Bill Doherty

Structures and Materials Research

Department of Civil Engineering

Division of Structural Engineering

and

Structural Mechanics

Report No. 68 - 7

THERMOMECHANICAL ANALYSIS OF VISCOELASTIC SOLIDS

by

Robert L. Taylor

Associate Professor of Civil Engineering

Karl S. Pister

Professor of Civil Engineering

Gerald L. Goudreau

Graduate Student in Civil Engineering

Final Report

Lockheed Propulsion Company

Redlands, California

Purchase Order No. 65972-838-NIR

Structural Engineering Laboratory

University of California

Berkeley, California

June 1968

Acknowledgment

This work has been supported by Lockheed Propulsion Company Redlands, California under contract F04611-67-C-0100

with: Air Force Rocket Propulsion Laboratory

Research and Technology Division

Edwards, California 93523

Air Force Systems Command, United

States Air Force

1. Introduction

In observing the mechanical behavior of viscoelastic solids it is common experience that the stress at a particle depends on both the history of (localized) motion of the solid as well as the temperature history at the particle. Although a general theory of thermomechanical behavior of materials has been developed by Coleman, among others, the application to engineering practice seems remote at the present time. However, if limitations on the generality of the theory are introduced, it is possible to develop more specialized methods of characterizing thermomechanical behavior leading to computational techniques for boundary value problems. This report is concerned with such a procedure. A thermomechanical constitutive equation appropriate to viscoelastic solids undergoing small, quasi-static deformations is utilized, along with field equations for this class of deformation. The heat conduction equation is assumed to be unaffected by the deformation and is therefore solved separately, but simultaneously, with the mechanical field problem.

The report begins with a discussion of a mechanical constitutive equation for solids undergoing small deformations and subjected to temperature changes. A constitutive functional linear in deformation but nonlinear in temperature is adopted and various representations are discussed. Specialization appropriate to thermorheologically simple solids is indicated.

In Section 3 field equations are adjoined to the constitutive equation and the uncoupled, quasi-static boundary value problem for a class of viscoelastic solids is posed. A functional whose stationary value is equivalent to the indicated boundary value problem is stated in Section 4.

In Sections 5 and 6 a computational algorithm based upon the finite

element technique of discretizing the stationary value problem of a functional is described. As an application of the method a problem recently studied by Morland and Lockett [7] is examined. Agreement with their results, which depend on a method of more limited scope, is generally good.

A User's Manual and Program Listing are included as appendices.

2. Constitutive Equations for a Class of Viscoelastic Solids

Consider a body undergoing small deformations from an unstressed reference state and simultaneously subjected to temperature changes relative to the same reference state. The theory of simple materials postulates that the stress at a particle of the body is determined by the histories of deformation and temperature at the particle. In the present context in symbolic form* this is expressed by the equation

$$\underset{\sim}{\sigma}(\underset{\sim}{x},\, t) = \underset{\sim}{F}_{s=-\infty}^{s=t} [\underset{\sim}{\varepsilon}(\underset{\sim}{x},\, s),\, T(\underset{\sim}{x},\, s);\, \underset{\sim}{x},\, t] \qquad (2.1)$$

where $\underset{\sim}{\sigma}$, $\underset{\sim}{\varepsilon}$ are the stress and (small) strain tensors at the place $\underset{\sim}{x}$ at time t; T is the temperature at $\underset{\sim}{x}$ and $\underset{\sim}{F}$ is the thermomechanical response functional of the material of the body, i.e., the functional that assigns to every small strain history and temperature history the value of the stress tensor at $\underset{\sim}{x}$. Guided by experience with many engineering applications of viscoelastic solids we introduce the assumption that the thermomechanical response functional is linear in strain and nonlinear in temperature. With this in mind and with further restriction to homogeneous,

*For the present direct notation will be employed, i.e., symbols underlined with a tilde are tensors of order indicated by the context. Further, for the class of small deformations considered, no distinction between "particle" and "place" need be made.

non-aging materials we replace (2.1) by the hereditary integral representation*

$$\underset{\sim}{\sigma}(t) = \int_{\tau=-\infty}^{\tau=t} \underset{\sim}{C}_{s=0}^{s=t}[T(s);\ t-\tau]\frac{\partial}{\partial\tau}[\underset{\sim}{\varepsilon}(\tau) - \underset{\sim}{\theta}(\tau)]d\tau \qquad (2.2)$$

In (2.2) we have introduced the pseudo-temperature

$$\underset{\sim}{\theta}(\tau) = \int_{T_0}^{T} \underset{\sim}{\alpha}(T')dT' \qquad (2.3)$$

where $\underset{\sim}{\alpha}$ is the temperature-dependent thermal coefficient of expansion tensor and the kernel

$$\underset{\sim}{C} = \underset{\sim}{C}_{s=0}^{s=t}[T(s);\ t-\tau] \qquad (2.4)$$

is a fourth-rank relaxation modulus tensor whose value depends upon the temperature history of the material. For a <u>prescribed</u> temperature history the relaxation modulus reduces to

$$\underset{\sim}{C} = \underset{\sim}{C}[t;\ t-\tau] \qquad (2.5)$$

Thie form resembles the kernel of an aging linear viscoelastic solid and emphasizes the role of temperature history on viscoelastic material properties, i.e., temperature has an effect equivalent to "aging" of the material whose relaxation modulus is of the form of (2.4). Development of related computational algorithms is the subject of another report and will not be discussed further here [2]. Instead we first return to (2.4) and examine the case where the temperature is constant, but different from <u>the reference temperature</u> T_0.

*Dependence of field variables on $\underset{\sim}{x}$ is understood.

We then adopt the postulate for thermorheologically simple (TS) materials, i.e.,

$$\underset{\sim}{C}(t;\ T) = \underset{\sim}{C}[\xi(t);\ T_0] \tag{2.6}$$

where the reduced time ξ is defined by

$$\xi(t) = t\ \varphi(T) \tag{2.7}$$

and the temperature shift function $\varphi(T)$ is assumed to be an intrinsic material property normalized by the condition $\varphi(T_0) = 1$. Using (2.7) it is possible to compare the mechanical behavior of a material at different <u>constant</u> temperatures. To extend the idea of a temperature shift function to non-isothermal applications a further postulate is required. In the past, more from lack of contrary evidence than from experimental confirmation, it has been assumed that (2.4) could be replaced by

$$\underset{\sim}{C}_{s=0}^{t}[t - \tau;\ T(s)] = \underset{\sim}{C}[\xi(t) - \xi(\tau);\ T_0] \tag{2.8}$$

where the reduced time is now defined by

$$\xi(\tau) = \int_0^{\tau} \varphi[T(s)]ds \tag{2.9}$$

For constant temperature (2.9) clearly reduces to (2.6). Equation (2.9) is only one of many postulates that might be used to extend the notion of a TS material to non-isothermal cases. For example, we might assume that the reduced time depends on both temperature and temperature rate histories, i.e.,

$$\xi^*(\tau) = \int_0^{\tau} f\left[T(s),\ \frac{\partial T}{\partial s}\right]ds \tag{2.10}$$

where

$$\underset{\sim}{\varepsilon}^{(1)} \equiv \frac{\partial \underset{\sim}{\varepsilon}}{\partial \tau} \quad , \quad \text{etc.},$$

and the symbol ($\otimes$) denotes the integral of the composition of the adjoined tensors. In (2.14) since $\underset{\sim}{C}$ is of rank four and $\underset{\sim}{\varepsilon}$, $\underset{\sim}{\theta}$ of rank two, the symbol in (2.15) denotes the integral of a doubly-contracted composition whose value is a second rank stress tensor. In component form (2.15) is

$$\sigma_{ij}(t) = \int_{-\infty}^{t} C_{ij}^{kl}[\xi(t) - \xi(\tau); T_0]\frac{\partial}{\partial \tau}[\varepsilon_{kl}(\tau) - \theta_{kl}(\tau)]d\tau$$

3. Formulation of the Boundary Value Problem

By a thermomechanical boundary value problem for a viscoelastic solid we understand the following: a mechanically linear, TS material undergoing quasistatic deformation and subjected to an independently determined temperature field satisfies the equilibrium equations

$$\begin{aligned} \underset{\sim}{\nabla} \cdot \underset{\sim}{\sigma} + \underset{\sim}{f} &= \underset{\sim}{0}, \\ \underset{\sim}{\sigma} &= \underset{\sim}{\sigma}^{T}, \end{aligned} \tag{3.1}$$

where $\underset{\sim}{f}$ is a prescribed body force vector,

the strain-displacement equations

$$2\underset{\sim}{\varepsilon} = [\nabla \underset{\sim}{u} + (\nabla \underset{\sim}{u})^{T}], \tag{3.2}$$

where $\underset{\sim}{u}$ is the displacement vector,

and constitutive equations

$$\underset{\sim}{\sigma} = \underset{\sim}{C} \otimes (\underset{\sim}{\varepsilon}^{(1)} - \underset{\sim}{\theta}^{(1)}) \tag{3.3}$$

in a region of space R occupied by the body. To these equations are adjoined the following boundary conditions:

$$\begin{aligned} \underset{\sim}{T}(\underset{\sim}{x}, t) &= \underset{\sim}{n} \cdot \underset{\sim}{\sigma} = \overline{\underset{\sim}{T}} \quad \text{on } S_\sigma \\ \underset{\sim}{u}(\underset{\sim}{x}, t) &= \overline{\underset{\sim}{u}} \quad \text{on } S_u \end{aligned} \tag{3.4}$$

In (3.4) $\underset{\sim}{n}$ is the outward unit vector normal to the boundary surface of the body, and $\overline{\underset{\sim}{T}}$, $\overline{\underset{\sim}{u}}$ are prescribed values of the surface traction vector and displacement vector on complementary parts of the boundary of the body, S_σ and S_u, respectively. The temperature of the body is assumed to be a prescribed function of position and time.

A direct computational method for attacking the boundary value problem follows in the next sections.

4. A Variational Theorem

For computational purposes it is expedient to recast the boundary value problem posed in Section 3 as a stationary value problem for a functional. Accordingly, we define a thermomechanical state functional $V\{\underset{\sim}{u}\}$ through the equation

$$\begin{aligned} V\{\underset{\sim}{u}\} = \int_R \left[\frac{1}{2} \underset{\sim}{C} \otimes \underset{\sim}{\varepsilon} * \underset{\sim}{\varepsilon} - \underset{\sim}{C} \otimes \underset{\sim}{\theta} * \underset{\sim}{\varepsilon} - h * \underset{\sim}{f} * \underset{\sim}{u} \right] dv \\ - \int_{S_\sigma} h * \overline{\underset{\sim}{T}} * \underset{\sim}{u} \, ds \end{aligned} \tag{4.1}$$

In (4.1) h is the Heaviside step function defined as unity for $t > 0$ and zero for $t < 0$; the star symbol $(*)$ denotes the convolution of two functions in the sense

$$\underset{\sim}{f} * \underset{\sim}{g} = \int_{-\infty}^{t} \underset{\sim}{f}(t - \tau) \, \underset{\sim}{g}(\tau) \, d\tau \tag{4.2}$$

We assume that the body is undisturbed over the interval $-\infty < \tau < 0^+$; consequently an explicit statement of initial conditions is not required.

We define an admissible thermomechanical state associated with the functional $V\{\underset{\sim}{u}\}$ as follows:

(1) the (symmetric) stress tensor is determined by the constitutive equation (3.3)

(2) the strain-displacement equations (3.2) are satisfied

(3) the displacement vector satisfies $(3.4)_2$

(4) the pseudo temperature $\underset{\sim}{\theta}$ is a prescribed function of place and time associated with a solution of the heat conduction boundary value problem for the body.

We now state the variational theorem: Among all admissible thermomechanical states, that which satisfies the equilibrium equations (3.1), and stress boundary conditions $(3.4)_2$ is given by

$$\delta V = 0 \tag{4.3}$$

Executing the variation of (4.1), using the Divergence Theorem and (3.3) leads to

$$\begin{aligned} \delta V = -\int_R \{h * [\underset{\sim}{\nabla} \cdot (\underset{\sim}{C} \otimes \underset{\sim}{\varepsilon}^{(1)} - \underset{\sim}{C} \otimes \underset{\sim}{\theta}^{(1)}) + \underset{\sim}{f}] * \delta\underset{\sim}{u}\} dv \\ + \int_{S_\sigma} [h * (\underset{\sim}{T} - \overline{\underset{\sim}{T}}) * \delta\underset{\sim}{u}] ds = 0 \end{aligned} \tag{4.4}$$

Application of a corollary of Titchmarsh's Theorem [12], (i.e., $f * g = 0$ implies either $f = 0$ or $g = 0$) and using the constitutive equation (3.3) in the volume integral yields the equilibrium equation (3.1) and stress boundary condition $(3.4)_1$.

In the sequel, along with (4.4) we will adopt a variational theorem for obtaining solutions of the heat conduction equation presented by Wilson and Nickell [13]. The finite element computer algorithm developed therein, along with the algorithm for the thermomechanical problem to be developed in Sections 5 and 6 form the basis of the computational work reported here.

5. Specialization for Axisymmetric Deformation of Isotropic Viscoelastic Solids--Finite Element Solution

In this section the previous results are specialized for a particular class of problems for isotropic solids. For axisymmetric solids subjected to axisymmetric loads (both mechanical and thermal) response occurs in the r, z plane (r, θ, z-coordinates) only, hence u_θ is zero and u_r, u_z are functions of r and z only. From (3.2) the non zero strains are (in terms of physical components)

$$\begin{aligned} \varepsilon_{rr} &= \frac{\partial u_r}{\partial r} \\ \varepsilon_{\theta\theta} &= \frac{u_r}{r} \\ \varepsilon_{zz} &= \frac{\partial u_z}{\partial z} \\ \varepsilon_{rz} &= \frac{1}{2}\left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r}\right) \end{aligned} \tag{5.1}$$

For isotropic thermomechanical response both $\underset{\sim}{C}$ and $\underset{\sim}{\theta}$ appearing in (3.3) are isotropic functions and the following constitutive equations result:

$$\begin{aligned} \sigma_{rr} &= \tfrac{1}{3}(3K - 2G) \otimes \varepsilon^{(1)} + 2G \otimes \varepsilon_{rr}{}^{(1)} - K \otimes \theta \\ \sigma_{\theta\theta} &= \tfrac{1}{3}(3K - 2G) \otimes \varepsilon^{(1)} + 2G \otimes \varepsilon_{\theta\theta}{}^{(1)} - K \otimes \theta \end{aligned} \tag{5.2}$$

$$\sigma_{zz} = \frac{1}{3}(3K - 2G) \circledast \varepsilon^{(1)} + 2G \circledast \varepsilon_{zz}^{(1)} - K \circledast \theta$$

$$\sigma_{rz} = 2G \circledast \varepsilon_{rz}^{(1)}$$

where

$$\varepsilon = \varepsilon_{rr} + \varepsilon_{\theta\theta} + \varepsilon_{zz} \tag{5.3}$$

and

$$\theta = \int_{T_0}^{T} \alpha(T')\, dt'$$

and the material functions K, G, have the form (2.8). Equation (4.1) can now be more conveniently written*

$$V = \int_R \left[\frac{1}{2} A_{ij} \circledast S_i * S_j - \theta_i * S_i - h * f_\alpha * u_\alpha\right] r\, dr\, dz - \int_{S_r} h * \bar{T}_\alpha * u_\alpha\, r\, dr\, dz \tag{4.1}_a$$

where

$$S_i = \left(\frac{u_r}{r};\ \frac{\partial u_r}{\partial r};\ \frac{\partial u_z}{\partial r};\ \frac{\partial u_r}{\partial z}\right) . \tag{5.4}$$

A_{ij} is a 5 x 5 symmetric array whose non-zero components are

$$A_{11} = A_{22} = A_{33} = \frac{1}{3}(3K + 4G)$$

$$A_{12} = A_{13} = A_{23} = \frac{1}{3}(3K - 2G)$$

$$A_{44} = A_{55} = A_{45} = G \tag{5.5}$$

---

*Latin indices range from 1 to 5 while Greek indices range over 1, 2. Summation convention is implied.

θ_i denotes thermal terms whose non-zero components are

$$\theta_1 = \theta_2 = \theta_3 = K \circledast \theta$$

$$u_\alpha = (u_r, u_z)$$

$$f_\alpha = (f_r, f_z)$$

and

$$\bar{T}_\alpha = (\bar{T}_r, \bar{T}_z)$$

A Ritz-type solution to (4.1a) may be obtained by a finite element method. To this end, the volume and surface integrals are expressed as a sum of integrals over a set of subregions (finite elements) defining R. Assumed solutions are taken for each element in such a way that displacement continuity is maintained between contiguous elements. In the present development triangular elements are used together with a linear expansion of the displacement field in each element.

Use of standard finite element procedures for spatial discretization [15] and application of the first variation of V yields a set of 2N linear integral equations in terms of the 2N nodal displacements (N equals the number of nodes). These may be expressed as:

$$\int_{-\infty}^{t} K_{mn}(\xi - \xi') \frac{\partial u_n}{\partial t'} \, dt' = R_m(t) \tag{5.6}$$

where K_{mn} is an assemblage of element stiffness relaxation functions for the body; similarly R_m is an assemblage of surface and body loads. The solution of (5.6) yields the nodal point displacement history. The strain and stress histories can then be computed from (5.1) and (5.2).

6. Solution of Simultaneous Integral Equations

The approximate spatial reduction by a finite element method leads to a set of simultaneous integral equations. In the absence of variable temperature history (5.6) reduces to a set of Volterra integral equations of the second kind which theoretically can be solved by integral transform methods. However, with variable temperature history these equations are no longer tractable by transform techniques. Therefore, in order to solve (5.6) direct numerical methods will be employed. A standard numerical technique for solving this class of equations is a step-forward integration procedure. In connection with viscoelastic analyses a finite difference technique has been used to solve the convolution inter-relationship between creep and relaxation [3, 5, 10]. Stress analyses have also been performed utilizing a finite difference numerical step-forward integration procedure (e.g., see [1, 11, 14]). The numerical integration procedure consists in expansions into a series of time increments where integrations are performed over each increment according to some difference approximation. The great disadvantage of this method (in connection with computer applications) is that all past solutions are required. Thus, in the case of (5.6) extensive amounts of information are required to obtain solutions over extended time periods. Also, considerable computer time is required (it is to be anticipated that a viscoelastic analysis will require a considerable increase in computer time over that of a similar elastic analysis).

Recently some modifications to the above procedure have been proposed. One is based upon the premise of a finite memory in the material, hence, the solution at any time involves only knowledge of a limited history of the past deformations [9]. Thus it is necessary to retain only a

finite number of past solutions to obtain a solution at any time. This modification achieves a considerable saving in computation time. However, for large numbers of nodes, the finite element reduction to (5.6) results in computation times which are generally still prohibitive.

A second alternative which is used herein, is to represent the material property functions assuming that the kernel functions of the integral equations are degenerate, that is we assume each coefficient in (5.6) has the property

$$K(\xi - \xi') = \sum_{i=1}^{I} K_i \, f_i(\xi) \, g_i(\xi') \tag{6.1}$$

Generalized Maxwell materials have often been used to approximate the viscoelastic response of real materials. When the stress-strain equations are expressed in integral form the kernel function then has the series representation

$$G(\xi) = \sum_{i=1}^{I} G_i \, e^{-\xi/\lambda_i} + G_0 \tag{6.2}$$

where G_i are constants associated with instantaneous response, λ_i are constants associated with a discrete relaxation spectrum (each λ_i can be called a relaxation time) and I is the number of Maxwell elements used to approximate the material's relaxation modulus. From (6.2) it follows that

$$G(\xi - \xi') = \sum_{i=1}^{I} G_i \, e^{-\xi/\lambda_i} \, e^{\xi'/\lambda_i} + G_0 \tag{6.3}$$

The above decomposition has previously been used in connection with a finite difference integration technique for approximate solution of

viscoelasticity problems [1, 14]. The method is used herein as a finite element concept for the approximate solution of viscoelastic problems which include temperature effects through the thermorheologically simple postulate. The solution technique is similar to that used in [8, 13]. In order to discuss the solution technique we consider the single integral equation

$$G(\xi)u(0) + \int_0^t G(\xi - \xi') \frac{\partial u}{\partial t'} dt' = r(t) \tag{6.4}$$

Equation (6.4) may be considered as a typical term in (5.6), consequently, any conclusions obtained from (6.4) are directly applicable to the solution of (5.6). In (6.4) $G(\xi)$ represents a particular relaxation modulus function (i.e. shear or bulk), $r(t)$ is a known forcing function and $u(t)$ is the sought solution. Substituting into (6.4) the material property representation given by (6.3) we obtain

$$\begin{aligned} G_0 \int_0^t \frac{\partial u}{\partial t'} dt' + \sum_{i=1}^{I} G_i e^{-\xi/\lambda_i} \int_0^t e^{\xi'/\lambda_i} \frac{\partial u}{\partial t'} dt' \\ = r(t) - u(0)\left[G_0 + \sum_{i=1}^{I} G_i e^{-\xi/\lambda_i}\right] \end{aligned} \tag{6.5}$$

As in the spatial reduction by a finite element method, (6.5) may be discretized by piecewise expansions in time of the dependent variable $u(t)$. A continuous time response may be obtained by assuming a polynomial time expansion and matching nodal displacements between each succeeding time expansion. The simplest expansion is given by the linear Lagrangian interpolation function

$$u(t) = \frac{1}{\Delta t_n}\left[(t_n - t)u_{n-1} + (t - t_{n-1})u_n\right]; \quad t_{n-1} \le t \le t_n \tag{6.6}$$

where

$$u_n = u(t_n)$$

$$\Delta t_n = t_n - t_{n-1}$$

Consequently the time derivative of u during each time increment is constant and is expressed by

$$\frac{\partial u}{\partial t} = \frac{u_n - u_{n-1}}{\Delta t_n} = \frac{\Delta u_n}{\Delta t_n} ; \quad t_{n-1} \leq t \leq t_n \tag{6.8}$$

If we introduce the notation*

$$h_i(\Delta t_j) = \frac{1}{\Delta t_j} \int_{t_{j-1}}^{t_j} e^{-(\xi_j - \xi')/\lambda_i} \, dt' = \frac{1}{\Delta t_j} \int_0^{\Delta t_j} e^{-\xi'/\lambda_i} \, dt' \tag{6.9}$$

then at time t_n an approximate solution to (6.4) is given by

$$\sum_{j=1}^{n} \left[G_0 + \sum_{i=1}^{I} G_i \, e^{-(\xi_n - \xi_j)/\lambda_i} \, h_i(\Delta t_j) \right] \Delta u_j = r(t_n) - \left(G_0 + \sum_{i=1}^{I} G_i \, e^{-\xi_n/\lambda_i} \right) u_0 \tag{6.10}$$

where $t_0 = 0$; all other t_j represent previous or present discrete solution points and ξ_n is the value of the reduced time at the present real time t_n. It is possible to rewrite (6.10) such that each new solution may be computed directly from the previous solution. To this end we let

$$g_i(t_n) = G_i \left[e^{-\xi_n/\lambda_i} \, u_0 + \sum_{j=1}^{n-1} e^{-(\xi_n - \xi_j)/\lambda_i} \, h_i(\Delta t_j) \Delta u_j \right] \tag{6.11}$$

*It should be noted that $h_i(0) = 1$, hence, instantaneous loading and unloading may be considered by setting Δt to zero.

and note that a recursion formula may be deduced as

$$g_i(t_n) = e^{-\Delta\xi_n/\lambda_i}\left[g_i(t_{n-1}) + G_i h_i(\Delta t_{n-1})\Delta u_{n-1}\right]; \quad n \geq 1 \qquad (6.12)$$

where $g_i(t_0) = 0$, $\Delta u_0 = u_0$, and $\Delta\xi_n = \xi_n - \xi_{n-1}$

Equation (6.10) may now be written as

$$\left[G_0 + \sum_{i=1}^{I} G_i h_i(\Delta t_n)\right]\Delta u_n = r(t_n) - G_0 u_{n-1} - \sum_{i=1}^{I} g_i(t_n) \qquad (6.13)$$

For a single integral equation (6.13) is an efficient solution algorithm for both short and long duration loads. The solution effort at each discrete time is proportional to the number of Maxwell elements used in the material characterization, whereas, in previous developments (e.g., see [5]) the solution effort was proportional to the number of previous solution points. In connection with a finite element method of spatial discretization (6.13) may be applied to each term of (5.6) separately and the resulting simultaneous linear algebraic equations may be solved by standard techniques (e.g., Gauss elimination is used in the program listed in the appendices).

The discretization errors involved in the above process are related to the order of the time interpolation polynomials. It is possible to increase the discretization error by using poor approximations to (6.9). The evaluation of (6.9) in closed form is in general not possible for non-uniform temperature states. An approximation may be obtained by some numerical integration or other approximation which will allow a closed form evaluation. If it is assumed, as in [6], that ξ is linear in time between t_{j-1} and t_j, which corresponds to the assumption of constant

temperature in the time interval, the integrals (6.9) may be evaluated, yielding

$$h_i(\Delta t_j) \doteq \lambda_i(1 - e^{-\Delta\xi_j/\lambda_i})/\Delta\xi_j \tag{6.14}$$

An alternative to (6.14) is the simple trapezoidal integration procedure. This method has previously been used with a finite difference approximation to effect solutions of integral equations [1, 14]. In this procedure each integral is approximated by

$$h_i(\Delta t_j) \doteq \frac{1}{2}(1 + e^{-\Delta\xi j/\lambda_i}) \tag{6.15}$$

For isothermal problems (6.14) represents an exact evaluation to (6.9) while (6.15) is in all cases an approximation (see Fig. 1). By considering the solution to a single integral equation we can illustrate that for a given time discretization careful evaluation of (6.9) is much more crucial in controlling the numerical error than the approximation of the dependent variable. It should be noted at the outset that our argument is based upon the fact that material characterization uses relaxation modulus functions together with a displacement method. Thus, in (6.4) if we consider a unit step forcing function, $u(t)$ is the creep compliance. For arbitrary inputs $u(t)$ will have time variations that are related to the time characteristics of the creep compliance. Consequently the numerical determination of the creep compliance serves as a check on the accuracy of (6.13). From the interrelationship between the relaxation modulus, $G(t)$, and the creep compliance, $J(t)$, it may be shown that the retardation time, τ is always greater than the relaxation time, λ, by the amount

$$\tau = \frac{G_0}{G_\infty}\lambda = \frac{J_\infty}{J_0}\lambda \tag{6.17}$$

where G_0, J_0 are initial values and G_∞, J_∞ are final values of the relaxation and creep functions, respectively (see Fig. 2). Consequently for relaxation moduli with low relative equilibrium values the creep compliance and consequently u(t) will have significant time effects long after the relaxation modulus has reached a near equilibrium value. To illustrate this effect we consider the relaxation modulus

$$G(t) = 0.75 \times 10^7 + 8.2925 \times 10^9\, e^{-t/2} \tag{6.18}$$

where the known solution is

$$u(t) = J(t) = \frac{4}{3} \times 10^{-7}\left[1 - \frac{8.2925}{8.3}\, e^{-\frac{3t}{6640}}\right]$$

The retardation time is 6640/3 and the relaxation time is 2. The numerical solution for J(t) using exact and trapezoidal integrals to represent h_i is shown in Table 1. The necessity of accurate evaluation of (6.9) is clearly illustrated by the results in Table 1. The trapezoidal integration scheme depends upon accurate estimates of the relaxation modulus integrals; these are obviously related to the λ_i and Δt as seen in Fig. 1, whereas if (6.14) is used for the relaxation modulus the solution increment is primarily related to the retardation times, τ_i and these are always larger than the λ_i. Consequently extensive reduction in computational effort is possible by using approximations to (6.9) which are accurate for large increments of time. This is especially important for problems involving change in temperature since the reduced time increment may be several orders of magnitude greater than the real time increment. A consequence of poor approximation to (6.9) is illustrated in the next section.

Creep Compliance - $J(t) \times 10^9$

t \ Δt	2		200		1000	
	Trapezoidal	Exact	Trapezoidal	Exact	Trapezoidal	Exact
0	.12	.12	.12	.12	.12	.12
200	10.79	11.63	.36	11.17		
400	20.61	22.14	.60	21.30		
600	29.64	31.75	.84	30.59		
800	37.95	40.53	1.08	39.12		
1000	45.59	48.54	1.32	46.93	.36	41.60
1200	52.62	55.87	1.56	54.10		
1400	59.09	62.56	1.80	60.67		
1600	65.04	68.68	2.03	66.70		
1800	70.51	74.26	2.27	72.22		
2000	75.54	79.37	2.51	77.29	.60	70.17
% Error @2000	5%	~ 0%	97%	2.5%	99+%	11.5%

Table 1 - Creep Compliance Computation Comparison for Exact and Trapezoidal Integration of Equation (6.9).

7. Numerical Example

As an application of the present development the thermal stress analysis of a thin-walled infinite cylinder under conditions of plane strain with a time dependent boundary temperature was investigated. This problem was considered by Lockett and Morland [7]. Previous closed form solutions of thermorheologically simple problems by integral transform methods have been limited to one dimensional slabs and spheres where symmetry was used to uncouple the single integral law in reduced time from the remaining field equations in real time. More general axisymmetric geometries are not amenable to this method of solution. Lockett and Morland have shown that for thin-walled cylinders, a perturbation scheme in the thinness parameter permits a similar uncoupling at each stage of the solution.

The numerical example presented in [7] shows only the first order solution, which the authors state should be valid for a sufficiently thin cylinder. In general this problem admits four characteristic times, that of the applied boundary temperature, the diffusion time of heat transfer, the relaxation time of the viscoelastic material and the time of the solution (retardation time). Since the primary purpose of the example was to show the effect of temperature-dependent material properties, the diffusion time was assumed to be negligible. As illustrated in the last section the retardation time is even more significant than the relaxation time for numerical applications.

For a time-dependent inner boundary temperature and a prescribed zero outer boundary temperature, the first order steady state temperature solution is linear and assumed to be achieved instantaneously. Using the notation of [7] with $\bar{\theta}$, ρ, and x normalized temperature, time, and

distance respectively, the applied boundary temperature is taken as

$$\bar{\theta}(0, \rho) = 1 - e^{-2\rho} \tag{7.1}$$

Consequently, the instantaneous steady state temperature is given by

$$\bar{\theta}(x, \rho) = (1 - x)(1 - e^{-2\rho}) \tag{7.2}$$

The shift function, based on data for polymethylmethacrylate is taken as

$$\varphi(x, \rho) = 3981.1\, e^{-6.2172(1-\bar{\theta})(1.333+\bar{\theta}+1.095\bar{\theta}^2)} \quad , \tag{7.3}$$

where the strong dependence on temperature is seen from the following table

$\bar{\theta}$	φ	$\rho(x=0)$
0.0	1.0	0
0.4	3.22	0.25
0.6	12.2	0.45
0.8	118.	0.8
0.9	572.	1.15
1.0	3981.	2.5

(7.4)

The approximate times at which the temperature is attained at the inner boundary is indicated in the third column.

The material moduli are characterized by an elastic bulk modulus and a standard solid shear relaxation time:

$$K = 2.50 \times 10^{10}$$

$$G(\xi) = 0.75 \times 10^{7} + 829.25 \times 10^{7}\, e^{-\xi} \tag{7.5}$$

In [7] the numerical solutions were obtained by using finite difference methods for both the spatial and time variables. The authors state that it was necessary to use 40 spatial points and 100 time points to attain a solution with an overall estimated error of two per cent. A finite element solution based upon the present development was obtained using only twelve nodal points (11 elements as shown at the top of Fig. 3) and 40 time points. The results of both analyses are reproduced in Figs. 3 and 4. In Fig. 3 a plot of the normalized hoop stress is shown for three times. Both analyses compare very favorably at $\rho = 0.5$. This corresponds to a normalized temperature of about 0.6 and a reduced time increment of about 25 times the real time increment. Based upon results from the previous section and the time increment used, we estimated that trapezoidal integration should be adequate. On the other hand for $\rho = 0.79$ and 1.26 the reduced time increments are about 110 and 600 times the real time increment respectively. For these time increments estimates show that the trapezoidal integration should lag behind the expected solution. The estimates are confirmed by the results obtained in [7]. It is physically impossible to obtain their response as shown near the inner radius (i.e., for small x). The stresses at the high temperature will decrease faster than at lower temperatures; consequently, due to the discretization error the results of [7] are erroneous. To verify this conclusion the solution was obtained for temperature independent properties and compared to the results of [7]. Here excellent agreement has been obtained as seen in Fig. 4. The results for the temperature dependent solution are also replotted in Fig. 4.

The above example serves to illustrate further the superiority of the time discretization presented herein since accurate estimates to the

response time can be estimated beforehand and used for the selection of the time solution points.

References

[1] Chang, T. Y., "Approximate Solutions in Linear Viscoelasticity," Structural Engineering Report 66-8, University of California, July 1966.

[2] Dong, R. G., Pister, K. S. and Dunham, R. S., "Mechanical Characterization of Nonlinear Viscoelastic Solids for Iterative Solution of Boundary Value Problems," (in preparation).

[3] Hopkins, I. L. and Hamming, R. W., "On Creep and Relaxation," J. Appl. Phys., v 28, p. 906, 1957.

[4] Huang, N. C., Lee, E. H., and Rogers, T. G., "On the Influence of Viscoelastic Compressibility in Stress Analysis," Proc 4th Int. Cong. of Rheology, Pt 2, 1963.

[5] Lee, E. H. and Rogers, T. G., "Solution of Viscoelastic Stress Analysis Problems Using Measured Creep or Relaxation Functions," J. Appl. Mech., v 30, p. 1277, 1963.

[6] Lianis, G. and Valanis, K. C., "The Linearization of the Problem of Transient Thermoviscoelastic Stresses," Tech. Rept. A & ES 62-2, Purdue University, April 1962.

[7] Lockett, F. J. and Morland, L. W., "Thermal Stresses in a Viscoelastic Thin-Walled Tube with Temperature Dependent Properties," Int. J. of Eng. Sci., v 5, n 12, p. 879, Dec. 1967.

[8] Nickell, R. E. and Sackman, J. L., "The Extended Ritz Method Applied to Transient, Coupled Thermoelastic Boundary-Valve Problems," SEL Report 67-3, University of California, February 1967.

[9] Selna, L. G., "Time-Dependent Behavior of Reinforced Concrete Structures," SEL Report 67-19 University of California, August 1967.

[10] Taylor, R. L., "Note on Creep and Relaxation," AIAA J., v 2, pp. 1659, 1964.

[11] Taylor, R. L. and Chang, T. Y., "An Approximate Method for Thermo-viscoelastic Stress Analysis," Nucl. Eng. and Des., v 4, p. 21, 1966.

[12] Titchmarsh, E. C., "The Zeros of Certain Integral Functions," Proc. London Math. Soc., v 25, 4, p. 283, 1926.

[13] Wilson, E. L. and Nickell, R. E., "Application of the Finite Element Method to Heat Conduction Analysis," Nucl.Eng. and Des., v 4, 3, p. 276, 1966.

[14] Zak, A. R., "Structural Analysis of Realistic Solid Propellant Materials," Tech. Rept. AAE 67-4, University of Illinois, May 1967.

[15] Zienkiewicz, O. C. and Cheung, Y. K., "The Finite Element Method in Structural and Continuum Mechanics," McGraw Hill Book Co., London, 1967.

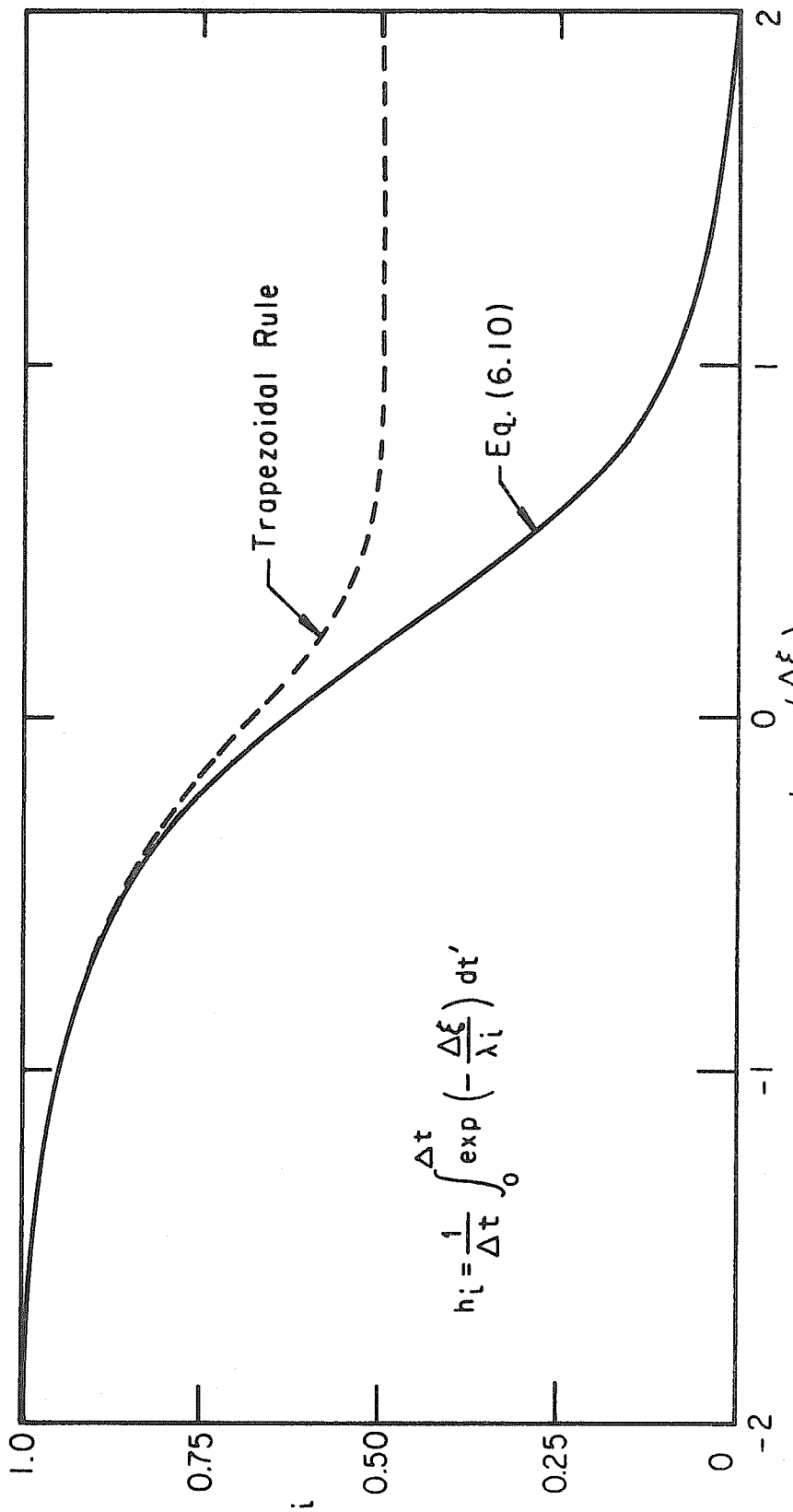

FIG. 1 EVALUATION OF INTEGRALS EQ. (6.10) vs TRAPEZOIDAL RULE

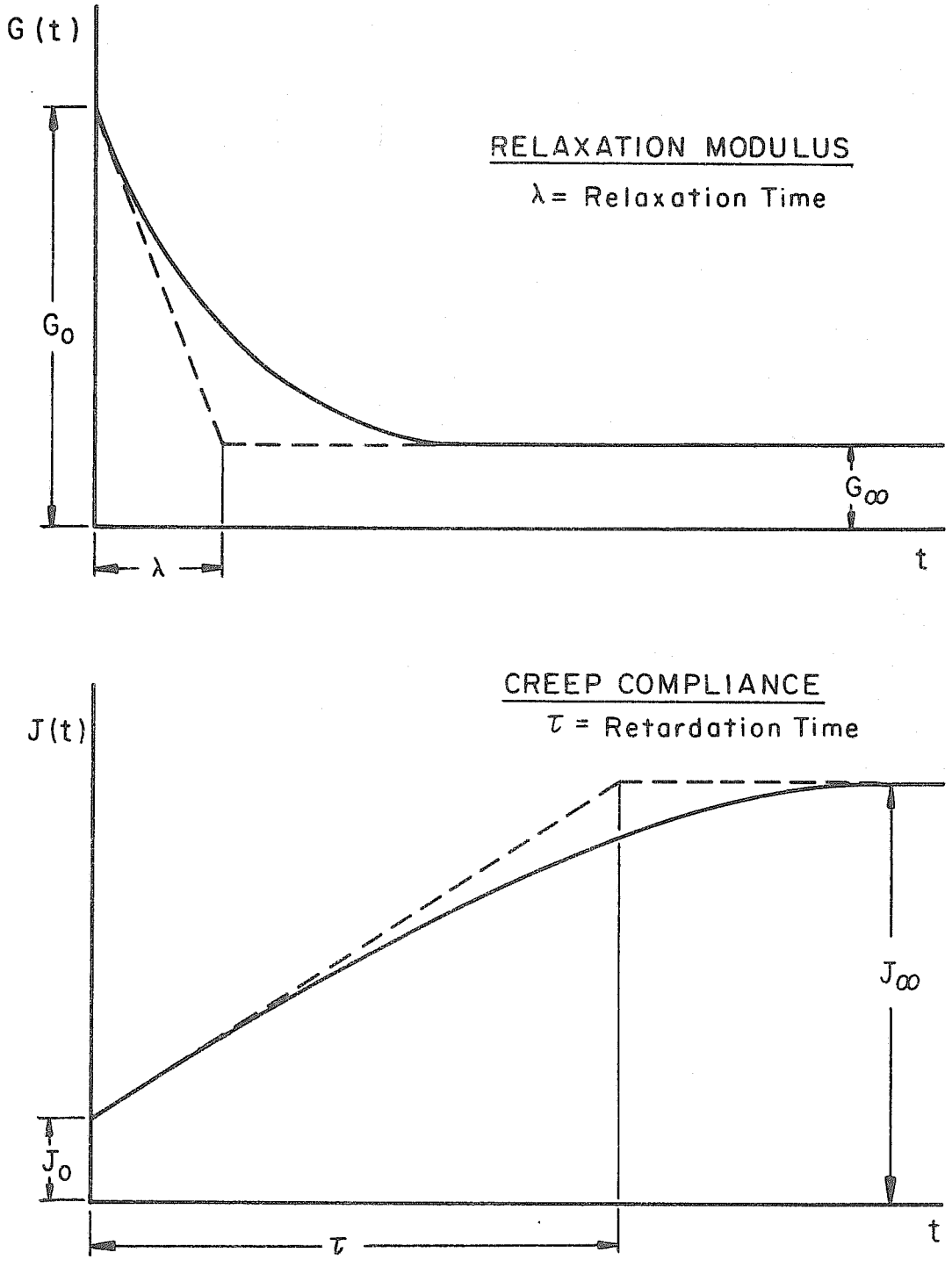

FIG. 2 COMPARISON OF RELAXATION AND RETARDATION TIMES

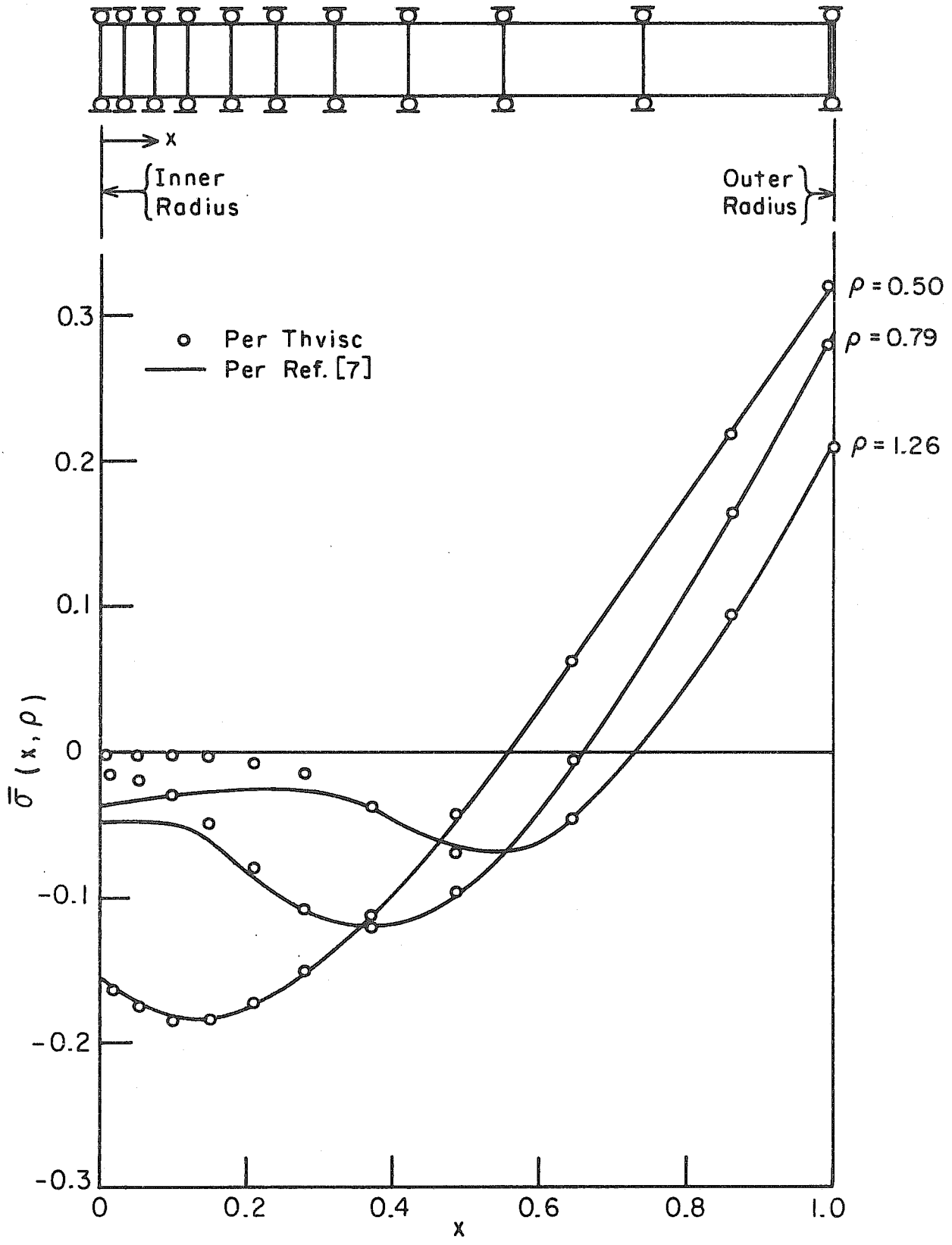

FIG. 3 HOOP STRESS vs TIME, TEMPERATURE-DEPENDENT PROPERTIES

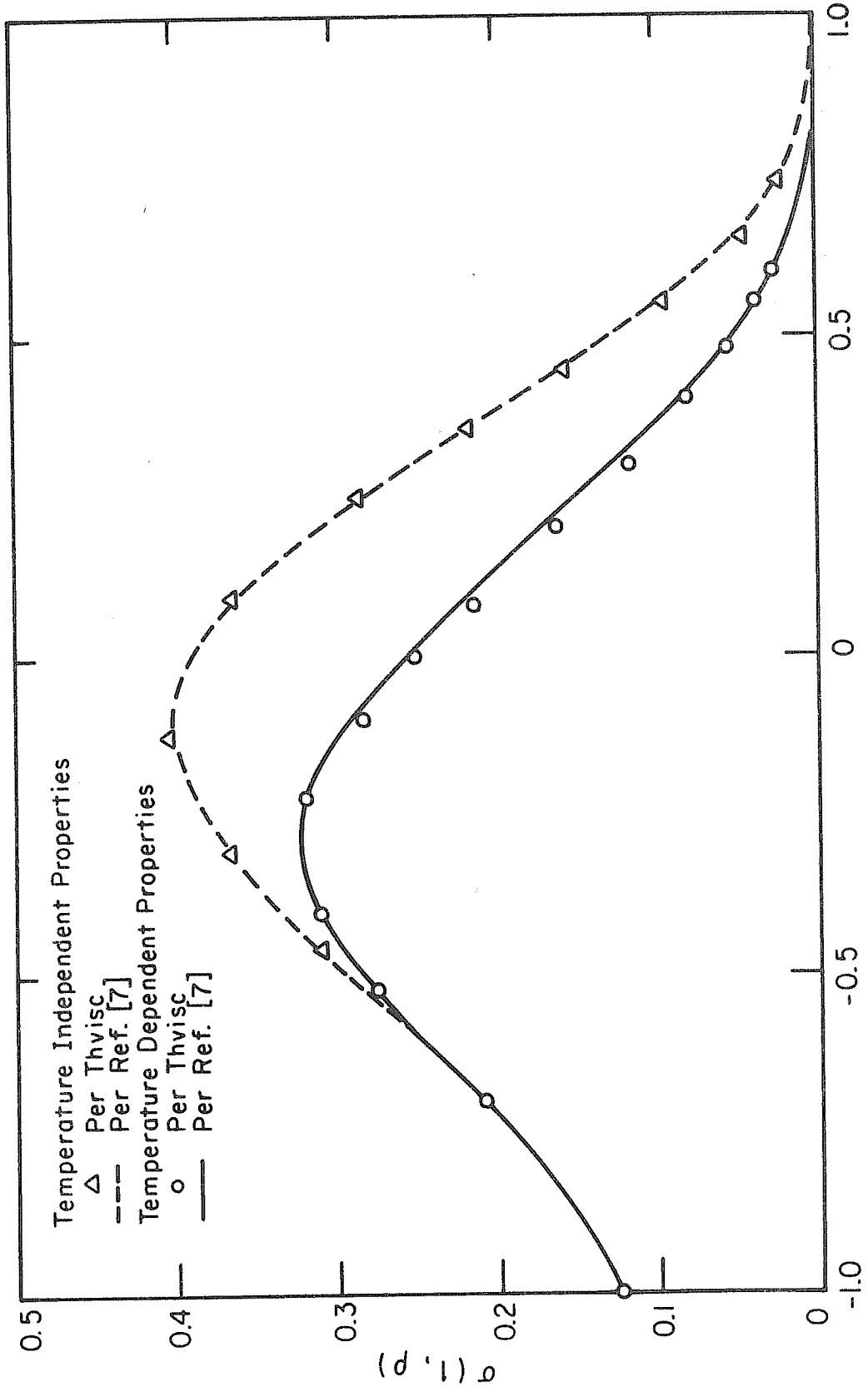

FIG. 4 HOOP STRESS AT OUTER RADIUS vs LOG TIME

APPENDIX A - USER'S MANUAL

IDENTIFICATION

THVISC - THERMOVISCOELASTIC STRESS ANALYSIS

Programmed - R. L. Taylor and G. L. Goudreau

University of California, Berkeley, June 1968

PURPOSE

The purpose of this computer program is to determine temperatures, deformations, and stresses in solids of revolution loaded axisymmetrically. The meridional cross section may have arbitrary shape and include multiple materials. The effects of arbitrary time dependent temperature, heat flux, displacement, or stress boundary conditions are included. Elastic bulk modulus and viscoelastic shear modulus specify the viscoelastic properties of isotropic linear viscoelastic materials. The effect of temperature on the shear modulus is determined for thermorheologically simple materials thru the use of a shift function. Variable time steps may be used in the step forward integration.

INPUT DATA

The first step in the structural analysis of an axisymmetric viscoelastic body is to spatially discretize the meridional plane into a mesh of finite elements. Elements and nodal points are numbered in two numerical sequences, each starting with one. To minimize the bandwidth, and thus the computational effort, nodes should be numbered across the narrower of the rows comprising a mesh.

The next step is the choice of time steps, defining the time discretization. This must be chosen with regard to the characteristic times of variable applied load (including temperature), thermal diffusion,

Subroutine TABLE can be replaced by the user's own program generating the exact functional form of φ if known.

Following cards - one for each temperature (3 F 10.0)

Columna 1-10 Temperature

11-20 Shift function

21-30 Coefficient of linear expansion

D. NODAL POINT CARDS (2I5, 5 F 10.0, I 5)

Columns 1-5 Nodal point number

10 Number which indicates if displacements or forces are to be specified

11-20 R - ordinate

21-30 Z - ordinate

31-40 UR

41-50 UZ

51-60 T

65 Number which indicates if temperature or heat flux is to be specified.

If the number in column 10 is

0 UR is the specified R load per radian, and
UZ is the specified Z load per radian

1 UR is the specified R displacement, and
UZ is the specified Z load per radian

2 UR is the specified R load per radian, and
UZ is the specified Z displacement

3 UR is the specified R displacement
UZ is the specified Z displacement

If the number in column 65 is

0 T is the specified heat flux per radian

1 T is the specified temperature

All loads and fluxes are considered total quantities acting on one radian of circumference. For insulated nodal points this external heat flow is zero.

Nodal points must be in numerical sequence. If cards are omitted, the omitted nodal points are generated at equal intervals along a straight line between the defined end nodal points. For the generated points, UR, UZ, and T are set equal to zero. If the boundary code in column 10 is the same for the end points of the interval, the value will be assigned to the generated points. If different, the code is set to zero. Thus, zero boundary forces or displacements can be generated, but non-zero ones cannot.

E. ELEMENT CARDS (6I5)

One card for each element

Columns 1-5 Element number

6-10 Nodal point I

11-15 Nodal point J

16-20 Nodal point K

21-25 Nodal point L

26-30 Material identification

If the R axis transforms to the Z axis by a counter clockwise rotation, then the sequence (I, J, K, L) must be counterclockwise. Maximum difference between nod al point numbers must be less than 15.

Element cards must be in element number sequence. If element cards are omitted, the program automatically generates the omitted information

by incrementing the preceding I, J, K, and L, setting the material number equal to the previous value. The last element card must always be supplied.

Triangular elements may be prescribed by repeating the last nodal number (I, J, K, K).

F. CONVECTION BOUNDARY CONDITION CARDS (2I5, 2F 10.0)

One card must be supplied for each boundary segment for which the following heat transfer equation applies:

$$q = h\Delta t$$

where q is the rate of heat transferred to the element per unit area, h the heat transfer coefficient of the boundary layer, and Δt the difference i n temperature between the external environment and the body. Each card contains the following information:

Column 1-5 I } Boundary nodes
6-10 J }

11-20 Boundary layer conductance

21-30 Temperature of external environment

G. TIME DEPENDENT BOUNDARY CONDITIONS

After NDT time steps, if less than NTIME, subroutine load is called and the following cards are read.

First Card (2I5, F10.0)

Column 1-5 Number of nodal point cards to be altered (NNP)

6-10 Number of time intervals for which new data is valid (NDT)

11-20 New time interval

Following NNP cards (if > 0), same as in D. Only cards to be altered need be included. Boundary codes, forces, displacements, fluxes or temperatures may be changed.

If a functional form of a time dependent boundary condition is preferred, subroutine LOAD can be replaced by the user. Set NDT = 1 on the control card and LOAD will be called after each time step.

APPENDIX B - PROGRAM LISTING

```
      PROGRAM THVISC (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE3,TAPE8,
     1 TAPE9,TAPE7)
C**** MAIN PROGRAM FOR SOLUTION OF THERMOVISCOELASTICITY PROBLEMS BY A
C**** FINITE ELEMENT DEVELOPMENT ----- TAYLOR 8/66
      COMMON HED(12),NUMNP,NUMEL,MAXBAN,INC,NTIME,NDT,TINC,
     1 TP(300),BP(300),B(300),A(30,300)
      COMMON/ND/CODE(150),R(150),Z(150),UR(150),UZ(150),T(150),KODE(150)
      COMMON /ELEM/ NTRANS,NUMCBC,TO,G(8,12),RO(12),COND(12),SPHT(12),
     1 QX(12),IX(150,5)
      COMMON/TAPE/DISTIF(317)
      DIMENSION D(1)
      EQUIVALENCE (D,B(201))
   10 CALL MESH
      INC=0
      TIME=-TINC
      NNN=2
      NN=NUMNP*NNN
      MB=MAXBAN*NNN
      NCOUNT=400+30*NUMNP
C**** FORM VISCOELASTIC STIFFNESS FOR EACH TIME
  250 CONTINUE
      DO 20 N=1,NUMNP
   20 TP(N)=T(N)
      DO 900 M=1,NDT
      TIME=TIME+TINC
      IF (NTRANS.NE.0.OR.TINC.EQ.0.0) GO TO 700
      REWIND 7
      REWIND 8
      IF (INC.GT.0) GO TO 280
      CALL THERM
      WRITE (8) (B(I),I=1,NCOUNT)
      GO TO 700
  280 IF (M.GT.1) GO TO 330
C**** TEMPERATURE BOUNDARY CONDITIONS
      READ  (8) (B(I),I=1,NCOUNT)
      DO 300 N=1,NUMNP
      B(N)=B(N)+T(N)
      IF (KODE(N).EQ.0) GO TO 300
      CALL MODIFY (N,T(N),A,B,NUMNP,MAXBAN,30)
      D(N)=0.0
  300 CONTINUE
C**** FORM EFFECTIVE CONDUCTIVITY MATRIX FOR STANDARD TIME INCREMENT
      DT2=1.0/TINC
      DO 320 N=1,NUMNP
      D(N)=DT2*D(N)
  320 A(1,N)=A(1,N)+D(N)
C**** DECOMPOSE CONDUCTIVITY MATRIX AND WRITE ON TAPE
      CALL SYMBC (A,B,NUMNP,MAXBAN,30,1)
      IF (NDT.GT.1) WRITE (7) (B(I),I=1,NCOUNT)
      GO TO 340
C**** CALCULATE TEMPERATURES AT END OF TIME INCREMENT
  330 READ (7) (B(I),I=1,NCOUNT)
  340 CONTINUE
      DO 400 I=1, NUMNP
  400 B(I)=B(I)+D(I)*TP(I)
```

```
      CALL SYMBC (A,B,NUMNP,MAXBAN,30,2)
      DO 500 I=1,NUMNP
  500 TP(I)=B(I)
  700 CONTINUE
      CALL BLOCKS (JSTOP)
      IF (JSTOP.NE.0) GO TO 10
      IF (INC.GT.NTIME) GO TO 10
      CALL SYMBC (A,B,NN,MB,30,1)
      CALL SYMBC (A,B,NN,MB,30,2)
      WRITE(6,2000) TIME,(N,R(N),Z(N), B(2*N-1), B(2*N),TP(N),N=1,NUMNP)
      DO 800 N=1,NN
  800 BP(N)=B(N)
      INC=INC+1
      DT=TINC
      IF (INC.GT.1) GO TO 900
      DT=0.0
      IF (NDT.EQ.1.AND.NTIME.GT.1) CALL LOAD
      GO TO 250
  900 CONTINUE
      IF (INC.LE.NTIME) CALL LOAD
      GO TO 250
 2000 FORMAT (18H1SOLUTION AT TIME=F10.3/  12HONODAL POINT,5X,5HR-ORD,
     1 5X,5HZ-ORD,6X,14HR-DISPLACEMENT,6X,14HZ-DISPLACEMENT,9X,11HTEMPER
     2ATURE/
     3 (I12,0P2F10.3,1P3E20.7))
      END
```

```
      SUBROUTINE MESH
      COMMON HED(12),NUMNP,NUMEL,MAXBAN,INC,NTIME,NDT,TINC,
     1 TP(300),BP(300),B(300),A(30,300)
      COMMON/ND/CODE(150),R(150),Z(150),UR(150),UZ(150),T(150),KODE(150)
      COMMON /ELEM/ NTRANS,NUMCBC,TO,G(8,12),RO(12),COND(12),SPHT(12),
     1 QX(12),IX(150,5)
      DATA THVISC /6HTHVISC/,STOP/6HSTOP  /,MBAND/15/,MAXNP/150/
      PHE=0.0
  950 READ (5,1000) HED
      IF (HED(1).EQ.STOP) STOP
      IF (HED(1).NE.THVISC) GO TO 950
C**** INPUT CONTROL INFORMATION AND MATERIAL PROPERTIES
      READ (5,1001)     NUMNP,NUMEL,NUMMAT,NUMPC,NUMCBC,NTIME,TINC,TO
     1 ,NTRANS,NDT
      IF (NDT.EQ.0) NDT=NTIME
      WRITE(6,2001) HED,NUMNP,NUMEL,NUMMAT,NUMPC,NUMCBC,NTIME,TINC,TO
      IF(NUMNP.GT.MAXNP.OR.NUMEL.GT.MAXNP) GO TO 950
      DO 50 M=1,NUMMAT
      READ (5,1002) MTYPE,NUMTC,     RO(MTYPE),COND(MTYPE),SPHT(MTYPE),
     1QX(MTYPE)
      WRITE(6,2002) MTYPE,NUMTC,     RO(MTYPE),COND(MTYPE),SPHT(MTYPE),
     1QX(MTYPE)
      READ (5,1003) (G(N,MTYPE),N=1,8)
      WRITE(6,2003) (G(N,MTYPE),N=1,8)
      CALL TABLE (NUMTC,MTYPE,TM,PHE,ALPHA)
   50 CONTINUE
C**** INPUT NODAL INFORMATION
      WRITE(6,2010)
      NEQ=2*NUMNP
      L=0
   60 READ (5,1004) N,CODE(N),R(N),Z(N),UR(N),UZ(N),T(N),KODE(N)
      NL=L+1
      ZX=N-L
      DR=(R(N)-R(L))/ZX
      DZ=(Z(N)-Z(L))/ZX
   70 L=L+1
      IF(N-L) 100,90,80
   80 CODE(L)=0.0
      IF (CODE(L-1).EQ.CODE(N)) CODE(L)=CODE(L-1)
      R(L)=R(L-1)+DR
      Z(L)=Z(L-1)+DZ
      UR(L)=0.0
      UZ(L)=0.0
      T(L)=0.0
      KODE(L)=0.
      GO TO 70
   90 WRITE(6,2004)(K,CODE(K),R(K),Z(K),UR(K),UZ(K),T(K),KODE(K),K=NL,N)
      IF(NUMNP-N) 100,110,60
  100 WRITE(6,2030) N
      GO TO 950
  110 CONTINUE
C**** INPUT ELEMENT INFORMATION
      WRITE(6,2011)
      N=0
```

```
      K=0
  130 READ (5,1005) M,(IX(M,I),I=1,5)
      DO 340 I1=1,4
      DO 325 L1=1,4
      KK=IABS(IX(M,I1)-IX(M,L1))
      IF(K.LE.KK) K=KK
  325 CONTINUE
  340 CONTINUE
  140 N=N+1
      IF(M.LE.N) GO TO 170
      DO 150 L=1,4
  150 IX(N,L)=IX(N-1,L)+1
      IX(N,5)=IX(N-1,5)
  170 WRITE(6,2005) N,(IX(N,I),I=1,5)
      IF(M-N) 185,180,140
  185 WRITE(6,2031) N
      GO TO 950
  180 IF(NUMEL.GT.N) GO TO 130
      MAXBAN=K+1
      IF(MAXBAN.LE.MBAND) RETURN
      WRITE(6,2032) MAXBAN
      GO TO 950
 1000 FORMAT (12A6)
 1001 FORMAT (6I5,2F10.0,2I5)
 1002 FORMAT (2I5,4F10.0)
 1003 FORMAT (8F10.0)
 1004 FORMAT (I5,F5.0,5F10.0,1I5)
 1005 FORMAT (6I5)
 2001 FORMAT (1H1,12A6/
     1    30H0 NUMBER OF NODAL POINTS------, I3/
     2    30H0 NUMBER OF ELEMENTS----------, I3/
     3    30H0 NUMBER OF DIFF. MATERIALS---, I3/
     5    30H0 NUMBER OF PRESS/SHEAR BC----,I3/
     4    30H0 NUMBER OF CONVECTION BC-----,I3/
     6    30H0 NUMBER OF TIME INCREMENTS---, I3/
     7    30H0 TIME INCREMENTS------------, F12.6/
     8    30H0 REFERENCE TEMPERATURE-------, F12.2/)
 2002 FORMAT (18H0 MATERIAL NUMBER=I3,30H, NUMBER OF TEMPERATURE CARDS=
     1I3/
     2 10H0 DENSITY=F10.5,15H, CONDUCTIVITY=F10.5,16H, SPECIFIC HEAT=
     3 F10.5,4H, Q=F10.5/)
 2003 FORMAT (14H0    K-ELASTIC,6X,9HG-ELASTIC,4X,11HG1-VISCOEL.,11X,
     X 4HTAU1,4X,11HG2-VISCOEL.,11X,4HTAU2,4X,11HG3-VISCOEL.,11X,4HTAU3/
     X (8E15.5))
 2004 FORMAT (I12,F12.2,2F12.3,2E24.7,1F13.4,I6)
 2005 FORMAT (I13,4I6,I13)
 2010 FORMAT (12H1NODAL POINT,8X,4HTYPE,12H  R-ORDINATE,12H  Z-ORDINATE,
     X 24H  R LOAD OR DISPLACEMENT,24H  Z LOAD OR DISPLACEMENT,
     X 13H  TEMPERATURE,6H  T-BC//)
 2011 FORMAT (50H1     ELEMENT     I     J     K     L     MATERIAL  )
 2030 FORMAT (26H0NODAL POINT CARD ERROR N=I5)
 2031 FORMAT (24H0 ELEMENT CARD ERROR, N=I5)
 2032 FORMAT (27H0BANDWIDTH EXCEEDED, MBAND=I5)
      END
```

```
      SUBROUTINE LOAD
      COMMON HED(12),NUMNP,NUMEL,MAXBAN,INC,NTIME,NDT,TINC
      COMMON/ND/CODE(150),R(150),Z(150),UR(150),UZ(150),T(150),KODE(150)
      READ (5,1000) NNP,NDT,TINC
      IF (NNP.EQ.0) RETURN
      WRITE (6,2000)
      DO 100 M=1,NNP
      READ  (5,1001) N,CODE(N),R(N),Z(N),UR(N),UZ(N),T(N),KODE(N)
  100 WRITE (6,2001) N,CODE(N),R(N),Z(N),UR(N),UZ(N),T(N),KODE(N)
      RETURN
 1000 FORMAT (2I5,F10.0)
 1001 FORMAT (I5,F5.0,5F10.0,I5)
 2000 FORMAT (12H1NODAL POINT,8X,4HTYPE,12H  R-ORDINATE,12H  Z-ORDINATE,
     X 24H  R LOAD OR DISPLACEMENT,24H  Z LOAD OR DISPLACEMENT,
     X 13H  TEMPERATURE,6H  T-BC//)
 2001 FORMAT (I12,F12.2,2F12.3,2E24.7,1F13.4,I6)
      END
```

```
      SUBROUTINE THERM
      COMMON HED(12),NUMNP,NUMEL,MAXBAN,INC,NTIME,NDT,TINC,
     1 TP(300),BP(300),B(300),A(30,300)
      COMMON/ND/CODE(150),R(150),Z(150),UR(150),UZ(150),T(150),KODE(150)
      COMMON /ELEM/ NTRANS,NUMCBC,TO,G(8,12),RO(12),COND(12),SPHT(12),
     1 QX(12),IX(150,5)
      DIMENSION D(1),TDOT(1),LM(3),E(3,3),EE(3,3),P(5),S(5,5),DD(5)
      DIMENSION X(1),Y(1)
      EQUIVALENCE (MBAND,MAXBAN),(D,B(201)),(X,R),(Y,Z)
      DATA EE/2.,3*1.,2.,3*1.,2./
      DO 110 I=1,NUMNP
      B(I)=0.0
      D(I)=0.0
      TP(I)=T(I)
      DO 110 J=1,MBAND
  110 A(J,I)=0.0
      DO 200 N=1,NUMEL
C**** FORM ELEMENT CONDUCTIVITY MATRIX
      DO 150 I=1,5
      P(I)=0.0
      DD(I)=0.0
      DO 150 J=1,5
  150 S(I,J)=0.0
      MTYPE=IX(N,5)
      I=IX(N,1)
      J=IX(N,2)
      K=IX(N,3)
      L=IX(N,4)
      IX(N,5)=I
      XX=(X(I)+X(J)+X(K)+X(L))/4.
      YY=(Y(I)+Y(J)+Y(K)+Y(L))/4.
      DO 152 K=1,4
      I=IX(N,K)
      J=IX(N,K+1)
      IF (I-J) 135,152,135
  135 AJ=X(J)-X(I)
      AK=XX-X(I)
      BJ=Y(J)-Y(I)
      BK=YY-Y(I)
      C=BJ-BK
      DX=AK-AJ
      XMUL=(X(I)+X(J)+XX)/3.0
  137 XLAM=AJ*BK-AK*BJ
      COMM=0.5*XMUL*COND(MTYPE)/XLAM
      QQ=XLAM*XMUL*QX(MTYPE)/4.
      QSTORE=XLAM*XMUL*SPHT(MTYPE)*RO(MTYPE)/4.
      E(1,1)= C*C+DX*DX
      E(1,2)= C*BK-DX*AK
      E(1,3)= DX*AJ-C*BJ
      E(2,1)=E(1,2)
      E(2,2)= BK*BK+AK*AK
      E(2,3)=-BK*BJ-AJ*AK
      E(3,1)=E(1,3)
      E(3,2)=E(2,3)
```

```
      E(3,3)= BJ*BJ+AJ*AJ
      LM(1)=K
      LM(2)=K+1
      IF (K-4) 145,140,145
  140 LM(2)=1
  145 LM(3)=5
      DO 151 I=1,3
      II=LM(I)
      P(II)=P(II)+QQ
      DD(II)=DD(II)+QSTORE
      DO 151 J=1,3
      JJ=LM(J)
  151 S(II,JJ)=S(II,JJ)+E(I,J)*COMM
  152 CONTINUE
      IX(N,5)=MTYPE
      DO 143 I=1,4
      DO 143 J=1,4
  143 S(I,J)=S(I,J)-S(I,5)*S(J,5)/S(5,5)
C**** ADD ELEMENT CONDUCTIVITY TO COMPLETE CONDUCTIVITY
      DO 175 L=1,4
      I=IX(N,L)
      B(I)=B(I)+P(L)
      D(I)=D(I)+DD(L)
      DO 175 M=1,4
      J=IX(N,M)-I+1
      IF(J.LE.0) GO TO 175
      A(J,I)=A(J,I)+S(L,M)
  175 CONTINUE
  200 CONTINUE
C**** CONVECTION BOUNDARY CONDITIONS
      IF (NUMCBC) 220,220,205
  205 WRITE (6,2006)
      DO 215 N=1,NUMCBC
      READ (5,1007)  I,J,H,TEMC
      WRITE (6,2007) I,J,H,TEMC
      XL=SQRT((X(J)-X(I))**2+(Y(J)-Y(I))**2)*(X(I)+X(J))*0.5
      TEMC=0.5*H*XL*TEMC
      H=H*XL/6.0
      B(I)=B(I)+TEMC
      B(J)=B(J)+TEMC
      A(1,I)=A(1,I)+2.0*H
      A(1,J)=A(1,J)+2.0*H
      K=J-I+1
      IF (K) 212,212,210
  210 A(K,I)=A(K,I)+H
      GO TO 215
  212 K=I-J+1
      A(K,J)=A(K,J)+H
  215 CONTINUE
  220 CONTINUE
      RETURN
 1007 FORMAT (2I5,2F10.0)
 2006 FORMAT (40H0   I     J                H     TEMPERATURE )
 2007 FORMAT (2I5,2E15.6)
      END
```

```
      SUBROUTINE BLOCKS (JSTOP)
      COMMON HED(12),NUMNP,NUMEL,MAXBAN,INC,NTIME,NDT,TINC,
     1 TP(300),BP(300),B(300),A(30,300)
      COMMON/ND/CODE(150),R(150),Z(150),UR(150),UZ(150),T(150),KODE(150)
      COMMON/TAPE/DISTIF(317)
      JSTOP=0
      NNN=2
      NNP=NNN*NUMNP
      MBA=NNN*MAXBAN
      DO 300 II=1,NNP
      B(II)=0.0
      DO 300 JJ=1,MBA
  300 A(JJ,II)=0.0
      REWIND 3
      REWIND 9
      IF (INC.GT.0) WRITE (6,2001)
      IND=1
      IF((INC/2)*2-INC.EQ.0) IND=0
      DO 500 N=1,NUMEL
      IF (INC.EQ.0) GO TO 400
      IF (IND.NE.0) READ (3) DISTIF
      IF (IND.EQ.0) READ (9) DISTIF
      CALL STRESS (N)
      IF (INC.GT.NTIME) GO TO 500
  400 CALL QUAD (N,ISTOP)
      IF (IND.EQ.0) WRITE (3) DISTIF
      IF (IND.NE.0) WRITE (9) DISTIF
      IF(ISTOP.EQ.0) GO TO 500
      WRITE(6,2000) N
      JSTOP=ISTOP
  500 CONTINUE
      IF (INC.GT.NTIME) RETURN
      IF(JSTOP.NE.0) RETURN
      DO 700 N=1,NUMNP
      B(2*N-1)=B(2*N-1)+UR(N)
      B(2*N  )=B(2*N  )+UZ(N)
      C=CODE(N)
      IF (C.LE.0.0) GO TO 700
      IF (C.EQ.1.0.OR.C.EQ.3.0) CALL MODIFY (2*N-1,UR(N),A,B,NNP,MBA,30)
      IF (C.EQ.2.0.OR.C.EQ.3.0) CALL MODIFY (2*N  ,UZ(N),A,B,NNP,MBA,30)
  700 CONTINUE
      RETURN
 2000 FORMAT (26HONEGATIVE OR ZERO AREA, N=I5)
 2001 FORMAT (8H1ELEMENT,5X,5HR-ORD,5X,5HZ-ORD,7X,8HR-STRESS,7X,
     X 8HT-STRESS,7X,8HZ-STRESS,6X,9HRZ-STRESS,11X,4HSIG1,11X,4HSIG2,5X,
     X 5HANGLE//)
      END
```

```
      SUBROUTINE STRESS (NL)
      COMMON HED(12),NUMNP,NUMEL,MAXBAN,INC,NTIME,NDT,TINC,
     1 TP(300),BP(300),B(300),A(30,300)
      COMMON/ND/CODE(150),R(150),Z(150),UR(150),UZ(150),T(150),KODE(150)
      COMMON /ELEM/ NTRANS,NUMCBC,TO,G(8,12),RO(12),COND(12),SPHT(12),
     1 QX(12),IX(150,5)
      COMMON/TAPE/ESTIG(10,10),ESTIK(10,10),TEMP(10),DU(10),UPR(10),
     1 GUN(3,11),ST(5,10),AT,RM,ZM,THETA
      DIMENSION TAU(7)
      MAT=IX(NL,5)
      DO 500 I=1,4
      II=2*I
      JJ=2*IX(NL,I)
      DU(II-1)=BP(JJ-1)-UPR(II-1)
      DU(II)=BP(JJ)-UPR(II)
      UPR(II-1)=BP(JJ-1)
  500 UPR(II)=BP(JJ)
      NN=4
      IF(IX(NL,3).EQ.IX(NL,4)) NN=3
      XNN=NN
      DU(9)=0.
      DU(10)=0.
      UPR(9)=0.
      UPR(10)=0.
      DO 510 J=1,NN
      DO 510 I=9,10
      IJ=2*J+I-10
      DU(I)=DU(I)+DU(IJ)/XNN
  510 UPR(I)=UPR(I)+UPR(IJ)/XNN
      DO 200 I=1,3
  200 TAU(I)=-3.*G(1,MAT)*THETA
      TAU(4)=0.0
      DO 303 II=1,10
      DO 301 JJ=1,3
  301 TAU(JJ)=TAU(JJ)+(3.*G(1,MAT)*ST(5,II)+2.0*G(2,MAT)*ST(JJ,II))
     1 *UPR(II)
  303 TAU(4)=TAU(4)+2.*G(2,MAT)*ST(4,II)*UPR(II)
      DO 300 M=1,3
      GAM=G(2*M+2,MAT)
      IF(GAM.EQ.0.) GO TO 300
      DO 302 II=1,10
      GUN(M,II)=GUN(M,II)+GUN(M,11)*DU(II)
      DO 302 JJ=1,4
  302 TAU(JJ)=TAU(JJ)+ST(JJ,II)*GUN(M,II)*2.0
  300 CONTINUE
C****COMPUTE PRINCIPAL STRESSES
      CC=0.5*(TAU(1)+TAU(3))
      CR=SQRT(0.25*(TAU(3)-TAU(1))**2+TAU(4)**2)
      TAU(5)=CC+CR
      TAU(6)=CC-CR
      TAU(7)=0.
      IF(TAU(4).EQ.0.0.AND.TAU(1)-TAU(3).EQ.0.0) GO TO 305
      TAU(7)=28.648*ATAN2(2.*TAU(4),TAU(1)-TAU(3))
  305 CONTINUE
```

```
      WRITE (6,2001) NL,RM,ZM,(TAU(II),II=1,7)
      RETURN
2001  FORMAT (I8,2F10.3,6E15.6,F10.3/)
      END
```

```
      SUBROUTINE QUAD (NL,ISTOP)
      COMMON HED(12),NUMNP,NUMEL,MAXBAN,INC,NTIME,NDT,TINC,
     1 TP(300),BP(300),B(300),A(30,300)
      COMMON/ND/CODE(150),R(150),Z(150),UR(150),UZ(150),T(150),KODE(150)
      COMMON /ELEM/ NTRANS,NUMCBC,TO,G(8,12),RO(12),COND(12),SPHT(12),
     1 QX(12),IX(150,5)
      COMMON/TAPE/ESTIG(10,10),ESTIK(10,10),TEMP(10),DU(10),UPR(10),
     1 GUN(3,11),ST(5,10),AT,RM,ZM,THETA
      DIMENSION ESTIF(10,10),FORCE(10),RR(4),ZZ(4)
      REAL KA,MU
C**** QUADRILATERAL STIFFNESS FOR AXISYMMETRIC VISCOELASTIC ELEMENT
C**** COMPUTE CENTER NODAL POSITION
      NN=4
      IF(IX(NL,3).EQ.IX(NL,4)) NN=3
      XNN=NN
      RM=0.
      ZM=0.
      TM=0.
      DO 760 J=1,NN
      I=IX(NL,J)
      RM=RM+R(I)/XNN
      ZM=ZM+Z(I)/XNN
  760 TM=TM+TP(I)/XNN
      IF (NN.EQ.3) NN=1
C**** COMPUTE ELEMENT DATA
      MAT=IX(NL,5)
C**** REDUCED TIME COMPUTATIONS FOR MATERIAL PROPERTIES
      CALL TABLE (0,MAT,TM,PHE,ALPHA)
      TMM=TM
      IF (INC.GT.0) TMM=0.5*(TM+AT)
      AT=TM
      CALL TABLE (0,MAT,TMM,PHE,ALPHA)
      RTINC=PHE*TINC
      THETA=ALPHA*(TM-TO)
      IF (INC.EQ.0.AND.NTRANS.EQ.0) THETA=0.0
      KA=G(1,MAT)
      MU=G(2,MAT)
      DO 330 II=1,10
  330 FORCE(II)=3.*KA*THETA*TEMP(II)
      IF (INC.GT.0) GO TO 320
      DO 700 N=1,4
      I=IX(NL,N)
      RR(N)=R(I)
  700 ZZ(N)=Z(I)
      CALL QSTIF (RR,ZZ,NN,ISTOP)
      IF (ISTOP.NE.0) RETURN
      DO 810 M=1,3
      GUN(M,11)=G(2*M+1,MAT)
  810 MU=MU+G(2*M+1,MAT)
      GO TO 350
C**** COMPUTE EFFECTIVE LOADS
  320 DO  300 M=1,3
      GAM=G(2*M+2,MAT)
      IF (GAM.EQ.0.0) GO TO 300
```

```
      COM1=G(2*M+1,MAT)*DINT(-RTINC/GAM)
      GUN(M,11)=COM1
      COM=EXP(-RTINC/GAM)
      MU=MU+COM1
      DO  310 II=1,10
      GUN(M,II)=COM*GUN(M,II)
      DO 310 JJ=1,10
  310 FORCE(JJ)=FORCE(JJ)+ESTIG(JJ,II)*(COM1*UPR(II)-GUN(M,II))
  300 CONTINUE
  350 DO 410 II=1,10
      DO 410 JJ=1,10
  410 ESTIF(II,JJ)=MU*ESTIG(II,JJ)+KA*ESTIK(II,JJ)
C      QUADRILATERAL STIFFNESS IS FORMED, ELIMINATE CENTER NODE
      IF (NN.EQ.1) GO TO 170
      NN=9
  160 DO 150 II=1,NN
      CC=ESTIF(II,NN+1)/ESTIF(NN+1,NN+1)
      ESTIF(II,NN+1)=CC
      FORCE(II)=FORCE(II)-CC*FORCE(NN+1)
      DO 150 JJ=1,NN
  150 ESTIF(II,JJ)=ESTIF(II,JJ)-CC*ESTIF(NN+1,JJ)
      FORCE(NN+1)=FORCE(NN+1)/ESTIF(NN+1,NN+1)
      NN=NN-1
      IF(NN.GE.8) GO TO 160
  170 CONTINUE
C**** ADD ELEMENT STIFFNESS TO TOTAL STIFFNESS
      DO 200 I=1,4
      DO 200 K=1,2
      II=2*IX(NL,I)+K-2
      KK=2*I+K-2
      B(II)=B(II)+FORCE(KK)
      DO 200 J=1,4
      DO 200 L=1,2
      JJ=2*IX(NL,J)+L-II-1
      MM=2*J+L-2
      IF(JJ.LE.0) GO TO 200
      A(JJ,II)=A(JJ,II)+ESTIF(KK,MM)
  200 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TABLE (NUMTC,MAT,TM,PHE,ALPHA)
      DIMENSION E(20,3,12)
      DATA MAXPHI/20/
      IF (NUMTC.EQ.0) GO TO 30
      DO 10 L=1,NUMTC
   10 READ (5,1000) (E(L,N,MAT   ),N=1,3)
      WRITE(6,2000)((E(L,N,MAT   ),N=1,3),L=1,NUMTC)
      DO 20 I=NUMTC,MAXPHI
      DO 20 J=1,2
   20 E(I,J,MAT   )=E(NUMTC,J,MAT   )
      RETURN
   30 DO 40 M=2,MAXPHI
      IF(E(M,1,MAT).GE.TM) GO TO 50
   40 CONTINUE
   50 RATIO=0.0
      IF(E(M,1,MAT).EQ.E(M-1,1,MAT)) GO TO 60
      RATIO = (TM-E(M-1,1,MAT))/(E(M,1,MAT)-E(M-1,1,MAT))
   60 PHE=E(M-1,2,MAT)+RATIO*(E(M,2,MAT)-E(M-1,2,MAT))
      ALPHA=E(M-1,3,MAT)+RATIO*(E(M,3,MAT)-E(M-1,3,MAT))
      RETURN
 1000 FORMAT (3F10.0)
 2000 FORMAT (14H0  TEMPERATURE,5X,15H SHIFT FUNCTION,5X,15HCOEFF. OF. E
     1XP./(F15.2,2E20.5))
      END
```

```
      FUNCTION DINT(X)
      DINT=1.0
      IF (X.EQ.0.0) RETURN
      IF (ABS(X).LE.0.1) GO TO 100
      DINT=-(1.0-EXP(X))/X
      RETURN
  100 DO 200 II=1,10
      XM=12-II
  200 DINT=1.0+X/XM*DINT
      RETURN
      END
```

```
      SUBROUTINE QSTIF (R,Z,NN,ISTOP)
      COMMON/TAPE/ESTIG(10,10),ESTIK(10,10),TEMP(10),DU(10),UPR(10),
     1 GUN(3,11),ST(5,10),AT,RM,ZM,THETA
      DIMENSION S(6,6),SK(6,6),SG(6,6),SKG(6,6),H(3,3),PHI(3,3),
     1 IPHI(3,4),C(5,5),TH(6),IG(7),JG(7),KG(7),R(4),Z(4),RC(3),ZC(3)
      DATA IG/1,1,1,2,3,4,5/
      DATA JG/2,1,3,1,1,1,1/
      DATA KG/1,2,3,2,6,5,3/
      DATA IPHI/1,3,9,3,5,9,5,7,9,7,1,9/
      DATA C/4.,-2.,-2.,2*0.,-2.,4.,-2.,2*0.,-2.,-2.,4.,5*0.,2*3.,3*0.,
     X 2*3./
C**** SET UP MATRIX
      ISTOP=0
      DO 750 I=1,10
      TEMP(I)=0.
      DO 750 J=1,10
      ESTIG(I,J)=0.
  750 ESTIK(I,J)=0.
      DO 101 II=1,10
      DU(II)=0.0
      UPR(II)=0.0
      DO 101 JJ=1,3
  101 GUN(JJ,II)=0.0
      DO 740 I=1,5
      DO 740 J=1,10
  740 ST(I,J)=0.0
      AREA=0.0
C**** LOOP ON ELEMENTS FOR QUAD
      DO 1000 NT=1,NN
      I=NT
      J=NT+1
      IF (J.GT.4) J=1
      COM=R(J)*(ZM-Z(I))+RM*(Z(I)-Z(J))+R(I)*(Z(J)-ZM)
      IF (COM.GT.0.0) GO TO 190
      ISTOP=1
      RETURN
  190 PHI(1,1)=(R(J)*ZM-RM*Z(J))/COM
      PHI(1,2)=(RM*Z(I)-R(I)*ZM)/COM
      PHI(1,3)=(R(I)*Z(J)-R(J)*Z(I))/COM
      PHI(2,1)=(Z(J)-ZM)/COM
      PHI(2,2)=(ZM-Z(I))/COM
      PHI(2,3)=(Z(I)-Z(J))/COM
      PHI(3,1)=(RM-R(J))/COM
      PHI(3,2)=(R(I)-RM)/COM
      PHI(3,3)=(R(J)-R(I))/COM
C**** INITIALIZE
      DO 100 II=1,6
      TH(II)=0.
      DO 100 JJ=1,6
      SKG(II,JJ)=0.0
      SG(II,JJ)=0.0
      SK(II,JJ)=0.0
  100 S(II,JJ)=0.
      RC(1)=R(I)
```

```
      RC(2)=R(J)
      RC(3)=RM
      ZC(1)=Z(I)
      ZC(2)=Z(J)
      ZC(3)=ZM
      CALL INTER (H,RC,ZC)
      DO 110 IA=1,5
      I=IG(IA)
      J=JG(IA)
      K=KG(IA)
      TH(K)=TH(K)+H(J,1)
      DO 110 JA=1,5
      L=IG(JA)
      M=JG(JA)
      N=KG(JA)
      SG(K,N)=SG(K,N)+H(J,M)*C(I,L)/3.
  110 S(K,N)=S(K,N)+H(J,M)
      DO 120 IA=6,7
      I=IG(IA)
      J=JG(IA)
      K=KG(IA)
      DO 120 JA=6,7
      L=IG(JA)
      M=JG(JA)
      N=KG(JA)
  120 SG(K,N)=SG(K,N)+H(J,M)*C(I,L)/3.
      DO 130 I=1,3
      DO 130 J=1,3
      DO 130 K=1,3
      SKG(I,J)=SKG(I,J)+SG(I,K)*PHI(K,J)
      SKG(I,J+3)=SKG(I,J+3)+SG(I,K+3)*PHI(K,J)
      SKG(I+3,J+3)=SKG(I+3,J+3)+SG(I+3,K+3)*PHI(K,J)
      SK(I,J)=SK(I,J)+S(I,K)*PHI(K,J)
      SK(I,J+3)=SK(I,J+3)+S(I,K+3)*PHI(K,J)
  130 SK(I+3,J+3)=SK(I+3,J+3)+S(I+3,K+3)*PHI(K,J)
      DO 140 I=1,3
      IS=IPHI(I,NT)
      IT=IS+1
      DO 140 J=1,3
      TEMP(IS)=TEMP(IS)+PHI(J,I)*TH(J)
      TEMP(IT)=TEMP(IT)+PHI(J,I)*TH(J+3)
      JS=IPHI(J,NT)
      JT=JS+1
      DO 140 K=1,3
      ESTIG(IS,JS)=ESTIG(IS,JS)+PHI(K,I)*SKG(K,J)
      ESTIG(IS,JT)=ESTIG(IS,JT)+PHI(K,I)*SKG(K,J+3)
      ESTIG(JT,IS)=ESTIG(IS,JT)
      ESTIG(IT,JT)=ESTIG(IT,JT)+PHI(K,I)*SKG(K+3,J+3)
      ESTIK(IS,JS)=ESTIK(IS,JS)+PHI(K,I)*SK(K,J)
      ESTIK(IS,JT)=ESTIK(IS,JT)+PHI(K,I)*SK(K,J+3)
      ESTIK(JT,IS)=ESTIK(IS,JT)
  140 ESTIK(IT,JT)=ESTIK(IT,JT)+PHI(K,I)*SK(K+3,J+3)
      IF (NN.NE.1) GO TO 160
      DO 150 I=1,4
      DO 150 J=5,6
```

```
      ESTIG(I,J)=ESTIG(I,J+4)
      ESTIK(I,J)=ESTIK(I,J+4)
      ESTIG(J,I)=ESTIG(I,J)
  150 ESTIK(J,I)=ESTIK(I,J)
      ESTIG(5,5)=ESTIG(9,9)
      ESTIK(5,5)=ESTIK(9,9)
      ESTIG(5,6)=ESTIG(9,10)
      ESTIK(5,6)=ESTIK(9,10)
      ESTIG(6,6)=ESTIG(10,10)
      ESTIK(6,6)=ESTIK(10,10)
  160 CONTINUE
C**** INITIALIZE STRAIN DISPLACEMENT MATRIX
      AREA=AREA+COM
      I=NT
      J=NT+1
      IF (J.GT.4) J=1
      II=2*NT-1
      JJ=2*NT+1
      IF (NT.EQ.4) JJ=1
      ST(1,II  )=2.*(Z(J)-ZM)/3.          +ST(1,II  )
      ST(2,II  )=   (ZM-Z(J))/3.          +ST(2,II  )
      ST(3,II  )=   (ZM-Z(J))/3.          +ST(3,II  )
      ST(4,II  )=   (RM-R(J))/2.          +ST(4,II  )
      ST(5,II  )=   (Z(J)-ZM)/3.          +ST(5,II  )
      ST(1,II+1)=   (R(J)-RM)/3.          +ST(1,II+1)
      ST(2,II+1)=   (R(J)-RM)/3.          +ST(2,II+1)
      ST(3,II+1)=2.*(RM-R(J))/3.          +ST(3,II+1)
      ST(4,II+1)=   (Z(J)-ZM)/2.          +ST(4,II+1)
      ST(5,II+1)=   (RM-R(J))/3.          +ST(5,II+1)
      ST(1,JJ  )=2.*(ZM-Z(I))/3.          +ST(1,JJ  )
      ST(2,JJ  )=   (Z(I)-ZM)/3.          +ST(2,JJ  )
      ST(3,JJ  )=   (Z(I)-ZM)/3.          +ST(3,JJ  )
      ST(4,JJ  )=   (R(I)-RM)/2.          +ST(4,JJ  )
      ST(5,JJ  )=   (ZM-Z(I))/3.          +ST(5,JJ  )
      ST(1,JJ+1)=   (RM-R(I))/3.          +ST(1,JJ+1)
      ST(2,JJ+1)=   (RM-R(I))/3.          +ST(2,JJ+1)
      ST(3,JJ+1)=2.*(R(I)-RM)/3.          +ST(3,JJ+1)
      ST(4,JJ+1)=   (ZM-Z(I))/2.          +ST(4,JJ+1)
      ST(5,JJ+1)=   (R(I)-RM)/3.          +ST(5,JJ+1)
      ST(1,9)=(2.*(Z(I)-Z(J))-COM/RM)/3.  +ST(1,9)
      ST(2,9)=(2.*COM/RM+Z(J)-Z(I))/3.    +ST(2,9)
      ST(3,9)=(Z(J)-Z(I)-COM/RM)/3.       +ST(3,9)
      ST(4,9)=(R(J)-R(I))/2.              +ST(4,9)
      ST(5,9)=(COM/RM+Z(I)-Z(J))/3.       +ST(5,9)
      ST(1,10)=(R(I)-R(J))/3.             +ST(1,10)
      ST(2,10)=(R(I)-R(J))/3.             +ST(2,10)
      ST(3,10)=2.*(R(J)-R(I))/3.          +ST(3,10)
      ST(4,10)=(Z(I)-Z(J))/2.             +ST(4,10)
      ST(5,10)=(R(J)-R(I))/3.             +ST(5,10)
 1000 CONTINUE
      DO 580 II=1,5
      DO 580 JJ=1,10
  580 ST(II,JJ)=ST(II,JJ)/AREA
      RETURN
      END
```

```
      SUBROUTINE INTER (H,RC,ZC)
      DIMENSION SS(10),W(10)
      DIMENSION RR(4),ZZ(4),H(3,3)
      DIMENSION RC(3),ZC(3)
      DATA W/0.555555556,0.888888888,0.555555556/
      DATA SS/-.77459667,0.0,+.77459667/,LIM/3/
C**** COMPUTE INTEGRALS
      IH=0
      DO 200 II=1,3
      DO 200 JJ=II,3
  200 H(II,JJ)=0.
      RMIN=AMIN1(RC(1),RC(2),RC(3))
      RMAX=AMAX1(RC(1),RC(2),RC(3))
      DO 100 II=1,3
      IF(RC(II).EQ.RMIN) IL=II
      IF(RC(II).EQ.RMAX) IU=II
  100 CONTINUE
      DO 110 II=1,3
      IF(II.NE.IL.AND.II.NE.IU) IM=II
  110 CONTINUE
      RR(1)=RC(IL)
      RR(2)=RC(IM)
      RR(3)=RR(1)
      RR(4)=RC(IM)
      ZZ(1)=ZC(IL)
      ZZ(3)=ZZ(1)
      IF(ZC(IM).LE.ZC(IU).AND.ZC(IM).LT.ZC(IL)) GO TO 150
      ZZ(2)=ZC(IM)
      ZZ(4)=ZC(IL)+(RR(2)-RR(1))*(ZC(IU)-ZZ(1))/(RC(IU)-RR(1))
      GO TO 160
  150 ZZ(4)=ZC(IM)
      ZZ(2)=ZC(IL)+(RR(2)-RR(1))*(ZC(IU)-ZZ(1))/(RC(IU)-RR(1))
  160 IF(RR(2).EQ.RR(1)) GO TO 220
  230 HH=(RR(2)-RR(1))*0.5
      DO 210 II=1,LIM
      HR=RR(1)+HH*(1.0+SS(II))
      ZT=ZZ(1)+0.5*(SS(II)+1.)*(ZZ(2)-ZZ(1))
      ZB=ZZ(3)+0.5*(SS(II)+1.0)*(ZZ(4)-ZZ(3))
      H(1,1)=H(1,1)+HH*HR*(ZT-ZB)*W(II)
      H(1,2)=H(1,2)+HH*(ZT-ZB)*W(II)
      H(1,3)=H(1,3)+0.5*HH*(ZT*ZT-ZB*ZB)*W(II)
      H(2,2)=H(2,2)+HH*(ZT-ZB)*W(II)/HR
      H(2,3)=H(2,3)+0.5*HH*(ZT*ZT-ZB*ZB)*W(II)/HR
  210 H(3,3)=H(3,3)+HH*(ZT*ZT*ZT-ZB*ZB*ZB)*W(II)/(3.*HR)
  220 IH=IH+1
      IF(IH.GE.2) GO TO 240
      RR(1)=RR(2)
      RR(2)=RC(IU)
      IF(RR(2).EQ.RR(1)) GO TO 240
      RR(3)=RR(4)
      RR(4)=RC(IU)
      ZZ(1)=ZZ(2)
      ZZ(2)=ZC(IU)
      ZZ(3)=ZZ(4)
```

```
      ZZ(4)=ZC(IU)
      GO TO 230
  240 CONTINUE
      DO 215 II=2,3
      IIM1=II-1
      DO 215 JJ=1,IIM1
  215 H(II,JJ)=H(JJ,II)
      RETURN
      END
```

```
      SUBROUTINE MODIFY (N,Q,A,B,NN,MB,MMAX)
      DIMENSION A(MMAX,1),B(1)
      DO 100 J=2,MB
      L=N-J+1
      IF (L.LE.0) GO TO 50
      B(L)=B(L)-A(J,L)*Q
      A(J,L)=0.0
   50 L=N+J-1
      IF (L.GT.NN) GO TO 100
      B(L)=B(L)-A(J,N)*Q
  100 A(J,N)=0.0
      B(N)=Q
      A(1,N)=1.0
      RETURN
      END
```

```
      SUBROUTINE SYMBC (A,B,NN,MB,MMAX,KK)
      DIMENSION A(MMAX,1),B(1)
      GO TO (1000,2000),KK
 1000 NNN=NN-1
      DO 300 N=1,NNN
      DO 200 J=2,MB
      IF (A(J,N).EQ.0.0) GO TO 200
      C=A(J,N)/A(1,N)
      L=N+J-1
      IF (L.GT.NN) GO TO 150
      DO 100 I=J,MB
      K=I-J+1
  100 A(K,L)=A(K,L)-C*A(I,N)
  150 A(J,N)=C
  200 CONTINUE
  300 CONTINUE
      RETURN
 2000 DO 500 N=1,NN
      DO 400 J=2,MB
      L=N+J-1
      IF (L.GT.NN) GO TO 500
  400 B(L)=B(L)-A(J,N)*B(N)
  500 B(N)=B(N)/A(1,N)
      DO 700 I=2,NN
      N=NN-I+1
      DO 600 J=2,MB
      L=N+J-1
      IF (L.GT.NN) GO TO 700
  600 B(N)=B(N)-A(J,N)*B(L)
  700 CONTINUE
      RETURN
      END
```