3(I,J,K,IK),I=1,8),J=1,8),K=1,8),IK=1,3)
      WRITE(6,604)
  604 FORMAT(/20X,' MATRIX S4 '/)
      WRITE(6,600) ((( S4(I,J,IK),I=1,8),J=1,8),IK=1,3)
  605 FORMAT(/20X,' MATRIX EDF '/)
  606 FORMAT(6E13.5)
  501 CONTINUE
      RETURN
      END

END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
      SUBROUTINE  DTHETA(DFB,DF,MLOAD,DTO,DT1,TP)
      COMMON /BLKA/NELEMT,INODE,NB,NBHC,LHEAT,LSTRES,NN,NM,NNH,NMH
      COMMON/BLD/SK(20000),HK(2000),CK(2000),HCK(2000),IBND(100),
     *   IHBND(50)
      DIMENSION DF(1),DFB(1),DTO(1),DT1(1),DX(60),DY(60)
      DIMENSION TP(60)
      CALL MATMPY(CK,NNH,DT1,DFB,60)
      WRITE(6,100)(DFB(I),I=1,NNH)
      DO 304 I=1,INODE
  304 DFB(I)=DFB(I)+DF(I)
      CALL REDUCE(DFB,IHBND,NNH,NBHC,60)
      IF(MLOAD.EQ.0) GO TO 10
      DO 30 I=1,NNH
      DY(I)=0.0
      DO 30 J=1,NBHC
```

```
      NK=IHBND(J)
   30 IF(I.EQ.NK) DY(I)=TP(I)
      WRITE(6,100)(DY(I),I=1,NNH)
      CALL MATMPY(HK,NNH,DY ,DX,60)
      WRITE(6,100)(DX(I),I=1,NNH)
      CALL REDUCE(DX,IHBND,NNH,NBHC,60)
      DO 20 I=1,NNH
   20 DFB(I)=DFB(I)-DX(I)
   10 CONTINUE
      CALL ZERO(DX,60,1)
      CALL MATMPY(HCK,NMH,DFB,DX,60)
      CALL REDUCE(DTI,IHBND,NNH,NBHC,60)
      DO 307 I=1,NMH
  307 DX(I)=DX(I)-DTI(I)
      CALL RSTORI(DX,IHBND,NNH,NBHC,60)
      CALL RSTORI(DTI,IHBND,NNH,NBHC,60)
      CALL RSTORI(DFB,IHBND,NNH,NBHC,60)
      DO 306 I=1,NNH
  306 DTO(I)=DX(I)
      DO 308 I=1,NNH
      DTO(I)=DTO(I)+TP(I)
  308 DTI(I)=DTI(I)+TP(I)
      NOPRNT=1
      NOPRNT=0
      IF(NOPRNT.EQ.0) GO TO 50
      WRITE(6,103)
  103 FORMAT(/' MATRIX DFB '/)
      WRITE(6,100) DFB
      WRITE(6,104)
  104 FORMAT(/' MATRIX DX '/)
      WRITE(6,100)(DX(I),I=1,NNH)
      WRITE(6,101)
  101 FORMAT(' MATRIX DTI ')
      WRITE(6,100) DTI
  100 FORMAT(10(1PE13.5))
      WRITE(6,102)
  102 FORMAT(/' MATRIX DTO '/)
      WRITE(6,100) DTO
   50 CONTINUE
      RETURN
      END

END OF COMPILATION:         NO  DIAGNOSTICS.
```

```
      SUBROUTINE STRSEL(DUO,DTO,IYIELD,IN)
      REAL LAMDA,MU
      REAL JCOB
      COMMON/BLK1/A(10),B(10),C(10),P(33),Q(33),R(33),W(6),H(6),IPT
      COMMON /BLK2/H1(8),H2(8),H3(8),H4(8),H5(8),H6(8),H7(8),H8(8),
     *H9(8),H10(8),GA(8),GB(8),GC(8),GD(8),GE(8),GF(8),GH(8),GI(8)
      COMMON/BLK6/D(6,6),E(10),EP(10),SYIELD(10),XNU(10),TKX(10),
     *              SH(10),ALPHA(10),DENSTY(10),NY(30),DELT,RT
      COMMON/BLK7/X(60),Y(60),Z(60),MNODE(8),MKODE(24)
      COMMON /BLK9/DSIGMA(6),SIGMA(6,30),SIGMAI(6,30)
      COMMON/BLK10/SIGMAB(30),SIGM1B(30),DSIG1B(30),DSIG2B(30),
     *              SIGMAX(30)
      COMMON/BLK11/STRN(6,30),DSTRN(6,30),STRNP(6,30),ABC
      DIMENSION LAMDA(8),MU(8),PHI(8),SS(6),DEPS(6),EDR(24)
      DIMENSION DUO(1),DTO(1)
      NN=NY(IN)
      BIJ=-E(NN)*ALPHA(NN)/(1.-2.*XNU(NN))
      CALL ZERO(EDR,24,1)
      CALL ZERO(LAMDA,8,1)
      CALL ZERO(MU,8,1)
      CALL ZERO(PHI,8,1)
      CALL ZERO(DSIGMA,6,1)
      CALL ZERO(DEPS,6,1)
      DO 10 I=1,24
      NK=MKODE(I)
   10 EDR(I)=DUO(NK)
      DO 20 I=1,8
      LAMDA(I)=H1(I)*P(1)+H4(I)*P(9)+H7(I)*P(17)
      MU(I)=H1(I)*Q(1)+H4(I)*Q(9)+H7(I)*Q(17)
   20 PHI(I)=H1(I)*R(1)+H4(I)*R(9)+H7(I)*R(17)
      DO 30 I=1,8
      I1=3*I-2
      I2=3*I-1
      I3=3*I
      DEPS(1)=LAMDA(I)*EDR(I1)+DEPS(1)
      DEPS(2)=MU(I)*EDR(I2)+DEPS(2)
      DEPS(3)=PHI(I)*EDR(I3)+DEPS(3)
      DEPS(4)=MU(I)*EDR(I1)+LAMDA(I)*EDR(I2)+DEPS(4)
      DEPS(5)=PHI(I)*EDR(I2)+MU(I)*EDR(I3)+DEPS(5)
   30 DEPS(6)=LAMDA(I)*EDR(I3)+PHI(I)*EDR(I1)+DEPS(6)
      JCOB=A(1)*P(1)+B(1)*Q(1)+C(1)*R(1)
      DETJ=1./(JCOB*8.)
      DO 40 I=1,6
   40 DEPS(I)=DEPS(I)*DETJ
      DT=0.
      DO 45 I=1,8
      MN=MNODE(I)
   45 DT=DT+DTO(MN)/8.
      DO 50 I=1,6
      DO 60 J=1,6
   60 DSIGMA(I)=DSIGMA(I)+D(I,J)*DEPS(J)
      IF(I.LE.3) DSIGMA(I)=DSIGMA(I)+BIJ*DT
```

```
      50 SIGMA(I,IN)=            DSIGMA(I)
         SG=(SIGMA(1,IN)+SIGMA(2,IN)+SIGMA(3,IN))/3.
         SS(1)=SIGMA(1,IN)-SG
         SS(2)=SIGMA(2,IN)-SG
         SS(3)=SIGMA(3,IN)-SG
         SS(4)=SIGMA(4,IN)
         SS(5)=SIGMA(5,IN)
         SS(6)=SIGMA(6,IN)
         S1=1.5*(SS(1)**2+SS(2)**2+SS(3)**2+2.*(SS(4)**2+SS(5)**2+SS(6)**2)
        *  )
         SIGMAB(IN)=SQRT(S1)
         IF(SIGMAB(IN).GT.SIGMAX(IN)) IYIELD=1
         DO 70 I=1,6
      70 STRN(I,IN)=DEPS(I)
         NOPRNT=1
         NOPRNT=0
         IF(NOPRNT.EQ.0) GO TO 51
         WRITE(6,81)
         WRITE(6,80)(EDR(I),I=1,24)
         WRITE(6,90)
         WRITE(6,83)(LAMDA(I),MU(I),PHI(I),I=1,8)
         WRITE(6,90)
         WRITE(6,84)
         WRITE(6,80)(DEPS(I) ,I=1,6)
         WRITE(6,86) DETJ
      51 CONTINUE
      80 FORMAT(1PE13.5)
      81 FORMAT(' LOCAL DISPLACEMENT MATRIX ')
      83 FORMAT(3(1PE13.5))
      84 FORMAT(' ELEMENT STRAIN MATRIX ')
      86 FORMAT(' JCOBIAN VALUE = ',1PE13.5)
      90 FORMAT(//)
         RETURN
         END

END OF COMPILATION:          NO  DIAGNOSTICS.
```

```
         SUBROUTINE STRSPL(DUO,DTO,AJ,IN)
         REAL JCOB
         REAL LAMDA,MU
         COMMON/BLK1/A(10),B(10),C(10),P(33),Q(33),R(33),W(6),H(6),IPT
         COMMON /BLK2/H1(8),H2(8),H3(8),H4(8),H5(8),H6(8),H7(8),H8(8),
        *H9(8),H10(8),GA(8),GB(8),GC(8),GD(8),GE(8),GF(8),GH(8),GI(8)
         COMMON/BLK6/D(6,6),E(10),EP(10),SYIELD(10),XNU(10),TKX(10),
        *             SH(10),ALPHA(10),DENSTY(10),NY(30),NELT,RT
         COMMON/BLK7/X(60),Y(60),Z(60),MNODE(8),MKODE(24)
         COMMON /BLK9/DSIGMA(6),SIGMA(6,30),SIGMA1(6,30)
         COMMON/BLK10/SIGMAB(30),SIGM1B(30),DSIG1B(30),DSIG2B(30),
        *              SIGMAX(30)
         COMMON/BLK11/STRN(6,30),DSTRN(6,30),STRNP(6,30),ABC
         DIMENSION LAMDA(8),MU(8),PHI(8),SS(6),DEPS(6),EDR(24)
         DIMENSION DTO(1),DUO(1),AJ(6,6)
         NN=NY(IN)
         BIJ=-E(NN)*ALPHA(NN)/(1.-2.*XNU(NN))
         CALL ZERO(EDR,24,1)
         CALL ZERO(LAMDA,8,1)
         CALL ZERO(MU,8,1)
         CALL ZERO(PHI,8,1)
         CALL ZERO(DSIGMA,6,1)
         CALL ZERO(DEPS,6,1)
         DO 10 I=1,24
         NK=MKODE(I)
      10 EDR(I)=DUO(NK)
         DO 20 I=1,8
         LAMDA(I)=H1(I)*P(1)+H4(I)*P(9)+H7(I)*P(17)
         MU(I)=H1(I)*Q(1)+H4(I)*Q(9)+H7(I)*Q(17)
      20 PHI(I)=H1(I)*R(1)+H4(I)*R(9)+H7(I)*R(17)
         DO 30 I=1,8
         I1=3*I-2
         I2=3*I-1
         I3=3*I
         DEPS(1)=LAMDA(I)*EDR(I1)+DEPS(1)
         DEPS(2)=MU(I)*EDR(I2)+DEPS(2)
         DEPS(3)=PHI(I)*EDR(I3)+DEPS(3)
         DEPS(4)=MU(I)*EDR(I1)+LAMDA(I)*EDR(I2)+DEPS(4)
         DEPS(5)=PHI(I)*EDR(I2)+MU(I)*EDR(I3)+DEPS(5)
      30 DEPS(6)=LAMDA(I)*EDR(I3)+PHI(I)*EDR(I1)+DEPS(6)
         JCOB=A(1)*P(1)+B(1)*Q(1)+C(1)*R(1)
         DETJ=1./(JCOB*8.)
         DO 40 I=1,6
      40 DEPS(I)=DEPS(I)*DETJ
         DT=0.
         DO 45 I=1,8
         MN=MNODE(I)
      45 DT=DT+DTO(MN)/8.
         DO 50 I=1,6
         DO 60 J=1,6
      60 DSIGMA(I)=DSIGMA(I)+(D(I,J)-AJ(I,J))*DEPS(J)
         IF(I.LE.3) DSIGMA(I)=DSIGMA(I)+BIJ*DT
      50 SIGMA(I,IN)=SIGMA1(I,IN)+DSIGMA(I)
         SG=(SIGMA(1,IN)+SIGMA(2,IN)+SIGMA(3,IN))/3.
         SS(1)=SIGMA(1,IN)-SG
         SS(2)=SIGMA(2,IN)-SG
```

```
      SS(3)=SIGMA(3,IN)-SG
      SS(4)=SIGMA(4,IN)
      SS(5)=SIGMA(5,IN)
      SS(6)=SIGMA(6,IN)
      DSIG2B(IN)=0.0
      DO 70 I=1,3
      J=I+3
      DSIG2B(IN)=DSIG2B(IN)+(1.5*SS(I)*DSIGMA(I)+3.0*SS(J)*DSIGMA(J))
     *          /SIGMAB(IN)
   70 CONTINUE
      DO 75 I=1,6
   75 DSTRN(I,IN)=DEPS(I)
      NOPRNT=1
      NOPRNT=0
      IF(NOPRNT.EQ.0) GO TO 51
      WRITE(6,81)
      WRITE(6,80)(EDR(I),I=1,24)
      WRITE(6,90)
      WRITE(6,90)
      WRITE(6,83)(LAMDA(I),MU(I),PHI(I),I=1,8)
      WRITE(6,84)
      WRITE(6,80)(DEPS(I) ,I=1,6)
      WRITE(6,86) DETJ
   80 FORMAT(1PE13.5)
   81 FORMAT(' LOCAL DISPLACEMENT MATRIX ')
   83 FORMAT(3(1PE13.5))
   84 FORMAT(' ELEMENT STRAIN MATRIX ')
   86 FORMAT(' JCOBIAN VALUE = ',1PE13.5)
   90 FORMAT(//)
   51 CONTINUE
      RETURN
      END

END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
      COMMON /BLK9/DSIGMA(6),SIGMA(6,30),SIGMAI(6,30)
      COMMON/BLK10/SIGMAB(30),SIGMIB(30),DSIG1B(30),DSIG2B(30),
     *             SIGMAX(30)
      DIMENSION AJ(6,6),SS(6)
      REWIND 2
      DO 100 I=1,NELEMT
      NN=NY(I)
      G=E(NN)/2./(1.+XNU(NN))
      CALL ZERO(AJ,6,6)
      IF(DSIG2B(I).LT.0.0) GO TO 50
      IF(SIGMAB(I).LT.SIGMAX(I)*0.99) GO TO 50
      S0=2./9./G*SIGMAB(I)**2*(EP(NN)+3.*G)
      IN=I
      SG=(SIGMA(1,IN)+SIGMA(2,IN)+SIGMA(3,IN))/3.
      SS(1)=SIGMA(1,IN)-SG
      SS(2)=SIGMA(2,IN)-SG
      SS(3)=SIGMA(3,IN)-SG
      SS(4)=SIGMA(4,IN)
      SS(5)=SIGMA(5,IN)
      SS(6)=SIGMA(6,IN)
      DO 120 J1=1,6
      DO 120 J2=1,6
  120 AJ(  J1,J2)=SS(J1)*SS(J2)*2.*G/S0
   50 CONTINUE
      WRITE (6,1) I,AJ
    1 FORMAT(1H0,'JMATRIX',I5,6(/1H ,6E13.5))
      WRITE(2) AJ
  100 CONTINUE
      RETURN
      END

END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
      SUBROUTINE SKAPPA
      COMMON /BLKA/NELEMT,INODE,NB,NBHC,LHEAT,LSTRES,NN,NM,NNH,NMH
      COMMON/BLK6/D(6,6),E(10),EP(10),SYIELD(10),XNU(10),TKX(10),
     *            SH(10),ALPHA(10),DENSTY(10),NY(30),NELT,RT
```

```
      SUBROUTINE STIFF(D,NX,NOPRNT)
      COMMON /BLKC/ESK(24,24),HC(8,8),HSK(8,8),TYPEH(8,8),TYPEI(8,8),
     *TYPEJ(8,8),TYPEM(8,8),TYPEN(8,8),TYPES(8,8)
      DIMENSION D(6,6)
      DIMENSION HSKI(8,8),HCI(8,8)
      IF(NX.NE.0) GO TO 350
      CALL ZERO(ESK,24,24)
      CALL ZERO(HSK,8,8)
      DO 200 M=1,8
      DO 200 N=M,8
      CALL GAUSS(1,AA,M,N,1)
  200 TYPEH(M,N)=AA
      DO 210 M=1,8
      DO 210 N=M,8
      CALL GAUSS(2,AA,M,N,1)
  210 TYPEI(M,N)=AA
      DO 220 M=1,8
      DO 220 N=M,8
      CALL GAUSS(3,AA,M,N,1)
  220 TYPEJ(M,N)=AA
      DO 260 M=1,8
      DO 260 N=M,8
      TYPEH(N,M)=TYPEH(M,N)
      TYPEI(N,M)=TYPEI(M,N)
  260 TYPEJ(N,M)=TYPEJ(M,N)
      DO 230 M=1,8
      DO 230 N=1,8
      CALL GAUSS(4,AA,M,N,1)
  230 TYPEM(M,N)=AA
      DO 240 M=1,8
      DO 240 N=1,8
      CALL GAUSS(5,AA,M,N,1)
  240 TYPES(M,N)=AA
      DO 250 M=1,8
      DO 250 N=1,8
      CALL GAUSS(6,AA,M,N,1)
  250 TYPEN(M,N)=AA
      DO 300 I=1,8
      DO 300 J=1,8
      ESK(I,J)=D(1,1)*TYPEH(I,J)+D(4,4)*TYPEI(I,J)+D(6,6)*TYPEJ(I,J)
      ESK(I,J+8)=D(1,2)*TYPEM(I,J)+D(4,4)*TYPEM(J,I)
      ESK(I,J+16)=D(1,3)*TYPEN(I,J)+D(6,6)*TYPEN(J,I)
      ESK(I+8,J+8)=D(2,2)*TYPEI(I,J)+D(4,4)*TYPEH(I,J)+D(5,5)*TYPEJ(I,J)
      ESK(I+8,J+16)=D(2,3)*TYPES(I,J)+D(5,5)*TYPES(J,I)
      ESK(I+16,J+16)=D(3,3)*TYPEJ(I,J)+D(5,5)*TYPEI(I,J)+
     *D(6,6)*TYPEH(I,J)
  300 CONTINUE
      DO 310 I=1,24
      DO 310 J=1,24
  310 ESK(J,I)=ESK(I,J)
   10 FORMAT(6(1PE13.5))
      DO 900 I=1,8
      DO 900 J=1,8
  900 HSK(I,J)=HSK(I,J)+TYPEH(I,J)+TYPEI(I,J)+TYPEJ(I,J)
      DO 270 I=1,8
      DO 270 J=1,8
      CALL GAUSS(11,AA,I,J,1)
  270 HC(I,J)=AA
      NOPRNT=1
      NOPRNT=0
      IF(NOPRNT.EQ.0) GO TO 30
      WRITE(6,601)
  601 FORMAT(//20X,5HTYPEH)
      WRITE(6,600)((TYPEH(I,J),I=1,8),J=1,8)
      WRITE(6,602)
  602 FORMAT(//20X,5HTYPEI)
      WRITE(6,600)((TYPEI(I,J),I=1,8),J=1,8)
      WRITE(6,603)
  603 FORMAT(//20X,5HTYPEJ)
      WRITE(6,600)((TYPEJ(I,J),I=1,8),J=1,8)
      WRITE(6,604)
  604 FORMAT(//20X,5HTYPEM)
      WRITE(6,600)((TYPEM(I,J),I=1,8),J=1,8)
      WRITE(6,606)
  606 FORMAT(//20X,5HTYPEN)
      WRITE(6,600)((TYPEN(I,J),I=1,8),J=1,8)
      WRITE(6,605)
  605 FORMAT(//20X,5HTYPES)
      WRITE(6,600)((TYPES(I,J),I=1,8),J=1,8)
  600 FORMAT(8(1PE13.5))
      WRITE(6,599)
  599 FORMAT(1H1/,5X,24HELEMENT STIFFNESS MATRIX/)
      WRITE(6,600)((ESK(I,J),I=1,24),J=1,24)
      WRITE(6,701)
  701 FORMAT(10X,' THERMAL STIFFNESS MATRIX '/)
      WRITE(6,600)((HSK(I,J),I=1,8),J=1,8)
      WRITE(6,271)
  271 FORMAT(/1X,' TRANSIENT HEAT CONDUCTION MATRIX '/)
      WRITE(6,600)((HC(I,J),I=1,8),J=1,8)
   30 CONTINUE
      DO 400 I=1,8
      DO 400 J=1,8
      HSKI(I,J)=HSK(I,J)
  400 HCI(I,J)=HC(I,J)
      RETURN
  350 CONTINUE
      DO 410 I=1,8
      DO 410 J=1,8
      HSK(I,J)=HSKI(I,J)
  410 HC(I,J)=HCI(I,J)
      RETURN
      END

END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
      SUBROUTINE PSTIFF(PK,PJ,IN,IGE)
      COMMON /BLKA/NELEMT,INODE,NB,NBHC,LHEAT,LSTRES,NN,NM,NNH,NMH
      COMMON /BLKC/ESK(24,24),HC(8,8),HSK(8,8),TYPEH(8,8),TYPEI(8,8),
     *TYPEJ(8,8),TYPEM(8,8),TYPEN(8,8),TYPES(8,8)
      COMMON/BLK9/DSIGMA(6),SIGMA(6,30),SIGMAI(6,30)
      DIMENSION PK(24,24),PJ(6,6)
      DO 300 I=1,8
      DO 300 J=1,8
      PK(I,J)= PJ(1,1)*TYPEH(I,J)+PJ(1,4)*TYPEM(I,J)+PJ(1,6)*TYPEN(I,J)+
     *         PJ(4,1)*TYPEM(J,I)+PJ(4,4)*TYPEI(I,J)+PJ(4,6)*TYPES(I,J)+
     *         PJ(6,1)*TYPEN(J,I)+PJ(6,4)*TYPES(J,I)+PJ(6,6)*TYPEJ(I,J)
      PK(I,J+8)=PJ(1,2)*TYPEM(I,J)+PJ(1,4)*TYPEH(I,J)+PJ(1,5)*TYPEN(I,J)
     *         +PJ(4,2)*TYPEI(I,J)+PJ(4,4)*TYPEM(J,I)+PJ(4,5)*TYPES(I,J)
     *         +PJ(6,2)*TYPES(J,I)+PJ(6,4)*TYPEN(J,I)+PJ(6,5)*TYPEJ(I,J)
       PK(I,J+16)= PJ(1,3)*TYPEN(I,J)+PJ(1,5)*TYPEM(I,J)+PJ(1,6)*TYPEH
     *(I,J)+PJ(4,3)*TYPES(I,J)+PJ(4,5)*TYPEI(I,J)+PJ(4,6)*TYPEM(J,I)+
     1 PJ(6,3)*TYPEJ(I,J)+PJ(6,5)*TYPES(J,I)+PJ(6,6)*TYPEN(J,I)
       PK(I+8,J+8)=PJ(2,2)*TYPEI(I,J)+PJ(2,4)*TYPEM(J,I)+PJ(2,5)*TYPES
     *(I,J)+PJ(4,2)*TYPEM(I,J)+PJ(4,4)*TYPEH(I,J)+PJ(4,5)*TYPEN(I,J)+
     1 PJ(5,2)*TYPES(J,I)+PJ(5,4)*TYPEN(J,I)+PJ(5,5)*TYPEJ(I,J)
       PK(I+8,J+16)=PJ(2,3)*TYPES(I,J)+PJ(2,5)*TYPEI(I,J)+PJ(2,6)*TYPEM
     *(J,I)+PJ(4,3)*TYPEN(I,J)+PJ(4,5)*TYPEM(I,J)+PJ(4,6)*TYPEH(I,J)+
     2PJ(5,3)*TYPEJ(I,J)+PJ(5,5)*TYPES(J,I)+PJ(5,6)*TYPEN(J,I)
       PK(I+16,J+16)=PJ(3,3)*TYPEJ(I,J)+PJ(3,5)*TYPES(J,I)+PJ(3,6)*TYPEN
     *(J,I)+PJ(5,3)*TYPES(I,J)+PJ(5,5)*TYPEI(I,J)+PJ(5,6)*TYPEM(J,I)+
     2PJ(6,3)*TYPEN(I,J)+PJ(6,5)*TYPEM(I,J)+PJ(6,6)*TYPEH(I,J)
  300 CONTINUE
       DO 301 I=1,24
       DO 301 J=1,24
  301 PK(J,I)=PK(I,J)
       NOPRNT=1
       NOPRNT=0
       IF(NOPRNT.NE.1) GO TO 500
  600 FORMAT(8(1PE13.5))
       WRITE(6,601)
  601 FORMAT(//20X,5HTYPEH/)
       WRITE(6,600)((TYPEH(I,J),I=1,8),J=1,8)
       WRITE(6,605)
  605 FORMAT(//20X,5HTYPEI/)
       WRITE(6,600)((TYPEI(I,J),I=1,8),J=1,8)
       WRITE(6,602)
  602 FORMAT(//20X,5HTYPEJ/)
       WRITE(6,600)((TYPEJ(I,J),I=1,8),J=1,8)
       WRITE(6,603)
  603 FORMAT(//20X,5HTYPEM/)
       WRITE(6,600)((TYPEM(I,J),I=1,8),J=1,8)
       WRITE(6,604)
  604 FORMAT(//20X,5HTYPEN/)
       WRITE(6,600)((TYPEN(I,J),I=1,8),J=1,8)
       WRITE(6,606)
  606 FORMAT(//20X,5HTYPES/)
       WRITE(6,600)((TYPES(I,J),I=1,8),J=1,8)
       WRITE(6,610)
  610 FORMAT(/20X,' MATRIX J '/)
       WRITE(6,611)((PJ(I,J),I=1,6),J=1,6)
  611 FORMAT(6(1PE13.5))
       WRITE(6,620)
  620 FORMAT(//20X,' MATRIX PK '/)
       WRITE(6,600)((PK(I,J),I=1,24),J=1,24)
  500 CONTINUE
       RETURN
       END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE PRNTDT(A,B,X,Y,Z,M,NJ,NP,NN)
C  PRINTS OUT JOINT FORCES AND DISPLACEMENTS FOR 3-D  STRUCTURES        11000050
      DIMENSION A(NN),B(NN),X(1),Y(1),Z(1)
      NP=NP+1                                                           11000070
      LINES=1                                                           11000080
      WRITE(6,1)NP                                                      11000090
      WRITE(6,2)                                                        11000100
      DO 15 I=1,NJ                                                      11000110
      IF(LINES-49)4,3,3
    3 LINES=1                                                           11000130
      NP=NP+1                                                           11000160
      WRITE(6,1 ) NP                                                    11000140
      WRITE(6,2 )                                                       11000150
    4 KH=M *I                                                           11000170
      KHH=KH-M+1                                                        11000180
    5 WRITE(6,16)I,X(I),Y(I),Z(I),(A(J),J=KHH,KH),(B(J),J=KHH,KH)
   14 LINES=LINES+1
   15 CONTINUE                                                          11000310
    1 FORMAT(1H1,115X,5HPAGE ,I3/)
    2 FORMAT(// 5X,5HJOINT,3X,7HCOORD-X,3X,7HCOORD-Y,3X,7HCOORD-Z,6X,  11000330
     17HFORCE-X,4X,7HFORCE-Y,4X,7HFORCE-Z,10X,7HDISPL-X,7X,7HDISPL-Y,  11000340
     27X,7HDISPL-Z/   )                                                 11000350
   16 FORMAT(/ 5X,I4,2X,F9.3,1X,F9.3,1X,F9.3,3X,F10.3,1X,F10.3,1X,F10.3,11000360
     16X,1PE13.5,1X,1PE13.5,1X,1PE13.5)                                 11000370
      RETURN                                                            11000430
      END                                                               11000440

END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
      SUBROUTINE PRINTT(A,B,X,Y,Z,M,NJ,NP,NN)
      COMMON/BLK4/TD(60),T1(60),T2(60),DT0(60),DT1(60),DT2(60)
C ***** PRINTS OUT NODAL TEMPERATURES FOR 3-D STRUCTURES  **********    1100050
      DIMENSION A(NN),B(NN),X(1),Y(1),Z(1)
      NP=NP+1                                                           11000070
      LINES=1                                                           11000080
      WRITE(6,1)NP                                                      11000090
      WRITE(6,2)                                                        11000100
      DO 15 I=1,NJ                                                      11000110
      IF(LINES-49)4,3,3
    3 LINES=1                                                           11000130
      NP=NP+1                                                           11000160
      WRITE(6,1 ) NP                                                    11000140
      WRITE(6,2 )                                                       11000150
    4 CONTINUE
    5 WRITE(6,16)I,X(I),Y(I),Z(I),A(I),B(I),T2(I),T1(I),T0(I)
   14 LINES = LINES + 1
   15 CONTINUE                                                          11000310
    1 FORMAT(1H1,115X,5HPAGE ,I3/)
    2 FORMAT(/ 5X,5HJOINT,3X,7HCOORD-X,3X,7HCOORD-Y,3X,7HCOORD-Z,
     *9X'DF',10X,'DFB',10X,'T2 ',10X,'T1 ',10X,'T0'/)
   16 FORMAT(/5X,I4,2X,F9.3,1X,F9.3,1X,F9.3,5X,5(1PE13.5))
      RETURN                                                            11000430
      END                                                               11000440

END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
      SUBROUTINE PRINTK(A,B,N)                                          10700060
      DIMENSION  A(1), B(1)                                             10700070
      WRITE(6,3)                                                        10700080
      DO 2 I=1,N                                                        10700090
      WRITE(6,4) I                                                      10700100
      IN=N                                                              10700110
      JN=I                                                              10700120
      DO 1 J=1,I                                                        10700130
      B(J) =A(JN)                                                       10700140
      IN=IN-1                                                           10700150
    1 JN=JN+IN                                                          10700160
      WRITE(6,5)  (B(J),J=1,I)                                          10700170
    2 CONTINUE                                                          10700180
    3 FORMAT(1H1//, 2X,33HSTIFFNESS MATRIX (LOWER TRIANGLE) ///)        10700190
    4 FORMAT(/2X, 5HROW =,I4)                                           10700200
    5 FORMAT(1X,1P10E13.4)                                              10700210
      RETURN                                                            10700220
      END                                                               10700230

END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
      SUBROUTINE ELASTC(D,E,XNU)
      DIMENSION D(6,6)
      CONST=E/((1.+XNU)*(1.-2.*XNU))
      CALL ZERO(D,6,6)
      D(1,1)=1.-XNU
      D(1,2)=XNU
      D(1,3)=XNU
      D(2,1)=XNU
      D(2,2)=D(1,1)
      D(2,3)=XNU
      D(3,1)=XNU
      D(3,2)=XNU
      D(3,3)=D(1,1)
      D(4,4)=0.5*(1.-2.*XNU)
      D(5,5)=D(4,4)
      D(6,6)=D(4,4)
      DO 20 I=1,6
      DO 20 J=1,6
   20 D(I,J)=CONST*D(I,J)
      NOPRNT=1
      NOPRNT=0
      IF(NOPRNT.EQ.0) GO TO 399
      WRITE(6,100)
  100 FORMAT(/10X,14HMATRIX ELASTIC/)
      WRITE(6,200)((D(I,J),I=1,6),J=1,6)
  200 FORMAT(6(1PE13.5))
  399 CONTINUE
      RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE ZERO(A,M,N)
      DIMENSION A(1)
      K=M*N
      DO 10 I=1,K
   10 A(I)=0.0
      RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE REDUCE(F,IB,N,NB,NN)
      DIMENSION F(NN),IB(1)                                             10500070
      IH=NB                                                             10500080
      NH=N                                                              10500090
    1 I=IB(IH)                                                          10500100
      IF(I-NH) 2,4,4                                                    10500110
    2 NH1=NH-1                                                          10500120
      DO 3 K=I,NH1                                                      10500130
      K1=K+1
    3 F(K)=F(K1)                                                        10500150
    4 IH=IH-1                                                           10500160
      NH=NH-1                                                           10500170
      IF(IH.EQ.0) GO TO 5                                               10500180
      GO TO1                                                            10500190
    5 CONTINUF                                                          10500200
      RETURN                                                            10500210
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE RSTORI(D,IB,N,NB,NN)
      DIMENSION D(NN),IB(1)
      NH=N-NB                                                           10600060
      IH=1                                                              10600070
    1 I=IB(IH)                                                          10600080
      IF(I.GT.NH) GO TO 7                                               10600090
      TDR1=D(I)
    2 D(I)=0.0
    3 J=I+1                                                             10600130
      IF(J.GT.NH) GO TO 5                                               10600140
      TDR2=D(J)
    5 D(J)=TDR1
      TDR1=TDR2
      IF(I.GE.NH) GO TO 9                                               10600200
      I=I+1                                                             10600210
      GO TO 3                                                           10600220
    7 D(I)=0.0
    9 IF(IH.GF.NB) GO TO 10                                             10600250
      IH=IH+1                                                           10600260
      NH=NH+1                                                           10600270
      GO TO 1                                                           10600280
   10 CONTINUF                                                          10600290
      RETURN                                                            10600300
      END                                                               10600310
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE CONTH
      COMMON /BLK2/H1(8),H2(8),H3(8),H4(8),H5(8),H6(8),H7(8),H8(8),
     *H9(8),H10(8),G1(8),G2(8),G3(8),G4(8),G5(8),G6(8),G7(8),G8(8)
      H1(1)=-1.0
      H1(2)= 1.0
      H1(3)=1.0
      H1(4)=-1.0
      H1(5)=-1.0
      H1(6)= 1.0
      H1(7)= 1.0
      H1(8)=-1.0
      H2(1)= 1.0
      H2(2)=-1.0
      H2(3)= 1.0
      H2(4)=-1.0
      H2(5)= 1.0
      H2(6)=-1.0
      H2(7)= 1.0
      H2(8)=-1.0
      H3(1)= 1.0
      H3(2)=-1.0
      H3(3)=-1.0
      H3(4)= 1.0
      H3(5)=-1.0
      H3(6)= 1.0
      H3(7)= 1.0
      H3(8)=-1.0
```

69

```
      H4(1)=-1.0
      H4(2)=-1.0
      H4(3)= 1.0
      H4(4)= 1.0
      H4(5)=-1.0
      H4(6)=-1.0
      H4(7)= 1.0
      H4(8)= 1.0
      H5(1)= 1.0
      H5(2)= 1.
      H5(3)=-1.0
      H5(4)=-1.
      H5(5)=-1.0
      H5(6)=-1.
      H5(7)= 1.0
      H5(8)= 1.
      H7(1)=-1.0
      H7(2)=-1.0
      H7(3)=-1.0
      H7(4)=-1.0
      H7(5)= 1.0
      H7(6)= 1.0
      H7(7)= 1.0
      H7(8)= 1.0
      H10(1)=-1.0
      H10(2)= 1.0
      H10(3)=-1.0
      H10(4)= 1.0
      H10(5)= 1.0
      H10(6)=-1.0
      H10(7)= 1.0
      H10(8)=-1.0
      DO 10 I=1,8
      H6(I)=H2(I)
      H8(I)=H3(I)
   10 H9(I)=H5(I)
      DO 20 I=1,8
   20 G1(I)= 1.0
      G2(1)=-1.0
      G2(2)= 1.0
      G2(3)= 1.0
      G2(4)=-1.0
      G2(5)=-1.0
      G2(6)= 1.0
      G2(7)= 1.0
      G2(8)=-1.0
      G3(1)=-1.0
      G3(2)=-1.0
      G3(3)= 1.0
      G3(4)= 1.0
      G3(5)=-1.0
      G3(6)=-1.0
      G3(7)= 1.0
      G3(8)= 1.0
      G4(1)=-1.0
      G4(2)=-1.0
      G4(3)=-1.0
      G4(4)=-1.0
      G4(5)= 1.0
      G4(6)= 1.0
      G4(7)= 1.0
      G4(8)= 1.0
      G5(1)= 1.0
      G5(2)=-1.0
      G5(3)= 1.0
      G5(4)=-1.0
      G5(5)= 1.0
      G5(6)=-1.0
      G5(7)= 1.0
      G5(8)=-1.0
      G6(1)= 1.0
      G6(2)= 1.0
      G6(3)=-1.0
      G6(4)=-1.0
      G6(5)=-1.0
      G6(6)=-1.0
      G6(7)= 1.0
      G6(8)= 1.0
      G7(1)= 1.0
      G7(2)=-1.0
      G7(3)=-1.0
      G7(4)= 1.0
      G7(5)=-1.0
      G7(6)= 1.0
      G7(7)= 1.0
      G7(8)=-1.0
      G8(1)=-1.0
      G8(2)= 1.0
      G8(3)=-1.0
      G8(4)= 1.0
      G8(5)= 1.0
      G8(6)=-1.0
      G8(7)= 1.0
      G8(8)=-1.0
      RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE MATMPY(A,N,X,Y,NN)
      DIMENSION A(1),X(NN),Y(NN)
      DO 4 I=1,N
      NA=I
      NB=N-1
    1 Y(I)=0.0
      DO 4 J=1,N
    5 Y(I)=Y(I)+A(NA)*X(J)
      IF (J-I) 2,3,3
    2 NA=NA+NB
      NB=NB-1
      GO TO 4
    3 NA=NA+1
    4 CONTINUE
      RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE BOUND(A,IB,N,NB)
      DIMENSION A(1),IB(1)
      IH=NB
      NH=N
    1 I=IB(IH)
      J=1
      KA=I
      IF(I .LE. 1) GO TO 6
    2 KD=((2*NH-J+1)*J)/2+(I-J-1)
    3 KC=KA+J
      A(KA)=A(KC)
      IF(KA .GE. (KD-J)) GO TO 4
      KA=KA+1
      GO TO 3
    4 IF( J .GE. (I-1)) GO TO 5
      KA=KA+1
      J=J+1
      GO TO 2
    5 IF(J .GE. (NH-1)) GO TO 8
      KA=KA+1
    6 KD=(NH*(NH+1))/2
    7 KC=KA+NH
      A(KA)=A(KC)
      IF(KC .GE. KD) GO TO 8
      KA=KA+1
      GO TO 7
    8 IF(IH .LE. 1) GO TO 9
      IH=IH-1
      NH=NH-1
      GO TO 1
    9 CONTINUE
      RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE  MATINV(A,C,N)
C     INVERSION SUBROUTINE FOR SYMMETRICAL MATRICES
C
C     REPLACES MATRIX BY ITS INVERSE
C
C     GOOD ONLY FOR WELL CONDITIONED MATRICES
C
C ****METHOD .. CROUTS REDUTION (CHOLESKYS METHOD) MODIFIED FOR
C                SYMMETRICAL MATRICES
C
C     A  IS THE MATRIX TO BE INVERTED
C ****C IS A DUMMY COLUMN OR ROW VECTOR OF SIZE N
C     N  IS THE SIZE OF THE MATRIX TO BE INVERTED
C
C ****INPUT LOWER OR UPPER TRIANGLE MATRIX AS A SINGLE ARRAY
C     NUMBER COLUMN (OR ROW) WISE, STARTING WITH DIAGONAL ELEMENT
C
C ****LIMIT ON THE SIZE OF MATRIX A IS SLIGHTLY MORE THAN 200
C
C     A  AND C SHOULD BE DIMENSIONED    AS SINGLE ARRAYS IN THE
C     MAIN PROGRAM
C
C ****NO PROVISION FOR TRANSFER OF CONTROL IN CASE OF SINGULAR MATRICES
C
C     CALL MATINV(A,C,N)
C
C
      DIMENSION A(1),C(1)
      IX(I,J)= (J-1)*N-((J-1)*(J-2))/2 +I-J+1
      DO 6 J=2,N
      DO 6 I=J,N
```

63

```
      JX=J-1
      SUM=0
      DO 5 K=1,JX
      IXX=IX(I,K)
      JXX=IX(J,K)
      KXX=IX(K,K)
    5 SUM=SUM+(A(IXX)*A(JXX))/A(KXX)
      IXX=IX(I,J)
    6 A(IXX)=A(IXX)-SUM
      NB=N+1
      N1=N-1
      DO 15 J=1,N1
      JXX=IX(J,J)
      C(J)=1.0/A(JXX)
      J1=J+1
      DO 10 I=J1,N
      SUM=0
      I1=I-1
      DO 9 K=J,I1
      IXX=IX(I,K)
    9 SUM=SUM+A(IXX)*C(K)
      IXX=IX(I,I)
   10 C(I)=-SUM/A(IXX)
      I=N
      NB=NB-1
      DO 12 JA=2,NB
      I=I-1
      IXX=IX(I,I)
      SUM=0
      I1=I+1
      KXX=IX(I1,I)-1
      DO 11 K=I1,N
      KXX=KXX+1
   11 SUM=SUM+A(KXX)*C(K)
   12 C(I)=C(I)-SUM/A(IXX)
      DO 13 I=J,N
      A(JXX)=C(I)
   13 JXX=JXX+1
   15 CONTINUE
      JXX=IX(N,N)
      A(JXX)=1.0/A(JXX)
      RETURN
      END

END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
      SUBROUTINE ASSIGN(ESK,MX,NX,KODE,NE,N)
      DIMENSION ESK(N,N),MX(1),X(24),XSK(24,24)
      MM=3
      IF(NX.EQ.0) GO TO 330
      DO 340 I=1,24
      DO 340 J=1,24
  340 ESK(I,J)=XSK(I,J)
      GO TO 350
  330 CONTINUE
      DO 30 J=1,N
      DO 10 I=1,KODE
      I1=I+KODE
      I2=I1+KODE
      K=I*MM
      L=K-1
      M=L-1
      X(M)=ESK(I,J)
      X(L)=ESK(I1,J)
   10 X(K)=ESK(I2,J)
      DO 20 I=1,N
   20 ESK(I,J)=X(I)
   30 CONTINUE
      DO 40 I=1,N
      DO 50 J=1,KODE
      J1=J+KODE
      J2=J1+KODE
```

```
      K=J*MM
      L=K-1
      M=L-1
      X(M)=ESK(I,J)
      X(L)=ESK(I,J1)
   50 X(K)=ESK(I,J2)
      DO 60 J=1,N
   60 ESK(I,J)=X(J)
   40 CONTINUE
  350 CONTINUE
      DO 320 I=1,24
      DO 320 J=1,24
  320 XSK(I,J)=ESK(I,J)
      NOPRNT=1
      NOPRNT=0
      IF(NOPRNT.EQ.0) GO TO 399
      WRITE(6,611)NE,(MX(I),I=1,8)
  611 FORMAT(/5X,14HMEMBER NUMBER=,I3,2X,3HMA=,I3,2X,3HMB=,I3,2X,3HMC=,
     *I3,2X,3HMD=,I3,2X,3HMP=,I3,2X,3HMQ=,I3,2X,3HMR=,I3,2X,3HMS=,I3/)
      WRITE(6,4)
    4 FORMAT(//4X,24HELEMENT STIFFNESS MATRIX/)
      WRITE(6,2)((ESK(I,J),I=1,N),J=1,N)
    2 FORMAT(//1X,8(1PE13.5))
  399 CONTINUE
      RETURN
      END

END OF COMPILATION:        NO  DIAGNOSTICS.

      SUBROUTINE PQR(X,Y,Z,MX,NX)
      COMMON /BLK1/A(10),B(10),C(10),P(33),Q(33),R(33),W(6),H(6),IPT
      DIMENSION X(1),Y(1),Z(1),MX(1)
      IF(NX.NE.0) GO TO 350
      CALL ZERO(W,6,1)
      CALL ZERO(H,6,1)
      IF(IPT.NE.6) GO TO 20
      W(1)=0.1713244924
      W(2)=0.3607615730
      W(3)=0.4679139344
      W(4)=W(3)
      W(5)=W(2)
      W(6)=W(1)
      H(1)=0.9324695142
      H(2)=0.6612093865
      H(3)=0.2386619186
      H(4)=-H(3)
      H(5)=-H(2)
      H(6)=-H(1)
      GO TO 115
   20 IF(IPT.NE.4) GO TO 40
      H(1)=0.8611363115
      H(2)=0.3399810435
      H(3)=-H(2)
      H(4)=-H(1)
      W(1)=0.3478548451
      W(2)=0.6521451548
      W(3)=W(2)
      W(4)=W(1)
      GO TO 115
   40 CONTINUE
      W(1)=1.0
      W(2)=1.0
      H(1)=0.5773502691
      H(2)=-H(1)
  115 CONTINUE
      I1=MX(1)
      I2=MX(2)
      I3=MX(3)
      I4=MX(4)
      I5=MX(5)
      I6=MX(6)
      I7=MX(7)
      I8=MX(8)
      A(1)=-X(I1)+X(I2)+X(I3)-X(I4)-X(I5)+X(I6)+X(I7)-X(I8)
      A(2)=-X(I1)-X(I2)+X(I3)+X(I4)-X(I5)-X(I6)+X(I7)+X(I8)
      A(3)=-X(I1)-X(I2)-X(I3)-X(I4)+X(I5)+X(I6)+X(I7)+X(I8)
      A(4)= X(I1)-X(I2)+X(I3)-X(I4)+X(I5)-X(I6)+X(I7)-X(I8)
      A(5)= X(I1)+X(I2)-X(I3)-X(I4)-X(I5)-X(I6)+X(I7)+X(I8)
      A(6)= X(I1)-X(I2)-X(I3)+X(I4)-X(I5)+X(I6)+X(I7)-X(I8)
      A(7)=A(6)
      A(8)=A(4)
      A(9)=A(5)
      A(10)=-X(I1)+X(I2)-X(I3)+X(I4)+X(I5)-X(I6)+X(I7)-X(I8)
      B(1)=-Y(I1)+Y(I2)+Y(I3)-Y(I4)-Y(I5)+Y(I6)+Y(I7)-Y(I8)
      B(2)=-Y(I1)-Y(I2)+Y(I3)+Y(I4)-Y(I5)-Y(I6)+Y(I7)+Y(I8)
      B(3)=-Y(I1)-Y(I2)-Y(I3)-Y(I4)+Y(I5)+Y(I6)+Y(I7)+Y(I8)
      B(4)= Y(I1)-Y(I2)+Y(I3)-Y(I4)+Y(I5)-Y(I6)+Y(I7)-Y(I8)
      B(5)= Y(I1)+Y(I2)-Y(I3)-Y(I4)-Y(I5)-Y(I6)+Y(I7)+Y(I8)
      B(6)= Y(I1)-Y(I2)-Y(I3)+Y(I4)-Y(I5)+Y(I6)+Y(I7)-Y(I8)
      B(7)=B(6)
      B(8)=B(4)
      B(9)=B(5)
      B(10)=-Y(I1)+Y(I2)-Y(I3)+Y(I4)+Y(I5)-Y(I6)+Y(I7)-Y(I8)
      C(1)=-Z(I1)+Z(I2)+Z(I3)-Z(I4)-Z(I5)+Z(I6)+Z(I7)-Z(I8)
      C(2)=-Z(I1)-Z(I2)+Z(I3)+Z(I4)-Z(I5)-Z(I6)+Z(I7)+Z(I8)
      C(3)=-Z(I1)-Z(I2)-Z(I3)-Z(I4)+Z(I5)+Z(I6)+Z(I7)+Z(I8)
      C(4)= Z(I1)-Z(I2)+Z(I3)-Z(I4)+Z(I5)-Z(I6)+Z(I7)-Z(I8)
      C(5)= Z(I1)+Z(I2)-Z(I3)-Z(I4)-Z(I5)-Z(I6)+Z(I7)+Z(I8)
      C(6)= Z(I1)-Z(I2)-Z(I3)+Z(I4)-Z(I5)+Z(I6)+Z(I7)-Z(I8)
      C(7)=C(6)
      C(8)=C(4)
      C(9)=C(5)
      C(10)=-Z(I1)+Z(I2)-Z(I3)+Z(I4)+Z(I5)-Z(I6)+Z(I7)-Z(I8)
      DO 10 I=1,9
      A(I)=0.125*A(I)
```

```
      B(I)=0.125*B(I)
   10 C(I)=0.125*C(I)
      P(1)=B(2)*C(3)-C(2)*B(3)
      P(2)=B(2)*C(6)-C(2)*B(6)+B(8)*C(3)-C(8)*B(3)
      P(3)=B(2)*C(9)-C(2)*B(9)
      P(4)=B(5)*C(3)-C(5)*B(3)
      P(5)=B(8)*C(6)-C(8)*B(6)
      P(6)=B(2)*C(10)-C(2)*B(10)+B(8)*C(9)-C(8)*B(9)
      P(7)=B(5)*C(9)-C(5)*B(9)
      P(8)=B(5)*C(6)-C(5)*B(6)+B(10)*C(3)-C(10)*B(3)
      P(25)=B(5)*C(10)-C(5)*B(10)+B(10)*C(9)-C(10)*B(9)
      P(26)=B(8)*C(10)-C(8)*B(10)
      P(27)=B(10)*C(6)-C(10)*B(6)
      P(9)=C(1)*B(3)-B(1)*C(3)
      P(10)=C(1)*B(6)-B(1)*C(6)
      P(11)=C(1)*B(9)-B(1)*C(9)+C(4)*B(3)-B(4)*C(3)
      P(12)=B(3)*C(7)-C(3)*B(7)
      P(13)=C(4)*B(9)-B(4)*C(9)
      P(14)=C(1)*B(10)-C(10)*B(1)+C(4)*B(6)-B(4)*C(6)
      P(15)=C(7)*B(9)-B(7)*C(9)+C(10)*B(3)-B(10)*C(3)
      P(16)=C(7)*B(6)-B(7)*C(6)
      P(28)=C(7)*B(10)-B(7)*C(10)+C(10)*B(6)-B(10)*C(6)
      P(29)=C(4)*B(10)-B(4)*C(10)
      P(30)=C(10)*B(9)-B(10)*C(9)
      P(17)=B(1)*C(2)-B(2)*C(1)
      P(18)=B(1)*C(8)-C(1)*B(8)
      P(19)=B(4)*C(2)-C(4)*B(2)
      P(20)=B(1)*C(5)-C(1)*B(5)+B(7)*C(2)-C(7)*B(2)
      P(21)=B(7)*C(5)-C(7)*B(5)
      P(22)=B(4)*C(8)-C(4)*B(8)
      P(23)=B(4)*C(5)-C(4)*B(5)+B(10)*C(2)-C(10)*B(2)
      P(24)=B(1)*C(10)-C(1)*B(10)+B(7)*C(8)-C(7)*B(8)
      P(31)=B(4)*C(10)-C(4)*B(10)+B(10)*C(8)-C(10)*B(8)
      P(32)=B(7)*C(10)-C(7)*B(10)
      B(33)=B(10)*C(5)-B(5)*C(10)
      Q(1)=A(3)*C(2)-A(2)*C(3)
      Q(2)=A(3)*C(8)-A(8)*C(3)+A(6)*C(2)-A(2)*C(6)
      Q(3)=A(9)*C(2)-A(2)*C(9)
      Q(4)=A(3)*C(5)-A(5)*C(3)
      Q(5)=A(6)*C(8)-A(8)*C(6)
      Q(6)=C(2)*A(10)-A(2)*C(10)+C(8)*A(9)-A(8)*C(9)
      Q(7)=A(9)*C(5)-A(5)*C(9)
      Q(8)=C(5)*A(6)-A(5)*C(6)+C(10)*A(3)-A(10)*C(3)
      Q(25)=C(5)*A(10)-A(5)*C(10)+C(10)*A(9)-A(10)*C(9)
      Q(26)=C(8)*A(10)-A(8)*C(10)
      Q(27)=C(10)*A(6)-A(10)*C(6)
      Q(9)=A(1)*C(3)-C(1)*A(3)
      Q(10)=A(1)*C(6)-C(1)*A(6)
      Q(11)=A(1)*C(9)-C(1)*A(9)+A(4)*C(3)-C(4)*A(3)
      Q(12)=A(7)*C(3)-C(7)*A(3)
      Q(13)=A(4)*C(9)-C(4)*A(9)
      Q(14)=A(1)*C(10)-C(1)*A(10)-A(4)*C(6)-C(4)*A(6)
      Q(15)=A(7)*C(9)-C(7)*A(9)+A(10)*C(3)-C(10)*A(3)
      Q(16)=A(7)*C(6)-C(7)*A(6)
      Q(28)=A(7)*C(10)-C(7)*A(10)+A(10)*C(6)-C(10)*A(6)
      Q(29)=A(4)*C(10)-C(4)*A(10)
      Q(30)=A(10)*C(9)-C(10)*A(9)
      Q(17)=A(2)*C(1)-A(1)*C(2)
      Q(18)=A(8)*C(1)-A(1)*C(8)
      Q(19)=A(2)*C(4)-A(4)*C(2)
      Q(20)=A(2)*C(7)-C(2)*A(7)+A(5)*C(1)-C(5)*A(1)
      Q(21)=A(5)*C(7)-A(7)*C(5)
      Q(22)=A(8)*C(4)-A(4)*C(8)
      Q(23)=C(4)*A(5)-A(4)*C(5)+C(10)*A(2)-A(10)*C(2)
      Q(24)=C(1)*A(10)-A(1)*C(10)+C(7)*A(8)-A(7)*C(8)
      Q(31)=C(4)*A(10)-A(4)*C(10)+C(10)*A(8)-A(10)*C(8)
      Q(32)=C(7)*A(10)-A(7)*C(10)
      Q(33)=C(10)*A(5)-A(10)*C(5)
      R(1)=A(2)*B(3)-B(2)*A(3)
      R(2)=A(2)*B(6)-A(6)*B(2)+A(8)*B(3)-A(3)*B(8)
      R(3)=A(2)*B(9)-A(9)*B(2)
      R(4)=A(5)*B(3)-A(3)*B(5)
      R(5)=A(8)*B(6)-A(6)*B(8)
      R(6)=A(2)*B(10)-B(2)*A(10)+A(8)*B(9)-B(8)*A(9)
      R(7)=A(5)*B(9)-A(9)*B(5)
      R(8)=A(5)*B(6)-B(5)*A(6)+A(10)*B(3)-A(3)*B(10)
      R(25)=A(5)*B(10)-A(10)*B(5)+A(10)*B(9)-B(10)*A(9)
      R(26)=A(8)*B(10)-B(8)*A(10)
      R(27)=A(10)*B(6)-B(10)*A(6)
      R(9)=A(3)*B(1)-A(1)*B(3)
      R(10)=A(6)*B(1)-A(1)*B(6)
      R(11)=A(3)*B(4)-A(4)*B(3)+A(9)*B(1)-A(1)*B(9)
      R(12)=A(3)*B(7)-A(7)*B(3)
      R(13)=A(9)*B(4)-A(4)*B(9)
      R(14)=B(1)*A(10)-A(1)*B(10)+B(4)*A(6)-A(4)*B(6)
      R(15)=B(7)*A(9)-A(7)*B(9)+B(10)*A(3)-A(10)*B(3)
      R(16)=A(6)*B(7)-A(7)*B(6)
      R(28)=B(7)*A(10)-A(7)*B(10)+B(10)*A(6)-A(10)*B(6)
      R(29)=B(4)*A(10)-A(4)*B(10)
      R(30)=B(10)*A(9)-A(10)*B(9)
      R(17)=A(1)*B(2)-A(2)*B(1)
      R(18)=A(1)*B(8)-B(1)*A(8)
      R(19)=A(4)*B(2)-B(4)*A(2)
      R(20)=A(1)*B(5)-B(1)*A(5)+A(7)*B(2)-B(7)*A(2)
      R(21)=A(7)*B(5)-B(7)*A(5)
      R(22)=A(4)*B(8)-B(4)*A(8)
      R(23)=A(10)*B(2)-B(10)*A(2)+A(4)*B(5)-B(4)*A(5)
      R(24)=A(1)*B(10)-B(1)*A(10)+A(7)*B(8)-B(7)*A(8)
      R(31)=A(4)*B(10)-B(4)*A(10)+A(10)*B(8)-B(10)*A(8)
      R(32)=A(7)*B(10)-B(7)*A(10)
      R(33)=A(10)*B(5)-B(10)*A(5)
      NOPRNT=0
      NOPRNT=1
      IF(NOPRNT.EQ.0) GO TO 399
      WRITE(6,30)(MX(I),I=1,8)
      WRITE(6,100)
      WRITE(6,200)(A(I),I=1,10)
      WRITE(6,300)
      WRITE(6,200)(B(I),I=1,10)
      WRITE(6,400)
      WRITE(6,200)(C(I),I=1,10)
      WRITE(6,600)
      WRITE(6,700)(P(I),I=1,33)
      WRITE(6,800)
```

```
      WRITE(6,700)(Q(I),I=1,33)
      WRITE(6,900)
      WRITE(6,700)(R(I),I=1,33)
      WRITE(6,402)
      WRITE(6,401)(W(I),H(I),I=1,IPT)
   30 FORMAT(8I5)
  100 FORMAT(/10X,8HMATRIX-A)
  200 FORMAT(10(1PE13.5))
  300 FORMAT(/10X,8HMATRIX-B)
  400 FORMAT(/10X,8HMATRIX-C)
  401 FORMAT(2(1PE13.5))
  402 FORMAT(/20X,' GAUSS  QUDRATURE CONSTANTS ')
  600 FORMAT(/10X,8HMATRIX-P)
  700 FORMAT(8(1PE13.5))
  800 FORMAT(/10X,8HMATRIX-Q)
  900 FORMAT(/10X,8HMATRIX-R)
  399 CONTINUE
  350 CONTINUE
      RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE MCODE(MKODE,MX,L,MM)
      DIMENSION MKODE(1),MX(1)
      COMMON/BLK/MA(30),MB(30),MC(30),MD(30),MP(30),MQ(30),MR(30),MS(30)
      MX(1)=MA(L)
      MX(2)=MB(L)
      MX(3)=MC(L)
      MX(4)=MD(L)
      MX(5)=MP(L)
      MX(6)=MQ(L)
      MX(7)=MR(L)
      MX(8)=MS(L)
      N1=MM*(MA(L)-1)+1
      N2=MM*(MB(L)-1)+1
      N3=MM*(MC(L)-1)+1
      N4=MM*(MD(L)-1)+1
      N5=MM*(MP(L)-1)+1
      N6=MM*(MQ(L)-1)+1
      N7=MM*(MR(L)-1)+1
      N8=MM*(MS(L)-1)+1
      DO 20 I=1,MM
      J1=I
      J2=J1+MM
      J3=J2+MM
      J4=J3+MM
      J5=J4+MM
      J6=J5+MM
      J7=J6+MM
      J8=J7+MM
      MKODE(J1)=N1
      N1=N1+1
      MKODE(J2)=N2
      N2=N2+1
      MKODE(J3)=N3
      N3=N3+1
      MKODE(J4)=N4
      N4=N4+1
      MKODE(J5)=N5
      N5=N5+1
      MKODE(J6)=N6
      N6=N6+1
      MKODE(J7)=N7
      N7=N7+1
      MKODE(J8)=N8
      N8=N8+1
   20 CONTINUE
      RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE KSTACK(SK,C,KODE,MM,MKODE,N1,NN)
      DIMENSION MKODE(1),SK(1),C(N1,N1),KM(8)
      NX(I,J,N)=(J-1)*N-((J-1)*(J-2))/2+I-J+1
      DO 60 I=1,KODE
      J=MM*(I-1)+1
      K=MKODE(J)
   60 KM(I)=K
      DO 20 K=1,KODE
      DO 20 J=1,MM
      ND=MM*(K-1)+J
      NK=MKODE(ND)
      NA=NX(NK,NK,NN)
      DO 20 I=J,MM
      NC=MM*(K-1)+I
      SK(NA)=SK(NA)+C(NC,ND)
   20 NA=NA+1
      KZ=0
      KX=1
      DO 40 M=2,KODE
      KX=KX+1
      KZ=KZ+1
      DO 31 K=KX,KODE
      KY=K-KZ
      J1=KM(KY)
      J2=KM(K)
      IF(J2-J1)2,4,3
    2 N1=J1
      NK=J2-1
      GO TO 4
    3 N1=J2
```

```
      NK=J1-1
    4 DO 31 J=1,MM
      NK=NK+1
      NA=NX(NI,NK,NN)
      DO 31 I=1,MM
      IF(J1.GT.J2) GO TO 50
      ND=MM*(K-KX)+J
      NC=MM*(K-1)+I
      GO TO 101
   50 CONTINUE
      ND=MM*(K-1)+J
      NC=MM*(K-KX)+I
  101 CONTINUE
      SK(NA)=SK(NA)+C(NC,ND)
   31 NA=NA+1
   40 CONTINUE
      RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      SUBROUTINE GAUSS(IT,AA,M,N,L)
      COMMON /BLK1/A(10),B(10),C(10),P(33),Q(33),R(33),W(6),H(6),IPT
      EXTERNAL F
      AA=0.0
      DO 100 I=1,IPT
      X=H(I)
      DO 100 J=1,IPT
      Y=H(J)
      DO 100 K=1,IPT
      Z=H(K)
  100 AA=AA+W(I)*W(J)*W(K)*F(X,Y,Z,M,N,L,IT)
      RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      FUNCTION F(X,Y,Z,M,N,L,IT)
      COMMON /BLK2/H1(8),H2(8),H3(8),H4(8),H5(8),H6(8),H7(8),H8(8),
     *H9(8),H10(8),G1(8),G2(8),G3(8),G4(8),G5(8),G6(8),G7(8),G8(8)
      EXTERNAL PLAMDA,PMU,PHI,PJCOB
      G=(G1(M)+G2(M)*X+G3(M)*Y+G4(M)*Z+G5(M)*X*Y+G6(M)*Y*Z+G7(M)*Z*X+
     * G8(M)*X*Y*Z)/8.0
      D=(G1(N)+G2(N)*X+G3(N)*Y+G4(N)*Z+G5(N)*X*Y+G6(N)*Y*Z+G7(N)*Z*X+
     * G8(N)*X*Y*Z)/8.0
      E=(G1(L)+G2(L)*X+G3(L)*Y+G4(L)*Z+G5(L)*X*Y+G6(L)*Y*Z+G7(L)*Z*X+
     * G8(L)*X*Y*Z)/8.0
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,
     *170,180),IT
   10 F=PLAMDA(X,Y,Z,M)*PLAMDA(X,Y,Z,N)*PJCOB(X,Y,Z)
      GO TO 300
   20 F=PMU(X,Y,Z,M)*PMU(X,Y,Z,N)*PJCOB(X,Y,Z)
      GO TO 300
   30 F=PHI(X,Y,Z,M)*PHI(X,Y,Z,N)*PJCOB(X,Y,Z)
      GO TO 300
   40 F=PLAMDA(X,Y,Z,M)*PMU(X,Y,Z,N)*PJCOB(X,Y,Z)
      GO TO 300
   50 F=PMU(X,Y,Z,M)*PHI(X,Y,Z,N)*PJCOB(X,Y,Z)
      GO TO 300
   60 F=PLAMDA(X,Y,Z,M)*PHI(X,Y,Z,N)*PJCOB(X,Y,Z)
  300 CONTINUE
      F=F/64.
      RETURN
   70 F=PLAMDA(X,Y,Z,M)*0.125
      RETURN
   80 F=PMU(X,Y,Z,M)*0.125
      RETURN
   90 F=PHI(X,Y,Z,M)*0.125
      RETURN
  100 F=G/PJCOB(X,Y,Z)
      RETURN
  110 F=G*D/PJCOB(X,Y,Z)
      RETURN
  120 F= G*D*F/PJCOB(X,Y,Z)
      RETURN
  130 F= G*PLAMDA(X,Y,Z,N)*0.125
      RETURN
  140 F= G*PMU(X,Y,Z,N)*0.125
      RETURN
  150 F= G* PHI(X,Y,Z,N)*0.125
      RETURN
  160 F= G*D*PLAMDA(X,Y,Z,L)*0.125
      RETURN
  170 F= G*D*PMU(X,Y,Z,L)*0.125
      RETURN
  180 F= G*D* PHI(X,Y,Z,L)*0.125
      RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      FUNCTION PLAMDA(X,Y,7,N)
      COMMON /BLK1/A(10),B(10),C(10),P(33),Q(33),R(33),W(6),H(6),IPT
      COMMON /BLK2/H1(8),H2(8),H3(8),H4(8),H5(8),H6(8),H7(8),H8(8),
     *H9(8),H10(8),G1(8),G2(8),G3(8),G4(8),G5(8),G6(8),G7(8),G8(8)
      E1=(P(1)+P(2)*X+P(3)*Y+P(4)*Z+P(5)*X*X+P(6)*X*Y+P(7)*Y*Z+P(8)*Z*X)
      E4=(P(9)+P(10)*X+P(11)*Y+P(12)*Z+P(13)*Y*Y+P(14)*X*Y+P(15)*Y*Z+
     *P(16)*Z*X)
      E7=(P(17)+P(18)*X+P(19)*Y+P(20)*Z+P(21)*Z*Z+P(22)*X*Y+P(23)*Y*Z+
     *P(24)*Z*X)
      E1=E1+P(25)*X*Y*Z+P(26)*X*X*Y+P(27)*X*X*Z
      E4=E4+P(28)*X*Y*Z+P(29)*Y*Y*X+P(30)*Y*Y*Z
      E7=E7+P(31)*X*Y*Z+P(32)*Z*Z*X+P(33)*Z*Z*Y
      E10=E1*Y*Z+E4*Z*X+E7*X*Y
      E2=E1*Y
      E3=E1*Z
      E5=E4*Z
      E6=E4*X
      E8=E7*X
      E9=E7*Y
      PLAMDA=(H1(N)*E1+H2(N)*E2+H3(N)*E3+H4(N)*E4+H5(N)*E5+
     *H6(N)*E6+H7(N)*E7+H8(N)*E8+H9(N)*E9)
      PLAMDA=PLAMDA+H10(N)*E10
      RETURN

      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      FUNCTION PMU(X,Y,Z,N)
      COMMON /BLK1/A(10),B(10),C(10),P(33),Q(33),R(33),W(6),H(6),IPT
      COMMON /BLK2/H1(8),H2(8),H3(8),H4(8),H5(8),H6(8),H7(8),H8(8),
     *H9(8),H10(8),G1(8),G2(8),G3(8),G4(8),G5(8),G6(8),G7(8),G8(8)
      F1=(Q(1)+Q(2)*X+Q(3)*Y+Q(4)*Z+Q(5)*X*X+Q(6)*X*Y+Q(7)*Y*Z+Q(8)*Z*X)
      F4=(Q(9)+Q(10)*X+Q(11)*Y+Q(12)*Z+Q(13)*Y*Y+Q(14)*X*Y+Q(15)*Y*Z+
     *Q(16)*Z*X)
      F7=(Q(17)+Q(18)*X+Q(19)*Y+Q(20)*Z+Q(21)*Z*Z+Q(22)*X*Y+Q(23)*Y*Z+
     *Q(24)*Z*X)
      F1=F1+Q(25)*X*Y*Z+Q(26)*X*X*Y+Q(27)*X*X*Z
      F4=F4+Q(28)*X*Y*Z+Q(29)*Y*Y*X+Q(30)*Y*Y*Z
      F7=F7+Q(31)*X*Y*Z+Q(32)*Z*Z*X+Q(33)*Z*Z*Y
      F10=F1*Y*Z+F4*Z*X+F7*X*Y
      F2=F1*Y
      F3=F1*Z
      F5=F4*Z
      F6=F4*X
      F8=F7*X
      F9=F7*Y
      PMU=(H1(N)*F1+H2(N)*F2+H3(N)*F3+H4(N)*F4+H5(N)*F5+H6(N)*F6+
     *H7(N)*F7+H8(N)*F8+H9(N)*F9)
      PMU=PMU+H10(N)*F10
      RETURN

      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      FUNCTION PHI(X,Y,Z,N)
      COMMON /BLK1/A(10),B(10),C(10),P(33),Q(33),R(33),W(6),H(6),IPT
      COMMON /BLK2/H1(8),H2(8),H3(8),H4(8),H5(8),H6(8),H7(8),H8(8),
     *H9(8),H10(8),GA(8),GB(8),GC(8),GD(8),GE(8),GF(8),GH(8),GI(8)
      G1=(R(1)+R(2)*X+R(3)*Y+R(4)*Z+R(5)*X*X+R(6)*X*Y+R(7)*Y*Z+R(8)*Z*X)
      G4=(R(9)+R(10)*X+R(11)*Y+R(12)*Z+R(13)*Y*Y+R(14)*X*Y+R(15)*Y*Z+
     *R(16)*Z*X)
      G7=(R(17)+R(18)*X+R(19)*Y+R(20)*Z+R(21)*Z*Z+R(22)*X*Y+R(23)*Y*Z+
     *R(24)*Z*X)
      G1=G1+R(25)*X*Y*Z+R(26)*X*X*Y+R(27)*X*X*Z
      G4=G4+R(28)*X*Y*Z+R(29)*Y*Y*X+R(30)*Y*Y*Z
      G7=G7+R(31)*X*Y*Z+R(32)*Z*Z*X+R(33)*Z*Z*Y
      G10=G1*Y*Z+G4*Z*X+G7*X*Y
      G2=G1*Y
      G3=G1*Z
      G5=G4*Z
      G6=G4*X
      G8=G7*X
      G9=G7*Y
      PHI=(H1(N)*G1+H2(N)*G2+H3(N)*G3+H4(N)*G4+H5(N)*G5+H6(N)*G6+
     *H7(N)*G7+H8(N)*G8+H9(N)*G9)
      PHI=PHI+H10(N)*G10
      RETURN

      END
```

END OF COMPILATION: NO DIAGNOSTICS.

```
      FUNCTION PJCOB(X,Y,Z)
      COMMON /BLK1/A(10),B(10),C(10),P(33),Q(33),R(33),W(6),H(6),IPT
      COMMON /BLK2/H1(8),H2(8),H3(8),H4(8),H5(8),H6(8),H7(8),H8(8),
     *H9(8),H10(8),GA(8),GB(8),GC(8),GD(8),GE(8),GF(8),GH(8),GI(8)
      E1=(P(1)+P(2)*X+P(3)*Y+P(4)*Z+P(5)*X*X+P(6)*X*Y+P(7)*Y*Z+P(8)*Z*X)
      E1=E1+P(25)*X*Y*Z+P(26)*X*X*Y+P(27)*X*X*Z
      F1=(Q(1)+Q(2)*X+Q(3)*Y+Q(4)*Z+Q(5)*X*X+Q(6)*X*Y+Q(7)*Y*Z+Q(8)*Z*X)
      F1=F1+Q(25)*X*Y*Z+Q(26)*X*X*Y+Q(27)*X*X*Z
      G1=(R(1)+R(2)*X+R(3)*Y+R(4)*Z+R(5)*X*X+R(6)*X*Y+R(7)*Y*Z+R(8)*Z*X)
      G1=G1+R(25)*X*Y*Z+R(26)*X*X*Y+R(27)*X*X*Z
      D1=A(1)*E1+B(1)*F1+C(1)*G1
      D2=A(4)*E1+B(4)*F1+C(4)*G1
      D3=A(7)*E1+B(7)*F1+C(7)*G1
      D4=A(10)*E1+B(10)*F1+C(10)*G1
      XJCOB=D1+D2*Y+D3*Z
      XJCOB=XJCOB+D4*Y*Z
      PJCOB=1./XJCOB
      RETURN
      END
```

END OF COMPILATION: NO DIAGNOSTICS.