(2)

r Research Center

Bethesda, MD 20084-5000

DTRC-SME-91/34 July 1991

Ship Materials Engineering Department

Research and Development Report

Analysis of Strain Dependent Damping in Materials via Modeling of Material Point Hysteresis

by

E.J. Graesser*

C.R. Wong

*ASEE/ONT Postdoctoral Fellow

DTRC-SME-91/34 Analysis of Strain Dependent Damping in Materials via Modeling of Material Point Hysteresis

Approved for public release; distribution unlimited.

MAJOR DTRC TECHNICAL COMPONENTS

CODE	011	DIRECTOR OF TECHNOLOGY, PLANS AND ASSESSMENT
	12	SHIP SYSTEMS INTEGRATION DEPARTMENT
	14	SHIP ELECTROMAGNETIC SIGNATURES DEPARTMENT
	15	SHIP HYDROMECHANICS DEPARTMENT
	16	AVIATION DEPARTMENT
	17	SHIP STRUCTURES AND PROTECTION DEPARTMENT
	18	COMPUTATION, MATHEMATICS & LOGISTICS DEPARTMENT
	19	SHIP ACOUSTICS DEPARTMENT
	27	PROPULSION AND AUXILIARY SYSTEMS DEPARTMENT
	28	SHIP MATERIALS ENGINEERING DEPARTMENT

DTRC ISSUES THREE TYPES OF REPORTS:

1. **DTRC reports, a formal series,** contain information of permanent technical value. They carry a consecutive numerical identification regardless of their classification or the originating department.

2. **Departmental reports, a semiformal series,** contain information of a preliminary, temporary, or proprietary nature or of limited interest or significance. They carry a departmental alphanumerical identification.

3. **Technical memoranda, an informal series,** contain technical documentation of limited use and interest. They are primarily working papers intended for internal use. They carry an identifying number which indicates their type and the numerical code of the originating department. Any distribution outside DTRC must be approved by the head of the originating department on a case-by-case basis.

David Taylor Research Center

Bethesda, MD 20084-5000

DTRC-SME-91/34 July 1991

Ship Materials Engineering Department

Research & Development Report

Analysis of Strain Dependent Damping in Materials via Modeling of Material Point Hysteresis

by

E.J. Graesser*

C.R. Wong

*ASEE/ONT Postdoctoral Fellow

CONTENTS

FIGURES

ABSTRACT

A constitutive relationship was used to model the cyclic material response of damping test samples in separate bending and torsion configurations. This was done in order to better understand variations in reported values of damping for materials possessing strain dependent characteristics. The constitutive equations are based on a model of shape memory alloy stress-strain behavior and have been adapted especially for the study of nonlinear hysteresis and the problem of strain dependent damping. Experimental measurements and analytical material response analyses of separate bending and torsion test samples indicated that when the damping of a single nonlinear material is plotted against the one-dimensional local strain of the sample, results are produced which are difficult to compare. However, when the same results are plotted against an invariant measure of three-dimensional distortion the means by which one may compare the data is more straightforward. Also, the approach allows for a quantitative comparison of the damping at a material point to the overall damping. The method can be applied to any homogeneous isotropic nonlinear damping material.

ADMINISTRATIVE INFORMATION

This research described in this paper was performed at David Taylor Research Center, under the supervision of Dr. O.P. Arora, DTRC Code 2812. It was supported in part by an Office of Naval Technology (ONT) postdoctoral fellowship under the administration of the American Society for Engineering Education (ASEE) and by the Quiet Alloys Program which is part of the Functional Materials Block Program sponsored by Mr. Ivan Caplan (DTRC Code 0115) Program Element 62254N Task Area RS3454, Work Unit 1-2812-949. This report satisfies milestone 53SR1/4.

INTRODUCTION

In addition to add-on damping techniques currently being used in the Navy, material damping is being investigated as a potential means of further reducing machine vibration, noise, and sound emission in seafaring vessels. Ideally a high damping structural material provides a sufficient amount of both stiffness and damping so as to be used as a sole machine part or vibrating element without added treatments. Such materials are most useful for oscillating parts or elements that cannot be damped by conventional external treatments. Also these materials can be useful in situations where heat or other environmental factors (e.g. moisture, corrosion) have to be considered. This approach is also useful in damping longitudinal vibrations which cannot be effectively controlled by external treatments.

Because high stiffness and strength are required in many important applications, metals which possess a large inherent damping capacity have been extensively sought [1,2]. Some specific applications include gears and gear webs, pump casings, engine parts, propellers, and others (see [1]). High damping metals are also used as plug inserts and cladding, and such applications can provide a reduction of resonant amplification factors as well as the attenuation of ringing [1].

Generally "high damping" in metal is a measured peak loss factor or phase lag with a value of 10^{-2} or higher. Indeed many alloy compositions have been studied and found to possess such levels of damping (e.g. see [3-8]). Mechanisms which give rise to damping in metals include: movement of point defects, dislocations or domain walls. These effects give rise to macroscopic hysteresis and thus damping. The damping capacity of high damping metals is strain dependent because the primary damping mechanisms function over a finite strain range. Such effects give rise to a well defined peak in the plot of measured damping vs. specimen strain amplitude. Examples of magnetostrictive metallic materials exhibiting strain-dependent damping are given in Fig. 1. This type of response is termed "nonlinear" because the measured damping capacity varies with specimen strain amplitude.

A generalized stress-strain diagram corresponding to a relatively large category of nonlinear damping mechanisms is illustrated in Fig. 2. Note from this figure that the damping mechanism is activated near a critical stress σ_c and becomes saturated at a strain of ε_0. This type of hysteretic response can be associated with a number of nonlinear anelastic damping mechanisms. For example, in dislocation breakaway a minimum stress is required to force dislocations over nearby pinning points during loading. Upon unloading the elastic strain energy stored in the lattice of the material may be sufficient to move the dislocations back to their original positions. The net effect of this process is an elastic response with internal friction.

Other nonlinear anelastic damping mechanisms include the movement of mobile domain boundaries. Ferromagnetic domain walls, twin boundaries, antiferromagnetic domain walls, and phase domain walls fall into this category. Usually, a finite amount of stress (σ_c) is required to initiate this type of mechanism, i.e. a specific amount of stress is required to overcome an energy barrier so that the boundaries may move. Release of the applied stress subsequently causes the mechanism to act in reverse because the elastic strain energy stored in the material is sufficient to move the boundaries back across the energy barrier. Again the net effect is an elastic response with energy dissipation due to internal friction. Also, these mechanisms may become saturated at a limiting value of strain. For example, the magnetic domains in high damping ferromagnetic alloys are arranged in a randomly oriented pattern when the material is unstressed. However, upon application of a uniaxial stress the domains change their orientation and tend to align themselves in the direction of loading as the stress is increased. Once these domains become fully aligned any further stress cannot cause relative motion of the domains and the mechanism is said to be saturated. The amount of energy that can be dissipated by this type of damping mechanism is therefore limited to a fixed value for cyclic strain amplitudes greater than the limiting value of strain corresponding to saturation (ε_0).

The data obtained for a single strain dependent material in varied test configurations is often difficult to compare because of the inherent strain distributions that arise from the loading. Data from separate bending and torsion tests [3] given in Fig. 3 shows this effect; indeed the results indicate that the torsional tests produce significantly higher values of damping for common levels of peak sample strain. Also, the damping rises more steeply with peak sample strain in the torsion test. However it is important to note that the strains on the abscissa are shear strains in the case of torsional data and axial strains in the case of bending data and these separate strains are not equivalent; indeed axial strains give a measure of length change while shear strains refer to the distortion of right angles This is an important aspect of the problem which will be discussed in the analysis section of this paper.

Strain dependent materials are, at best, difficult to model analytically because of their nonlinear characteristics. Early work in this area concentrated on evaluating the damping of members by combining material energy absorbing properties with geometric and stress distribution factors [9,10]. Another approach is to use a constitutive law which describes nonlinear material behavior and hysteresis at a point, and this approach will be used here. Many such laws exist (e.g. see [11,12]), but these are usually specific to postyielding viscoplastic behavior and large strain levels. In this paper a proposed constitutive law [13,14] for the stress-strain behavior of shape memory alloys is adapted to the case of nonlinear damping. The equations of this law were applied to the cases of simple uniaxial tension-compression and shear loading of materials. Solid geometries of beam and shaft test samples in bending and torsion were also considered. The strain dependent nature of each test configuration was computed, and because this behavior was of primary interest, temperature and frequency effects were not considered.

ANALYSIS

A three-dimensional constitutive law of hysteretic material behavior was employed so that a useful study of strain dependent damping could be made. This law is based on the three dimensional generalization [14] of a one-dimensional model of shape memory alloy (SMA) stress-strain behavior [13], where the extension from one to three dimensions follows a method originally developed by Prager [15] (for a detailed development regarding this extension method see [16,17,14]). This choice of modeling schemes was pursued because the hysteretic response of superelastic SMA's is very similar in character to that of high damping metals (see Fig. 2), except that the stress and strain levels are different by many orders of magnitude. This does not prevent the use of the constitutive law, however, as long as the material properties of the law can be scaled to accommodate the lower stress and strain levels associated with the dissipative mechanisms of the damping material.

The constitutive law is for homogeneous and isotropic material behavior and is based upon a separation of strain and strain rate into elastic and inelastic components:

$$\varepsilon_{ij} = \varepsilon_{ij}^{el} + \varepsilon_{ij}^{in} \tag{1a}$$

$$\dot{\varepsilon}_{ij} = \dot{\varepsilon}_{ij}^{el} + \dot{\varepsilon}_{ij}^{in} \tag{1b}$$

Here an overhead dot represents differentiation with respect to time. Thus ε_{ij} and $\dot{\varepsilon}_{ij}$ are the three-dimensional tensors of strain and strain rate, and the superscripts "el" and "in" designate the respective elastic and inelastic components of each. The elastic component follows directly from the theory of isotropic elasticity [17]:

$$\varepsilon_{ij}^{el} = \frac{1+\nu}{E}\sigma_{ij} - \frac{\nu}{E}\sigma_{kk}\,\delta_{ij} \tag{2}$$

where σ_{ij} is the stress tensor, δ_{ij} is the Kronecker delta[1], and where E and ν are the elastic material constants.

The basic equations for the evolution of inelastic strain were taken from the previously cited model of shape memory alloy behavior. In this model the growth of inelastic strain is a function of a backstress tensor β_{ij}, which is a variable that accounts for internal stress fields in the material, as described by the following set of equations:

$$\dot{\varepsilon}_{ij}^{in} = \sqrt[+]{3K_2}\left[\sqrt[+]{3J_2}\right]^{n-1}\left[\frac{s_{ij} - b_{ij}}{\sigma_c}\right] \tag{3}$$

$$b_{ij} = \frac{2}{3}E\alpha\left[\varepsilon_{ij}^{in} + f_T\frac{e_{ij}}{\frac{2}{3}\sqrt[+]{3I_2}}\operatorname{erf}\left[\frac{2}{3}a\sqrt[+]{3I_2}\right]\left\{u\left[-\dot{I}_2\right]\right\}\right] \tag{4}$$

Here e_{ij}, s_{ij}, and b_{ij} are the deviatoric tensors of strain, stress and backstress respectively; the difference $s_{ij} - b_{ij}$ is often referred to as the effective stress. The quantities I_2, J_2, and K_2 are the second order invariants of the deviatoric tensors of strain, dimensionless effective stress, and strain rate, respectively; these quantities are defined below:

$$e_{ij} = \varepsilon_{ij} - \frac{1}{3}\varepsilon_{kk}\,\delta_{ij} \quad , \qquad I_2 = \frac{1}{2}e_{ij}\,e_{ij} \quad , \qquad K_2 = \frac{1}{2}\dot{e}_{ij}\,\dot{e}_{ij}$$

$$s_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\,\delta_{ij}$$

[1] $\delta_{ij}=1$ if $i=j$, $\delta_{ij}=0$ if $i\neq j$, $i,j=1,2,3$

$$b_{ij} = \beta_{ij} - \frac{1}{3}\beta_{kk}\,\delta_{ij} \quad , \qquad J_2 = \frac{1}{2}\frac{s_{ij} - b_{ij}}{\sigma_c}\frac{s_{ij} - b_{ij}}{\sigma_c}$$

Thus the growth of inelastic strain is a function of stress, backstress, and strain rate. Also, note here that plus sign appearing with the radical sign of the square root of the invariants in Eqs. (3)-(4) indicates that the square root, once taken, is to be positive (i.e. the absolute value of the square root). The tensor e_{ij} is known as the distortional component of strain because, by definition, it subtracts the dilatational component of deformation out of the strain tensor ε_{ij}. Therefore, I_2 represents a measure of volumetric distortion that is invariant with respect to coordinate transformations, and this will be an important quantity in the forthcoming discussion. The material constants in Eqs. (2)-(4) are:

- E: Young's extensional elastic modulus
- ν: Poisson ratio of elastic material
- σ_c: minimum axial stress necessary to activate the damping mechanism
- α: constant which determines the slope of the inelastic region $= E_y/(E - E_y)$, where E_y is the inelastic slope
- n: constant controlling the sharpness of transition from elastic to inelastic behavior
- f_T: constant controlling the size of the hysteresis loop
- a: constant controlling the amount of elastic recovery during unloading

Also, Eqs. (3) and (4) contain two special functions: the error function, erf( ), and the unit step function, {u( )}. Simply stated the purpose of the error and unit step functions contained in Eq. (4) is to allow for the recovery of accumulated inelastic strain during unloading, and thus simulate the unique behavior of superelastic materials [13,14].

Let us take a moment to explain the role of the inelastic response in the modeling of strain dependent damping. The inelastic component of strain is responsible for the dissipation of energy that takes place in cyclic loading. Equations (1)-(4) have been used to represent the macroscopic stress-strain behavior of shape memory alloys, and especially superelastic materials [13,14]. The hysteretic character of superelasticity is macroscopically similar to that of nonlinear anelasticity except that the respective stress and strain levels of each type of response are different by many orders of magnitude. Therefore, the inelastic response governed by Eqs. (3)-(4) can be used to represent the macroscopic effect of a nonlinear anelastic damping mechanism.

By using Eqs. (1)-(4), a number of special cases can be considered. First let us consider the cases of uniaxial tension-compression and pure shear loading. The state of uniaxial loading (superscript u) is described by:

$$\varepsilon_{ij}^{u} = \begin{bmatrix} \varepsilon & 0 & 0 \\ 0 & -\mu\varepsilon & 0 \\ 0 & 0 & -\mu\varepsilon \end{bmatrix} \qquad \dot{\varepsilon}_{ij}^{u} = \begin{bmatrix} \dot{\varepsilon} & 0 & 0 \\ 0 & -\rho\dot{\varepsilon} & 0 \\ 0 & 0 & -\rho\dot{\varepsilon} \end{bmatrix} \qquad \sigma_{ij}^{u} = \begin{bmatrix} \sigma & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \beta_{ij}^{u} = \begin{bmatrix} \beta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Here ε, σ, and β are the axial strain, stress, and backstress in the x direction of Cartesian space, respectively. Also the lateral strain and strain rate induced by the Poisson effect ($-\mu\varepsilon$ and $-\rho\dot{\varepsilon}$) are associated with the coefficients μ and ρ respectively. Strictly speaking, μ and ρ are neither constant nor equal due to the nonlinear effect induced by the damping mechanism. In order to evaluate these coefficients the lateral strain and strain rate are decomposed into elastic and inelastic parts and we will assume that volume changes induced by axial loading are associated only with elastic deformation. Thus the inelastic part of the strain and strain rate are associated with incompressible behavior as is done in the theories of plasticity and viscoplasticity [17]. This assumption is plausible since the damping mechanisms involve movement of dislocations, point defects, domain walls, or polymer chains none of which induce a change in volume. Using this assumption, the elastic component of the lateral strain is related to the elastic axial strain by the elastic Poisson ratio ν, and the inelastic component is related to the axial inelastic strain by the Poisson coefficient of incompressible deformation (which is 0.5); therefore $-\mu\varepsilon = -\nu\varepsilon^{el} - .5\varepsilon^{in}$. Similarly, the lateral strain rate is $-\rho\dot{\varepsilon} = -\nu\dot{\varepsilon}^{el} - .5\dot{\varepsilon}^{in}$. Using these relations it can be shown (Appendix A) that μ and ρ are:

$$\mu = \frac{1}{2} - \frac{1}{E}\left[\frac{1}{2} - \nu\right]\frac{\sigma}{\varepsilon}$$

$$\rho = \frac{1}{2} - \frac{1}{E}\left[\frac{1}{2} - \nu\right]\frac{d\sigma}{d\varepsilon}$$

Thus μ and ρ are clearly variable coefficients which are not necessarily equal to each other.

For the conditions of pure shear loading (superscript s) we have:

$$\varepsilon_{ij}^{s} = \begin{bmatrix} 0 & \frac{\gamma}{2} & 0 \\ \frac{\gamma}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \dot{\varepsilon}_{ij}^{s} = \begin{bmatrix} 0 & \frac{\dot{\gamma}}{2} & 0 \\ \frac{\dot{\gamma}}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \sigma_{ij}^{s} = \begin{bmatrix} 0 & \tau & 0 \\ \tau & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \beta_{ij}^{s} = \begin{bmatrix} 0 & \xi & 0 \\ \xi & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Here γ and $\dot{\gamma}$ are the engineering shear strain and strain rate, τ is the shear stress, and ξ is the shear backstress in the xy plane of Cartesian space.

By using the appropriate stress, backstress, and strain tensors, as well as their respective deviators and associated invariants, Eqs. (1)-(4) produce the following uniaxial equations:

$$\dot{\sigma} = E\left[\dot{\varepsilon} - \frac{2(1+\rho)}{3}|\dot{\varepsilon}|\left|\frac{\sigma - \beta}{\sigma_c}\right|^{n-1}\left[\frac{\sigma - \beta}{\sigma_c}\right]\right]$$

$$\beta = E\alpha\left[\varepsilon - \frac{\sigma}{E} + f_T \operatorname{erf}\left[\frac{2(1+\mu)}{3}a\varepsilon\right]\{u(-\varepsilon\dot{\varepsilon})\}\right]$$

If the behavior is only a small departure from elasticity then $\mu \approx \rho \approx \nu$; conversely if a condition of strain and strain rate exists where inelastic behavior dominates and where $\sigma/\varepsilon \ll E$ and $d\sigma/d\varepsilon \ll E$ then the response is essentially incompressible with $\mu \approx \rho \approx .5$. Now, since μ and ρ appear only in terms associated with inelastic strain, the previous equations can be simplified by setting both Poisson coefficients equal to 0.5. This does not greatly affect the numerical results since these terms will be significant only in the inelastic region (i.e. when $\sigma_c/E \leq \varepsilon$). Thus we obtain:

$$\dot{\sigma} = E\left[\dot{\varepsilon} - |\dot{\varepsilon}|\left|\frac{\sigma - \beta}{\sigma_c}\right|^{n-1}\left[\frac{\sigma - \beta}{\sigma_c}\right]\right] \tag{5}$$

$$\beta = E\alpha\left[\varepsilon - \frac{\sigma}{E} + f_T \operatorname{erf}(a\varepsilon)\{u(-\varepsilon\dot{\varepsilon})\}\right] \tag{6}$$

For the case of pure shear loading the governing differential equations are:

$$\dot{\tau} = G\left[\dot{\gamma} - |\dot{\gamma}| \left|\frac{\tau - \xi}{\tau_c}\right|^{n-1} \left(\frac{\tau - \xi}{\tau_c}\right)\right] \tag{7}$$

$$\xi = \frac{E\alpha}{3}\left[\gamma - \frac{\tau}{G} + \sqrt{3}\, f_T \,\mathrm{erf}\left(\frac{a\gamma}{\sqrt{3}}\right) \{u(-\gamma\dot{\gamma})\}\right] \tag{8}$$

where

$G = \dfrac{E}{2(1 + \nu)}$ is the elastic shear modulus

$\tau_c = \dfrac{\sigma_c}{\sqrt{3}}$ is the shear stress whereupon the damping mechanism is activated

Note that τ_c falls out of the formulation automatically in a manner that is consistent with the theory of maximum distortional strain energy [18]. This is because Eq. (3) is dependent on the stress gradient of a potential function [14,16] that contains a Bingham type condition for the onset of the inelastic damping mechanism.

As noted before, the damping mechanisms become saturated at an axial strain of ε_0. This differs from SMA hysteretic behavior and therefore the model must be modified. The modification consists of specifying that the growth of inelastic strain be stopped at a saturation limit beyond which linear elastic behavior once again takes place. To do this the second term in Eq. (5) is multiplied by the unit step function $\{u(\varepsilon_0 - |\varepsilon|)\}$. At levels where $\varepsilon < \varepsilon_0$, $\{u(\varepsilon_0 - |\varepsilon|)\}=1$ and inelastic growth of strain may proceed, but when $\varepsilon \geq \varepsilon_0$, $\{u(\varepsilon_0 - |\varepsilon|)\}=0$ and continued loading beyond ε_0 is elastic. Similarly, for saturation of damping mechanisms in shear the second term on the right hand side of Eq. (7) is multiplied by $\{u(\gamma_0 - |\gamma|)\}$, where γ_0 is the shear strain of saturation.

Since we are modeling materials which are both homogeneous and isotropic, the saturation process in more general three-dimensional loadings can be represented by a value of the invariant I_2 which is determined from either ε_0 or γ_0; we will call this value I_{2_0}. Thus saturation is determined by a specific amount of distortion, and the unit step functions $\{u(\varepsilon_0-|\varepsilon|)\}$ and $\{u(\gamma_0-|\gamma|)\}$ can be obtained from $\{u(\sqrt{3I_{2_0}} - \sqrt{3I_2}\}$.

To summarize, the equations that are set forth to represent the respective three dimensional, axial, and shear responses of nonlinear anelastic materials are as follows:

3D:
$$\dot{\varepsilon}_{ij}^{in} = \sqrt[+]{3\dot{K}_2}\left[\sqrt[+]{3J_2}\right]^{n-1}\left[\frac{s_{ij} - b_{ij}}{\sigma_c}\right]\left\{u\left[\sqrt{3I_{2_0}} - \sqrt[+]{3I_2}\right]\right\} \tag{9}$$

$$b_{ij} = \frac{2}{3}E\alpha\left[\varepsilon_{ij}^{in} + f_T\frac{e_{ij}}{\frac{2}{3}\sqrt[+]{3I_2}}\operatorname{erf}\left[\frac{2}{3}a\sqrt[+]{3I_2}\right]\left\{u\left[-\dot{I}_2\right]\right\}\right] \tag{10}$$

Axial:
$$\dot{\sigma} = E\left[\dot{\varepsilon} - |\dot{\varepsilon}|\left|\frac{\sigma - \beta}{\sigma_c}\right|^{n-1}\left[\frac{\sigma - \beta}{\sigma_c}\right]\{u(\varepsilon_0 - |\varepsilon|)\}\right] \tag{11}$$

$$\beta = E\alpha\left[\varepsilon - \frac{\sigma}{E} + f_T\operatorname{erf}(a\varepsilon)\{u(-\varepsilon\dot{\varepsilon})\}\right] \tag{12}$$

Shear:
$$\dot{\tau} = G\left[\dot{\gamma} - |\dot{\gamma}|\left|\frac{\tau - \xi}{\tau_c}\right|^{n-1}\left[\frac{\tau - \xi}{\tau_c}\right]\{u(\gamma_0 - |\gamma|)\}\right] \tag{13}$$

$$\xi = \frac{E\alpha}{3}\left[\gamma - \frac{\tau}{G} + \sqrt{3}\, f_T\operatorname{erf}\left[\frac{a\gamma}{\sqrt{3}}\right]\{u(-\gamma\dot{\gamma})\}\right] \tag{14}$$

where

$$\gamma_0 = \frac{2(1 + \nu)}{\sqrt{3}}\varepsilon_0$$ is strain limiting inelastic growth in shear

The hysteresis loops associated with axial and shear behavior are produced by numerical integration of Eqs. (11)-(12) and (13)-(14), and by specifying a sinusoidal history of strain input with amplitudes of ε_p in the axial case and γ_p in the shear case (Appendices B and C). These amplitudes were specified to be greater than the saturation strains so that the full character of the predicted response could be plotted. The results of calculations for the axial and shear loading conditions are given in Figs. 4 and 5 respectively. Both figures possess the same characteristics:

elastic behavior dominates in the region of the origin as well as outside the limiting strain, and a hysteresis loop is manifested in the full cycle of strain application. The area enclosed by the hysteresis loop represents the energy absorbed by the material per cycle of oscillation, and this quantity is denoted as ΔW. An elastic modulus of $E = 28.5 \times 10^6$ psi and saturation strain of $\varepsilon_0 = 0.0001$ were selected based on the elastic modulus and approximate strain of peak damping in Fe-Cr alloys (see Fig. 1). The remaining material constants used in the calculations which generated Figs. 4 and 5 were not selected to reproduce the behavior of any specified damping material; rather they were selected to approximate a typical pattern of hysteresis in nonlinear damping materials and to permit investigative analyses. It should be noted that the numerical results generated by the constitutive equations were obtained by Runge-Kutta fourth order integration, and these results for the material response were strain rate independent. Also the elastic and inelastic material properties (i.e. E, E_y, and σ_c) are accurately reproduced in numerical calculations [14].

By having numerical results of the type just presented, we can now compute the damping according to the definition of the loss factor, η, defined as

$$\eta = \frac{\Delta W}{2\pi W} \tag{15}$$

where ΔW is defined as given above, and W is a measure of stored energy most often selected as

$$W = \frac{1}{2}\varepsilon_{max}\,\sigma\Big|_{\varepsilon_{max}} \tag{16}$$

By using Eqs. (11)-(12) and (13)-(14) in calculations for the cyclic material response over a range of peak axial and shear strains, and computing the loss factor associated with each peak strain according to Eqs. (15) and (16), the general character of the damping vs. strain diagram of nonlinear materials was produced (computer algorithms are given in Appendices D and E); this is shown in Fig. 6. Note that both curves possess the characteristic damping peak associated with nonlinear damping materials.

However the separate curves in Fig. 6 representing axial and shear loading differ significantly with respect to one another, and this can make the results difficult to interpret. In an effort to understand these results, the amount of energy absorbed in each loading configuration was evaluated. The amount of energy absorbed is plotted as a function of peak strain in Fig. 7. Above the respective axial and shear saturation strains the amount of energy absorbed by the material is essentially the same for both loading configurations, their difference being less than 1%. Therefore the difference in the character of the two separate responses must be due to other factors.

It turns out that the plots given in both Figs. 6 and 7 are misleading because the abscissa of these figures represents values of strain associated with separate axial and shear loading conditions, and the strains associated with these separate conditions are not equivalent. Since damping is a material property, it is desirable to plot strain dependent material damping values using a measure that will unify the curves from separate tests. Such a method would also provide a basis by which to present and compare damping data for a variety of materials. Therefore another measure of deformation equivalent to both types of loading needs to be employed. One such possibility is to use a measure of distortion rather than strain. Let us define an equivalent strain $\bar{\varepsilon}$ as follows:

$$\bar{\varepsilon} = \sqrt{3I_2} \tag{17}$$

This measure is similar to the effective plastic strain in plastically deforming materials [17]. It is clear that $\bar{\varepsilon}$ has a physical meaning that is independent of the choice of coordinate axes since it is based on the invariant I_2, which is the second invariant of the deviatoric strain e_{ij}. Therefore $\bar{\varepsilon}$ is an invariant measure of distortion.

By considering the separate conditions of axial and shear loading, and by taking the variable Poisson coefficient to be the elastic constant ν for axial case, the equivalent strains for each condition are:

$$\bar{\varepsilon}^{u} = (1 + \nu)\,\varepsilon \qquad \text{(uniaxial loading)}$$

$$\bar{\varepsilon}^{s} = \frac{\sqrt{3}}{2}\gamma \qquad \text{(shear loading)}$$

Using the peak equivalent strains of axial and shear loading in place of the peak strains used in Figs. 6 and 7 produces a more consistent pattern of results. This is shown in Fig. 8 where the energy absorbed as a function of $\bar{\varepsilon}$ is in very good agreement along the entire abscissa for both cases. In Fig. 9 the loss factors of the axial and shear loading cases are also plotted against the peak equivalent strain. Even though the peaks of the separate curves are not of equal magnitude the results in this figure are now very similar; indeed the rise and fall of each curve follow the same trend and the peak of each damping curve occurs at approximately the same level of distortion.

The peak of the loss factor curve for shear loading is slightly higher than the peak of the axial loss factor curve due to a smaller value of stored energy in the shear loading case. This is another complication that arises from the nonlinear nature of the of the stress-strain material response and (for equal amounts of distortion) causes the value of peak shear stress to be lower than the peak axial stress. Consequently, at equal levels of distortion, the measure of stored energy (Eq. (17)) will be larger in axial loading than in shear and this will cause the loss modulus in shear to be greater than the loss modulus in axial loading.

Bending and torsion are common configurations in which to measure damping. The solid beam has length L and rectangular cross-section of width b and thickness h. Although the stress-strain response is nonlinear we can consider both cases in a simple fashion because the response takes place in a manner which gives symmetric behavior for positive and negative strains. When considering bending and torsion problems with pronounced plastic deformation and nonsymmetrical stress-strain responses, then special considerations must be made when computing the acting moments [17].

Schematic illustrations of bending and torsion are shown in Fig. 10. The torsional solid shaft has length L and circular cross-section of radius R. Note that the strain profiles in each geometry are linear, passing through zero at the position of the neutral axis of the beam and starting at zero at the center of the shaft. Also note that ε_p is the value of the axial strain at the beam surface while γ_p is the value of the engineering shear strain at the shaft surface. The angles θ and ϕ are the curvature of the bending beam and the angle of twist of the shaft respectively. Because the problems under consideration involve only small strain, the following simple relations can be used to compute the moments and angular displacements for the beam and shaft geometries respectively:

$$M = -\int_A y\, \sigma\, dA \qquad \text{and} \qquad \theta = \frac{2\varepsilon_p L}{h} \qquad \text{(beam)}$$

$$T = \int_A r\, \tau\, dA \qquad \text{and} \qquad \phi = \frac{\gamma_p L}{R} \qquad \text{(shaft)}$$

Here y is the vertical distance from the neutral axis of the beam cross-section, σ is the axial stress in the longitudinal fibers of the beam, and M is the resultant moment bending the beam; for the shaft r is the distance from the center of the circular cross-section, τ is the shear stress due to torsion, and T is the resultant torque twisting the shaft. The equations given above relating θ and ϕ to sample dimensions and surface strain amplitude are easily deducible from simple geometrical arguments which involve the knowledge of a linear strain profile and the restriction of small strains.

In the numerical analyses associated with these bending and torsion tests, the surface strains of each test sample were specified to act sinusoidally in time. Each geometry was subdivided into a large number of finite, but thin, subsections (i.e. the infinitesimal distances dy and dr in Fig. 10 were replaced by small but finite distance Δy and Δr respectively). Also, the strain distribution for each finite subsection was assumed to be constant over the subsection thickness and the value of the strain was taken as the value of the strain profile at the center of the subsection. Knowing the strain profile of the cross-section of each geometry and the history of the respective surface strains, the stress history for each subsection of the geometry was computed numerically. Specifically, Eqs. (11)-(12) were integrated to give the stress profile time history of the bending beam and (13)-(14) were integrated for the shear stress profile time history of the shaft (see Appendices F and G). Then the following formulas were used to compute the resultant moment and torque histories of the beam and shaft:

$$M = -b \sum_{i=1}^{N} y_i \, \sigma_i \, \Delta y$$

$$T = 2\pi \sum_{i=1}^{N} (r_i)^2 \, \tau_i \, \Delta r$$

where N is the number of subdivisions making up the cross-sectional geometry and where the subscript i indicates reference to the location of a single subsection.

The loss factor of each sample geometry was then calculated for a specified value of surface strain amplitude according to Eq. (15) where ΔW was determined by the area enclosed by the resultant moment vs. angular displacement hysteretic response and W was determined by

$$W = \frac{1}{2} \theta_{max} \, M \Big|_{\theta_{max}} \qquad \text{(beam)}$$

$$W = \frac{1}{2} \phi_{max} \, T \Big|_{\phi_{max}} \qquad \text{(shaft)}$$

The damping values which were computed in this way were found to be independent of sample geometry, i.e. for a given surface amplitude the ratio of ΔW to W remained constant for changes in cross-sectional size, sample length or both. This effect is due to the fact that a proportional, volume of material is undergoing deformation wherein the damping mechanism is activated and this proportion of volume is constant irrespective of sample size for both simple bending and torsion.

In Fig. 11 the loss factor was plotted against the surface amplitude for both the bending and torsion cases by repeating the calculations over a range of surface amplitudes. Note that the character of the damping vs. surface amplitude curves are different with respect to one another. This is analogous the trend shown earlier in Fig. 6 for one dimensional behavior. Also, by comparing Fig. 11 to Fig. 6 it is clear that the character of the damping vs. peak strain curve of each sample is quite different than that corresponding to the respective one-dimensional material point responses. This is due to the strain dependent nature of the damping and the fact that strain is nonuniformly distributed throughout the sample; therefore some regions of the geometry may be contributing significantly to the overall damping of the solid sample while others are not.

Using Eq. (11) to calculate the amounts of peak equivalent strain at the surface of the bending and torsion samples, Fig. 12 shows that the use of peak equivalent strain gives an improved measure of correlation in the same manner that was exhibited earlier for the one-dimensional cases.

Thus presentation of nonlinear damping data as a function of equivalent strain rather than as a function of sample strain can be very useful. It is probably most useful in comparing damping data obtained by different test methods. It may also be useful in design work where the dynamic strains in a vibrating part or member are known. Along this line, let us briefly consider an example where a designer wishes to use a high damping, but nonlinear, material in an application where bending is the primary mode of deformation, and suppose that damping data is available only from torsional tests. If the vibrational strain levels to be expected in service can be deduced from load and design analyses, then these strain levels can be converted to the measure of equivalent strain introduced in this paper. The designer would then be able to estimate whether or not the material damping will be in a range of peak performance for the application of interest by applying the same conversion to the peak shear strains of the torsional damping data.

SUMMARY

The work presented in this paper includes three major aspects; 1) modeling of nonlinear (or strain dependent) damping behavior via constitutive equations, 2) relating the damping of the material to the damping of a test specimen, and 3) a way of improving correlation of nonlinear damping data via use of equivalent measures of distortion. These efforts were conducted in order to gain a better understanding of macroscopic nonlinear high damping material behavior and also to obtain a means in which to better correlate damping data from tests which use different sample geometries. The modeling scheme applies to homogeneous isotropic materials and is adapted from a constitutive model of the viscoplastic type through incorporation of constants that represent the onset of damping mechanisms. Also the model was modified to include damping mechanisms that become saturated after a given amount of strain. Analyses were made to calculate the loss factor of the common damping test configurations of bending and torsion. To do this material point relationships were used at a large number of points making up the cross-sectional geometry. In this way it was possible to relate the damping of the material to the damping of the specimen. The results did not depend on the relative dimensions of the sample geometry; rather the calculated loss factors depended only on the mode of deformation. The results showed that the strain dependent damping associated with each test were difficult to compare when plotted solely against the peak surface strain of the sample geometry. This is because the peak strains that correspond to each of these test configurations, namely axial and shear strain, are different from one another. However if an invariant measure of peak sample distortion is used in place of peak sample strain, then the correlation of the nonlinear damping of separate bending and torsion samples improves considerably. Such an improved capacity for the correlation of nonlinear damping data is very useful for comparison of data obtained from different tests. Future research will include the modeling of specific

nonlinear damping data. Also, constitutive model material parameters that are physically motivated by the microstructure will be studied.

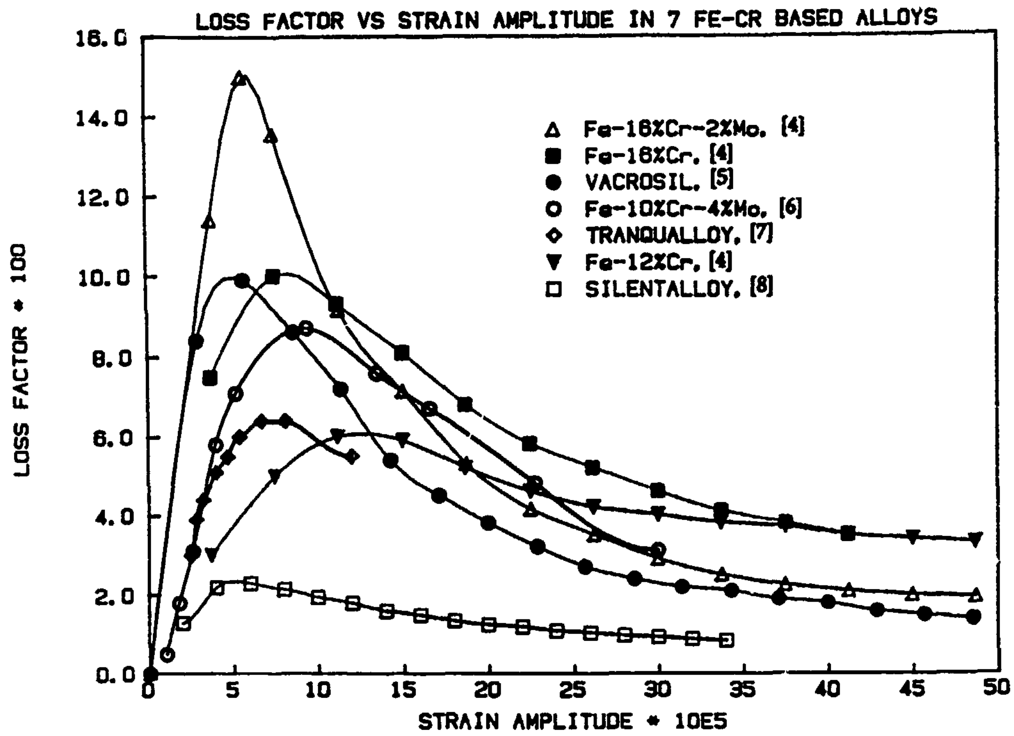


Fig. 1. Strain Amplitude Dependent Damping in Fe-Cr Based High Damping Alloys.

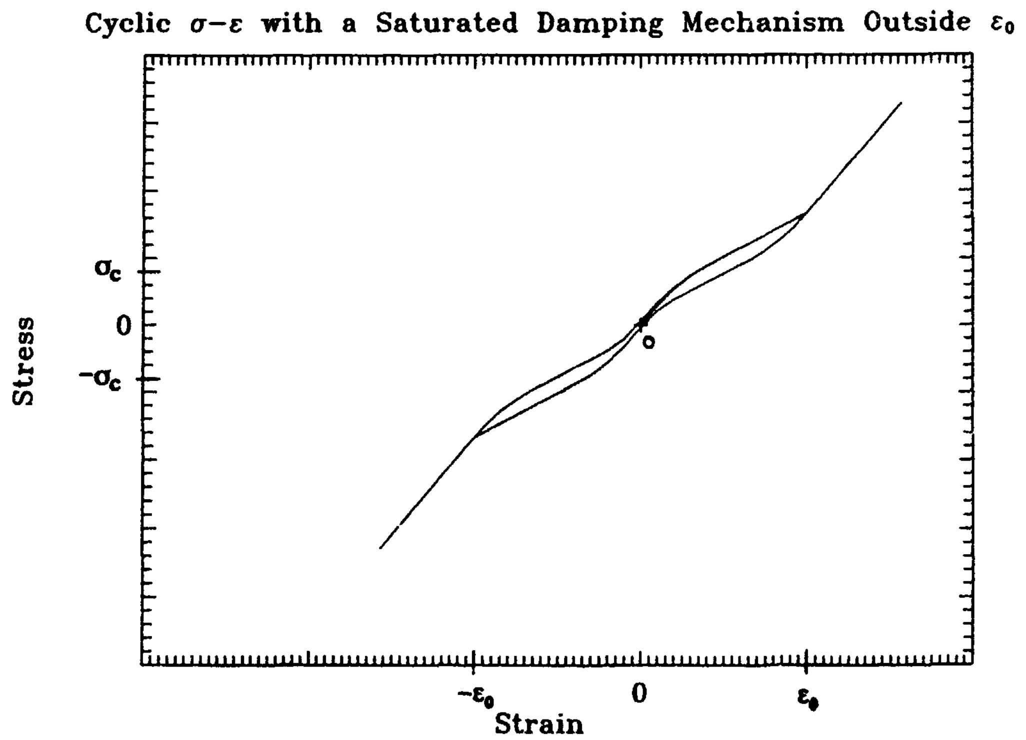


Fig. 2. Generalized Macroscopic Hysteresis of Nonlinear Damping Materials.

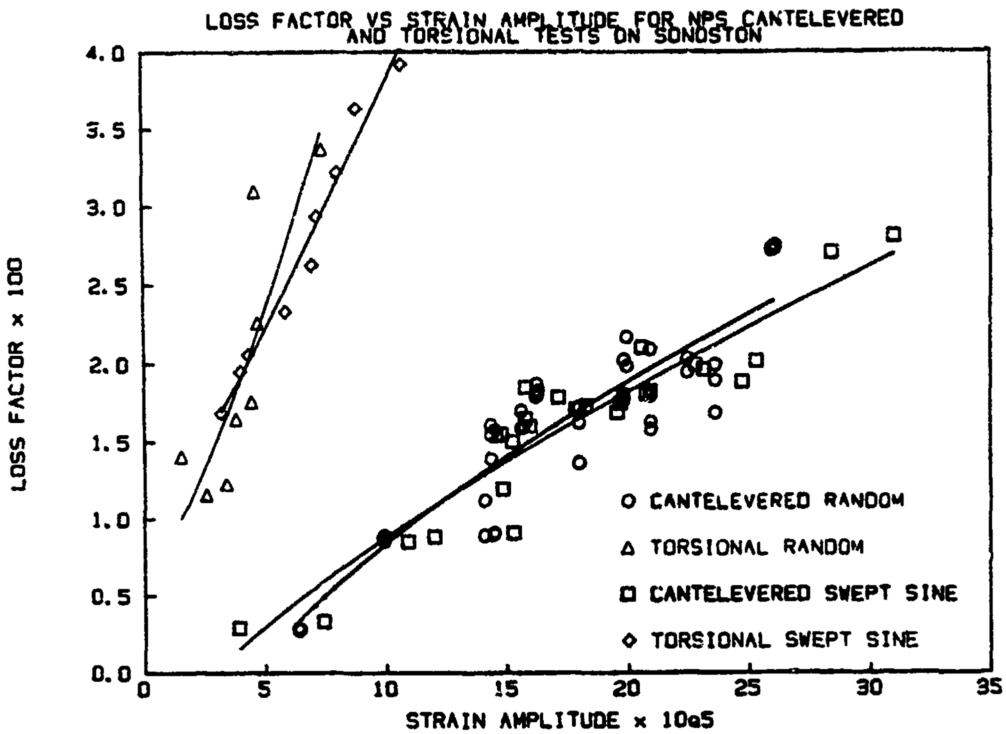


Fig 3. Strain Dependent Damping of Cu-Mn in Separate Bending and Torsion Tests [3].

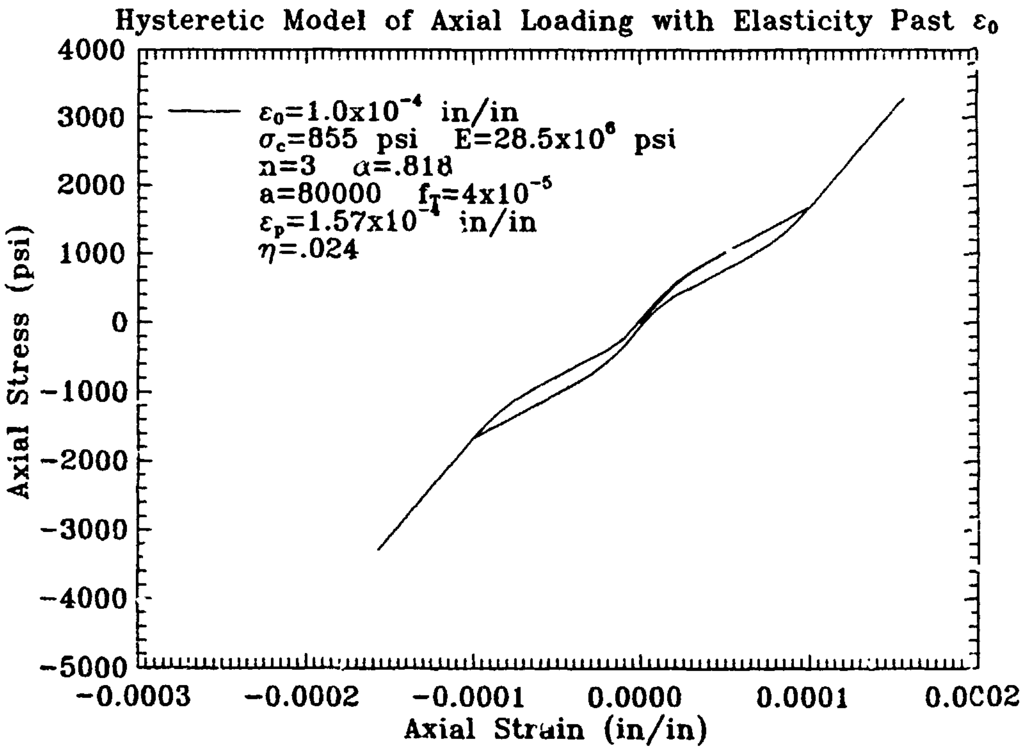


Fig. 4. Hysteretic Behavior Calculated for Pure Axial Loading.

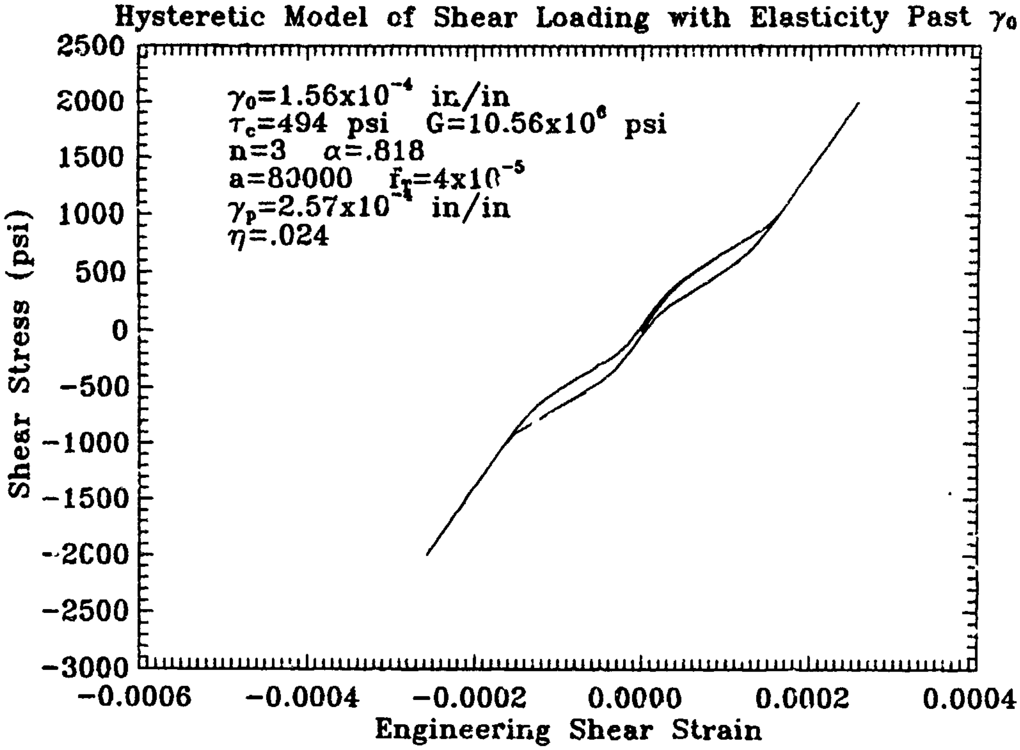


Fig. 5. Hysteretic Behavior Calculated for Pure Shear Loading.

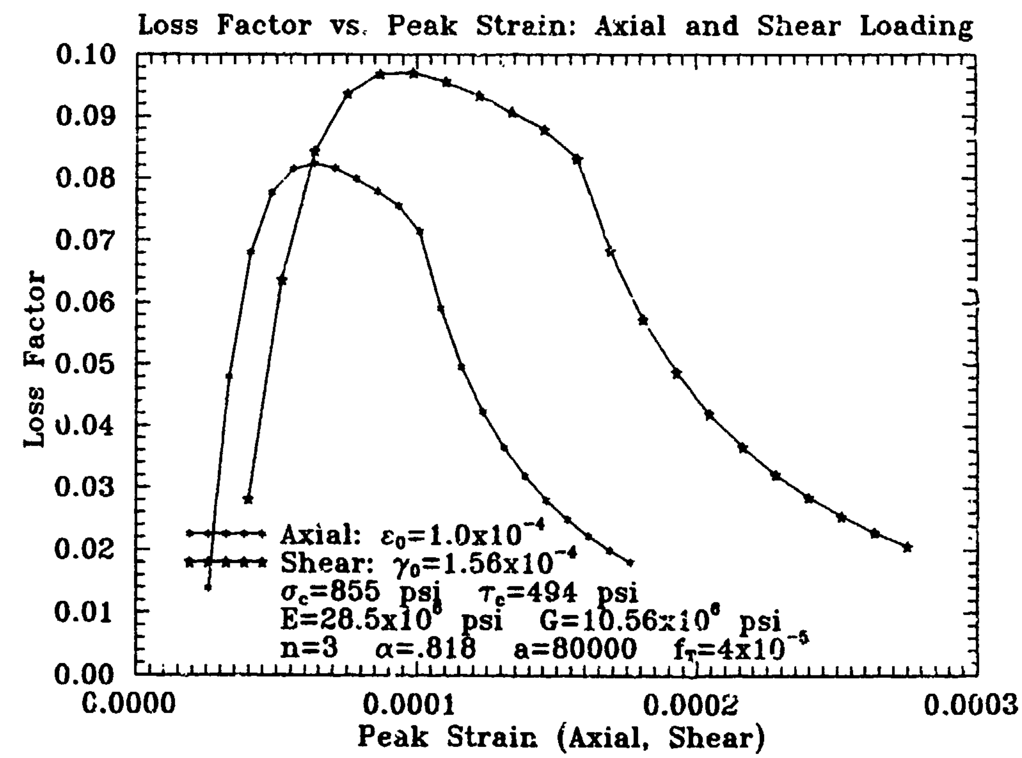


Fig. 6. Strain Amplitude Dependent Damping for Pure Axial and Shear Loading.

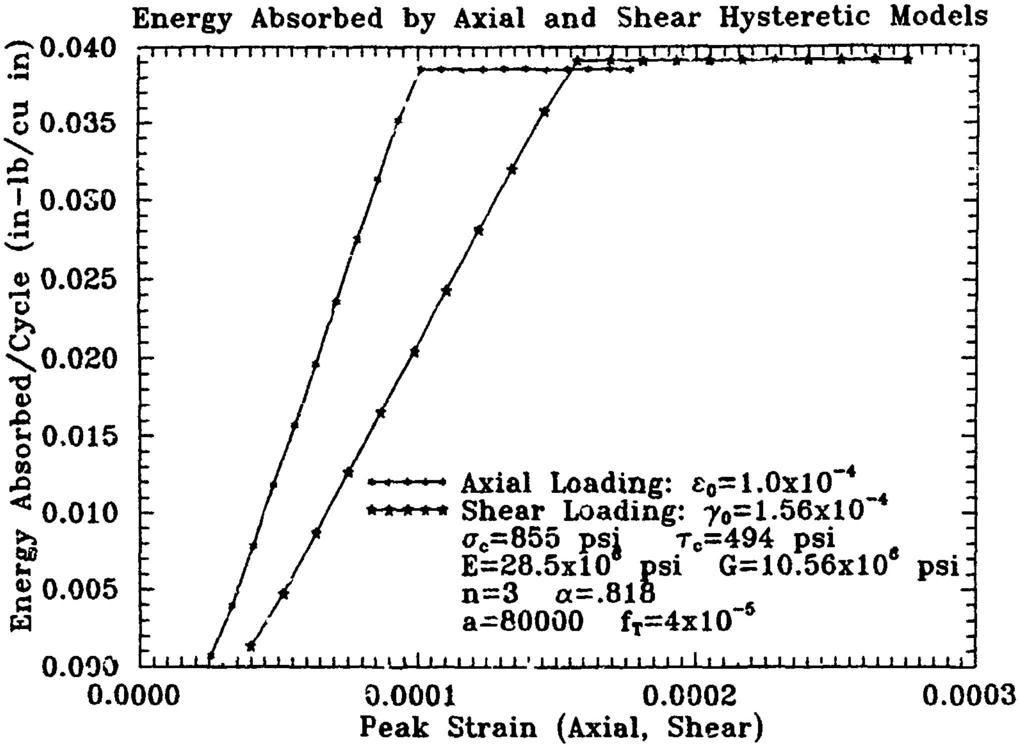


Fig. 7. Energy Absorbed in Axial and Shear vs. Peak Axial and Shear Strains.

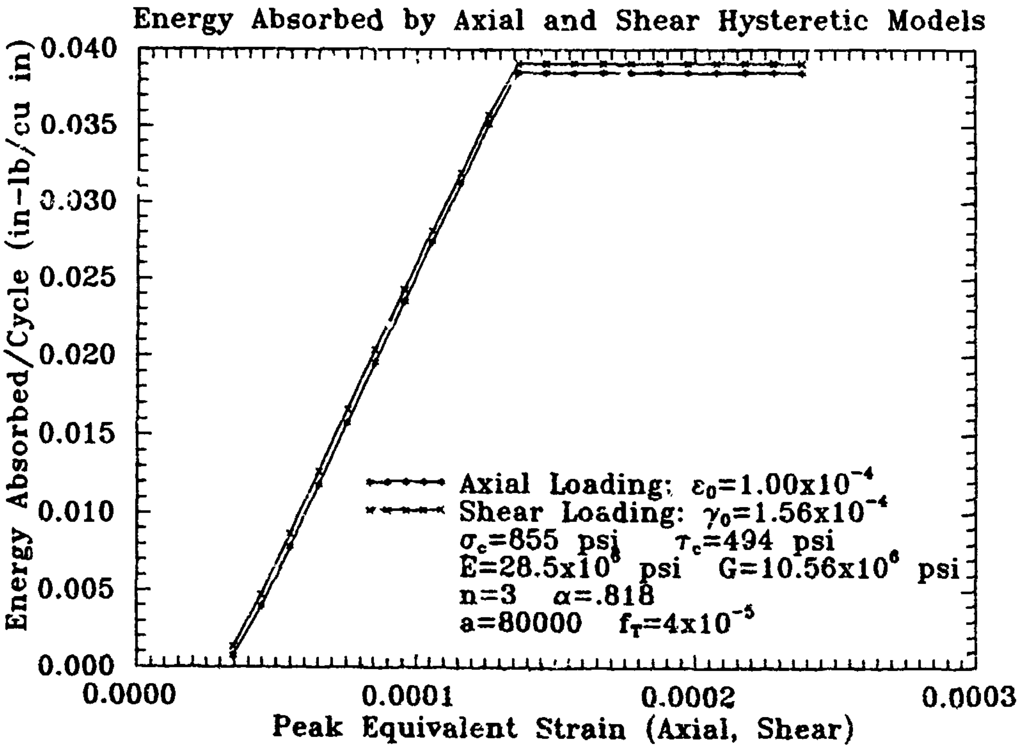


Fig. 8. Energy Absorbed in Axial and Shear vs. Peak Equivalent Strain.

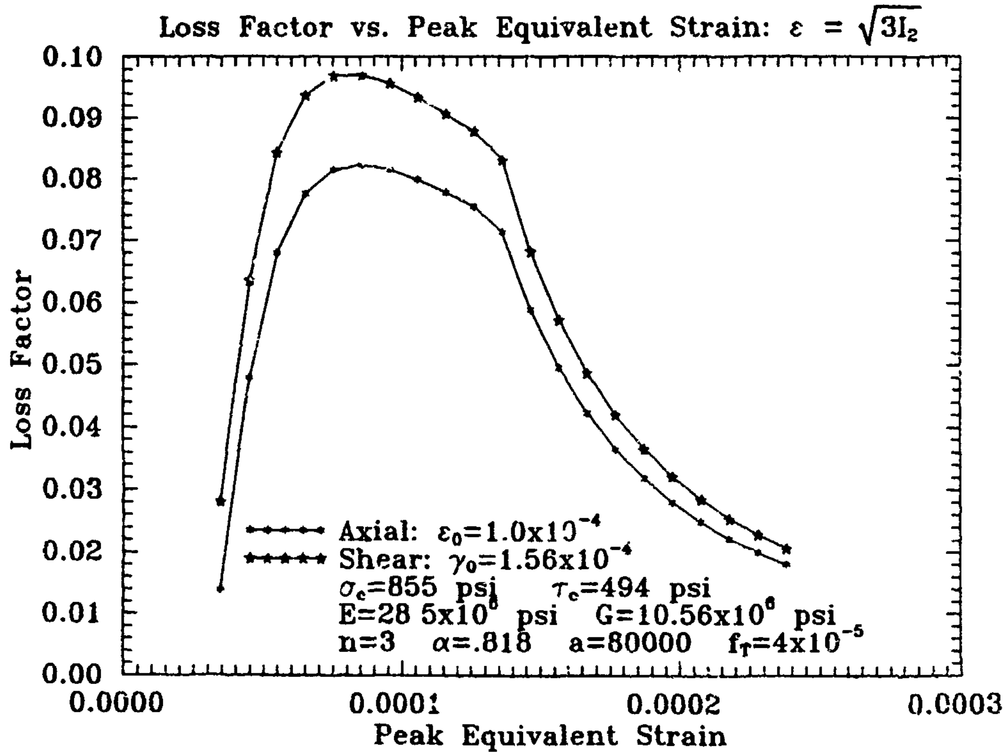
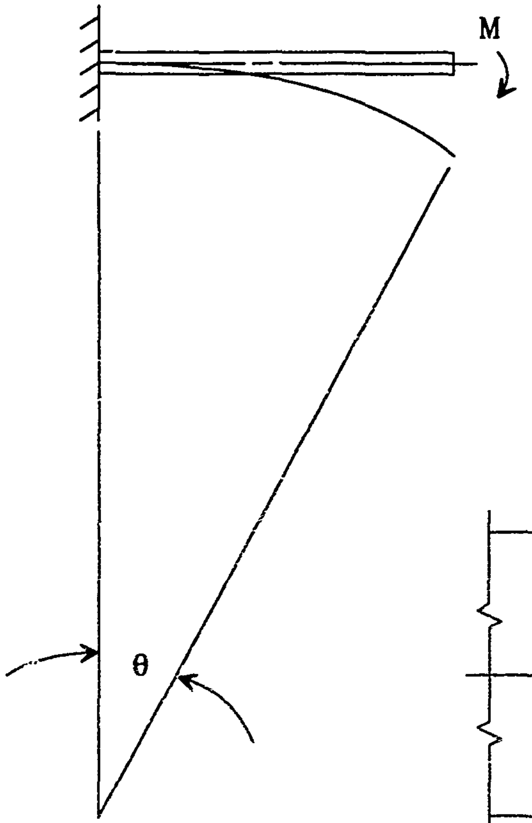


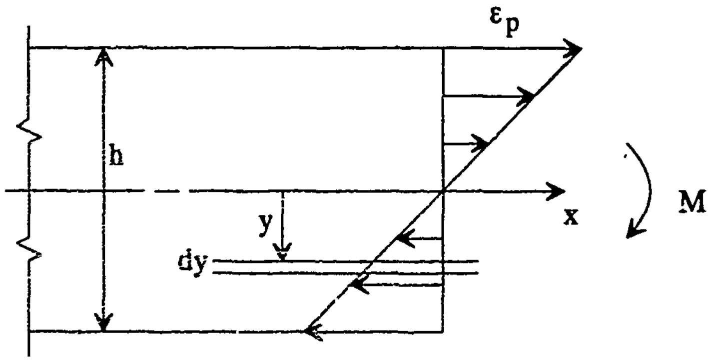
Fig. 9. Amplitude Dependent Damping in Axial and Shear vs. Peak Equivalent Strain.

Bending Beam (Length: L, Width: b, Thickness: h)

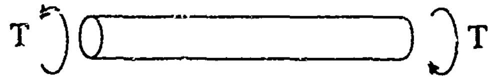


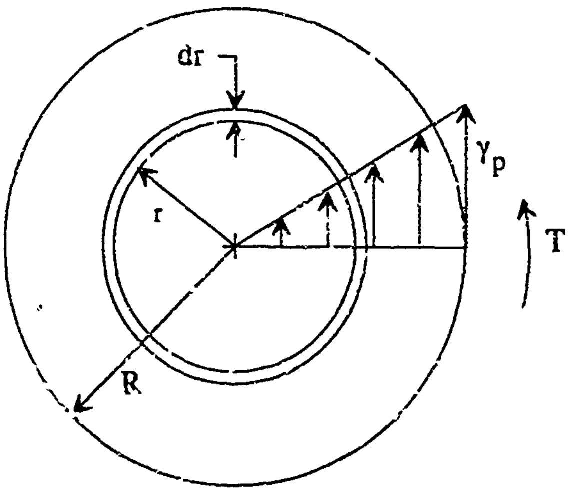
$$\theta = \frac{2\varepsilon_p L}{h}$$

$$M = -\int_A y\,\sigma\,dA \approx -b\sum_{i=1}^{N} y_i\,\sigma_i\,\Delta y$$



Twisting Shaft (Length: L, Radius R)





$$\phi = \frac{\gamma_p L}{R}$$

$$T = \int_A r\,\tau\,dA \approx 2\pi\sum_{i=1}^{N} (r_i)^2\,\tau_i\,\Delta r$$

Fig. 10. Schematic Drawing of Strain Profile in Bending and Torsion Geometries.

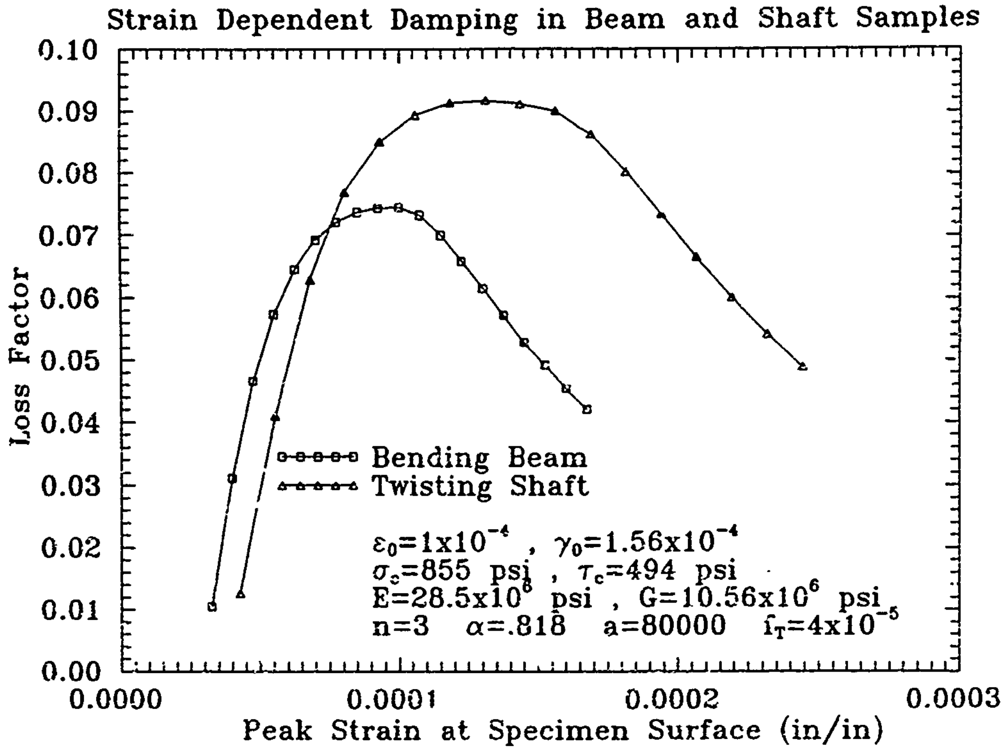


Fig. 11. Amplitude Dependent Damping in Bending and Torsion vs. Peak Surface Strain.

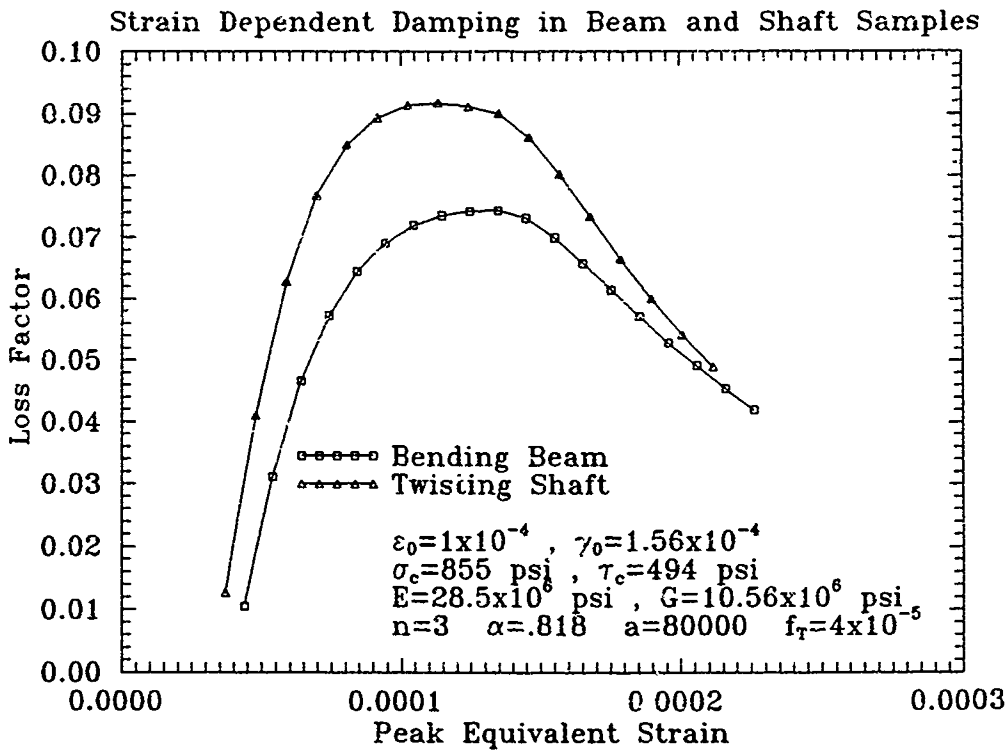


Fig. 12. Amplitude Dependent Damping in Bending and Torsion vs. Peak Equivalent Strain.

THIS
PAGE
IS
MISSING
IN
ORIGINAL
DOCUMENT

REFERENCES

1. Perkins, A.J., G.R. Edwards, and N.A. Hills, "Materials Approaches to Ship Silencing," Report No. NPS-59Ps74061, Naval Postgraduate School, Monterey, California, June, 1974.

2. Schetky, L.M. and J. Perkins, "The 'Quiet' Alloys," *Machine Design*, April 6, 1978, pp. 202-206.

3. Dew, D., "Strain Dependent Damping Characteristics of a High Damping Manganese-Copper Alloy," Report No. NPS69-86-007, Naval Postgraduate School, Monterey, California, September, 1986.

4. Suzuki, K., T. Fujita, and M. Hasebe, "Damping Capacity and Mechanical Properties of Sintered Fe-Cr-Mo High Damping Alloys," *Proceedings of the 6th International Conference on Internal Friction and Ultrasonic Attenuation in Solids*, Hasiguti, R. and Mikoshiba, N. (ed.), University of Tokyo Press, 1977, pp. 757-761.

5. Schneider, W., P. Schrey, G. Hausch, and E. Török, "Damping Capacity of Fe-Cr and Fe-Cr Based High Damping Alloys," *Journal De Physique*, Colloque C5, Supplément au n° 10, Tome 42, October, 1981, pp. C5-635 - C5-639.

6. Masumoto, M., M. Hinai, and S. Sawaya, "Damping Capacity and Pitting Corrosion Resistance of Fe-Mo-Cr Alloys," *Transactions of the Japan Institute of Metals*, Vol. 25, No. 12, 1984, pp. 891-899.

7. Nippon Kokan K.K., Tranqaloy Data Sheets, 1981.

8. Kawabe, H. and K. Kuwahara, "A Consideration of the Strain Dependent Damping and Modulus in Ferromagnetic Metals," *Transactions of the Japan Institute of Metals*, Vol. 22, No. 5, 1981, pp. 301-308.

9. Lazan, B.J., "Effect of Damping Constants and Stress Distribution on the Resonance Response of Members," *Journal of Applied Mechanics*, Transactions of the ASME, Vol. 75, 1953, pp. 201-209.

10. Cochardt, A.W., "A Method for Determining the Internal Damping of Machine Members," *Journal of Applied Mechanics*, Transactions of the ASME, Vol. 76, 1954, pp. 257-262.

11. Krempl, E., "Models of Viscoplasticity: Some Comments on Equilibrium (Back) Stress and Drag Stress," *Acta Mechanica*, Vol. 69, 1987, pp. 25-42.

12. Miller, A.K. (ed.), *Unified Constitutive Equations for Creep and Plasticity*, Elsevier Applied Science, 1987.

13. Graesser, E.J. and Cozzarelli, F.A., "Shape Memory Alloys as New Materials for Aseismic Isolation," ASCE, *Journal of Engineering Mechanics*, scheduled for publication in Nov. 1991.

14. Graesser E.J., "Multi-Dimensional Modeling of Hysteretic Materials Including Shape Memory Alloys: Theory and Experiment," Ph.D. Dissertation, Dept. of Mechanical and Aerospace Engineering, State University of New York at Buffalo, Buffalo, New York, February, 1990.

15. Prager, W., "Strain Hardening Under Combined Stresses," *Journal of Applied Physics*, Vol. 16, pp. 837-840.

16. Graesser, E.J. and F.A. Cozzarelli, "Extension of a One-Dimensional Model of Hysteresis to Three Dimensions: Procedure and Verification," *Proceedings of the ASME Conference on High Temperature Constitutive Modeling: Theory and Application*, edited by A.D. Freed and K.P. Walker, part of the ASME Winter Meeting, Dec., 1991, Atlanta, Georgia.

17. Shames, I.H. and F.A. Cozzarelli, *Elastic and Inelastic Stress Analysis*, Prentice-Hall, 1991.

18. Timoshenko, S.P. and J.N. Goodier, *Theory of Elasticity*, Third Edition, McGraw-Hill, 1970.

APPENDIX A

In this appendix, we will examine the Poisson effect associated with isotropic material deformation with nonlinear inelastic effects. To do this first consider the strain tensor ε_{ij} associated with pure axial stressing, i.e. without off-diagonal (shear) strains. The total strain is made up of separate elastic and inelastic parts as follows:

$$\varepsilon_{ij} = \varepsilon_{ij}^{el} + \varepsilon_{ij}^{in}$$

where the elastic part (superscript el) is related to the stress by elasticity theory and where the inelastic part is associated with incompressible deformation mechanisms (i.e. mechanisms which operate without any associated volume changes; e.g. dislocation glide). Therefore, for the case of uniaxial (superscript u) stressing

$$\varepsilon_{ij} = \varepsilon_{ij}^{u} = \begin{bmatrix} \varepsilon^{el} & 0 & 0 \\ 0 & -\nu\varepsilon^{el} & 0 \\ 0 & 0 & -\nu\varepsilon^{el} \end{bmatrix} + \begin{bmatrix} \varepsilon^{in} & 0 & 0 \\ 0 & -\frac{1}{2}\varepsilon^{in} & 0 \\ 0 & 0 & -\frac{1}{2}\varepsilon^{in} \end{bmatrix}$$

Note that the total axial strain is specified to be aligned along the x coordinate of Cartesian space with $\varepsilon^{el} = \sigma/E$. This in turn leads to

$$\varepsilon_{ij} = \varepsilon_{ij}^{u} = \begin{bmatrix} \varepsilon^{el} + \varepsilon^{in} & 0 & 0 \\ 0 & -\nu\varepsilon^{el} - \frac{1}{2}\varepsilon^{in} & 0 \\ 0 & 0 & -\nu\varepsilon^{el} - \frac{1}{2}\varepsilon^{in} \end{bmatrix}$$

Thus it is clear that the Poisson induced strains are composed of an elastic part which is related to the elastic strain by Poisson's ratio and an inelastic part which is related to the inelastic strain by the coefficient 1/2. By this process it is clear that the inelastic part of strain satisfies the condition of incompressibility.

Now, by using $\varepsilon^{el} = \sigma/E$ and $\varepsilon = \varepsilon^{el} + \varepsilon^{in}$ let us evaluate the Poisson induced strains in more detail. First note that

$$-\nu\varepsilon^{el} - \frac{1}{2}\varepsilon^{in} = -\nu\frac{\sigma}{E} - \frac{1}{2}\left[\varepsilon - \frac{\sigma}{E}\right] = \left[\frac{1}{E}\left(\frac{1}{2} - \nu\right)\frac{\sigma}{\varepsilon} - \frac{1}{2}\right]\varepsilon$$

Thus, by defining a variable Poisson coefficient μ as

$$\mu = \frac{1}{2} - \frac{1}{E}\left(\frac{1}{2} - \nu\right)\frac{\sigma}{\varepsilon} \tag{A1}$$

it follows that

$$-\mu\varepsilon = -\nu\varepsilon^{el} - \frac{1}{2}\varepsilon^{in} \tag{A2}$$

$$\varepsilon^{u}_{ij} = \begin{bmatrix} \varepsilon & 0 & 0 \\ 0 & -\mu\varepsilon & 0 \\ 0 & 0 & -\mu\varepsilon \end{bmatrix} \tag{A3}$$

Now, from Eq (A3) we can compute the strain rate tensor $\dot{\varepsilon}^{u}_{ij}$

$$\dot{\varepsilon}^{u}_{ij} = \begin{bmatrix} \dot{\varepsilon} & 0 & 0 \\ 0 & -\mu\dot{\varepsilon} - \dot{\mu}\varepsilon & 0 \\ 0 & 0 & -\mu\dot{\varepsilon} - \dot{\mu}\varepsilon \end{bmatrix}$$

From this let us proceed to evaluate the Poisson induced strain rates in more detail. Considering Eq. (A1) it follows that

$$\dot{\mu} = -\frac{1}{E}\left[\frac{1}{2}-\nu\right]\left[-\frac{\sigma}{\varepsilon^2}\dot{\varepsilon}+\frac{1}{\varepsilon}\dot{\sigma}\right]$$

and this leads to

$$-\mu\dot{\varepsilon}-\dot{\mu}\varepsilon = -\frac{1}{2}\dot{\varepsilon}+\frac{1}{2}\left[\frac{1}{2}-\nu\right]\dot{\sigma} = -\frac{1}{2}\dot{\varepsilon}+\left[\frac{1}{2}-\nu\right]\dot{\varepsilon}^{el}$$

$$= -\nu\,\dot{\varepsilon}^{el}-\frac{1}{2}(\dot{\varepsilon}-\dot{\varepsilon}^{el}) = -\nu\,\dot{\varepsilon}^{el}-\frac{1}{2}\dot{\varepsilon}^{in} \tag{A4}$$

Thus the Poisson induced strain rate decomposes into two parts: one part is an elastic component and is related to the axial strain rate by the Poisson ratio, and an inelastic part that is associated with the inelastic strain rate in a manner consistent with incompressibility.

By using $\dot{\varepsilon}^{el} = \dot{\sigma}/E$ and $\dot{\varepsilon} = \dot{\varepsilon}^{el} + \dot{\varepsilon}^{in}$ Eq. (A4) becomes

$$-\nu\,\dot{\varepsilon}^{el}-\frac{1}{2}\dot{\varepsilon}^{in} = \frac{1}{E}\left[\frac{1}{2}-\nu\right]\dot{\sigma}-\frac{1}{2}\dot{\varepsilon}$$

or

$$-\mu\dot{\varepsilon}-\dot{\mu}\varepsilon = -\left[\frac{1}{2}-\frac{1}{E}\left[\frac{1}{2}-\nu\right]\frac{d\sigma}{d\varepsilon}\right]\dot{\varepsilon}$$

Thus the variable Poisson coefficient associated with strain rate (which is denoted here as ρ) is as follows:

$$\rho = \frac{1}{2}-\frac{1}{E}\left[\frac{1}{2}-\nu\right]\frac{d\sigma}{d\varepsilon}$$

and

$$\dot{\varepsilon}^{u}_{ij} = \begin{bmatrix} \dot{\varepsilon} & 0 & 0 \\ 0 & -\rho\dot{\varepsilon} & 0 \\ 0 & 0 & -\rho\dot{\varepsilon} \end{bmatrix}$$

for axial stressing.

APPENDIX B

FORTRAN MAIN PROGRAM (all subroutines and function subprograms that are not specifically related to the calculations being made here are given in Appendix H):

```
C==========> Program Name: UNIAXIAL.FOR
C
C======> This program will carry out a Runge-Kutta integration on the
C        stress-strain equations of the SMA hysteretic model for axial
C        loading. Note that inelastic behavior is suppressed beyond EPSLNO.
C        The strain history is prescribed (i.e. input) as sinusoidal.
C        Stress is the output variable. Strain and stress pairs (i.e. the
C        hysteresis curve) are written to data files for plotting
C        with GRAPHER (which reads ASCII data arranged in column pairs).
C
C==========================================================================
C
C   NOMENCLATURE:
C
C     REAL CONSTANTS:
C       A       Material constant controlling shape of hysteresis
C       ALFA    =Ey/(E-Ey) where Ey is the inelastic modulus
C       AMP     Amplitude of strain input
C       E       Young's modulus
C       EPSLNO  Strain beyond which inelastic growth is supressed
C       FREQ    Frequency of cyclic strain input
C       FT      Material constant controlling size of hysteresis
C       N       Overstress power (controls sharpness of transition to inel.)
C       VNEW    Poisson's ratio
C       VNEWIN  Inelatic Poisson coefficient
C       Y       Stress where damping mechanisms are activated
C
C     INTEGER CONSTANTS:
C       NCYCLE  No. of cycles of oscillation
C       NPPCYC  No. of points per cycle to be used in integration
C
C     CHARACTER STRINGS:
C       FILENAME string for filename assignment to a FORTRAN unit number
C       SALF     string for material constant ALFA
C       SA       string for material constant A
C       SE       string for Young's modulus E
C       SE0      string for limiting strain EPSLNO
C       SEP      string for peak strain AMP
C       SFT      string for material constant FT
C       SN       string for overstress power N
C       SY       string for stress where damping mechanisms activate: Y
C       TITLE    descriptive title for run
C
C     VARIABLES:
C       ENGABS          Energy absorbed per cycle of oscillation
C       ENGSTO          Energy stored (=.5 * max. strain * max. stress)
C       ETA             Loss factor
C       SDC             Specific damping capacity
C
C     VARIABLE ARRAYS:
C       EPSLON(K)       Axial strain (index K rep. time)
C       STRESS(K)       Axial stress (index K rep. time)
C       Z(K)            Axial stress passed from INTFUN (index K rep. time)
C
C     SUBROUTINES:
C       DERIV           Contains the differential eqs. (invoked by INTFUN)
C       INTFUN          Integration routine (4th order forward Runge-Kutta)
C       MAXIM           Determines the min. & max. values of an array
C       STRLEN          Counts the number of characters in a string
C
C     FUNCTION SUBPROGRAMS:
C       ERF(X)          Error function of X
C       FACT(K)         K factorial
C       SGN(V)          Signum function of V
C       UNIT(X)         Unit step function of X
C
C==========================================================================
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER NEQ
      REAL*8 DERIV,FLOAT,T,TEND,TOL,Z(20),N
      DIMENSION EPSLON(1001),STRESS(1001)
```

```
      CHARACTER*20 FILENAME
      CHARACTER*70 TITLE,SY,SE,SN,SALF,SEO,SA,SPT,SEP
      CHARACTER*(*) SAR1,SAR2,SAR3,SAR4,SAR5,SAR6,SAR7,SARP
      PARAMETER (SAR1='e0=')
      PARAMETER (SAR2='Y=')
      PARAMETER (SAR3='E=')
      PARAMETER (SAR4='n=')
      PARAMETER (SAR5='alfa=')
      PARAMETER (SAR6='a=')
      PARAMETER (SAR7='fT=')
      PARAMETER (SARP='eP=')
      COMMON/BLOK1/AMP,OMEGA,Y,E,VNEWIN,ALFA,N,EPSLNO,A,FT
C
      NEQ = 1
C
      T = 0.0
      Z(1) = 0.0
C
C=====> Interactive input of filenames for the material data input file
C         and for output files.
C
      WRITE(*,*)'Enter the name of your input data file.'
      IFL=9
    1 READ(*,2)FILENAME
    2 FORMAT(A)
      WRITE(*,3)FILENAME
    3 FORMAT(' ',3X,A20)
      IF(IFL.EQ.9)THEN
        OPEN(IFL,FILE=FILENAME,STATUS='UNKNOWN')
      ELSE
        OPEN(IFL,FILE=FILENAME,STATUS='UNKNOWN')
      ENDIF
      IF (IFL.EQ.9) THEN
        WRITE(*,*)'Enter name for the plot file for STRESS vs. EPSLON'
        IFL=10
        GO TO 1
      ENDIF
      IF (IFL.EQ.10) THEN
        WRITE(*,*)'Enter name for the summary text file'
        IFL=11
        GO TO 1
      ENDIF
      REWIND 09
      REWIND 10
      REWIND 11
C
C=====> Read input quantities from input file.
C
      READ(9,5)TITLE
      READ(9,'(A)')SY
      READ(9,'(A)')SE
      READ(9,'(A)')SN
      READ(9,'(A)')SALF
    5 FORMAT(A)
      READ(9,*)E,VNEW
      READ(9,*)Y
      READ(9,*)ALFA
      READ(9,*)N
C
C=====> Interactive input of other material parameters, strain amplitude,
C        and number of cycles of loading to be used in calculations.
C
      WRITE(*,*)' Enter EPSLNO'
      READ(5,*)EPSLNO
      WRITE(*,*)' Enter character string for EPSLNO'
      READ(5,'(A)')SEO
      WRITE(*,*)' Enter A AND FT'
      READ(5,*)A,FT
      WRITE(*,*)' Enter character string for a'
      READ(5,'(A)')SA
      WRITE(*,*)' Enter character string for fT'
      READ(5,'(A)')SPT
      WRITE(*,*)' Enter FREQ and AMP'
      READ(5,*)FREQ,AMP
      WRITE(*,*)' Enter character string for Peak Strain (i.e. AMP)'
      READ(5,'(A)')SEP
      WRITE(*,*)' Enter Number of Cycles and Number of Points per Cycle'
      READ(5,*)NCYCLE,NPPCYC
C
C=====> Determine length of various character strings for later use.
C
```

```
      CALL STRLEN(SE0,IBSE0,IESE0)
      CALL STRLEN(SEP,IBSEP,IESEP)
      CALL STRLEN(SY,IBSY,IESY)
      CALL STRLEN(SE,IBSE,IESE)
      CALL STRLEN(SN,IBSN,IESN)
      CALL STRLEN(SALF,IBSALF,IESALF)
      CALL STRLEN(SA,IBSA,IESA)
      CALL STRLEN(SFT,IBSFT,IESFT)
      CALL STRLEN(SAR1,IBSAR1,IESAR1)
      CALL STRLEN(SAR2,IBSAR2,IESAR2)
      CALL STRLEN(SAR3,IBSAR3,IESAR3)
      CALL STRLEN(SAR4,IBSAR4,IESAR4)
      CALL STRLEN(SAR5,IBSAR5,IESAR5)
      CALL STRLEN(SAR6,IBSAR6,IESAR6)
      CALL STRLEN(SAR7,IBSAR7,IESAR7)
      CALL STRLEN(SARP,IBSARP,IESARP)
C
      ISUM1 = IESAR1 + IESE0 + 2 + IESARP + IESEP
      ISUM2 = IESAR2 + IESY + 2 + IESAR3 + IESE
      ISUM3 = IESAR4 + IESN + 2 + IESAR5 + IESALF
      ISUM4 = IESAR6 + IESA + 2 + IESAR7 + IESFT
C
C=====> Calculate the quantity Pi=3.14159267..., and other parameters
C
      PI=DACOS(-1.0D00)
C
      OMEGA = 2.*PI*FREQ
      VNEWIN = .5 - ALFA*(.5-VNEW)/(1.+ALFA)
      NSTEPS = NCYCLE*NPPCYC
      PRINT *,' NSTEPS=',NSTEPS
      PERIOD = 1./FREQ
      DELT = PERIOD/NPPCYC
      PRINT *,' DELT=',DELT
C
C=====> Set initial conditions.
C
      STRESS(1) = 0.
      EPSLON(1) = 0.
C
      WRITE(*,99999)
99999 FORMAT(4X,'ISTEP',5X,'STRAIN',9X,'STRESS')
C
C=====> Carry out the numerical integration
C
      DO 10 ISTEP = 1,NSTEPS
          CALL INTFUN(Z,T,DELT,NEQ)
        STRESS(ISTEP+1) = Z(1)
        EPSLON(ISTEP+1) = AMP * DSIN(OMEGA*T)
        WRITE(*,'(3X,I6,4(3X,E10.4))')ISTEP,EPSLON(ISTEP+1),Z(1)
   10 CONTINUE
C
C=====> Compute damping.
C
      CALL MAXIM(STRESS,NSTEPS,STRMIN,STRMAX)
      ENGABS = 0.
      ISTART = NPPCYC/4. + 1
      IEND = ISTART + NPPCYC-1
      DO 50 I=ISTART,IEND
        ENGABS = ENGABS + .5 * (STRESS(I+1) + STRESS(I)) *
     >                              (EPSLON(I+1) - EPSLON(I))
  50  CONTINUE
      ENGSTO = .5*STRMAX*AMP
      ETA = ENGABS/(2.*PI*ENGSTO)
      SDC = ETA*2*PI
C
C=====> Write results to output data file (unit 10) and to output
C        text file (unit 11).
C
      WRITE(10,'(1X,2(E16.8,'','');')(EPSLON(I),STRESS(I),I=1,NSTEPS)
C
      WRITE(11,'(5X,''0 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM1,SAR1,SE0(IBSE0:IESE0),SARP,SEP(IBSEP:IESEP)
      WRITE(11,'(5X,''1 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM2,SAR2,SY(IBSY:IESY),SAR3,SE(IBSE:IESE)
      WRITE(11,'(5X,''2 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM3,SAR4,SN(IBSN:IESN),SAR5,SALF(IBSALF:IESALF)
      WRITE(11,'(5X,''3 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM4,SAR6,SA(IBSA:IESA),SAR7,SFT(IBSFT:IESFT)
      WRITE(11,'(5X,''4 28 "h='',E9.4,2X,''SDC='',E9.4,''"'')')ETA,SDC
      PRINT *,'  LOSS FACTOR=',ETA,'  SDC=',SDC
C
```

```
 1000 FORMAT(A70)
 1001 FORMAT(2A60)
 1200 FORMAT(' ',3X,'1',5X,I7)
 1201 FORMAT(' ',3X,'2',5X,I7)
 1500 FORMAT(' ',5X,E10.4,5X,E10.4)
 1501 FORMAT(' ',5X,E10.4,5X,E10.4,5X,E10.4)
      STOP
      END


      SUBROUTINE DERIV(Z,T,ZDOT)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER NEQ
      REAL*8 OMEGA,T,Z(20),ZDOT(20),NN,LOADD
      COMMON/BLOK1/AMP,OMEGA,Y,E,VNEWIN,ALFA,NN,EPSLN0,A,PT
C
      EPSLON = AMP * DSIN(OMEGA*T)
      EPDOT = AMP * OMEGA * DCOS(OMEGA*T)
      LOADD = EPSLON*EPDOT
      DLTAEP = DABS(EPSLON) - EPSLN0
      IF(DLTAEP.GT.0.0)THEN
        BETA=(E*ALFA)*( EPSLON - Z(1)/E +
     >          PT*ERF(2.*(1.+VNEWIN)/3.*A*EPSLON)*UNIT(-LOADD) )
        IF(NN.EQ.1.0)THEN
           ZDOT(1) = E*( EPDOT - 2.*(1.+VNEWIN)/3.*DABS(EPDOT)*
     >           (Z(1)-BETA)/Y )
        ELSE
           ZDOT(1) = E*( EPDOT - 2.*(1.+VNEWIN)/3.*DABS(EPDOT)*
     >           (DABS(Z(1)-BETA)/Y)**(NN-1)*(Z(1)-BETA)/Y )
        ENDIF
      ELSE
        ZDOT(1) = E*EPDOT
      ENDIF
      RETURN
      END
C
C==========> End of UNIAXIAL.FOR
```

EXAMPLE OF AN INPUT FILE:

```
SMA Hysteretic Model with Elasticity Outside e0
855 psi
28.5x106 psi
3
.818
 28.5E06  0.35
 855.
 0.818
 3
```

APPENDIX C

FORTRAN MAIN PROGRAM (all subroutines and function subprograms that are not specifically re[illegible] to the calculations being made here are given in Appendix H):

```
C=========>Program Name: SHEAR.FOR
C
C=====> This program will carry out a Runge-Kutta integration on the
C       stress-strain equations of the SMA hysterotic model for SHEAR
C       loading. Note that inelastic behavior is supressed beyond GAMMA0.
C       Calculated results for the stress-strain response are written
C       to user defined output data files where the data is arranged
C       in two colums with strain in the first column and stress in the
C       second. The loading sequence (hysteresis) starts with the first
C       line of the file and proceeds in order to the last line.
C       The resulting data files can be used to plot the curves via
C       GRAPHER, which reads ASCII data file with data
C       arranged in column pairs.
C
C=========================================================================
C
C   NOMENCLATURE:
C
C     REAL CONSTANTS:
C       A       Material constant controlling shape of hysteresis
C       ALFA    =Ey/(E-Ey) where Ey is the inelastic modulus
C       AMP     Amplitude of strain input
C       E       Young's modulus
C       G       Shear modulus
C       GAMMA0  Strain beyond which inelastic growth is supressed
C       FREQ    Frequency of cyclic strain input
C       PT      Material constant controlling size of hysteresis
C       N       Overstress power (controls sharpness of transition to inel.)
C       VNEW    Poisson's ratio
C       Y       Axial Stress where damping mechanisms are activated
C       YS      Shear Stress where damping mechanisms are activated
C
C     INTEGER CONSTANTS:
C       NCYCLE No. of cycles of oscillation
C       NPPCYC No. of points per cycle to be used in integration
C
C     CHARACTER STRINGS:
C       FILENAME string for filename assignment to a FORTRAN unit number
C       SALF     string for material constant ALFA
C       SA       string for material constant A
C       SG       string for the shear modulus G
C       SG0      string for limiting strain GAMMA0
C       SGP      string for peak strain AMP
C       SPT      string for material constant PT
C       SN       string for overstress power N
C       SYS      string for shear stress where damping mechanisms activate: YS
C       TITLE    descriptive title for run
C
C     VARIABLES:
C       ENGABS          Energy absorbed per cycle of oscillation
C       ENGSTO          Energy stored (=.5 * max. strain * max. stress)
C       ETA             Loss factor
C       SDC             Specific damping capacity
C
C     VARIABLE ARRAYS:
C       GAMMA(K)        Shear strain (index K rep. time)
C       TAU(K)          Shear stress (index K rep. time)
C       Z(K)            Shear stress passed from INTFUN (index K rep. time)
C
C     SUBROUTINES:
C       DERIV           Contains the differential eqs. (invoked by INTFUN)
C       INTFUN          Integration routine (4th order forward Runge-Kutta.
C       MAXIM           Determines the min. & max. values of an array
C       STRLEN          Counts the number of characters in a string
C
C     FUNCTION SUBPROGRAMS:
C       ERF(X)          Error function of X
C       FACT(K)         K factorial
C       SGN(V)          Signum function of V
C       UNIT(X)         Unit step function of X
C
C=========================================================================
```

```
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER NEQ
      REAL*8 DERIV,FLOAT,T,TEND,TOL,Z(20),N
      DIMENSION GAMMA(1001),TAU(1001)
      CHARACTER*20 FILENAME
      CHARACTER*70 TITLE,SYS,SG,SN,SALF,SG0,SGP,SA,SFT
      CHARACTER*(*) SAR1,SAR2,SAR3,SAR4,SAR5,SAR6,SAR7,SARP
      PARAMETER (SAR1='g0=')
      PARAMETER (SAR2='Ys=')
      PARAMETER (SAR3='G=')
      PARAMETER (SAR4='n=')
      PARAMETER (SAR5='alfa=')
      PARAMETER (SAR6='a=')
      PARAMETER (SAR7='fT=')
      PARAMETER (SARP='gP=')
      COMMON/BLOK1/AMP,OMEGA,YS,E,G,ALFA,N,GAMMA0,A,FT,RAD3
C
      NEQ = 1
C
      T = 0.0
      Z(1) = 0.0
C
C=====> Interactive input of filenames for material data input file, and
C       for output files contaning data and text.
C
      WRITE(*,*)'Enter the name of your input data file.'
      IFL=9
    1 READ(*,2)FILENAME
    2 FORMAT(A)
      WRITE(*,3)FILENAME
    3 FORMAT(' ',3X,A20)
      IF(IFL.EQ.9)THEN
        OPEN(IFL,FILE=FILENAME,STATUS='UNKNOWN')
      ELSE
        OPEN(IFL,FILE=FILENAME,STATUS='UNKNOWN')
      ENDIF
      IF (IFL.EQ.9) THEN
        WRITE(*,*)'Enter name for the plot file for TAU vs. GAMMA'
        IFL=10
        GO TO 1
      ENDIF
      IF (IFL.EQ.10) THEN
        WRITE(*,*)'Enter name for the summary text file'
        IFL=11
        GO TO 1
      ENDIF
      REWIND 09
      REWIND 10
      REWIND 11
C
C=====> Read input quantities from input file.
C
      READ(9,5)TITLE
      READ(9,'(A)')SYS
      READ(9,'(A)')SG
      READ(9,'(A)')SN
      READ(9,'(A)')SALF
    5 FORMAT(A)
C
C=====> Read in uniaxial properties; convert to shear properties later.
C
      READ(9,*)E,VNEW
      READ(9,*)Y
      READ(9,*)ALFA
      READ(9,*)N
C
C=====> Interactive input of other material parameters, strain amplitude,
C       and number of cycles of loading to be used in calculations.
C
      WRITE(*,*)' Enter GAMMA0'
      READ(5,*)GAMMA0
      WRITE(*,*)' Enter character string for GAMMA0'
      READ(5,'(A)')SG0
      WRITE(*,*)' Enter A AND FT'
      READ(5,*)A,FT
      WRITE(*,*)' Enter character string for a'
      READ(5,'(A)')SA
      WRITE(*,*)' Enter character string for fT'
      READ(5,'(A)')SFT
```

```
      WRITE(*,*)' Enter FREQ and AMP of shear strain loading'
      READ(5,*)FREQ,AMP
      WRITE(*,*)' Enter character string for peak shear strain'
      READ(5,'(A)')SGP
      WRITE(*,*)' Enter Number of Cycles and Number of Points per Cycle'
      READ(5,*)NCYCLE,NPPCYC
C
C=====> Determine length of various character strings for later use.
C
      CALL STRLEN(SG0,IBSG0,IESG0)
      CALL STRLEN(SGP,IBSGP,IESGP)
      CALL STRLEN(SYS,IBSYS,IESYS)
      CALL STRLEN(SG,IBSG,IESG)
      CALL STRLEN(SN,IBSN,IESN)
      CALL STRLEN(SALF,IBSALF,IESALF)
      CALL STRLEN(SA,IBSA,IESA)
      CALL STRLEN(SFT,IBSFT,IESFT)
      CALL STRLEN(SAR1,IBSAR1,IESAR1)
      CALL STRLEN(SARP,IBSARP,IESARP)
      CALL STRLEN(SAR2,IBSAR2,IESAR2)
      CALL STRLEN(SAR3,IBSAR3,IESAR3)
      CALL STRLEN(SAR4,IBSAR4,IESAR4)
      CALL STRLEN(SAR5,IBSAR5,IESAR5)
      CALL STRLEN(SAR6,IBSAR6,IESAR6)
      CALL STRLEN(SAR7,IBSAR7,IESAR7)
C
      ISUM1 = IESAR1 + IESG0 + 2 + IESARP + IESGP
      ISUM2 = IESAR2 + IESY + 2 + IESAR3 + IESE
      ISUM3 = IESAR4 + IESN + 2 + IESAR5 + IESALF
      ISUM4 = IESAR6 + IESA + 2 + IESAR7 + IESFT
C
C=====> Calculate the quantity Pi=3.14159267...., and other parameters.
C
      PI=DACOS(-1.0D00)
      RAD3=3.**.5
      G = E/(2.*(1+VNEW))
      YS = Y/RAD3
      OMEGA = 2.*PI*FREQ
      PRINT *,' PI=',PI,' OMEGA=',OMEGA
      NSTEPS = NCYCLE*NPPCYC
      PRINT *,' NSTEPS=',NSTEPS
      PERIOD = 1./FREQ
      DELT = PERIOD/NPPCYC
      PRINT *,' DELT=',DELT
C
C=====> Set initial conditions
C
      TAU(1) = 0.
      GAMMA(1) = 0.
C
      WRITE(*,99999)
99999 FORMAT(4X,'ISTEP',5X,'TIME',9X,'Z1',11X,'Z2')
C
C=====> Carry out the numerical integration
C
      DO 10 ISTEP = 1,NSTEPS
          CALL INTFUN(Z,T,DELT,NEQ)
        TAU(ISTEP+1) = Z(1)
        GAMMA(ISTEP+1) = AMP * DSIN(OMEGA*T)
        WRITE(*,'(5X,I6,4(3X,E10.4))')ISTEP,GAMMA(ISTEP+1),Z(1)
   10 CONTINUE
C
C
      CALL MAXIM(TAU,NSTEPS,TAUMIN,TAUMAX)
      ENGABS = 0.
      ISTART = NPPCYC/4,+1
      IEND = ISTART + NPPCYC-1
      DO 50 I=ISTART,IEND
        ENGABS = ENGABS + .5 * (TAU(I+1) + TAU(I)) *
     >                                (GAMMA(I+1) - GAMMA(I))
   50 CONTINUE
      ENGSTO = .5*TAUMAX*AMP
      ETA = ENGABS/(2.*PI*ENGSTO)
      SDC = ETA*2*PI
C
C=====> Write the results to output data file (unit 10) and to output
C         text file (unit 11).
C
      WRITE(10,'(1X,2(E16.8,'','','')))')(GAMMA(I),TAU(I),I=1,NSTEPS+1)
      WRITE(11,'(1X,''0 '',I2,'' "'',A,A,2X,A,A ''"'')')
     >          ISUM1,SAR1,SG0(IBSG0:IESG0),SARP,SGP(IBSGP:IESGP)
```

```
      WRITE(11,'(1X,''1 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM2,SAR2,SYS(IBSYS:IESYS),SAR3,SG(IBSG:IESG)
      WRITE(11,'(1X,''2 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM3,SAR4,SN(IBSN:IESN),SAR5,SALF(IBSALF:IESALF)
      WRITE(11,'(1X,''3 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM4,SAR6,SA(IBSA:IESA),SAR7,SPT(IBSPT:IESPT)
      WRITE(11,'(1X,''4 28 "ETA='',E9.4,2X,''SDC='',E9.4,''"'')')ETA,SDC
      PRINT *,'  LOSS FACTOR=',ETA,'  SDC=',SDC
C
 1000 FORMAT(A70)
 1001 FORMAT(2A60)
 1200 FORMAT(' ',3X,'1',5X,I7)
 1201 FORMAT(' ',3X,'2',5X,I7)
 1500 FORMAT(' ',5X,E10.4,5X,E10.4)
 1501 FORMAT(' ',5X,E10.4,5X,E10.4,5X,E10.4)
      STOP
      END


      SUBROUTINE DERIV(Z,T,ZDOT)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER NEQ
      REAL*8 OMEGA,T,Z(20),ZDOT(20),NN,LOADD
      COMMON/BLOK1/AMP,OMEGA,YS,E,G,ALFA,NN,GAMMA0,A,PT,RAD3
C
      GAMMA = AMP * DSIN(OMEGA*T)
      GAMDOT = AMP * OMEGA * DCOS(OMEGA*T)
      DLTAGA = DABS(GAMMA) - GAMMA0
      IF(DLTAGA.LT.0.0)THEN
        BETA=(1./3.)*(E*ALFA)*( GAMMA - Z(1)/G +
     >      RAD3*PT*ERF(A*GAMMA/RAD3)*UNIT(-GAMMA*GAMDOT) )
        IF(NN.EQ.1.0)THEN
          ZDOT(1) = G*( GAMDOT - DABS(GAMDOT)*(Z(1)-BETA)/YS )
        ELSE
          ZDOT(1) = G*( GAMDOT - DABS(GAMDOT)*
     >         (DABS(Z(1)-BETA)/YS)**(NN-1) * (Z(1)-BETA)/YS )
        ENDIF
      ELSE
        ZDOT(1) = G*GAMDOT
      ENDIF
      RETURN
      END
C
C=========> SHEAR.FOR
```

EXAMPLE OF AN INPUT FILE:

```
SMA Hysteretic Model for Shear with Elasticity Outside g0
494 psi
10.56x106 psi
3
.818
 28.5E06  0.35
 853.
 0.818
 3
```

APPENDIX D

<u>FORTRAN MAIN PROGRAM</u> (all subroutines and function subprograms that are not specifically related to the calculations being made here are given in Appendix H):

```
C=========> Program Name: SDUA.FOR (for Strain Dependent UniAxial response)
C
C=====> This program is different from UNIAXIAL.FOR in that it
C       repeats the cyclic strain application over a specified range
C       for a given number of peak strains, and it calculates the loss
C       factor associated with each. Runge-Kutta integration will be applied
C       to the stress-strain equations of the axial SMA hysteretic model.
C       Note that inelastic behavior is supressed beyond EPSLNO.
C       Files will be generated which contain data for the plotting
C       of Peak Strain vs. Energy Abrorbed, Peak Strain vs. Loss Factor,
C       Peak Equivalent Srain vs. Energy Absorbed, and Peak Equivalent
C       Strain vs. Loss Factor. File Format will be that of GRAPHER;
C       ASCII data is arranged in column pairs. Data is arranged in
C       two vertical columns with strain (or equivalent strain) being
C       in the first column and loss factor (or energy absorbed) being
C       in the second column. A short set of data summarizing the run
C       is written to a user defined text file.
C
C=========================================================================
C
C   NOMENCLATURE:
C
C     REAL CONSTANTS:
C       A       Material constant controlling shape of hysteresis
C       ALPA    =Ey/(E-Ey) where Ey is the inelastic modulus
C       AMP     Amplitude of strain input
C       E       Young's modulus
C       EPSLNO  Strain beyond which inelastic growth is supressed
C       EPSPK1  Minimum peak strain
C       EPSPK2  Maximum peak strain
C       FREQ    Frequency of cyclic strain input
C       PT      Material constant controlling size of hysteresis
C       N       Overstress power (controls sharpness of transition to inel.)
C       VNEW    Poisson's ratio
C       VNEWIN  Inelatic Poisson coefficient
C       Y       Stress where damping mechanisms are activated
C
C     INTEGER CONSTANTS:
C       NCYCLE  No. of cycles of oscillation
C       NPPCYC  No. of points per cycle to be used in integration
C       NINC    No. of increments for the range of peak strain
C
C     CHARACTER STRINGS:
C       FILENAME string for filename assignment to a FORTRAN unit number
C       SALF     string for material constant ALPA
C       SA       string for material constant A
C       SE       string for Young's modulus E
C       SEO      string for limiting strain EPSLNO
C       SPT      string for material constant PT
C       SN       string for overstress power N
C       SY       string for stress where damping mechanisms activate: Y
C       TITLE    descriptive title for run
C
C     VARIABLE ARRAYS:
C       ENRGAB(I)     Energy absorbed in one cycle (I represents position)
C       ENRGST(I)     Energy absorbed in one cycle (I represents position)
C       EPSLON(K)     Axial strain (index K rep. time)
C       ETA(I)        Loss factor (I represents position)
C       PEAKST(I)     Peak strain (index I represents position)
C       PKEQST(I)     Peak equivalent strain (index I represents position)
C       STRESS(K)     Axial stress (index K rep. time)
C       Z(K)          Axial stress passed from INTFUN (index K rep. time)
C
C     SUBROUTINES:
C       DERIV         Contains the differential eqs. (invoked by INTFUN)
C       INTFUN        Integration routine (4th order forward Runge-Kutta)
C       MAXIM         Determines the min. & max. values of an array
C       STRLEN        Counts the number of characters in a string
C
C     FUNCTION SUBPROGRAMS:
C       ERF(X)        Error function of X
C       FACT(K)       K factorial
```

```
C       SGN(V)          Signum function of V
C       UNIT(X)         Unit step function of X
C
C===========================================================================
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER NEQ
      REAL*8 DERIV,FLOAT,T,TEND,Z(20),N
      DIMENSION EPSLON(1001),STRESS(1001),
     >          PEAKST(200),ETA(200),ENRGAB(200),ENRGST(200),PKEQST(200)
      CHARACTER*20 FILENAME
      CHARACTER*70 TITLE,SY,SE,SN,SALF,SE0,SA,SFT
      CHARACTER*(*) SAR1,SAR2,SAR3,SAR4,SAR5,SAR6,SAR8
      PARAMETER (SAR1='e0=')
      PARAMETER (SAR2='Y=')
      PARAMETER (SAR3='E=')
      PARAMETER (SAR4='n=')
      PARAMETER (SAR5='a=')
      PARAMETER (SAR6='a=')
      PARAMETER (SAR8='fT=')
      COMMON/BLOK1/AMP,OMEGA,Y,E,VNEW,VNEWIN,EPY,ALFA,N,EPSLN0,A,PT
C
      NEQ = 1
C
C=====> Interactive input of filenames for the material data input file
C        and for output files.
C
      WRITE(*,*)'Enter the name of your input data file.'
      IFL=9
    1 READ(*,2)FILENAME
    2 FORMAT(A)
      WRITE(*,3)FILENAME
    3 FORMAT(' ',3X,A20)
      IF(IFL.EQ.9)THEN
        OPEN(IFL,FILE=FILENAME,STATUS='UNKNOWN')
      ELSE
        OPEN(IFL,FILE=FILENAME,STATUS='UNKNOWN')
      ENDIF
      IF (IFL.EQ.9) THEN
        WRITE(*,*)'Enter name of plot file for Loss Fac. vs. Peak Str.'
        IFL=10
        GO TO 1
      ENDIF
      IF (IFL.EQ.10) THEN
        WRITE(*,*)'Enter name of the GRAPHER legend text file'
        IFL=11
        GO TO 1
      ENDIF
      IF (IFL.EQ.11) THEN
        WRITE(*,*)'Enter filename for Energy Absorbed vs. Peak Strain.'
        IFL=12
        GO TO 1
      ENDIF
      IF (IFL.EQ.12) THEN
        WRITE(*,*)'Enter filename for Loss Factor vs. Peak Eqiv. Strain'
        IFL=13
        GO TO 1
      ENDIF
      IF (IFL.EQ.13) THEN
        WRITE(*,*)'Enter filename for Energy Abs. vs. Peak Eqiv. Strain'
        IFL=14
        GO TO 1
      ENDIF
      REWIND 09
      REWIND 10
      REWIND 11
      REWIND 12
      REWIND 13
      REWIND 14
C
C=====> Read input quantities from input file.
C
      READ(9,5)TITLE
      READ(9,'(A)')SY
      READ(9,'(A)')SE
      READ(9,'(A)')SN
      READ(9,'(A)')SALF
    5 FORMAT(A)
      READ(9,*)E,VNEW
      READ(9,*)Y
      READ(9,*)ALFA
```

```
      READ(9,*)N
C
C=====> Interactive input of other material parameters, strain amplitude,
C       and number of cycles of loading to be used in calculations
C
      WRITE(*,*)' Enter EPSLNO'
      READ(5,*)EPSLNO
      WRITE(*,*)' EPSLNO=',EPSLNO
      WRITE(*,*)' Enter character string for EPSLNO'
      READ(5,'(A)')SEO
      WRITE(*,'(1X,'' SEO='',A)',SEO
      WRITE(*,*)' Enter A AND PT'
      READ(5,*)A,PT
      WRITE(*,*)' A=',A,' PT=',PT
      WRITE(*,*)' Enter character string for a'
      READ(5,'(A)')SA
      WRITE(*,'(1X,'' SA='',A)')SA
      WRITE(*,*)' Enter character string for PT'
      READ(5,'(A)')SPT
      WRITE(*,'(1X,'' SPT='',A)')SPT
      WRITE(*,*)' Enter FREQ, EPSPK1, & EPSPK2 (min and max peak str.)'
      READ(5,*)FREQ,EPSPK1,EPSPK2
      WRITE(*,*)' FREQ=',FREQ,' EPSPK1=',EPSPK1,' EPSPK2=',EPSPK2
      WRITE(*,*)' Enter No. of Cycles and No. of Points per Cycle'
      READ(5,*)NCYCLE,NPPCYC
      WRITE(*,*)' NCYCLE=',NCYCLE,' NPPCYC=',NPPCYC
      WRITE(*,*)' Enter NINC (No. of increments bet. peak strains)'
      READ(5,*)NINC
      WRITE(*,*)' NINC=',NINC
C
C=====> Determine length of various character strings for later use.
C
      CALL STRLEN(SEO,IBSEO,IESEO)
      CALL STRLEN(SY,IBSY,IESY)
      CALL STRLEN(SE,IBSE,IESE)
      CALL STRLEN(SN,IBSN,IESN)
      CALL STRLEN(SALF,IBSALF,IESALF)
      CALL STRLEN(SA,IBSA,IESA)
      CALL STRLEN(SPT,IBSPT,IESPT)
      CALL STRLEN(SAR1,IBSAR1,IESAR1)
      CALL STRLEN(SAR2,IBSAR2,IESAR2)
      CALL STRLEN(SAR3,IBSAR3,IESAR3)
      CALL STRLEN(SAR4,IBSAR4,IESAR4)
      CALL STRLEN(SAR5,IBSAR5,IESAR5)
      CALL STRLEN(SAR6,IBSAR6,IESAR6)
      CALL STRLEN(SAR8,IBSAR8,IESAR8)
C
      ISUM1 = IESAR1 + IESEO
      ISUM2 = IESAR2 + IESY +2 +IESAR3 + IESE
      ISUM3 = IESAR4 + IESN + 2 + IESAR5 + IESALF
      ISUM4 = IESAR6 + IESA + 2 + IESAR8 +IESPT
C
C=====> Calculate the quantity Pi=3.14159267..., and other parameters
C
      PI=DACOS(-1.0D00)
      OMEGA = 2.*PI*FREQ
      PRINT *,' PI=',PI,' OMEGA=',OMEGA
      VNEWIN = .5 - ALFA*(.5-VNEW)/(1.+ALFA)
      EPY = Y/E
      PRINT *,' VNEWIN=',VNEWIN,' EPY=',EPY
      NSTEPS = NCYCLE*NPPCYC
      NSTPP1 = NSTEPS + 1
      PRINT *,' NSTEPS=',NSTEPS
      PERIOD = 1./FREQ
      DELT = PERIOD/NPPCYC
      PRINT *,' DELT=',DELT
      EPSINC = (EPSPK2-EPSPK1)/NINC
      ISTART = NPPCYC/4. + 1
      IEND = ISTART + NPPCYC-1
C
C=====> Set initial conditions.
C
      STRESS(1) = 0.
      EPSLON(1) = 0.
C
C=====> Set up the loop for carrying out integration for each step in peak strain.
C
      NINCP1 = NINC + 1
      DO 7777 J=1,NINCP1
C
C=====> Clear the arrays used in the loop marked by 30 at the continue statement
```

```
C       Reinitialize T and Z(1).
C
        DO 10 K=1,NSTPP1
          EPSLON(K)=0.0
   10     STRESS(K)=0.0
C
        T = 0.0
        Z(1) = 0.0
C
C=====> Update the peak strain
C
        AMP = EPSPK1 + EPSINC*(J-1)
        PEAKST(J) = AMP
        PKEQST(J) = (1+VNEW)*AMP
C
C=====> Carry out the integration for the current peak strain
C
        DO 30 ISTEP = 1,NSTEPS
            CALL INTFUN(Z,T,DELT,NEQ)
          STRESS(ISTEP+1) = Z(1)
          EPSLON(ISTEP+1) = AMP * DSIN(OMEGA*T)
   30   CONTINUE
C
C=====> Compute damping for the current peak strain.
C
        STRMIN = 0.
        STRMAX = 0.
        CALL MAXIM(STRESS,NSTPP1,STRMIN,STRMAX)
        ENGABS = 0.
C
        DO 50 I=ISTART,IEND
          ENGABS = ENGABS + .5 * (STRESS(I+1) + STRESS(I)) *
     >                               (EPSLON(I+1) - EPSLON(I))
   50 CONTINUE
        ENGSTO = .5*STRMAX*AMP
        ENRGAB(J) = ENGABS
        ENRGST(J) = ENGSTO
        ETA(J) = ENGABS/(2.*PI*ENGSTO)
        WRITE(*,'(3X,I3,3(3X,E10.4))')J,PEAKST(J),ENRGAB(J),ETA(J)
 7777 CONTINUE
      CALL MAXIM(ETA,NINCP1,ETAMIN,ETAMAX)
C
C=====> Write results to output data file (unit 10, 12, 13, and 14) and to
C       output text file (unit 11).
C
      WRITE(10,'(1X,2(E16.8,'','',))')(PEAKST(I),ETA(I),I=1,NINCP1)
      WRITE(11,'(1X,''0 '',I2,'' "'',A,A,''"'')')
     >          ISUM1,SAR1,SE0(IBSE0:IESE0)
      WRITE(11,'(1X,''1 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >          ISUM2,SAR2,SY(IBSY:IESY),SAR3,SE(IBSE:IESE)
      WRITE(11,'(1X,''2 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >          ISUM3,SAR4,SN(IBSN:IESN),SAR5,SALF(IBSALF:IESALF)
      WRITE(11,'(1X,''3 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >          ISUM4,SAR6,SA(IBSA:IESA),
     >          SAR8,SFT(IBSFT:IESFT)
      WRITE(11,'(1X,''4 38 "ETAmax='',E9.4,2X,''ETAmin='',E9.4,''"'')')
     >ETAMAX,ETAMIN
      WRITE(12,'(1X,2(E16.8,'','',))')(PEAKST(I),ENRGAB(I),I=1,NINCP1)
      WRITE(13,'(1X,2(E16.8,'','',))')(PKEQST(I),ETA(I),I=1,NINCP1)
      WRITE(14,'(1X,2(E16.8,'','',))')(PKEQST(I),ENRGAB(I),I=1,NINCP1)
      PRINT *,'  ETAMAX=',ETAMAX,'  ETAMIN=',ETAMIN
C
 1000 FORMAT(A70)
 1001 FORMAT(2A60)
 1200 FORMAT(' ',3X,'1',5X,I7)
 1201 FORMAT(' ',3X,'2',5X,I7)
 1500 FORMAT(' ',5X,E10.4,5X,E10.4)
 1501 FORMAT(' ',5X,E10.4,5X,E10.4,5X,E10.4)
      STOP
      END


      SUBROUTINE DERIV(Z,T,ZDOT)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 OMEGA,T,Z(20),ZDOT(20),NN,LOADD
      COMMON/BLOK1/AMP,OMEGA,Y,E,VNEW,VNEWIN,EPY,ALFA,NN,EPSLN0,A,FT
C
      EPSLON = AMP * DSIN(OMEGA*T)
      EPDOT = AMP * OMEGA * DCOS(OMEGA*T)
      LOADD = EPSLON * EPDOT
```

```
      DLTAEP = DABS(EPSLON) - EPSLNO
C
      IF(DLTAEP.LT.0.0)THEN
        IF(DABS(EPSLON).GE.EPY)THEN
          DELTA = .5 - DABS(Z(1)/EPSLON)*(.5-VNEW)/E
        ELSE
          DELTA = VNEW
        ENDIF
        BETA=(E*ALFA)*( EPSLON - Z(1)/E +
     >         FT*ERF((2.*(1.+DELTA)/3.)*A*EPSLON)*UNIT(-LOADD) )
        IF(NN.EQ.1.0)THEN
           ZDOT(1) = E*( EPDOT - (2.*(1.+VNEWIN)/3.)*DABS(EPDOT)*
     >       (Z(1)-BETA)/Y )
        ELSE
           ZDOT(1) = E*( EPDOT - (2.*(1.+VNEWIN)/3.)*DABS(EPDOT)*
     >       (DABS(Z(1)-BETA)/Y)**(NN-1)*(Z(1)-BETA)/Y )
        ENDIF
      ELSE
        ZDOT(1) = E*EPDOT
      ENDIF
      RETURN
      END
C
C==========>SDUA.FOR
```

EXAMPLE OF AN INPUT FILE:

```
SMA Axial Hysteretic Model with Elasticity Outside e0
855 psi
28.5x106 psi  Vnew=.35
3
.818
 28.5E06  0.35
 855.
 0.818
 3
```

APPENDIX E

FORTRAN MAIN PROGRAM (all subroutines and function subprograms that are not specifically related to the calculations being made here are given in Appendix H):

```
C=========> Program Name: SDSH.FOR (for Strain Dependent SHear response)
C
C=====> This program is different from UNIAXIAL.FOR in that it
C       repeats the cyclic strain application over a specified range
C       for a given number of peak strains, and it calculates the loss
C       factor associated with each. Runge-Kutta integration will be applied
C       to the stress-strain equations of the SMA hysteretic model.
C       Note that inelastic behavior is supressed beyond GAMMA0.
C       The applied loading is that of a pure state of shear stress.
C       We specify a maximum peak strain to be used in the sinusiodal
C       application of shear strain. The resulting curves for stress and
C       strain will be stored in data arrays and then will be
C       used to calculate the loss factor (damping). The data file will also
C       be in a format compatible for plotting with GRAPHER.
C       ASCII data is arranged in column pairs. Data is arranged in
C       two vertical columns with strain (or equivalent strain) being
C       in the first column and loss factor (or energy absorbed) being
C       in the second column. A short set of data summarizing the run
C       is written to a user defined text file.
C
C
C=========================================================================
C
C    NOMENCLATURE:
C
C      REAL CONSTANTS:
C        A       Material constant controlling shape of hysteresis
C        ALFA    =Ey/(E-Ey) where Ey is the axial inelastic modulus
C        AMP     Amplitude of strain input
C        E       Young's modulus
C        G       Shear modulus {=E/[2(1+VNEW)]}
C        GAMMA0  Strain beyond which inelastic growth is supressed
C        GAMPK1  Minimum peak strain
C        GAMPK2  Maximum peak strain
C        FREQ    Frequency of cyclic strain input
C        PT      Material constant controlling size of hysteresis
C        N       Overstress power (controls sharpness of transition to inel.)
C        VNEW    Poisson's ratio
C        Y       Axial stress where damping mechanisms are activated
C        YS      Shear stress where damping mechanisms are activated
C
C      INTEGER CONSTANTS:
C        NCYCLE  No. of cycles of oscillation
C        NPPCYC  No. of points per cycle to be used in integration
C        NINC    No. of increments for the range of peak strain
C
C      CHARACTER STRINGS:
C        FILENAME string for filename assignment to a FORTRAN unit number
C        SALF     string for material constant ALFA
C        SA       string for material constant A
C        SG       string for Young's modulus G
C        SG0      string for limiting strain GAMMA0
C        SPT      string for material constant PT
C        SN       string for overstress power N
C        SYS      string for shear stress where damping mechanisms activate
C        TITLE    descriptive title for run
C
C      VARIABLE ARRAYS:
C        ENRGAB(I)      Energy absorbed in one cycle (I represents position)
C        ENRGST(I)      Energy absorbed in one cycle (I represents position)
C        GAMMA(K)       Shear strain (index K rep. time)
C        ETA(I)         Loss factor (I represents position)
C        PEAKST(I)      Peak strain (index I represents position)
C        PKEQST(I)      Peak equivalent strain (index I represents position)
C        TAU(K)         Shear stress (index K rep. time)
C        Z(K)           Shear stress passed from INTFUN (index K rep. time)
C
C      SUBROUTINES:
C        DERIV          Contains the differential eqs. (invoked by INTFUN)
C        INTFUN         Integration routine (4th order forward Runge-Kutta)
C        MAXIM          Determines the min. & max. values of an array
C        STRLEN         Counts the number of characters in a string
```

```
C
C     FUNCTION SUBPROGRAMS:
C       ERF(X)          Error function of X
C       FACT(K)         K factorial
C       SGN(V)          Signum function of V
C       UNIT(X)         Unit step function of X
C
C======================================================================
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER NEQ
      REAL*8 DERIV,FLOAT,T,TEND,TOL,Z(20),N
      DIMENSION GAMMA(1001),TAU(1001),PEAKST(201),ETA(201),PKEQST(201),
     >          ENRGAB(201),ENRGST(201)
      CHARACTER*20 FILENAME
      CHARACTER*70 TITLE,SYS,SG,SN,SALF,SG0,SA,SFT
      CHARACTER*(*) SAR1,SAR2,SAR3,SAR4,SAR5,SAR6,SAR7
      PARAMETER (SAR1='g0=')
      PARAMETER (SAR2='Ys=')
      PARAMETER (SAR3='G=')
      PARAMETER (SAR4='n=')
      PARAMETER (SAR5='a=')
      PARAMETER (SAR6='a=')
      PARAMETER (SAR7='fT=')
      COMMON/BLOK1/AMP,OMEGA,YS,E,G,ALFA,N,GAMMA0,A,FT,RAD3
C
      NEQ = 1
C
C=====> Interactve input of filenames for the material data input file
C       and for output files.
C
      WRITE(*,*)'Enter the name of your input data file.'
      IFL=9
    1 READ(*,2)FILENAME
    2 FORMAT(A)
      WRITE(*,3)FILENAME
    3 FORMAT(' ',3X,A20)
      IF(IFL.EQ.9)THEN
        OPEN(IFL,FILE=FILENAME,STATUS='UNKNOWN')
      ELSE
        OPEN(IFL,FILE=FILENAME,STATUS='UNKNOWN')
      ENDIF
      IF (IFL.EQ.9) THEN
        WRITE(*,*)'Enter file name for Loss Factor vs. Peak Strain'
        IFL=10
        GO TO 1
      ENDIF
      IF (IFL.EQ.10) THEN
        WRITE(*,*)'Enter name for the summary file'
        IFL=11
        GO TO 1
      ENDIF
      IF (IFL.EQ.11) THEN
        WRITE(*,*)'Enter filename for energy absorbed vs. peak strain'
        IFL=12
        GO TO 1
      ENDIF
      IF (IFL.EQ.12) THEN
        WRITE(*,*)'Enter filename for loss factor vs peak equiv. strain'
        IFL=13
        GO TO 1
      ENDIF
      IF (IFL.EQ.13) THEN
        WRITE(*,*)'Enter filename for energy abs. vs peak equiv. strain'
        IFL=14
        GO TO 1
      ENDIF
      REWIND 09
      REWIND 10
      REWIND 11
      REWIND 12
      REWIND 13
      REWIND 14
C
C=====> Read input quantities from input file.
C
      READ(9,5)TITLE
      READ(9,'(A)')SYS
      READ(9,'(A)')SG
      READ(9,'(A)')SN
      READ(9,'(A)')SALF
```

```
    5 FORMAT(A)
      READ(9,*)E,VNEW
      READ(9,*)Y
      READ(9,*)ALFA
      READ(9,*)N
C
C=====> Interactive input of other material parameters, strain amplitude,
C       and number of cycles of loading to be used in calculations.
C
      WRITE(*,*)' Enter GAMMA0'
      READ(5,*)GAMMA0
      WRITE(*,*)' GAMMA0=',GAMMA0
      WRITE(*,*)' Enter character string for GAMMA0'
      READ(5,'(A)')SG0
      WRITE(*,'(1X,''SG0='',A)')SG0
      WRITE(*,*)' Enter A AND FT'
      READ(5,*)A,FT
      WRITE(*,*)' A=',A,' FT=',FT
      WRITE(*,*)' Enter character string for a'
      READ(5,'(A)')SA
      WRITE(*,'(1X,''SA='',A)')SA
      WRITE(*,*)' Enter character string for fT'
      READ(5,'(A)')SFT
      WRITE(*,'(1X,''SFT='',A)')SFT
      WRITE(*,*)' Enter FREQ, GAMPK1, & GAMPK2 (Min & Max Peak Strains)'
      READ(5,*)FREQ,GAMPK1,GAMPK2
      WRITE(*,*)' FREQ=',FREQ,' GAMPK1=',GAMPK1,' GAMPK2=',GAMPK2
      WRITE(*,*)' Enter Number of Cycles and Number of Points per Cycle'
      READ(5,*)NCYCLE,NPPCYC
      WRITE(*,*)' NCYCLE=',NCYCLE,' NPPCYC=',NPPCYC
      WRITE(*,*)' Enter NINC (No. of increments between peak strains)'
      READ(5,*)NINC
      WRITE(*,*)' NINC=',NINC
C
C=====> Determine length of various character strings for later use.
C
      CALL STRLEN(SG0,IBSG0,IESG0)
      CALL STRLEN(SYS,IBSYS,IESYS)
      CALL STRLEN(SG,IBSG,IESG)
      CALL STRLEN(SN,IBSN,IESN)
      CALL STRLEN(SALF,IBSALF,IESALF)
      CALL STRLEN(SA,IBSA,IESA)
      CALL STRLEN(SFT,IBSFT,IESFT)
      CALL STRLEN(SAR1,IBSAR1,IESAR1)
      CALL STRLEN(SAR2,IBSAR2,IESAR2)
      CALL STRLEN(SAR3,IBSAR3,IESAR3)
      CALL STRLEN(SAR4,IBSAR4,IESAR4)
      CALL STRLEN(SAR5,IBSAR5,IESAR5)
      CALL STRLEN(SAR6,IBSAR6,IESAR6)
      CALL STRLEN(SAR7,IBSAR7,IESAR7)
C
      ISUM1 = IESAR1 + IESG0
      ISUM2 = IESAR2 + 7 +IESAR3 + 6
      ISUM3 = IESAR4 + IESN + 7 + IESAR5 + IESALF
      ISUM4 = IESAR6 + IESA + 2 + IESAR7 + IESFT
C
C
C=====> Calculate the quantity Pi=3.14159267..., and other parameters
C
      PI=DACOS(-1.0D00)
      RAD3 = 3.**.5
      G = E/(2.*(1+VNEW))
      YS = Y/RAD3
      OMEGA = 2.*PI*FREQ
      PRINT *,' PI=',PI,' OMEGA=',OMEGA
      NSTEPS = NCYCLE*NPPCYC
      PRINT *,' NSTEPS=',NSTEPS
      PERIOD = 1./FREQ
      DELT = PERIOD/NPPCYC
      DELGAM = (GAMPK2-GAMPK1)/NINC
      ISTART = NPPCYC/4. + 1
      IEND = ISTART + NPPCYC-1
C
C=====> Set up the loop for carrying out integration for each step in
C       peak strain.
C
      NINCP1 = NINC + 1
      DO 7777 J=1,NINCP1
C
C=====> Clear the arrays for stress, strain, strain rate, and time.
C
```

```
      DO 10 K=1,NSTEPS
        TAU(K)=0.0
   10   GAMMA(K)=0.0
C
C=====> Update the peak strain and reinitialize time and Z(1).
C
      T = 0.0
      Z(1) = 0.0
      AMP = GAMPK1 + DELGAM*(J-1)
      PEAKST(J) = AMP
      PKEQST(J) = RAD3*AMP/2.
C
C=====> Carry out the integration for the current peak strain
C
      DO 30 K = 1,NSTEPS
          CALL INTFUN(Z,T,DELT,NEQ)
        TAU(K+1) = Z(1)
        GAMMA(K+1) = AMP * DSIN(OMEGA*T)
   30 CONTINUE
C
C=====> Compute damping for the current peak strain.
C
      TAUMIN = 0.0
      TAUMAX = 0.0
     CALL MAXIM(TAU,NSTEPS,TAUMIN,TAUMAX)
C
     ENGABS = 0.
C
     DO 50 I=ISTART,IEND
       ENGABS = ENGABS + .5 * (TAU(I+1) + TAU(I)) *
    >                                   (GAMMA(I+1) - GAMMA(I))
  50 CONTINUE
C
     ENGSTO = .5*TAUMAX*AMP
     ENRGAB(J) = ENGABS
     ENRGST(J) = ENGSTO
     ETA(J) = ENGABS/(2.*PI*ENGSTO)
     WRITE(*,'(3X,I3,3(3X,E10.4))')J,PEAKST(J),ENRGAB(J),ETA(J)
 777 CONTINUE
C
     CALL MAXIM(ETA,NINCP1,ETAMIN,ETAMAX)
C
C=====> Write results to output data file (unit 10, 12, 13, and 14) and to
C       output text file (unit 11).
C
     WRITE(10,'(1X,2(E16.8,'','')))')(PEAKST(J),ETA(J),J=1,NINCP1)
     WRITE(11,'(5X,''0 '',I2,'' "'',A,A,''"'')')
    >        ISUM1,SAR1,SG0(IBSG0:IESG0)
     WRITE(11,'(5X,''1 '',I2,'' "'',A,F5.1,2X,A,E6.1,''"'')')
    >        ISUM2,SAR2,YS,SAR3,G
     WRITE(11,'(5X,''2 '',I2,'' "'',A,A,2X,A,A,''"'')')
    >        ISUM3,SAR4,SN(IBSN:IESN),SAR5,SALF(IBSALF:IESALF)
     WRITE(11,'(5X,''3 '',I2,'' "'',A,A,2X,A,A,''"'')')
    >        ISUM4,SAR6,SA(IBSA:IESA),SAR7,SFT(IBSFT:IESFT)
     WRITE(11,'(5X,''4 38 "gMin='',E9.4,2X,''gMax='',E9.4,''"'')')
    >      ETAMIN,ETAMAX
     WRITE(12,'(1X,2(E16.8,'','')))')(PEAKST(J),ENRGAB(J),J=1,NINCP1)
     WRITE(13,'(1X,2(E16.8,'','')))')(PKEQST(J),ETA(J),J=1,NINCP1)
     WRITE(14,'(1X,2(E16.8,'','')))')(PKEQST(J),ENRGAB(J),J=1,NINCP1)
     PRINT *,'  MAX. LOSS FACTOR=',ETAMAX
C
1000 FORMAT(A79)
1001 FORMAT(2A60)
1200 FORMAT(' ',3X,'1',5X,I7)
1201 FORMAT(' ',3X,'2',5X,I7)
1500 FORMAT(' ',5X,E10.4,5X,E10.4)
1501 FORMAT(' ',5X,E10.4,5X,E10.4,5X,E10.4)
     STOP
     END


     SUBROUTINE DERIV(Z,T,ZDOT)
     IMPLICIT REAL*8 (A-H,O-Z)
     REAL*8 OMEGA,T,Z(1),ZDOT(1),NN
     COMMON/BLOK1/AMP,OMEGA,YS,E,G,ALFA,NN,GAMMA0,A,PT,RAD3
C
     GAMMA = AMP*DSIN(OMEGA*T)
     GAMDOT = AMP*OMEGA*DCOS(OMEGA*T)
     DLTAGA = DABS(GAMMA) - GAMMA0
     IF(DLTAGA.LT.0.0)THEN
```

```
      BETA=(1./3.)*(E*ALFA)*( GAMMA - Z(1)/G +
     >     RAD3*FT*ERF(A*GAMMA/RAD3)*UNIT(-GAMMA*GAMDOT) )
         IF(NN.EQ.1.0)THEN
            ZDOT(1) = G*( GAMDOT - DABS(GAMDOT)*(Z(1)-BETA)/YS )
         ELSE
            ZDOT(1) = G*( GAMDOT - DABS(GAMDOT)*
     >         (DABS(Z(1)-BETA)/YS)**(NN-1)*
     >         (Z(1)-BETA)/YS )
         ENDIF
      ELSE
         ZDOT(1) = G*GAMDOT
      ENDIF
      RETURN
      END
C
C==========>SDSE.FOR
```

EXAMPLE OF AN INPUT FILE:

```
SHA Shear Hysteretic Model with Elasticity Outside g0
494 psi
10.56x106 psi
3
.818
 28.5E06  0.35
 855.
 0.818
 3
```

APPENDIX F

FORTRAN MAIN PROGRAM (all subroutines and function subprograms that are not specifically related to the calculations being made here are given in Appendix H):

```
C========> Program Name: BENDR.FOR
C
C======> This program will carry out a Runge-Kutta integration on the
C       stress-strain equations of the SMA hysteretic model.
C       Note that inelastic behavior is supressed beyond EPSLNO.
C       The physical geometry is that of a rectangular beam under bending.
C       Plane sections are assumed assumed to remain plane.
C       We begin by specifying a maximum peak strain at the surface.
C       A linear strain distribution is assumed to exist about the
C       neutral axis. The height coordinate (i.e. y) is divided into fine
C       increments and the stress is computed for each of these increments.
C       Based on the nonlinear uniaxial stress distribution, the torque
C       is computed for each time instant. The resulting curves for torque and
C       bending angle will be stored in data files and then will be
C       used to calculate the loss factor (damping) for that particular
C       peak strain. The peak strain is then incremented and the process
C       is repeated until the final peak strain is used. The results for
C       loss factor vs. strain are then stored in an ASCII data file to be
C       used in plotting with GRAPHER; ASCII data is arranged in column pairs.
C
C===========================================================================
C
C   NOMENCLATURE:
C
C     REAL CONSTANTS:
C       A       Material constant controlling shape of hystoresis
C       ALFA    =Ey/(E-Ey) where Ey is the inelastic modulus
C       AMP     Amplitude of strain input
C       E       Young's modulus
C       EPSLNO  Strain beyond which inelastic growth is supressed
C       EPSPK1  Minimum peak strain
C       EPSPK2  Maximum peak strain
C       FREQ    Frequency of cyclic strain input
C       PT      Material constant controlling size of hysteresis
C       HEIGHT  Height (or thickness) of beam
C       LEN     Length of beam
C       N       Overstress power (controls sharpness of transition to inel.)
C       VNEW    Poisson's ratio
C       WIDTH   Width (or depth) of beam
C       Y       Stress where damping mechanisms are activated
C       YY      Height coordinate of beam cross-section.
C
C     INTEGER CONSTANTS:
C       NCYCLE  No. of cycles of oscillation
C       NPPCYC  No. of points per cycle to be used in integration
C       NINC    No. of increments for the range of peak strain
C
C     CHARACTER STRINGS:
C       FILENAME string for filename assignment to a FORTRAN unit number
C       SALF     string for material constant ALFA
C       SA       string for material constant A
C       SE       string for Young's modulus E
C       SE0      string for limiting strain EPSLNO
C       SEP      string for peak surface strain applied to beam
C       SPT      string for material constant PT
C       SHEI     string for beam height
C       SLEN     string for beam length
C       SN       string for overstress power N
C       SWID     string for beam width
C       SY       string for stress where damping mechanisms activate: Y
C       TITLE    descriptive title for run
C
C     VARIABLE ARRAYS:
C       ANGLE(K)         Angle of bending curvature (index K represents time)
C       ENRGAB(I)        Energy absorbed in one cycle (I represents position)
C       ENRGST(I)        Energy absorbed in one cycle (I represents position)
C       EPSLON(I,K)      Axial strain (index I rep. pos in cross-section,
C                                    index K rep. time)
C       ETA(I)           Loss factor (I represents position)
C       PEAKST(I)        Peak strain (index I represents position)
C       PKEQST(I)        Peak equivalent strain (index I represents position)
C       STRESS(I,K)      Axial stress (index I rep. pos in cross-section,
```

```
C                                        index K rep. time)
C       TORQUE(K)       End torque acting on beam cross-section (K rep. time)
C       Z(K)            Axial stress passed from INTFUN (index K rep. time)
C
C     SUBROUTINES:
C       DERIV           Contains the differential eqs. (invoked by INTFUN)
C       INTFUN          Integration routine (4th order forward Runge-Kutta)
C       MAXIM           Determines the min. & max. values of an array
C       STRLEN          Counts the number of characters in a string
C
C     FUNCTION SUBPROGRAMS:
C       ERF(X)          Error function of X
C       FACT(K)         K factorial
C       SGN(V)          Signum function of V
C       UNIT(X)         Unit step function of X
C
C=======================================================================
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER NEQ
      REAL*8 DERIV,FLOAT,T,TEND,Z(20),N,LEN
      DIMENSION EPSLON(101,1001),
     >          STRESS(101,1001),TORQUE(1001),ANGLE(1001),
     >          YY(101),ETA(200),PEAKST(200),ENGA(200),ENGS(200)
      CHARACTER*20 FILENAME
      CHARACTER*70 TITLE
      CHARACTER*70 SY,SE,SN,SALF,SE0,SA,SFT,SLEN,SHEI,SWID,SEP
      CHARACTER*(*) SAR1,SARP,SAR2,SAR3,SAR4,SAR5,SAR6,SAR7,SAR8,SAR9,
     >              SA10
      PARAMETER (SAR1='e0=')
      PARAMETER (SARP='eP=')
      PARAMETER (SAR2='Y=')
      PARAMETER (SAR3='E=')
      PARAMETER (SAR4='n=')
      PARAMETER (SAR5='alfa=')
      PARAMETER (SAR6='a=')
      PARAMETER (SAR7='fT=')
      PARAMETER (SAR8='Length=')
      PARAMETER (SAR9='Height=')
      PARAMETER (SA10='Width=')
      COMMON/BLOK1/AMP,OMEGA,Y,E,ALFA,N,EPSLN0,A,FT,RATIO
C
      NEQ = 1
C
C=====> Interactive input of filenames for the material data input file
C         and for output files.
C
      WRITE(*,*)'Enter the name of your input data file.'
      IFL=9
    1 READ(*,2)FILENAME
    2 FORMAT(A)
      WRITE(*,3)FILENAME
    3 FORMAT(' ',3X,A20)
      IF(IFL.EQ.9)THEN
        OPEN(IFL,FILE=FILENAME,STATUS='UNKNOWN')
      ELSE
        OPEN(IFL,FILE=FILENAME,STATUS='UNKNOWN')
      ENDIF
      IF (IFL.EQ.9) THEN
        WRITE(*,*)'Enter name of file for Loss Fact. vs. Peak Strain'
        IFL=10
        GO TO 1
      ENDIF
      IF (IFL.EQ.10) THEN
        WRITE(*,*)'Enter name for the summary file'
        IFL=11
        GO TO 1
      ENDIF
      IF (IFL.EQ.11) THEN
        WRITE(*,*)'Filename for ENGABS vs. ENGSTO at each peak strain'
        IFL=12
        GO TO 1
      ENDIF
      REWIND 09
      REWIND 10
      REWIND 11
      REWIND 12
C
C=====> Read input quantities from input file.
C
      READ(9,5)TITLE
```

```
      READ(9,'(A)')SY
      READ(9,'(A)')SE
      READ(9,'(A)')SN
      READ(9,'(A)')SALP
      READ(9,'(A)')SLEN
      READ(9,'(A)')SWID
      READ(9,'(A)')SHEI
    5 FORMAT(A)
      READ(9,*)E,VNEW
      READ(9,*)Y
      READ(9,*)ALFA
      READ(9,*)N
      READ(9,*)LEN
      READ(9,*)WIDTH,HEIGHT
C
C=====> Interactive input of other material parameters, strain amplitude,
C       and number of cycles of loading to be used in calculations.
C
      WRITE(*,*)' Enter EPSLN0'
      READ(5,*)EPSLN0
      WRITE(*,*)' EPSLN0=',EPSLN0
      WRITE(*,*)' Enter character string for EPSLN0'
      READ(5,'(A)')SE0
      WRITE(*,'(1X,''SE0='',A)')SE0
      WRITE(*,*)' Enter A AND FT'
      READ(5,*)A,FT
      WRITE(*,*)' A=',A,' FT=',FT
      WRITE(*,*)' Enter character string for a'
      READ(5,'(A)')SA
      WRITE(*,'(1X,''SA='',A)')SA
      WRITE(*,*)' Enter character string for fT'
      READ(5,'(A)')SFT
      WRITE(*,'(1X,''SFT='',A)')SFT
      WRITE(*,*)' Enter No. of points on linear strain profile'
      READ(5,*)NGRIDP
      WRITE(*,*)' NGRIDP=',NGRIDP
      WRITE(*,*)' Enter FREQ, EPSPK1 and EPSPK2 (Min, Max Peak Strains)'
      READ(5,*)FREQ,EPSPK1,EPSPK2
      WRITE(*,*)' FREQ=',FREQ,' EPSPK1=',EPSPK1,' EPSPK2=',EPSPK2
      WRITE(*,*)' Enter character string for EPSPK2'
      READ(5,'(A)')SEP
      WRITE(*,'(1X,''SEP='',A)')SEP
      WRITE(*,*)' Enter Number of Cycles and Number of Points per Cycle'
      READ(5,*)NCYCLE,NPPCYC
      WRITE(*,*)' NCYCLE=',NCYCLE,' NPPCYC=',NPPCYC
      WRITE(*,*)' Enter NINC (No. of oncrements between peak strains)'
      READ(5,*)NINC
      WRITE(*,*)' NINC=',NINC
C
C=====> Determine length of various character strings for later use.
C
      CALL STRLEN(SE0,IBSE0,IESE0)
      CALL STRLEN(SEP,IBSEP,IESEP)
      CALL STRLEN(SY,IBSY,IESY)
      CALL STRLEN(SE,IBSE,IESE)
      CALL STRLEN(SN,IBSN,IESN)
      CALL STRLEN(SALP,IBSALP,IESALP)
      CALL STRLEN(SA,IBSA,IESA)
      CALL STRLEN(SFT,IBSFT,IESFT)
      CALL STRLEN(SLEN,IBSLEN,IESLEN)
      CALL STRLEN(SHEI,IBSHEI,IESHEI)
      CALL STRLEN(SWID,IBSWID,IESWID)
      CALL STRLEN(SAR1,IBSAR1,IESAR1)
      CALL STRLEN(SARP,IBSARP,IESARP)
      CALL STRLEN(SAR2,IBSAR2,IESAR2)
      CALL STRLEN(SAR3,IBSAR3,IESAR3)
      CALL STRLEN(SAR4,IBSAR4,IESAR4)
      CALL STRLEN(SAR5,IBSAR5,IESAR5)
      CALL STRLEN(SAR6,IBSAR6,IESAR6)
      CALL STRLEN(SAR7,IBSAR7,IESAR7)
      CALL STRLEN(SAR8,IBSAR8,IESAR8)
      CALL STRLEN(SAR9,IBSAR9,IESAR9)
      CALL STRLEN(SA10,IBSA10,IESA10)
C
      ISUM1 = IESAR1 + IESE0 + 2 + IESARP + IESEP
      ISUM2 = IESAR2 + IESY + 2 +IESAR3 + IESE
      ISUM3 = IESAR4 + IESN + 2 + IESAR5 + IESALP
      ISUM4 = IESAR6 + IESA + 2 + IESAR7 + IESFT
      ISUM5 = IESAR8+IESLEN+2+IESAR9+IESHEI+2+IESA10+IESWID
C
C=====> Calculate the quantity Pi=3.14159267..., and other parameters
```

```
C
      PI=DACOS(-1.0D00)
      OMEGA = 2.*PI*FREQ
      PRINT *,' PI=',PI,' OMEGA=',OMEGA
      NSTEPS = NCYCLE*NPPCYC
      NSTPP1 = NSTEPS+1
      PRINT *,' NSTEPS=',NSTEPS
      PERIOD = 1./FREQ
      DELY = HEIGHT/NGRIDP
      DELT = PERIOD/NPPCYC
      DELEPS = (EPSPK2-EPSPK1)/NINC
      ISTART = NPPCYC/4. + 1
      IEND = ISTART + NPPCYC-1
C
C=====> Set up radial grid.
C
      DO 7, I=1,NGRIDP
    7   YY(I) = -(HEIGHT/2.) + DELY*(2.*I - 1.)/2.
C
C=====> Set up the loop for carrying out integration for each step in peak
C       strain and position in the cross-section.
C
      DO 7777 J=1,NINC+1
C
C=====> Reinitialize T and Z(1). Clear the arrays used in the integration loop.
C
      T = 0.
      Z(1) = 0.
      DO 9 K=1,NSTPP1
        DO 8 I=1,NGRIDP
          STRESS(I,K) = 0.
    8     EPSLON(I,K) = 0.
    9 CONTINUE
C
C=====> Update the peak surface strain.
C
      AMP = EPSPK1+DELEPS*(J-1)
      PEAKST(J) = AMP
C
C=====> Carry out the integration at each point in one half of the beam
C       cross-section for the current surface strain (only one-half of the
C       cross-section need be considered because of symmetry about the
C       neutral axis).
C
      DO 11 I=1,NGRIDP/2
      T = 0.
      Z(1) = 0.
      EPSLON(I,1) = 0.
      STRESS(I,1) = 0.
      RATIO = YY(I)/(HEIGHT/2.)
        DO 10 K = 1,NSTEPS
            CALL INTFUN(Z,T,DELT,NEQ)
          EPSLON(I,K+1) = -AMP * DSIN(OMEGA*T) * RATIO
          STRESS(I,K+1) = Z(1)
   10   CONTINUE
   11 CONTINUE
C
C=====> Using symmetry, determine stresses and strains for the other
C       half of the cross-section
C
      DO 13 I=NGRIDP/2+1,NGRIDP
        DO 12 K=1,NSTPP1
          EPSLON(I,K) = -EPSLON(NGRIDP-I+1,K)
   12     STRESS(I,K) = -STRESS(NGRIDP-I+1,K)
   13 CONTINUE
C
C=====> Compute the acting end torque and angle of beam curvature.
C
      DO 15 K=1,NSTPP1
   15   TORQUE(K)=0.0
C
      DO 25 K=1,NSTPP1
        DO 20 I=1,NGRIDP
   20     TORQUE(K)=TORQUE(K) + (-YY(I))*STRESS(I,K)*DELY
        TORQUE(K) = WIDTH*TORQUE(K)
        ANGLE(K) = 2.*LEN*EPSLON(1,K)/HEIGHT
   25 CONTINUE
C
C=====> Compute damping for the current peak surface strain.
C
      CALL MAXIM(TORQUE,NSTPP1,TORMIN,TORMAX)
```

```
      CALL MAXIM(ANGLE,NSTPP1,ANGMIN,ANGMAX)
      ENGABS = 0.
C
      DO 50 I=ISTART,IEND
        ENGABS = ENGABS + .5 * (TORQUE(I+1) + TORQUE(I)) *
     >                                  (ANGLE(I+1) - ANGLE(I))
   50 CONTINUE
C
      ENGSTO = .5*TORMAX*ANGMAX
      ENGA(J) = ENGABS
      ENGS(J) = ENGSTO
      ETA(J) = ENGABS/(2.*PI*ENGSTO)
      WRITE(*,'(3X,I3,2(3X,E10.4))')J,PEAKST(J),ETA(J)
 7777 CONTINUE
      CALL MAXIM(ETA,NINC,ETAMIN,ETAMAX)
C
C=====> Write results to output data file (unit 10) and to
C        output text file (unit 11).
C
      WRITE(10,'(1X,2(E16.8,'','',''))')(PEAKST(J),ETA(J),J=1,NINC)
      WRITE(12,'(1X,2(E16.8,'','',''))')(ENGS(J),ENGA(J),J=1,NINC)
      WRITE(11,'(5X,''1 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM1,SAR1,SE0(IBSE0:IESE0),SARP,SEP(IBSEP:IESEP)
      WRITE(11,'(5X,''1 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM2,SAR2,SY(IBSY:IESY),SAR3,SE(IBSE:IESE)
      WRITE(11,'(5X,''2 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM3,SAR4,SN(IBSN:IESN),SAR5,SALF(IBSALF:IESALF)
      WRITE(11,'(5X,''3 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM4,SAR6,SA(IBSA:IESA),SAR7,SPT(IBSPT:IESPT)
      WRITE(11,'(5X,''4 '',I2,'' "'',A,A,2X,A,A,2X,A,A,''"'')')
     >         ISUM5,SAR8,SLEN(IBSLEN:IESLEN),
     >         SAR9,SHEI(IBSHEI:IESHEI),
     >         SA10,SWID(IBSWID:IESWID)
      WRITE(11,'(5X,''5 26 "Peak Loss Factor='',E9.3,''"'')')ETAMAX
      PRINT *,'  MAX LOSS FACTOR=',ETAMAX
C
 1000 FORMAT(A70)
 1001 FORMAT(2A60)
 1200 FORMAT(' ',3X,'1',5X,I7)
 1201 FORMAT(' ',3X,'2',5X,I7)
 1500 FORMAT(' ',5X,E10.4,5X,E10.4)
 1501 FORMAT(' ',5X,E10.4,5X,E10.4,5X,E10.4)
      STOP
      END


      SUBROUTINE DERIV(Z,T,ZDOT)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER NEQ
      REAL*8 OMEGA,T,Z(20),ZDOT(20),NN
      COMMON/BLOK1/AMP,OMEGA,Y,E,ALFA,NN,EPSLN0,A,PT,RATIO
C
          EPSLON = -AMP * DSIN(OMEGA*T) * RATIO
          EPDOT = -AMP * OMEGA * DCOS(OMEGA*T) * RATIO
          DLTAEP = DABS(EPSLON) - EPSLN0
      IF(DLTAEP.LT.0.0)THEN
        BETA=(E*ALFA)*( EPSLON - Z(1)/E +
     >     PT*ERF(A*EPSLON)*UNIT(-EPSLON*EPDOT) )
        IF(NN.EQ.1.0)THEN
           ZDOT(1) = E*( EPDOT -
     >        DABS(EPDOT)*(Z(1)-BETA)/Y )
        ELSE
           ZDOT(1) = E*( EPDOT - DABS(EPDOT)*
     >        (DABS(Z(1)-BETA)/Y)**(NN-1)*
     >        (Z(1)-BETA)/Y )
        ENDIF
      ELSE
        ZDOT(1) = E*EPDOT
      ENDIF
      RETURN
      END
C
C=========>BENDR.FOR
```

EXAMPLE OF AN INPUT FILE:

```
SMA Hysteretic Model with Elasticity Outside e0
855 psi
```

```
28.5x106 psi
3
.818
20 in
1 in
.5 in
 28.5E06  0.35
 855.
 0.818
 3
 20.
 1. .5
```

EXAMPLE OF A BATCH FILE TO RUN BENDR ON THE CODE 281 MICROVAX:

```
set def [.hysteresis]
r bendr
bendr.inp
bendr.dat
bendr.txt
benrg.dat
1.e-4
1.0e-4
80000.  4.e-5
80000
4.0e-5
20
1.0  .25e-4  1.75e-4
1.75e-4
2 100
20
```

APPENDIX G

<u>FORTRAN MAIN PROGRAM</u> (all subroutines and function subprograms that are not specifically related to the calculations being made here are given in Appendix H):

```
C=========> Program Name: TORRANGE.FOR
C
C=====> This program will carry out a Runge-Kutta integration on the
C       stress-strain equations of the SMA hysteretic model.
C       Note that inelastic behavior is supressed beyond GAMMA0.
C       The physical geometry is that of a solid cylinder under torsion.
C       Plane stress conditions are assumed (pure shear to be specific).
C       We begin by specifying a minimum peak strain at the surface.
C       A linear strain distribution is assumed to exist in the circular
C       cross section. The radial coordinate is divided into fine
C       increments and the stress is computed for each of these increments.
C       Based on the shear stress distribution, the resulting torque is
C       computed for each time instant. The resulting curves for torque and
C       overall sample twist will be stored in data files and then will be
C       used to calculate the loss factor (damping) for that particular
C       peak strain. The peak strain is then incremented and the process
C       is repeated until the final peak strain is used. The results for
C       loss factor vs. strain are then stored in an output file to be used
C       in plotting with GRAPHER; ASCII data is arranged in column pairs.
C
C==============================================================================
C
C   NOMENCLATURE:
C
C     REAL CONSTANTS:
C       A       Material constant controlling shape of hysteresis
C       ALFA    =Ey/(E-Ey) where Ey is the axial inelastic modulus
C       AMP     Amplitude of strain input
C       DIA     Diameter of shaft
C       E       Young's modulus
C       G       Shear modulus {=E/[2(1+VNEW)]}
C       GAMMA0  Strain beyond which inelastic growth is supressed
C       GAMPK1  Minimum peak strain
C       GAMPK2  Maximum peak strain
C       FREQ    Frequency of cyclic strain input
C       PT      Material constant controlling size of hysteresis
C       LEN     Length of Shaft
C       N       Overstress power (controls sharpness of transition to inel.)
C       RADIUS  Radius of shaft
C       VNEW    Poisson's ratio
C       Y       Axial stress where damping mechanisms are activated
C       YS      Shear stress where damping mechanisms are activated
C
C     INTEGER CONSTANTS:
C       NCYCLE  No. of cycles of oscillation
C       NPPCYC  No. of points per cycle to be used in integration
C       NINC    No. of increments for the range of peak strain
C
C     CHARACTER STRINGS:
C       FILENAME string for filename assignment to a FORTRAN unit number
C       SALF     string for material constant ALFA
C       SA       string for material constant A
C       SDIA     string for geometric constant DIA
C       SG       string for Young's modulus G
C       SG0      string for limiting strain GAMMA0
C       SPT      string for material constant PT
C       SLEN     string for geometric constant LEN
C       SN       string for overstress power N
C       SYS      string for shear stress where damping mechanisms activate
C       TITLE    descriptive title for run
C
C     VARIABLE ARRAYS:
C       ANGLE(K)        Angle of twist of shaft (index K represents time)
C       ENRGAB(I)       Energy absorbed in one cycle (I represents position)
C       ENRGST(I)       Energy absorbed in one cycle (I represents position)
C       GAMMA(I,K)      Shear strain (index I rep. position, K rep. time)
C       ETA             Loss factor (I represents position)
C       PEAKST(I)       Peak strain (index I represents position)
C       PKEQST(I)       Peak equivalent strain (index I represents position)
C       TAU(I,K)        Shear stress (index I rep. position, K rep. time)
C       TORQUE(K)       Resultant torque acting on shaft
C       Z(K)            Shear stress passed from INTFUN (index K rep. time)
```

```
C
C      SUBROUTINES:
C        DERIV          Contains the differential eqs. (invoked by INTFUN)
C        INTFUN         Integration routine (4th order forward Runge-Kutta)
C        MAXIM          Determines the min. & max. values of an array
C        STRLEN         Counts the number of characters in a string
C
C      FUNCTION SUBPROGRAMS:
C        ERF(X)         Error function of X
C        FACT(K)        K factorial
C        SGN(V)         Signum function of V
C        UNIT(X)        Unit step function of X
C
C==========================================================================
C
       IMPLICIT REAL*8 (A-H,O-Z)
       INTEGER NEQ
       REAL*8 DERIV,FLOAT,T,TEND,TOL,Z(20),N,LEN
       DIMENSION GAMMA(101,1001),TAU(101,1001),
     >           TORQUE(1001),TWIST(1001),
     >           R(101),ETA(200),PEAKST(200),ENGA(200),ENGS(200)
       CHARACTER*20 FILENAME
       CHARACTER*70 TITLE,SYS,SG,SN,SALF,SG0,SA,SFT,SLEN,SDIA,SGP
       CHARACTER*(*) SAR1,SAR2,SAR3,SAR4,SAR5,SAR6,SAR7,SAR8,SAR9,SARP
       PARAMETER (SAR1='g0=')
       PARAMETER (SARP='gP=')
       PARAMETER (SAR2='Ys=')
       PARAMETER (SAR3='G=')
       PARAMETER (SAR4='n=')
       PARAMETER (SAR5='a=')
       PARAMETER (SAR6='a=')
       PARAMETER (SAR7='fT=')
       PARAMETER (SAR8='Length=')
       PARAMETER (SAR9='Diameter=')
       COMMON/BLOK1/AMP,OMEGA,YS,E,G,ALFA,N,GAMMA0,A,PT,RAD3,RATIO
C
       NEQ = 1
C
C=====> Interactve input of filenames for the material data input file
C        and for output files.
C
       WRITE(*,*)'Enter the name of your input data file.'
       IFL=9
     1 READ(*,2)FILENAME
     2 FORMAT(A)
       WRITE(*,3)FILENAME
     3 FORMAT(' ',3X,A20)
       IF(IFL.EQ.9)THEN
         OPEN(IFL,FILE=FILENAME,STATUS='UNKNOWN')
       ELSE
         OPEN(IFL,FILE=FILENAME,STATUS='UNKNOWN')
       ENDIF
       IF (IFL.EQ.9) THEN
         WRITE(*,*)'Enter name of file for Loss Fact. vs. Peak Strain'
         IFL=10
         GO TO 1
       ENDIF
       IF (IFL.EQ.10) THEN
         WRITE(*,*)'Enter name for the summary file'
         IFL=11
         GO TO 1
       ENDIF
       IF (IFL.EQ.11) THEN
         WRITE(*,*)'Filename for ENGABS and ENGSTO at each peak strain.'
         IFL=12
         GO TO 1
       ENDIF
       REWIND 09
       REWIND 10
       REWIND 11
       REWIND 12
C
C=====> Read input quantities from input file.
C
       READ(9,5)TITLE
       READ(9,'(A)')SYS
       READ(9,'(A)')SG
       READ(9,'(A)')SN
       READ(9,'(A)')SALF
       READ(9,'(A)')SLEN
       READ(9,'(A)')SDIA
```

```
    5 FORMAT(A)
      READ(9,*)E,VNEW
      READ(9,*)Y
      READ(9,*)ALFA
      READ(9,*)N
      READ(9,*)LEN
      READ(9,*)DIA
C
C=====> Interactive input of other material parameters, strain amplitude,
C        and number of cycles of loading to be used in calculations.
C
      WRITE(*,*)' Enter GAMMA0'
      READ(5,*)GAMMA0
      WRITE(*,*)' GAMMA0=',GAMMA0
      WRITE(*,*)' Enter character string for GAMMA0'
      READ(5,'(A)')SG0
      WRITE(*,'(1X,''SG0='',A)')SG0
      WRITE(*,*)' Enter A AND FT'
      READ(5,*)A,FT
      WRITE(*,*)' A=',A,' FT=',FT
      WRITE(*,*)' Enter character string for a'
      READ(5,'(A)')SA
      WRITE(*,'(1X,''SA='',A)')SA
      WRITE(*,*)' Enter character string for fT'
      READ(5,'(A)')SFT
      WRITE(*,'(1X,''SFT='',A)')SFT
      WRITE(*,*)' Enter No. of points on linear strain profile'
      READ(5,*)NGRIDP
      WRITE(*,*)' NGRIDP=',NGRIDP
      WRITE(*,*)' Enter FREQ, GAMPK1 and GAMPK2 (Min, Max Peak Strains)'
      READ(5,*)FREQ,GAMPK1,GAMPK2
      WRITE(*,*)' FREQ=',FREQ,' GAMPK1=',GAMPK1,' GAMPK2=',GAMPK2
      WRITE(*,*)' Enter character string for GAMMAP'
      READ(5,'(A)')SGP
      WRITE(*,'(1X,''SGP='',A)')SGP
      WRITE(*,*)' Enter Number of Cycles and Number of Points per Cycle'
      READ(5,*)NCYCLE,NPPCYC
      WRITE(*,*)' NCYCLE=',NCYCLE,' NPPCYC=',NPPCYC
      WRITE(*,*)' Enter NINC (No. of increments between peak strains)'
      READ(5,*)NINC
      WRITE(*,*)' NINC=',NINC
      GAMMAP = GAMPK2
C
C=====> Determine length of various character strings for later use.
C
      CALL STRLEN(SG0,IBSG0,IESG0)
      CALL STRLEN(SGP,IBSGP,IESGP)
      CALL STRLEN(SYS,IBSYS,IESYS)
      CALL STRLEN(SG,IBSG,IESG)
      CALL STRLEN(SN,IBSN,IESN)
      CALL STRLEN(SALF,IBSALF,IESALF)
      CALL STRLEN(SA,IBSA,IESA)
      CALL STRLEN(SFT,IBSFT,IESFT)
      CALL STRLEN(SLEN,IBSLEN,IESLEN)
      CALL STRLEN(SDIA,IBSDIA,IESDIA)
      CALL STRLEN(SAR1,IBSAR1,IESAR1)
      CALL STRLEN(SARP,IBSARP,IESARP)
      CALL STRLEN(SAR2,IBSAR2,IESAR2)
      CALL STRLEN(SAR3,IBSAR3,IESAR3)
      CALL STRLEN(SAR4,IBSAR4,IESAR4)
      CALL STRLEN(SAR5,IBSAR5,IESAR5)
      CALL STRLEN(SAR6,IBSAR6,IESAR6)
      CALL STRLEN(SAR7,IBSAR7,IESAR7)
      CALL STRLEN(SAR8,IBSAR8,IESAR8)
      CALL STRLEN(SAR9,IBSAR9,IESAR9)
C
      ISUM1 = IESAR1 + IESG0 +2 + IESARP + IESGP
      ISUM2 = IESAR2 + IESYS  2 + IESAR3 + IESG
      ISUM3 = IESAR4 + IESN + 2 + IESAR5 + IESALF
      ISUM4 = IESAR6 + IESA + 2 + IESAR7 + IESFT
      ISUM5 = IESAR8 + IESLEN + 2 + IESAR9 + IESDIA
C
C=====> Calculate the quantity Pi=3.14159267..., and other parameters
C
      PI = DACOS(-1.0D00)
      RAD3 = 3.**.5
      G = E/(2.*(1+VNEW))
      YS = Y/RAD3
      OMEGA = 2.*PI*FREQ
      PRINT *,' PI=',PI,' OMEGA=',OMEGA
      NSTEPS = NCYCLE*NPPCYC
```

```
      NSTPP1 = NSTEPS+1
      PRINT *,' NSTEPS=',NSTEPS
      PERIOD = 1./FREQ
      RADIUS = DIA/2.
      DELR = (DIA/2.)/NGRIDP
      DELT = PERIOD/NPPCYC
      PRINT *,'DELR=',DELR,' DELT=',DELT
      DELGAM = (GAMPK2-GAMPK1)/NINC
      PRINT *,'DELGAM=',DELGAM
      ISTART = NPPCYC/4. + 1
      IEND = ISTART + NPPCYC-1
C
C=====> Set up radial grid.
C
      DO 7, I=1,NGRIDP-1
    7   R(I)=DELR*I
      R(NGRIDP)=RADIUS
C
C=====> Set up the loop for carrying out integration for each step in
C        peak strain.
C
      DO 7777 J=1,NINC+1
C
C=====> Clear the arrays for shear stress and strain.
C
      DO 9 K=1,NSTPP1
        DO 8 I=1,NGRIDP
          TAU(I,K) = 0.
    8     GAMMA(I,K) = 0.
    9 CONTINUE
C
C=====> Update the peak strain and reinitialize time and Z(1).
C
      T = 0.
      Z(1) = 0.
      AMP = GAMPK1+DELGAM*(J-1)
      PEAKST(J) = AMP
C
C=====> Carry out the integration at each point along the radial
C        coordinate of the circular cross-section, for the current
C        surface strain.
C
      DO 11 I=1,NGRIDP
      T = 0.
      Z(1) = 0.
      GAMMA(I,1) = 0.
      TAU(I,1) = 0.
      RATIO = R(I)/RADIUS
        DO 10 K = 1,NSTEPS
            CALL INTFUN(Z,T,DELT,NEQ)
          TAU(I,K+1) = Z(1)
          GAMMA(I,K+1) = AMP * DSIN(OMEGA*T) * RATIO
   10   CONTINUE
   11 CONTINUE
C
C=====> Compute the acting torque and angle of twist of the shaft.
C
      DO 15 K=1,NSTPP1
   15   TORQUE(K)=0.0
C
      DO 25 K=1,NSTPP1
        DO 20 I=1, NGRIDP-1
   20       TORQUE(K)=TORQUE(K) + R(I)*TAU(I,K)*2.*PI*R(I)*DELR
        TORQUE(K)=TORQUE(K) + (R(NGRIDP)-DELR/4.)*TAU(NGRIDP,K)*
     >                         2.*PI*(R(NGRIDP)-DELR/4.)*DELR/2.
        TWIST(K) = GAMMA(NGRIDP,K)*LEN/RADIUS
   25 CONTINUE
C
C=====> Compute damping for the current peak strain.
C
C
      CALL MAXIM(TORQUE,NSTPP1,TORMIN,TORMAX)
      CALL MAXIM(TWIST,NSTPP1,TWIMIN,TWIMAX)
      ENGABS = 0.
C
      DO 50 I=ISTART,IEND
        ENGABS = ENGABS + .5 * (TORQUE(I+1) + TORQUE(I)) *
     >                                  (TWIST(I+1) - TWIST(I))
   50 CONTINUE
C
      ENGSTO = .5*TORMAX*TWIMAX
```

```
      ENGA(J) = ENGABS
      ENGS(J) = ENGSTO
      ETA(J) = ENGABS/(2.*PI*ENGSTO)
      WRITE(*,'(3X,I3,2(3X,E10.4))')J,PEAKST(J),ETA(J)
 7777 CONTINUE
      CALL MAXIM(ETA,NINC,ETAMIN,ETAMAX)
C
C=====> Write results to output data file (unit 10, 12, 13, and 14) and to
C       output text file (unit 11).
C
      WRITE(10,'(1X,2(E16.8,'','',))')(PEAKST(J),ETA(J),J=1,NINC)
      WRITE(12,'(1X,2(E16.8,'','',))')(ENGS(J),ENGA(J),J=1,NINC)
      WRITE(11,'(5X,''0 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM1,SAR1,SG0(IBSG0:IESG0),SARP,SGP(IBSGP:IESGP)
      WRITE(11,'(5X,''1 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM2,SAR2,SYS(IBSYS:IESYS),SAR3,SG(IBSG:IESG)
      WRITE(11,'(5X,''2 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM3,SAR4,SN(IBSN:IESN),SAR5,SALF(IBSALF:IESALF)
      WRITE(11,'(5X,''3 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM4,SAR6,SA(IBSA:IESA),SAR7,SPT(IBSPT:IESPT)
      WRITE(11,'(5X,''4 '',I2,'' "'',A,A,2X,A,A,''"'')')
     >         ISUM5,SAR8,SLEN(IBSLEN:IESLEN),SAR9,SDIA(IBSDIA:IESDIA)
      WRITE(11,'(5X,''5 26 "g='',E9.4,2X,''SDC='',E9.4,''"'')')ETA,SDC
      WRITE(11,'(5X,''6 32 "Peak Loss Factor='',E9.3,''"'')')ETAMAX
      PRINT *,'  MAX LOSS FACTOR=',ETAMAX
C
 1000 FORMAT(A70)
 1001 FORMAT(2A60)
 1200 FORMAT(' ',3X,'1',5X,I7)
 1201 FORMAT(' ',3X,'2',5X,I7)
 1500 FORMAT(' ',5X,E10.4,5X,E10.4)
 1501 FORMAT(' ',5X,E10.4,5X,E10.4,5X,E10.4)
      STOP
      END


      SUBROUTINE DERIV(Z,T,ZDOT)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER NEQ
      REAL*8 OMEGA,T,Z(20),ZDOT(20),NN
      COMMON/BLOK1/AMP,OMEGA,YS,E,G,ALFA,NN,GAMMA0,A,PT,RAD3,RATIO
C
          GAMMA = AMP * DSIN(OMEGA*T) * RATIO
          GAMDOT = AMP * OMEGA * DCOS(OMEGA*T) * RATIO
          DLTAGA = DABS(GAMMA) - GAMMA0
      IF(DLTAGA.LT.0.0)THEN
        BETA=(1./3.)*(E*ALFA)*( GAMMA - Z(1)/G +
     >      RAD3*PT*ERF(A*GAMMA/RAD3)*UNIT(-GAMMA*GAMDOT) )
        IF(NN.EQ.1.0)THEN
          ZDOT(1) = G*( GAMDOT -  DABS(GAMDOT)*(Z(1)-BETA)/YS )
        ELSE
          ZDOT(1) = G*( GAMDOT - DABS(GAMDOT)*
     >         (DABS(Z(1)-BETA)/YS)**(NN-1)*
     >         (Z(1)-BETA)/YS )
        ENDIF
      ELSE
        ZDOT(1) = G*GAMDOT
      ENDIF
      RETURN
      END
C
C=========>TORRANGE.FOR
```

EXAMPLE OF AN INPUT FILE:

```
SMA Hysteretic Shear Model with Elasticity Outside g0
494 psi
10.56x106 psi
3
.818
2 in
.5 in
 28.5E06  0.35
 855.
 0.818
 3
 2.
 .5
```

EXAMPLE OF A BATCH FILE TO RUN TORRANGE ON THE CODE 281 MICROVAX:

```
set def [.hysteresis]
r torrange
torrange.inp
torrange.dat
torrange.txt
torrenrg.dat
1.56e-4
1.56e-4
80000.  4.e-5
80000
4.0e-5
40
1.0  .05e-4  2.57e-4
2.57e-4
2 100
20
```

APPENDIX H

GENERAL SUBROUTINES AND FUNCTION SUBPROGRAMS USED IN THE FORTRAN PROGRAMS IN APPENDICIES B-G:

```
c=============================================================================
c title: strlen.for
c
c author: shw
c       Turbulence Research Laboratory
c       SUNY @ Buffalo
c
c       trlscott@ubvms
c
c function:
c       find the first and last character in a string
c
c inputs:
c       string - character string
c
c outputs:
c       ib - first non-blank character
c       ie - last non-blank character
c
c subroutines required:
c
c modifications:
c
c
        subroutine strlen(string, ib, ie)
        character string*(*)
        j = len(string)
        do 10 i=j,1,-1
                if(string(i:i) .ne. ' ') goto 20
10      continue
        ie = 1
        ib = 1
        return
20      ie = i
        do 30  i=1,j
                if(string(i:i) .ne. ' ') goto 40
30      continue
40      ib = i
        return
        end


      SUBROUTINE MAXIM(F,NPOINT,FMIN,FMAX)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION F(1001)
      FMAX = 0.
      FMIN = 0.
      DO 10 I=1,NPOINT-1
        IF(F(I+1).GT.FMAX)THEN
          FMAX = F(I+1)
        ENDIF
        IF(F(I+1).LT.FMIN)THEN
          FMIN = F(I+1)
        ENDIF
  10  CONTINUE
      RETURN
      END


      REAL*8 FUNCTION UNIT(X)
      REAL*8 X
      IF(X.LT.0.0)THEN
        UNIT=0.0
      ELSE
        UNIT=1.0
      ENDIF
      RETURN
      END
```

```
      REAL*8 FUNCTION ERF(X)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER K
      PI = DACOS(-1.0D00)
C
      IF(X.GT.3.0D00)THEN
        ERF = 1.0D00
        GO TO 500
      ELSEIF(X.LT.-3.0D00)THEN
        ERF = -1.0D00
        GO TO 500
      ELSE
        TOL = .0001
        K = 0
        KMAX = 35
        SERIES = 1.
        TMULT = 2.*X/(PI**.5)
  100   K = K + 2
        IF(K.GT.KMAX)THEN
          PRINT *,' K = KMAX    ERF(',X,')=',ERF
          GO TO 500
        ENDIF
        PRIOR = SERIES
        ODDTRM = (-1)*( DABS(X)**(2.*(K-1)) )/
     >                   ( FACT(K-1)*(1+2*(K-1)) )
        EVNTRM = ( DABS(X)**(2.*K) )/( FACT(K)*(1+2*K) )
        SERIES = SERIES + ODDTRM + EVNTRM
        ERF1 = TMULT * PRIOR
        ERF2 = TMULT * SERIES
        DELTA = ERF2 - ERF1
        ADELTA = ABS(DELTA)
        IF(ADELTA.LE.TOL)THEN
          ERF = ERF2
          GO TO 500
        ELSE
          GO TO 100
        ENDIF
      ENDIF
  500 RETURN
      END


      REAL*8 FUNCTION FACT(K)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER I,K
      IF(K.EQ.0.OR.K.EQ.1)THEN
        FACT = 1.0
        GO TO 20
      ELSE
        FACT = 1.0
        DO 10,I=2,K
          FACT = FACT*I
   10   CONTINUE
      ENDIF
   20 RETURN
      END


      REAL*8 FUNCTION SGN(V)
      REAL*8 V
      IF(V.LT.0.0)THEN
        SGN=-1.0
      ELSE IF(V.EQ.0.0)THEN
        SGN=0.0
      ELSE
        SGN=1.0
      ENDIF
      RETURN
      END


        SUBROUTINE INTFUN(X,TIME,T,N)
        IMPLICIT REAL*8 (A-H,O-Z)
C       X - STATE VECTOR
C       TIME - RUNNING TIME
C       T - TIME INTERVAL
C       N - DIMENSION OF THE STATE VECTOR
```

```
      DIMENSION X(20),D(20),A(20,5),XB(20)
      DATA KUSE/1/
C     INITIALIZE
      KINC= 0
      HMIN= 0
      EMAX= 1.E-05
      EMIN= 1.E-07
      TIN= TIME
      TOUT= TIN + T
C     TIN - BEGINNING OF INTERVAL
C     TOUT - END OF THE INTERVAL
      IF(KUSE .NE. 0)H=T
      IF(KUSE .EQ. 0)H=HSAVE
      KH= 0
      KUSE= 0
C     INTEGRATION ALGORITHM BETWEEN 22 AND 19
      DO 101 I=1,N
101         XB(I)= X(I)
22    K=1
15    CALL DERIV(X,TIME,D)
      GO TO (100,200,300,400,500),K
100   DO 11 I=1,N
            A(I,K)= D(I)*H/3.
11          X(I)= XB(I)+A(I,1)
      TIME= TIN + H/3.00
      K= K+1
      GO TO 15
200   DO 12 I=1,N
            A(I,K)= D(I)*H/3.00
12          X(I)=XB(I)+0.50*( A(I,1)+A(I,2) )
      K=K+1
      GO TO 15
300   DO 17 I=1,N
            A(I,K)= D(I)*H/3.00
17          X(I)= XB(I)+(3.0 * A(I,1)+9.0*A(I,3))/8.00
      TIME= TIN + 0.50*H
      K= K+1
      GO TO 15
400   DO 18 I=1,N
            A(I,K)= D(I)*H/3.00
18          X(I)=XB(I)+(3.*A(I,1)-9.*A(I,3)+12.*A(I,4))/2.0
      TIME= TIN + H
      K=K+1
      GO TO 15
500   DO 19 I=1,N
            A(I,K)= D(I)*H/3.00
19          X(I)=XB(I)+.5*(A(I,1)+4.*A(I,4)+A(I,5))
C     COMPUTE THE TRUNCATION ERROR
C     INTEGRATION ALGORITHM BETWEEN 22 AND 19
      ERROR= 0.00
      DO 21 I=1,N
            TE=A(I,1)-(9.*A(I,3)-8.*A(I,4)+A(I,5))/2.00
21          ERROR= DMAX1(ERROR,DABS(TE))
      IF( ERROR .GE. EMAX )GO TO 33
      DO 32 I=1,N
32          XB(I)= X(I)
      TIN= TIME
      IF(TIME .EQ. TOUT)GO TO 39
```

```
        TREM= TOUT - TIME
        IF(TREM .GT. H)GO TO 31
        HSAVE= H
        KH= 1
        H= TREM
        GO TO 22

31      IF(TREM .LT. (2.00*H))GO TO 22
        IF(ERROR .GT. EMIN)GO TO 22

        KINC= KINC+1
        IF(KINC .LT. 3)GO TO 22
        H= 2.00*H
        KINC= 0
        GO TO 22

33      H= H/2.00
        IF(H .LT. HMIN)GO TO 35
        TIME= TIN

        DO 34 I=1,N
34             X(I)= XB(I)

        GO TO 22
35      WRITE(*,*)'H IS LT HMIN.  TERMINATED.'
        GO TO 40

39      IF(KH .EQ. 1)RETURN
        HSAVE= H
        RETURN
40      STOP
        END
```

INITIAL DISTRIBUTION

Copies	
2	NRL Code 6323 Code 6372
3	NAVPGSCOL Code 69Ps
7	NAVSEA SEA 514 SEA 5142 SEA 5143 SEA 5112 SEA 55N SEA 55N2 SEA 09B31
2	DTIC

CENTER DISTRIBUTION

Copies	Code	Name
1	0112	
1	0115	
1	1940	
1	1941	
1	1944	
1	1945	(RJD)
1	2742	
1	2749	
1	28	(Wacker)
1	2801	(Crisci)
1	2801	(Ventriglio)
1	2802	(Morton)
1	2803	(Cavalaro)
1	2809	(Malec)
5	281	(Holsberg)
5	2812	
10	2812	(CRW)
25	2812	(Graesser)
1	2813	
1	2814	
1	2815	
1	283	(Singerman)
1	284	(Fischer)
1	2844	
1	3421	
1	3422	

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1991	RDT&E 1/90 - 5/91

4. TITLE AND SUBTITLE
Analysis of Strain Dependent Damping in Materials via Modeling of Material Point Hysteresis

5. FUNDING NUMBERS
Program Element No. 62254N
Task No. RS3454
Work Unit Accession No. DN507603

6. AUTHOR(S)
E.J. Graesser
C.R. Wong

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
David Taylor Research Center
Code 2812
Bethesda, MD 20084-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
DTRC-SME-91/34

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Technology
Arlington, VA 22217-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION /AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13 ABSTRACT *(Maximum 200 words)*

A constitutive relationship was used to model the cyclic material response of damping test samples in separate bending and torsion configurations. This was done in order to better understand variations in reported values of damping for materials possessing strain dependent characteristics. The constitutive equations are based on a model of shape memory alloy stress-strain behavior and have been adapted especially for the study of nonlinear hysteresis and the problem of strain dependent damping. Experimental measurements and analytical material response analyses of separate bending and torsion test samples indicated that when the damping of a single nonlinear material is plotted against the one-dimensional local strain of the sample, results are produced which are difficult to compare. However, when the same results are plotted against an invariant measure of three-dimensional distortion the means by which one may compare the data is more straightforward. Also, the approach allows for a quantitative comparison of the damping at a material point to the overall damping. The method can be applied to any homogeneous isotropic nonlinear damping material.

14 SUBJECT TERMS
Damping, Strain Dependence, Nonlinear, Hysteresis, Modeling

15. NUMBER OF PAGES
67

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)