# Analysis of Adiabatic Shear Banding in a Thick-Walled Steel Tube by the Finite Element Method

by

Dean J. Rattazzi

Thesis submitted to the faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

**Master of Science**

in

Engineering Mechanics

APPROVED:

Prof. Romesh C. Batra, Chairman

Prof. Rakesh K. Kapania | Prof. Ronald D. Kriz

September, 1996

Blacksburg, Virginia

**keywords:** adiabatic shear band, high strain rate, thick-walled cylinder, DYNA3D.

# Analysis of Adiabatic Shear Banding in a Thick-Walled Steel Tube by the Finite Element Method

by

Dean J. Rattazzi

Committee Chairman: Prof. Romesh C. Batra

Engineering Mechanics

## (ABSTRACT)

The initiation and propagation of adiabatic shear bands is analyzed numerically for an impulsively loaded thick-walled steel tube. A circumferential V-notch located at the outer surface of the center of the tube provides a stress concentration. The material is modeled as strain hardening, strain-rate hardening and thermal softening. The dynamic loading conditions considered are pure torsion, axial pressure combined with torsion, and internal pressure combined with torsion. Because of the stress concentration, a shear band will first initiate in an element adjoining the notch tip and propagate radially inwards through the thickness of the tube. The speed of propagation and the amount of energy required to drive a shear band through the material are calculated. The effects of the pressure preload and the depth of the notch are studied. Also, the influence of thermal softening is investigated by modeling it after a relation proposed by Zhou et al.

# Acknowledgments

This work was supported by the Army Research Office grant DAAH04-95-1-0043, the National Science Foundation grant CMS9411383, and the Office of Naval Research grant N00014-94-1-1211 to Virginia Polytechnic Institute and State University.

# Contents


**1 Introduction** **1**

**2 Formulation of the Problem** **3**

**3 Computation and Discussion of Results** **6**

3.1 Torsion of the Tube . . . . . . . . . . . . . . . . . . . . . . . . . . 9

3.2 Effect of Notch Depth . . . . . . . . . . . . . . . . . . . . . . . . . 16

3.3 Effect of Prior Axial Loading . . . . . . . . . . . . . . . . . . . . . 19

3.4 Effect of Internal Pressure . . . . . . . . . . . . . . . . . . . . . . . 23

3.5 Torsion of a CR-300 Steel Tube . . . . . . . . . . . . . . . . . . . . 25

**4 Conclusions** **29**

**A FORTRAN Converter Programs** **60**

A.1 Program to Convert a PATRAN Neutral File to a DYNA3D Input File . . 60

A.2 Program to Convert Modified DYNA3D Output Files to PATRAN Results Files . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82

**B FORTRAN Utility Programs** **87**

B.1 Program to Calculate Torque on the Tube . . . . . . . . . . . . . . . . 87

B.2 Program to Calculate External Work Done by Torque . . . . . . . . . . 92

B.3 Program to Calculate External Work Done by Axial Pressure . . . . . . 94

B.4 Program to Calculate Shear Band Speed . . . . . . . . . . . . . . . . . 98

B.5 Program to Extract Data from the Modified DYNA3D Output Files . . . 105

B.6 Program to Convert Modified DYNA3D Binary Output Files to Ascii . . . 109

B.7 Program to Convert Stress and Displacement to Cylindrical Coordinates . 111

# List of Figures


1 (a) Finite element discretization of the tube, and (b) Details of the notch. . . . . . 34

2 Distribution of the effective plastic strain on a radial line for notch 4 with different values of the maximum applied angular speed; (a) $t = 20$ $\mu$s, (b) $t = 70$ $\mu$s. . . . . . 35

3 Time history of (a) the effective stress, (b) the effective plastic strain, and (c) the temperature at the centroids of eleven elements on a radial line through the notch tip, for notch 4 with $\omega_o = 6500$ rad/s. . . . . . 36

4 Dependence of the initiation of a shear band on the prescribed angular speed, calculated by three different criteria, for notch 4 and a load duration of 70 $\mu$s. . . . . . 37

5 Band speed according to different initiation criteria vs. the radial distance from the notch tip; The band initiates when (a) the effective stress attains its maximum value (b) the effective stress has dropped to 90% of its maximum value; (c) and (d) the effective plastic strain equals 0.46 and 0.90 respectively. Results are for notch 4 with $\omega_o =$ 6500 rad/s. . . . . . 38

6 Time history of the torque required to deform the tube with notch 4 for three different durations of the applied angular speed, with $\omega_o =$ 6500 rad/s. . . . . . 39

7 Deformed configuration of a plane passing through the tube's axis at $t = 100$ $\mu$s for notch 4 with $\omega_o = 6500$ rad/s and a load duration of 110 $\mu$s. . . . . . 40

8 Time history of the kinetic energy of the material to the left and right of the notch for notch 5 with $\omega_o = 6500$ rad/s. . . . . . . . . . . . . . 41
9 External work done vs. the radial distance through which a shear band (a contour of effective plastic strain of 0.50) has propagated for notch 4 with $\omega_o = 6500$ rad/s. . . . . . . . . . . . . . . . . . . . . . . . . . 42
10 Stress distribution on a radial line for notch 5 with $\omega_o = 6500$ rad/s at (a) $t = 10$ $\mu$s, and (b) $t = 40$ $\mu$s. . . . . . . . . . . . . . . . . . . . . . 43
11 Dependence of the time of initiation of a shear band upon the defect size as computed by three different criteria, for the case of $\omega_o = 6500$ rad/s. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
12 Time history of the torque required to deform the tube for five different notches with $\omega_o = 6500$ rad/s. . . . . . . . . . . . . . . . . . . . . . . . 45
13 External work done vs. the radial distance through which a shear band (a contour of effective plastic strain of 0.5) has propagated for notches 2, 3, 4 and 5 with $\omega_o = 6500$ rad/s. . . . . . . . . . . . . . . . . . . . 46
14 Band speed vs. the radial distance through which a shear band (a contour of effective plastic strain of 0.5) has propagated for notches 2, 3, 4 and 5 with $\omega_o = 6500$ rad/s. . . . . . . . . . . . . . . . . . . . . 47
15 Loading history for combined pressure and torsional loading. . . . . . 48
16 Time history of (a) the effective stress, (b) the effective plastic strain and (c) the temperature at the centroid of the element below the notch tip for notch 5 with combined axial and torsional loading. . . . . . . . 49
17 Band speed in the radial direction for different axial loads vs. the radial distance through which a shear band (a contour of effective plastic strain of 0.5) has propagated for notch 5 with combined axial and torsional loading. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50

18 Time history of (a) the effective stress and (b) the effective plastic strain at the centroids of eleven elements on a radial line through the notch tip for notch 5 for the case of combined tensile axial and torsional loading with $P_o = 500$ MPa. . . . . . . . . . . . . . . . . . . . . . . 51

19 External work done vs. the radial distance through which a shear band has propagated for notch 5 for the cases of pure torsional loading, and constant tensile axial loading combined with torsional loading. . . . . 52

20 Time history of (a) the effective stress and (b) the effective plastic strain at the centroids of eleven elements on a radial line through the notch tip for notch 5 with a constant internal pressure of 800 MPa and a torsional pulse starting at $t = 40\ \mu s$. . . . . . . . . . . . . . . . . . . 53

21 Variation of shear band (a contour of effective plastic strain of 0.5) speed with the radial distance from the notch tip for notch 5 with an applied internal pressure of 800 MPa followed by torsional loading. . . 54

22 Time history of (a) the effective stress, (b) the effective plastic strain and (c) the temperature at two elements - one just below the notch tip and the other adjoining the inner surface of a CR-300 steel thick-walled tube with notch 5 deformed in torsion. . . . . . . . . . . . . . . . . . . 55

23 Band speed vs. the radial distance from the notch tip of a shear band in a CR-300 steel thick-walled tube with notch 5 deformed in torsion, as computed by the criteria (a) a contour of effective plastic strain of 0.50, and (b) a drop to 90% of the maximum effective stress. . . . . . 56

24 External work done vs. the radial distance through which a shear band (a contour of effective plastic strain of 0.5) has propagated for torsional deformations of the CR-300 steel thick-walled tube with notch 5. . . . 57

25 Time history of the torque required to deform the CR-300 steel thick-walled tube with notch 5. . . . . . . . . . . . . . . . . . . . . . . . . . 58

26 Distribution of the different components of stress on a radial line through the notch tip for torsional deformations of the CR-300 steel thick-walled tube with notch 5 for $\kappa = 800$ K, at (a) $t = 10.5\ \mu s$ and (b) $t = 30.5\ \mu s$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59

# 1 Introduction

Since shear bands usually precede shear fractures in impulsively loaded ductile materials, their study has drawn considerable attention. Even though Tresca (1878) observed them over a century ago during the hot forging of a platinum bar, the activity in the field has picked up since 1944 when Zener and Hollomon (1944) reported observing shear bands during the punching of a hole in a low carbon steel plate. A shear band is a narrow region, usually a few microns (micrometers) wide, that forms in high strain-rate deformations of many metals and some polymers. The primary mode of deformation in these narrow regions is simple shearing. They are called adiabatic since they fully develop in a few microseconds and there is not enough time for the heat to be conducted out of these severely deformed regions. Zener and Hollomon (1944) postulated that the material becomes unstable when the softening caused by heating due to plastic deformation equals the hardening of the material due to strain and strain-rate effects. Analytical works (e.g. see Clifton (1980), Anand et al. (1987)) have assumed that a shear band initiates when the shear stress in a simple shearing problem or the effective stress in a three dimensional problem attains its maximum value. Much of the work done during the last two decades is reviewed in papers included in works edited by Zbib et al. (1992), Armstrong et al. (1994) and Batra and Zbib (1994). Two of the unresolved issues are the energy required to drive a shear band and the effect of a multiaxial state of stress on the initiation and propagation of an adiabatic shear band. We attempt to address these issues by studying the initiation and propagation of an adiabatic shear band in an impulsively loaded thick-walled steel tube with a $V$-notch at the midsection. The stress state near the notch root is expected to be triaxial and because of the stress concentration, a shear band should initiate there first. Previous experimental (e.g. see Marchand

and Duffy (1988)) works on the torsion of thin-walled steel tubes and their numerical simulation (Wright and Walter (1987), Batra and Kim (1992)) as simple shearing problems have revealed that the shear stress drops catastrophically at the instant of the initiation of a shear band. Batra and Zhang (1994) studied numerically the torsion of a thin-walled steel tube as a 3-dimensional problem and assumed that the yield stress of a small region near the center of the tube was 5% less than that of the rest of the material. They found that the torque required to deform the tube dropped very rapidly when a shear band, as evidenced by distortions of the deformed mesh, initiated.

## 2 Formulation of the Problem

We use the referential or Lagrangian description of motion to study dynamic thermo-mechanical deformations of a thick-walled steel tube; these deformations are governed by the following balance laws of mass, linear momentum, moment of momentum and internal energy (e.g. see Truesdell and Noll (1965)).

$$\rho J = \rho_o \tag{1}$$

$$\rho_o \dot{\mathbf{v}} = \text{Div } \mathbf{T} \tag{2}$$

$$\mathbf{F}^T \mathbf{T} = \mathbf{T}^T \mathbf{F} \tag{3}$$

$$\rho_o \dot{e} = tr(\mathbf{T} \dot{\mathbf{F}}^T) \tag{4}$$

Here $\rho_o$ is the mass density of a material particle in the reference configuration, $\rho$ its present mass density, $\mathbf{F}$ the deformation gradient, $J = \det \mathbf{F}$, $\mathbf{v}$ the present velocity of a material particle, a superimposed dot denotes the material time derivative, $\mathbf{T}$ the first Piola-Kirchhoff stress tensor, $\mathbf{F}^T$ the transpose of $\mathbf{F}$, $e$ the internal energy density, tr denotes the trace operator and Div signifies divergence with respect to coordinates in the reference configuration. In eqn. (4) the effect of heat conduction has been neglected. This is justified because a shear band develops in a few microseconds and there is not enough time for the heat to be conducted out of the band. Batra and Kim (1991) have shown through numerical experiments that heat conduction has a negligible effect on the time of initiation of a shear band. As is the case in continuum mechanics, we require that the balance of moment of momentum (3) be identically satisfied. In eqn. (4) we have assumed that all of the plastic working, rather than 90-95% of it as asserted by Farren and Taylor (1925) and Sulijoadikusumo and Dillon (1979), is converted into heating. We postulate the following constitutive relations for the material of the tube.

$$\mathbf{T} = J\boldsymbol{\sigma}(\mathbf{F}^{-1})^T, \qquad \boldsymbol{\sigma} = -p\mathbf{1} + \mathbf{S}, \qquad p = K(\rho/\rho_o - 1), \tag{5}$$

$$\overset{\triangledown}{\mathbf{S}} = 2\mu(\overline{\mathbf{D}} - \overline{\mathbf{D}}^p), \qquad \overset{\triangledown}{\mathbf{S}} = \dot{\mathbf{S}} + \mathbf{SW} - \mathbf{WS}, \tag{6}$$

$$\mathbf{D} = \frac{1}{2}(\text{grad } \mathbf{v} + (\text{grad } \mathbf{v})^T), \qquad \mathbf{W} = \frac{1}{2}(\text{grad } \mathbf{v} - (\text{grad } \mathbf{v})^T), \tag{7}$$

$$\overline{\mathbf{D}} = \mathbf{D} - \frac{1}{3}(tr\ \mathbf{D})\mathbf{1}, \qquad tr\mathbf{D}^p = 0, \qquad \mathbf{D}^p = \Lambda\mathbf{S}, \qquad S_e^2 \equiv \frac{3}{2}tr(\mathbf{SS}^T), \tag{8}$$

$$\dot{e} = c\dot{\theta} + tr(\boldsymbol{\sigma}\mathbf{D}^e), \qquad \sigma_y = (A + B(\epsilon^p)^n)(1 + C\ ln(\dot{\epsilon}^p/\dot{\epsilon}_o))(1 - T^m), \tag{9}$$

$$T = (\theta - \theta_o)/(\theta_m - \theta_o), \qquad (\dot{\epsilon}^p)^2 = \frac{2}{3}\ tr(\mathbf{D}^p\mathbf{D}^p). \tag{10}$$

Here $\boldsymbol{\sigma}$ is the Cauchy stress tensor, $\mathbf{S}$ its deviatoric part, $p$ the hydrostatic pressure taken to be positive in compression, $K$ the bulk modulus, $\mu$ the shear modulus, $\overset{\triangledown}{\mathbf{S}}$ the Jaumann derivative of $\mathbf{S}$, $\overline{\mathbf{D}}$ the deviatoric strain-rate, $\mathbf{D}^p$ the plastic strain-rate, $\mathbf{D}^e$ the elastic strain-rate, $\mathbf{D} = \mathbf{D}^e + \mathbf{D}^p$ the strain-rate tensor, $\mathbf{W}$ the spin tensor, $c$ the specific heat, $\theta$ the temperature of a material particle, $\theta_m$ its melting temperature, $\theta_o$ the room temperature, $T$ the homologous temperature, $S_e$ the effective stress, and $\epsilon^p$ the effective plastic strain. Equation $(5)_3$ implies that the volumetric response of the material is elastic. Equation $(6)_1$ is the constitutive relation in terms of deviatoric stresses for a linear isotropic hypoelastic material, grad $\mathbf{v}$ equals the gradient of the velocity field with respect to coordinates in the present configuration, eqn. $(8)_4$ signifies the von Mises yield criterion with isotropic hardening, and eqn. $(9)_2$ is the Johnson-Cook (1983) relation. The flow stress, $\sigma_y$, increases with an increase in the effective plastic strain and the effective plastic strain rate but decreases with an increase in the temperature of a material particle. Truesdell and Noll (1965) have pointed out that eqn. $(9)_1$ is not invariant with respect to the choice of different objective (or material frame indifferent) time derivatives of the stress tensor. In eqn. $(9)_2$ parameters $B$ and $n$ characterize the strain hardening of the material, $C$ and $\dot{\epsilon}_o$

its strain-rate hardening and $(1 - T^m)$ its thermal softening. Equation $(8)_3$ signifies that the plastic strain-rate is along the normal to the yield surface $(8)_4$, and the factor of proportionality $\Lambda$ is given by

$$\Lambda = 0 \text{ when either } S_e < \sigma_y, \text{ or } S_e = \sigma_y \text{ and } tr(\mathbf{S}\dot{\mathbf{D}}^p) < 0; \tag{11}$$

otherwise it is a solution of

$$S_e = (A + B(\epsilon^p)^n)(1 + C\ ln(\frac{2}{3}\Lambda S_e/\dot{\epsilon}_o))(1 - T^m). \tag{12}$$

Our constitutive hypotheses (5) and $(9)_2$ imply that the flow stress required to deform a material point plastically vanishes once its temperature equals the melting temperature of the material; the material point will then behave as a compressible, nonviscous fluid. In physical experiments, fracture in the form of a crack will ensue from the point much before it is heated up to the melting temperature of the material. Here we have not incorporated any fracture criterion into the problem formulation, and in our numerical simulations for the 4340 steel tube, no material point reaches its melting temperature.

We take the body to be initially at rest, stress free and at a uniform temperature $\theta_o$. All bounding surfaces of the tube are taken to be thermally insulated; this is consistent with the assumption of locally adiabatic deformations. One end face of the tube is rigidly clamped and the other end is loaded either by a prescribed angular speed, or a normal pressure followed by a prescribed angular speed. In another case, the angular speed is preceded by a pressure load on the inner wall of the tube. In each case, the prescribed variable increases linearly in time from zero to the assigned value in 20 $\mu$s, stays there for the desired duration, and then decreases linearly to zero in 20 $\mu$s. The time duration during which the prescribed quantity stays fixed is varied to change the energy input into the system. The rise time of 20 $\mu$s is typical of torsional tests conducted in a split Hopkinson bar.

# 3 Computation and Discussion of Results

We assigned the following values to various material and geometric parameters in order to compute numerical results presented and discussed below in sections 3.1 through 3.4.

$$
\begin{aligned}
&\rho_o = 7,840 \text{ kg/m}^3, \quad \mu = 76 \text{ GPa}, \quad K = 157 \text{ GPa} \\
&A = 792.2 \text{ MPa}, \quad B = 509.5 \text{ MPa}, \quad n = 0.26, \quad C = 0.014, \quad m = 1.03 \\
&\theta_m = 1793\ K, \quad \theta_o = 298\ K, \quad \dot{\epsilon}_o = 1/s, \quad c = 477 \text{ J/kg}^{\circ}C \\
&\text{Inner radius} = 1.27 \text{ mm}, \quad \text{Outer radius} = 4.445 \text{ mm}, \quad \text{Tube length} = 16.51 \text{ mm}.
\end{aligned}
\tag{13}
$$

The values of material parameters in the Johnson-Cook model for 4340 steel are taken from Rajendran (1992). The range of effective plastic strains, effective plastic strain rates and temperatures used to obtain these values is much smaller than that likely to occur in a shear band problem. Klepaczko et al. (1987) have pointed out that nearly all of the material parameters in (13) depend upon the temperature. However, such temperature dependence is not considered herein primarily because of the difficulty in finding test data over the wide range of strains, strain-rates and temperatures likely to occur within a shear band.

The coupled thermomechanical problem formulated in the preceding section is highly nonlinear and can not be solved analytically; therefore we seek its approximate solution by the finite element method and employ the large scale explicit finite element code DYNA3D developed by Whirley and Hallquist (1991). The code uses 8-noded brick elements, one-point quadrature rule, an hour-glass control to suppress spurious modes, and adjusts the time step adaptively to satisfy the Courant condition; thus the stability condition is satisfied. Because of the use of a one-point integration rule, stresses, strains and temperatures in an element are assumed to be constants.

In DYNA3D, artificial bulk viscosity is added to smear out the shocks; this may influence the time at which a shear band initiates. In this method, the pressure in elements being compressed is augmented by an artificial viscous term $q$ given by

$$q = \rho\hat{l}\ |tr(\mathbf{D})|\ (Q_1\hat{l}\ |tr(\mathbf{D})|\ +\ Q_2\hat{c}). \tag{14}$$

Here $Q_1$ and $Q_2$ are dimensionless constants which default to 1.5 and 0.06 respectively, $\hat{l}$ is the cube root of the volume of the element, $\hat{c}$ is the speed of sound in the material and equals $((K + 4\mu/3)/\rho_o)^{\frac{1}{2}}$. Batra and Adulla (1995) have shown that different values of $Q_1$ and $Q_2$ have virtually no effect on the instant of initiation of an adiabatic shear band. Results presented below are for default values of $Q_1$ and $Q_2$. The code neglects the effect of heat conduction which has the advantage that the time step, $\Delta t$, is controlled by the mechanical problem, and the temperature rise, $\Delta\theta$, at the centroid of an element is computed from

$$\rho c\Delta\theta = tr(\mathbf{S}\mathbf{D}^p)\Delta t \tag{15}$$

which follows from equations (5), $(8)_{1,2}$ and $(9)_1$.

The finite element mesh generally consisted of 11 elements in the radial direction, 71 elements in the axial direction and 108 elements in the circumferential direction; a finer mesh could not be used within available computational resources. The mesh was essentially uniform in the radial direction but was slightly graded in the axial direction away from the notch; the aspect ratio for elements near the notch was close to 1, while the remainder of the elements had aspect ratios no greater than 3.5. As shown by Batra and Ko (1992), an adaptively refined mesh gives sharper results for the rate of evolution of a quantity within a shear band but does not affect when a shear band initiates. Since most of the results presented herein involve the initiation of a shear band, they are representative of the problem studied. Because of our desire to directly evaluate quantities at points on the midsurface of the tube, the mesh had one row of elements with their centroids at the midsurface. This blunted the root of the $V$-notch causing the notch to look more like an open channel; a finite element discretization of the tube and a sketch of the $V$-notch are shown in Fig. 1. The notch shape was varied by changing the dimension $d$; it equals 0.142, 0.221, 0.3, 0.379 and 0.458 mm respectively, for notches 1, 2, 3, 4 and 5.

## 3.1 Torsion of the Tube

The tube is deformed by keeping one end face stationary and twisting the other end face by prescribing on it an angular speed that increases linearly from zero to the maximum value, $\omega_o$, in 20 $\mu$s, keeping it steady for $t_r$ microseconds, then decreasing it linearly to zero in 20 $\mu$s and keeping it at zero. The average shear strain-rate is least for points on the inner surface of the tube and most at points on the outer surface. As should be clear from the distribution of the effective plastic strain on a radial line plotted in Fig. 2 for different values of the maximum angular speed applied to the tube with notch 4, and for a 70 $\mu$s total duration of the loading pulse, at lower values of the prescribed maximum angular speed elements below the notch tip do not undergo much plastic deformation at $t = 20$ $\mu$s. Even at $t = 70$ $\mu$s and $\omega_o = 2786$ rad/s the material in nearly half of the thickness of the tube adjoining its inner surface has an effective plastic strain of at most 0.25. This is because the external work done during the loading of the body or the energy supplied by the loading pulse is not enough to cause intense plastic deformations of the material underneath the notch bottom. Figure 3 depicts the time history of the effective stress, effective plastic strain and temperature at the centroids of 11 elements on a radial line through the notch tip for $\omega_o = 6500$ rad/s and $t_r = 30$ $\mu$s; curve 1 is for the first element below the notch tip, and curve 11 is for the element adjoining the inner surface of the tube. Even though the problem was analyzed as three-dimensional, the deformations were found to be independent of the angular position of a material point. It is clear from these plots that once the elastic shear loading wave arrives at the centroid of the notch at about 2.65 $\mu$s, the effective stress in the element abutting the notch root rises sharply because of the stress concentration there. We note that the elastic shear loading wave arrives at these eleven elements at the same time. The different rate of rise of the effective stress in these elements is due to the fact that the average shear strain-rate and thus the average shear strain increases, at least during the initial elastic part of deformations, with the radial distance of a point from the tube's axis. Also, the effects of stress concentration and the triaxiality of deformations due to the notch diminish rapidly with the distance from the notch tip; we address these issues a little later. Since the effective stress after having reached its peak value drops rather gradually, it is hard to decide when a shear band initiates. We note that during torsional tests on thin-walled tubes, Marchand and Duffy (1988) observed a catastrophic drop in the shear stress at the initiation of a shear band; similar rapid drops in the shear stress were computed by Wright and Walter (1987) during their numerical simulations of the test as a simple shearing problem, and by Batra and Zhang (1994) who analyzed it as a three-dimensional problem. Batra and Kim (1992) studied the simple shearing problem for twelve materials and postulated that a shear band initiates in earnest when the shear stress has dropped to 90% of the peak value. Deltort (1994) has associated the initiation of severe localization of deformation with the instant when the shear stress has dropped to 80% of its peak value. Analytical works (e.g. see Clifton (1980)) usually assume that a shear band initiates when the shear stress in a simple shearing problem attains its peak value. Marchand and Duffy did not report the values of the effective plastic strain within a shear band at the instant of its initiation, but indicated a maximum shear strain of 20 within a fully developed band. Zhou et al. (1996) studied the development of shear bands in a single-notched CR-300 steel plate and assumed that a material particle within a shear band fails and subsequently behaves as a nonlinear viscous fluid once the effective plastic strain there equals 0.4. As is evident from the plots of the evolution of the effective plastic strain and the temperature, their values at the instant of initiation of a shear band will vary with the criterion used for the initiation of a shear band. For example, at the instant of the effective stress attaining its peak value in the element just below the notch tip, the effective plastic strain and the temperature in it equal 0.39 and 154°C respectively. However, when the effective stress has dropped to 90% of its peak value in that element, the effective plastic strain and the temperature there equal 1.26 and 436°C respectively. In either case, the temperature in the element at the instant a shear band initiates is nowhere close to its melting temperature. Of course, the temperature and the effective plastic strain there continue to increase at an increasing rate during the postlocalization period until the time the loaded end is brought to rest and is subsequently kept at rest. The temperature in the element below the notch tip at $t = 60$ $\mu$s reaches 1156°C, which equals 80% of the presumed melting temperature of the material.

By adopting a criterion for the initiation of a shear band, one can ascertain the time at which a shear band initiates in different elements on a radial line through the notch tip and thus compute the band speed in the radial direction. Figure 4 shows the dependence of the time of the initiation of a shear band in the first element below the notch tip on the prescribed angular speed as computed by the following three criteria: (i) the effective plastic strain equals 0.50, (ii) the effective stress attains its maximum value, and (iii) the effective stress has dropped to 90% of its maximum value. It is clear from the figure that the initiation time depends on the rate of loading as well as the definition chosen to characterize the initiation of a shear band. Figure 5 illustrates, for $\omega_o = 6500$ rad/s, the variation of the band speed with the radial distance from the notch tip computed by criterion (ii) and (iii) stated above, and by the following additional criteria: (iv) the effective plastic strain equals 0.46, and (v) the effective plastic strain equals 0.90. For each case, the band speed starts out at about 50 m/s when it initiates in an element just below the notch root, and begins to increase as it propagates radially inwards; the variation of speed with the radial distance is not monotonic. Both the band speed and its variation with the radial distance depend strongly upon the criterion used for the initiation of a shear band; the maximum band speed computed is 130 m/s. Possible reasons for getting an erratic variation of the band speed with the radial distance obtained by using criterion (ii) are the coarseness of the finite element mesh, the discrete times at which the output is printed, and the error in estimating exactly when the peak stress occurs in an element. Peng and Batra (1995) analyzed shear bands in a 4340 steel thick-walled tube with a hemispherical cavity centered at the outer surface of the midsection of the tube. In this case a shear band initiating at the bottom of the cavity could propagate in the axial, circumferential, and radial directions. They found that a contour of effective plastic strain of 0.7 propagated in the circumferential and radial directions at approximately 600 m/s and 60 m/s respectively, and the speed increased in both directions as the band propagated further. Marchand and Duffy (1988) reported that a shear band initiating from a point defect in a thin-walled HY-100 steel tube propagated in the circumferential direction either at an average speed of 520 m/s or at 260 m/s depending upon whether it traveled in one direction only, or in both directions at the same speed. Computational work of Batra and Zhang (1994) supports the second alternative suggesting that the average shear band speed in Marchand and Duffy's work was 260 m/s.

As can be inferred from the plots of Fig. 3, at the instant of the initiation of a shear band in the element just below the notch tip, according to any one of the aforementioned criteria, there is a steep gradient in the distribution of the temperature and the effective plastic strain on a radial line through the notch tip. Even when the shear band has propagated to the innermost surface of the tube, the temperature and the plastic strain vary through the thickness of the tube. These quantities become steady at a material point for $t \geq 70$ $\mu s$ since the two end faces of the tube are subsequently kept fixed. The consideration of heat conduction would tend to stabilize these fields sooner.

In Fig. 6 we have plotted the time history of the torque required to deform the tube with notch 4 for three different durations of the applied angular speed; in each case the angular speed increases linearly from zero to the steady value of 6500 rad/s in 20 $\mu$s and eventually decreases linearly to zero in 20 $\mu$s. There is no sharp drop in the torque, as in the case of a thin-walled tube, to signify the initiation of a shear band. In a thin-walled tube a shear band essentially initiates simultaneously at all points on a radial line through the point of minimum wall thickness. Once it happens, the load carrying capacity of the tube is significantly diminished. For a thick-walled tube even when a shear band initiates in an element on the outer surface of the tube, elements interior to it are either deforming elastically or even if deforming plastically may still be on the rising part of the effective stress vs. effective strain curve. Thus the torque required to deform the tube may even increase first and then decrease gradually. The maximum value of the torque occurs at about $t = 20$ $\mu$s when the prescribed angular speed attains its steady value. For this loading condition, at $t =$ 20 $\mu$s, elements just below the notch tip have started to soften in the sense that the effective stress there has peaked and has started to decrease. When the prescribed angular speed begins to decrease, the torque drops quickly to indicate the onset of an elastic unloading wave. The time period of the oscillations in the torque equals the time taken for an elastic wave to propagate through twice the length of the tube.

Figure 7 depicts the deformed configuration, at $t = 100$ $\mu s$, of a plane passing through the tube's axis when the total load duration equals 110 $\mu s$. It is clear that only the material layer beneath the notch tip undergoes severe plastic deformations; it was confirmed by plotting the distribution of the effective plastic strain on an axial line. The plot, not shown here for the sake of brevity, indicated that for $t \geq 30$ $\mu s$, the plastic strain accumulated only within this one element thick layer below the notch bottom. The band width, defined as the thickness in the axial direction of the severely deformed region, equals the axial dimension of the element abutting the bottom surface of the notch. Of course, one needs a much finer mesh to estimate the width of a shear band. As mentioned earlier, the available computational resources limited the fineness of the mesh. The time history of the kinetic energy of the material to the left and right of the one-element wide shear band, plotted in Fig. 8 for notch 5 with $\omega_o = 6500$ rad/s, indicates that subsequent to the initiation of a shear band at $t \simeq 20$ $\mu$s, the material to its right is virtually stationary and that to its left moves as a rigid body. Thus the tangential velocity is discontinuous across a shear band; a similar conclusion was drawn by Batra and Jin (1994) who studied the development of a shear band in a porous elasto-thermo-viscoplastic material deformed in plane strain tension and confirms Tresca's (1878) conjecture.

From the time history of the torque plotted in Fig. 6 and knowing the prescribed angular speed, we can compute the work done by external forces or energy input into the body. A part of this energy is used to change the kinetic energy of the body and the rest to deform it. The strain energy density of the elastic deformations is negligible as compared to the work done/volume required to deform the body plastically. After a shear band has initiated in elements below the notch tip, the kinetic energy of the tube remains essentially uniform until the prescribed angular speed begins to decrease. We have plotted in Fig. 9 the work done by external force versus the radial distance through which a shear band, defined as a contour of effective plastic strain of 0.5, has propagated for notch 4 with $\omega_o = 6500$ rad/s; the plot is essentially a straight line. The slope of this line, 274 kJ/m, equals the energy required to drive a shear band through a unit distance. It is reasonable to conjecture that this value depends upon the material of the tube, its inner and outer radii, the criterion used to define a shear band, and the loading conditions. Noting that the width of the band is 0.119 mm and the external work done between the time it initiates in the element just below the notch tip and propagates to the innermost surface of the tube is 704 J, the energy required to drive a shear band through the material equals 126.2 J/mm$^3$.

## 3.2 Effect of Notch Depth

Keeping the torsional loading pulse of 110 $\mu$s total duration fixed, we varied the notch size by assigning values of 0.142, 0.221, 0.3, 0.379 and 0.458 mm to the depth, $d$, of the notch (cf. Fig. 1b); these notches are identified as 1, 2, 3, 4 and 5 respectively. The results presented in this section are for a maximum applied angular speed of $\omega_o = 6500$ rad/s. Figure 10a illustrates for notch 5 the distribution of different stress components on a radial line through the notch tip at time $t = 10$ $\mu s$, when the deformations of the tube are expected to be elastic; it is evident that $\sigma_{z\theta}$ grows rapidly near the notch tip ($z$-axis is along the tube's axis) and all of the remaining stress components are at least an order of magnitude smaller. A closer look at their values reveals that $\sigma_{rr}$, $\sigma_{zz}$ and $\sigma_{\theta\theta}$ remain essentially zero through the thickness of the tube, and $\sigma_{zr}$ and $\sigma_{r\theta}$ while remaining small do exhibit a singular behavior. Extrapolating $\sigma_{z\theta}$ to the notch tip, and also evaluating it from $Tr/J_e$, where $T$ is the torque, $J_e = \pi(r_o^4 - r_i^4)/2$, $r_o$ and $r_i$ being the outer and inner radii of the tube, and $r$ is the radial distance of a point from the tube's axis, we obtain a stress concentration factor of 1.64 which agrees well with that given in Fig. 10.2 of Dowling's book (1993). A plot of $\sigma_{z\theta}$ vs. $1/\sqrt{s}$ where $s$ is the radial distance from the notch tip gives a stress-intensity factor of 6.61 MPa$\sqrt{m}$. In Fig. 10b we have plotted the distribution of stress components on a radial line at $t = 40$ $\mu$s when a part of the cross-section of the tube passing through the notch-tip has deformed plastically. Because of the softening of the material near the notch tip, caused by heating due to plastic deformation, $\sigma_{z\theta}$ near the notch tip is lower than that at other points on the radial line; the variation of $\sigma_{\theta\theta}$, $\sigma_{zz}$ and $\sigma_{rr}$ vs. $s$ exhibits a singular behavior and their maximum magnitudes at the notch tip equal approximately one-fourth the magnitude of $\sigma_{z\theta}$ there. As expected, the stress distribution on a radial line is quite different when the material is undergoing plastic deformations as compared to that when it is deforming elastically. The stress state at points ahead of the shear band is triaxial.

Figure 11 depicts the dependence of the time of initiation of a shear band in an element just below the notch tip upon the defect size defined as the notch depth, $d$, divided by the tube thickness at a cross-section away from the notch. Batra et al. (1996) used DYNA3D to study torsional deformations of a thin-walled tube whose thickness varied sinusoidally with the minimum thickness occurring at the midsection. They found that the average critical strain (or the time) at which a shear band initiated, as indicated by the sharp drop in the torque required to deform the tube, varied exponentially with the defect parameter $\epsilon$ = (1 - minimum tube thickness/maximum tube thickness). The slope of the average critical strain vs. log $\epsilon$ curve was found to be independent of the nominal strain. These results are in qualitative agreement with the test results of Chi (1990), Murphy (1990), and Deltort (1994), and analytical results of Molinari and Clifton (1987) and Wright (1994). For the thick-walled tube studied here, the curve representing the time of initiation of a shear band vs. log (defect size) is not a straight line. This could either be due to an improper definition of the defect size and/or the triaxiality of the stress-state near the notch tip; it does not depend upon the criterion used for the initiation of a shear band. We note that the analytical studies of Molinari and Clifton (1987) and Wright (1994) are for a simple shearing problem which is a good model for torsional deformations of a thin-walled tube.

In Fig. 12 we have plotted the time history of the torque required to deform the tube for each of the five notches. For notch 1 the torque continues to increase until $t = 90\ \mu$s when the prescribed angular speed begins to decrease, and no shear band initiates. The time when the torque starts to drop sharply decreases with an increase in the notch depth and for notch 5 the torque drops when the prescribed angular speed has reached its steady value of 6500 rad/s. For notch 2 a sharp drop in the torque commences at $t \simeq 62$ $\mu$s but a shear band initiates in an element below the notch tip at $t = 34,\ 36$ or $53$ $\mu$s according to criteria (i), (ii) and (iii) of section 3.1. A similar observation can be made for the other three notches. Thus the instant of the sharp drop in the torque required to deform the tube need not coincide with the time when a shear band first initiates at a point in the tube. Figure 13 illustrates, for notches 2 through 5, the external work done required to drive a shear band defined as the boundary of the region with an effective plastic strain of at least 0.5. We note that a part of the external work done is required to change the kinetic energy of the body. However, once a shear band has developed, the kinetic energy of the tube deformed by prescribing the angular speed at its end faces remains constant as long as the prescribed angular speed is uniform. During the time interval when the shear band propagates from $s = 1.0$ mm to $s = 2.5$ mm, the slopes of the curves for notches 3, 4 and 5 equal 316, 274 and 260 kJ/m respectively; they give the energy required to drive a shear band through a unit distance.

The variation of the band speed with the radial distance from the notch tip for notches 2, 3, 4 and 5 is plotted in Fig. 14; the band is taken to initiate at a point when the effective plastic strain there equals 0.5. Except for the initial period during which the band propagates away from the notch tip, the band speed can be considered as essentially uniform, independent of the notch depth, and equals 90 m/s. Thus the defect size influences the initial speed of a shear band but has very little effect upon its speed once it has propagated away from the defect.

## 3.3 Effect of Prior Axial Loading

For the 4340 steel thick-walled tube with notch 5 at its center, we consider the loading history shown in Fig. 15. That is, the tube is first loaded by an axial compressive pressure which increases from zero to the final value $P_o$ in 20 $\mu$s, is kept steady for 30 $\mu$s and then decreases to zero in 20 $\mu$s. Before the pressure begins to decrease at $t = 50$ $\mu$s, one end of the tube is held fixed and the other twisted by a prescribed angular speed that increases linearly from zero at $t = 40$ $\mu$s to 6500 rad/s at $t = 60$ $\mu$s, is held there till $t = 90$ $\mu$s and then decreases linearly to zero at $t = 110$ $\mu$s. Three steady values, 100, 500 and 1000 MPa, of the applied axial pressure are considered. We have also investigated two cases in which the axial traction is maintained at 500 MPa or -500 MPa, for $t \geq 20$ $\mu$s. Recalling that the quasistatic yield stress of the material at room temperature equals 792 MPa, the axial traction of 1000 MPa will deform the tube plastically prior to the application of the torsional loading. Because of stress concentration near the notch root, the axial traction of 500 MPa will also plastically deform the element abutting the notch tip. The above-mentioned loading histories should simulate those experienced by the rod deformed in a split compression and torsion Hopkinson bar. Since the compression wave travels faster than the torsion wave, the tube will first be deformed in compression and then in torsion; the delay time between the arrival of the two waves at a tube face depends upon the length of the incident bar. The combined loading considered herein is not meant to simulate any specific experimental set-up. Furthermore, in our simulations, one end face of the tube is kept fixed and the other loaded; this is dictated by the way the stipulated combined loading can be easily applied in DYNA3D. Preventing points on the fixed end from moving laterally, and those on the loaded end from moving radially, will induce additional stresses there. This local effect should not influence, for moderate values of the axial load, deformations of the material neighboring the notch at the midsection of the tube.

Figures 16a, 16b and 16c exhibit, for different values of the axial load, the time history of the effective stress, the effective plastic strain and the temperature at the centroid of an element just below the notch tip; we have also included results for the case when the pure torsional loading begins at $t = 40\ \mu$s, and there is no axial load applied. We first discuss results for axial loads of finite duration. Because of the stress concentration at the notch tip, for $P_o = 500$ MPa, the effective stress there exceeds the quasistatic yield stress of the material. However, since the axial load begins to decrease 10 $\mu s$ after the torsional load is applied, by the time the prescribed angular speed reaches its steady value the axial stress everywhere is below the yield stress of the material. Since $P_o < A$, not much plastic strain is accumulated prior to the application of the torsional loading. However, for $P_o = 1000$ MPa, the tube has been deformed plastically prior to being deformed in torsion. The plastic deformations caused by the torsional loading are significantly more than those induced by the axial compressive load. Except for $P_o = 1000$ MPa, the effect of prior axial pressure is minimal on the plastic deformations of the element under the subsequent torsional loading. For $P_o = 1000$ MPa, the prior plastic deformations of the element result in higher values of the effective plastic strain as compared to those affected by pure torsional loading.

For a constant pressure of $\pm$ 500 MPa, the effective stress drops gradually, and the effective plastic strain and the temperature rise slowly as compared to those for the other four cases; also the peak in the effective stress occurs later than that in the other four cases studied. Thus, the presence of axial stresses tends to delay the initiation of a shear band and the shear band grows less rapidly under combined axial and torsional loading as compared to that under the same pure torsional loading.

Murphy (1990) has tested thin-walled steel tubes loaded first quasistatically in simple compression and then twisted dynamically. He found that an increase in the prior compressive load increased the nominal shear strain at which a shear band initiated as indicated by the sharp drop in the shear stress or the torque required to deform the tube. Our computed result for the constant pressure of 500 MPa agrees qualitatively with Murphy's observations.

Figure 17 exhibits the variation of the band speed, defined as a contour of effective plastic strain of 0.5, as it propagates radially inwards through the thickness of the tube; it is evident that the prior axial loading does not significantly influence the speed of the band. However, when the axial traction is maintained, the band propagates radially through a much smaller distance and at a slower speed in comparison to that in the other four cases in which the axial load becomes zero by the time the applied angular speed attains its steady value. The band propagates a little further for tensile axial traction as compared to that for compressive loading of the same magnitude. We note that when the tube is loaded simultaneously in compression/tension and torsion, the maximum shear stress at a point need not occur on a surface perpendicular to the tube's axis. In order to compare results, we have determined in every case the speed of a shear band in the radial direction.

Figures 18a and 18b depict the time history of the effective stress and the effective plastic strain at the eleven elements on a radial line through the notch tip for the combined loading case with a constant tensile axial traction of $P_o = 500$ MPa. Figure 18a shows that before the torsional load begins to drop at 90 $\mu s$, the effective stress drops to 90% of its maximum value in only one element below the notch tip. It is evident from Fig. 18b that a contour of effective plastic strain of 0.5 propagates to five elements through the thickness during the first 90 $\mu s$. This provides us with an example where the two definitions of the initiation of a shear band yield totally different answers for the distance through which a band propagates.

We note that the work done by the applied axial traction is essentially negligible as compared to that done by the torque; for example, the total work done by the constant axial pressure of 500 MPa during the load duration of 110 $\mu s$ is only 3 J whereas that done by the torque is 1920 J. The results plotted in Fig. 19 clearly indicate that significantly more energy is required to drive a shear band for the case of combined loading as compared to that for pure torsional loading.

## 3.4 Effect of Internal Pressure

An internal pressure in a thick-walled tube will produce a biaxial state of stress with the maximum principal stress occurring at a point on the inner surface of the tube. Thus, for a thick-walled tube with a V-notch at the midsection, loaded first by an internal pressure (with no axial loads) and then by a torsional load, a shear band initiating from the notch tip will propagate radially inwards towards a region of increasing prior plastic deformation. We considered two types of loading; in the first case, the internal pressure increases linearly from 0 to 800 MPa in 20 $\mu s$, is kept steady for 30 $\mu s$ and then decreases linearly to zero in 20 $\mu s$. At $t = 40$ $\mu s$, one end face of the tube is kept fixed while the other end is twisted with an angular speed that increases linearly from zero to 6500 rad/s in 20 $\mu s$, is kept steady for 30 $\mu s$ and then decreases to zero in 20 $\mu s$. The second loading differs from the first only in one respect, the internal pressure is kept steady for $20 \leq t \leq 110$ $\mu s$. As in the case of the axial loading, constraining points on the fixed face from moving laterally and those on the loaded end from moving radially will induce additional stresses at points near the end faces. For moderate values of the internal pressure, these effects will stay local and will not significantly influence deformations of the material close to the midsection; these end effects are neglected in the following discussion. For these two loadings, the time histories of the effective stress, the effective plastic strain and the temperature at the centroids of eleven elements on a radial line through the notch tip essentially overlapped each other; those for the constant pressure case are depicted in Fig. 20 wherein we have omitted the plots of the time history of the temperature since they look similar to those of the effective plastic strain. Prior to the application of the torsional loading, the effective stress is maximum at a point on the inner surface of the tube, and the plastic deformation has propagated radially outwards only through two elements. With the application of the torsional loading, the effective stress increases sharply in the element below the notch-tip and a shear band initiates there first and propagates inwards. The effective stress in an element adjoining the inner surface of the tube drops much more gradually than that for the pure torsional loading. In Fig. 21, we have plotted the variation of the band speed (speed of a contour of effective plastic strain of 0.5) with the radial distance from the notch tip. It is evident that the multiaxial state of stress and/or the prior plastic deformation of the material decreases the speed of the shear band in the radial direction. It could be due to the fact that the maximum shear strain at a point on a cross-section perpendicular to the tube's axis is not in the radial direction.

## 3.5 Torsion of a CR-300 Steel Tube

Zhou et al. (1996) have recently reported observed and computed shear band speeds in impact loaded prenotched CR-300 steel plates. Depending upon the impact speed, the observed average shear band speed varied from 50 m/s to 1000 m/s, and the corresponding computed average shear band speed ranged between 70 m/s and 1200 m/s. The failure mode in the impact loaded prenotched plate is close to Mode II and that in the torsional loading of a thick-walled tube is more like Mode III. A major difference between our and their numerical simulations is in the thermal softening function used in the constitutive relation $(9)_2$; thus we replace it by

$$\sigma_y = max\{(A + B(\epsilon^p)^n)(1 + C\ ln(\dot{\epsilon}^p/\dot{\epsilon}_o))[1 - \delta(exp((T - T_o)/\kappa) - 1)], 0\} \tag{16}$$

where $T_o$ equals the room temperature. Values of material parameters A, B, n, C, $\dot{\epsilon}_o$, $\delta$, and $\kappa$ obtained by fitting curves to their data and of other material parameters used in the results reported herein are given below; geometric parameters were assigned values given in (13).

$$\begin{aligned} &\rho_o = 7,830\ \text{kg/m}^3, \quad \mu = 76.9\ \text{GPa}, \quad K = 164.7\ \text{GPa}, \quad \delta = 0.8 \\ &A = 2,000\ \text{MPa}, \quad B = 94.5\ \text{MPa}, \quad n = 0.2, \quad C = 0.0165, \\ &\theta_o = 293\ K, \quad \dot{\epsilon}_o = 1.3\ \text{x}\ 10^{-3}/s, \quad c = 448\ \text{J/kg}^\circ C, \quad \kappa = 500\text{K} \end{aligned} \tag{17}$$

For

$$T > \kappa\ \ ln(1 + \delta) + T_o \tag{18}$$

the expression on the right hand side of (16) will be negative; thus $\sigma_y = 0$. For the values of the material parameters given in (17), $\sigma_y = 0$ for $T = 698$ K which implies that the melting temperature of the material has been taken to be too low. We also investigate the problem for $\kappa = 800$ K which corresponds to a melting temperature of the material of 941 K. Zhou et al. (1996) also assumed that a material point failed if $\epsilon^p$ there equaled 0.40 and it subsequently behaved as a nonlinear viscous fluid. Here we do not adopt this failure criterion and assume that a material point behaves as a perfect fluid once $\sigma_y = 0$ there. Because of the Lagrangian formulation used in DYNA3D, a fluid element will undergo intense deformations in essentially no time and the finite element mesh will become severely distorted resulting in an unacceptable size of the time step. The thermal softening function of eqn. (16) was incorporated into the material subroutine in DYNA3D.

We investigate the torsional deformations of the CR-300 steel thick-walled tube under the loading conditions of section 3.1, with $\omega_o = 6500$ rad/s and a load duration of 70 $\mu$s. Figure 22 exhibits the time history of the effective stress, effective plastic strain and temperature at the centroids of two elements - one element just below the notch tip and the other adjoining the inner surface of the thick-walled tube with notch 5 at its midsection. For $\kappa = 500$ K, a shear band initiates, according to criterion (iii) of section 3.2, in the element below the notch tip at $t = \ 23.5\ \mu s$ and propagates to the inner surface of the tube in 2.8 $\mu s$. The stress drop in each element is quite rapid and resembles that observed by Marchand and Duffy (1988) in a thin-walled tube. For $\kappa = 800$ K, the shear band initiates at $t = 24.2\ \mu s$ and the stress drop is less catastrophic than that for $\kappa = 500$ K. Thus, the value of $\kappa$ in the thermal softening function slightly influences when a shear band initiates in an element, but strongly affects the subsequent rate of drop of the effective stress there. The time history plots of the temperature verify the assertion that once $\sigma_y = 0$ in an element, its temperature does not change. This is because there is no more plastic working for that element and the deformations are presumed to be locally adiabatic.

The variation of the band speed with the radial distance from the notch tip is shown in Fig. 23, for two different initiation criteria; as expected the results depend upon the definition of the initiation of a shear band. If a shear band initiates at a material point when the effective plastic strain there equals 0.50, then the band speed varies from 750 m/s (1000 m/s) to 450 m/s (700 m/s) for $\kappa = 500$ K (800 K). However, if a shear band is taken to initiate at a point when the effective stress there has dropped to 90% of its maximum value, then the band speed increases from 300 m/s (300 m/s) to 1500 m/s (900 m/s) for $\kappa = 500$ K (800 K) as it propagates from the notch tip to the inner surface. These values are in the same range as those observed by Zhou et al. , but the maximum temperatures computed herein are lower than those reported by them. Their plots of the time history of the speed of a shear band lend credence to the definition of a shear band as a contour of effective plastic strain of say 0.5. However, Marchand and Duffy (1988) suggest that a shear band initiates when the shear stress drops catastrophically.

Figure 24 depicts the external work done as a function of the radial distance, $s$, through which a shear band (a contour of effective plastic strain of 0.5) has propagated. As expected, less work is needed to drive the shear band for $\kappa = 500$ K as compared to that for $\kappa = 800$ K. The slopes of these curves for 1.0 mm $\leq s \leq$ 2.6 mm are 45.2 and 60.7 J/mm for $\kappa = 500$ K and $\kappa = 800$ K respectively; these equal the energy required to drive a shear band radially inwards through 1 mm on the cross-section through the notch tip. Since the shear band width equals 0.119 mm, the average energy required to shear band the material equals 23.4 and 34.9 J/mm$^3$ for the two values of $\kappa$.

We have plotted in Fig. 25 the time history of the torque required to deform the tube. Once a shear band has developed at the cross-section through the notch tip, the effective stress there drops sharply and an elastic unloading wave which emanates results in the oscillations in the applied torque. The amplitude of this unloading wave is related to the drop in the effective stress and is less for the larger value of $\kappa$ because of decreased thermal softening of the material. The period of these oscillations approximately equals the time taken for a shear wave to traverse through twice the length of the tube. We recall that one end of the tube is kept stationary and the other is twisted with a prescribed angular speed. The negative torque means that it acts in a direction opposite to that of twisting. Of course this is not possible for linear elastic materials, but here an elasto-thermo-visco-plastic material is undergoing large deformations. Batra and Kim (1990) seem to be the first to compute the elastic unloading wave in their study of the development of shear bands in a steel block undergoing simple shearing deformations; their computed shear stress on the shearing plane opposed the shearing direction. The oscillations in the torque will not occur for large values of $\kappa$ since the stress will drop gradually within the shear band. The amplitude of the oscillations decreases because of the energy dissipated due to plastic deformations of the tube.

The distribution at $t = 10.5\ \mu s$ and $30.5\ \mu s$ of different components of stress on a radial line through the notch tip for $\kappa = 800$ K is exhibited in Figs. 26a and 26b respectively. At $t = 10.5\ \mu s$, the deformations of the tube are essentially elastic and at $t = 30.5\ \mu s$ a shear band, as indicated by the accumulation of plastic strain, has initiated and propagated to the inner surface of the tube. Whereas at $t = 10.5\ \mu s$, $\sigma_{z\theta}$ (z-axis is along the tube's axis) is maximum at the notch tip, it is minimum there at $t = 30.5\ \mu s$; this is due to the softening of the material caused by heating from the intense plastic deformations. These plots suggest that the effective stress at the point on the inner surface of the tube that is radially below the notch tip has not dropped noticeably even though the effective plastic strain there is at least 0.5.

# 4 Conclusions

We have studied dynamic thermomechanical finite deformations of a thick-walled 4340 steel tube subjected to torsion, combined torsional and axial loading, and combined torsional and internal pressure loading. The tube material is modeled as elastic-thermo-viscoplastic with the flow stress depending upon the effective plastic strain, effective plastic strain-rate and temperature. The tube has a V-notch at the midsection which acts as a stress concentrator. The loading pulse is of finite duration with a rise and fall-off time of 20 $\mu$s each and stays steady for some time in between; thus a finite amount of energy is input into the body.

Under pure torsional loading a shear band, identified by a drop in the effective stress to 90% of its peak value or the effective plastic strain reaching a preassigned value, initiates first in an element below the notch tip and propagates radially inwards. Its speed of propagation varies from 50 m/s at the time of initiation to a maximum of 130 m/s as it reaches the innermost surface of the tube. The temperature rise at the initiation of a shear band is nearly 130°C but the temperature reaches approximately 95% of the melting temperature of the material 70 $\mu$s later. When the loaded end face of the tube has been brought to rest at $t = 110\ \mu s$, sharp gradients in the temperature and effective plastic strain develop through the shear-banded region. The energy required to shear band the material equals 126.2 J/mm$^3$.

During elastic deformations of the tube under pure torsional loading, only the $\sigma_{z\theta}$ component of the Cauchy stress exhibits noticeable stress concentration at the notch tip and other components of the stress tensor have negligible values everywhere. However, when the cross-section through the notch tip has deformed plastically, $\sigma_{z\theta}$ at the notch tip has the lowest value. Other components, $\sigma_{\theta\theta}$, $\sigma_{zz}$ and $\sigma_{rr}$ take on maximum values at the notch tip which equal approximately one-fourth of the maximum magnitude of $\sigma_{z\theta}$. The time of initiation of a shear band decreases exponentially with the depth of the notch; however, the notch depth influences only the initial speed of a shear band. After the band has propagated radially inwards, its speed is essentially independent of the notch depth and equals 90 m/s.

For combined torsional and axial loading, the shear band initiates later and propagates slower than that for pure torsional loading. When the tube is loaded by a constant internal pressure followed by torsional loading, the shear band propagates into a plastically deformed region. For this case as well, the band speed is found to be lower than that for the pure torsional case. Also, for these two cases of combined loads, different definitions of the initiation of a shear band give contradictory results in the sense that according to one criterion, a shear band initiates at a point because the effective plastic strain there reaches the preassigned value of 0.5, but according to another criterion, does not initiate because the effective stress there does not drop to 90% of its peak value for that material point.

For torsional loading of a CR-300 steel thick-walled tube with the thermal softening modeled by the relation proposed by Zhou et al. (1996), the computed maximum shear band speed of 1000 m/s agrees with that observed by them in their impact experiments on pre-notched plates. Because of the presumed enhanced thermal softening, the effective stress drops catastrophically in the shear banded region, and an elastic unloading wave emanates from there. It results in oscillations in the torque required to deform the tube, and during some time intervals, the torque acts in a direction opposite to that of the prescribed angular speed. The energy required to drive the shear band radially inwards on the cross-section through the notch tip strongly depends upon the parameters characterizing the thermal softening of the material. For $\kappa = 500$ K in equation (16), the energy required to shear band the material equals 23.4 J/mm$^3$.

# References

Anand, L., Kim, K.H. and Shawki, T.G. (1987) Onset of shear localization in viscoplastic solids. *J. Mechs. Phys. Solids* **35**, 381-399.

Armstrong, R., Batra, R.C., Meyers, M.A. and Wright, T.W. (Guest Editors) (1994) Special issue on shear instabilities and viscoplasticity theories. *Mech. Mater.* **17**, 83-327.

Batra, R.C. and Adulla, C. (1995) Effect of prior quasistatic loading on the initiation and growth of dynamic adiabatic shear bands. *Arch. Mechs.* **47**, 485-498.

Batra, R.C. and Kim, C.H. (1990) Adiabatic shear banding in elastic-viscoplastic nonpolar and dipolar materials. it Int. J. Platicity **6**, 127-141.

Batra, R.C. and Kim, C.H. (1991) Effect of thermal conductivity on the initiation, growth, and band width of adiabatic shear bands. *Int. J. Engng. Sci.* **29**, 949-960.

Batra, R.C. and Kim, C.H. (1992) Analysis of shear bands in twelve materials. *Int. J. Plasticity* **8**, 75-89.

Batra, R.C. and Zbib, H.M. (1994) Material instabilities: theory and applications. ASME Press, New York.

Batra, R.C. and Zhang, X. (1994) On the propagation of a shear band in a steel tube. *J. Eng'g Materials & Tech.* **116**, 155-161.

Batra, R.C., Adulla, C. and Wright, T.W. (1996) Effect of defect shape and size on the initiation of adiabatic shear bands. *Acta Mechanica* **116**, 239-243.

Batra, R.C. and Ko, K.J. (1992) An adaptive mesh refinement technique for the analysis of shear bands in plane strain compression of a thermoviscoplastic block. *Comp. Mechs.* **10**, 369-379.

Batra, R.C. and Jin, X.S. (1994) Analysis of dynamic shear bands in porous thermally softening viscoplastic materials. *Arch. Mechs.* **41**, 13-36.

Chi, Y.C. (1990) Measurement of the local strain and temperature during the formation of adiabatic shear bands in steel. Ph.D. Thesis, Brown University.

Clifton, R.J. (1980) Adiabatic shear in material response to ultrahigh loading rates. U.S. NRC Material Advisory Board Report NMAB-356, W. Herrman et al., eds.

Deltort, B. (1994) Experimental and numerical aspects of adiabatic shear in a 4340 steel. *J. de Physique*, Colloque C8 **4**, 447-452.

Dowling, N.E. (1993) Mechanical Behavior of Materials: engineering methods for deformation, fracture and fatigue. Prentice-Hall, Englewood Cliffs, NJ.

Farren, W.S. and Taylor, G.I. (1925) The heat developed during plastic extrusion of metal. *Proc. R. Soc.* **A207**, 422-426.

Johnson, G.R. and Cook, W.H. (1983) A constitutive model and data for metals subjected to large strains, high strain rates and high temperatures. *Proc. 7th Int. Symp. Ballistics*, The Hague, The Netherlands, 541-548.

Klepaczko, J.R., Lipinski, P. and Molinari, A. (1987) An analysis of the thermoplastic catastrophic shear in some metals. Impact Loading and Dynamic Behavior of Materials, C.Y. Chiem, H.-D. Kunze, and L.W. Meyer, Eds., Informationsgesellschaft, Verlag, Bremen, 695-704.

Marchand, A. and Duffy, J. (1988) An experimental study of the formation process of adiabatic shear bands in a structural steel. *J. Mech. Phys. Solids* **36**, 251-283.

Molinari, A. and Clifton, R.J. (1987) Analytical characterization of shear localization in thermoviscoplastic materials. *J. Appl. Mech.* **54**, 806-812.

Murphy, B.P. (1990) Shear band formation in a structural steel under a combined state of stress. M.S. Thesis, Brown University.

Peng, Z. and Batra, R.C. (1995) Propagation of shear bands in thick-walled RHA steel tubes. *Computational Mech* '95, S.N. Atluri, Y. Yagawa and T.A. Cruse, Eds., Springer-Verlag, 2034-2039.

Rajendran, A.M. (1992) High strain rate behavior of metals, ceramics and concrete. Report #WL-TR-92-4006, Wright Patterson Air Force Base.

Sulijoadikusumo, A.V. and Dillon, Jr., O.W. (1979) Temperature distribution for steady axisymmetric extrusion with an application to Ti-6Al-4V, part 1. *J. Thermal Stresses* **2**, 97-112.

Tresca, H. (1878) On further application of the flow of solids. *Proc. Inst. Mech. Engr.* **30**, 301-345.

Truesdell, C.A. and Noll, W. (1965) The nonlinear field theories of mechanics. Handbuch der Physik, Vol. III/3, (S. Flügge, Ed.), Springer, Berlin.

Whirley, R.G. and Hallquist, J.P. (1991) DYNA3D User's Manual, a nonlinear, explicit, three-dimensional finite element code for solid and structural mechanics. User Manual, Lawrence Livermore National Laboratory Report, UCRL-MA-107254.

Wright, T.W. and Walter, J.W. (1987) On stress collapse in adiabatic shear bands. *J. Mech. Phys. Solids* **35**, 701-716.

Wright, T.W. (1994) Toward a defect invariant basis for susceptibility to adiabatic shear bands. *Mechs. Materials* **17**, 215-222.

Zbib, H.M., Shawki, T. and Batra, R.C. (Eds.) (1992) Material instabilities. Special Issue of *Applied Mechanics Reviews* **45**, No. 3.

Zener, C. and Hollomon, J.H. (1944) Effect of strain rate on plastic flow of steel. *J. Appl. Phys.* **14**, 22-32.

Zhou, M., Rosakis, A.J. and Ravichandran, G. (1996) Dynamically propagating shear bands in impact-loaded prenotched plates II - numerical simulations, *J. Mechs. Phys. Solids* **44**, 1007-1032.

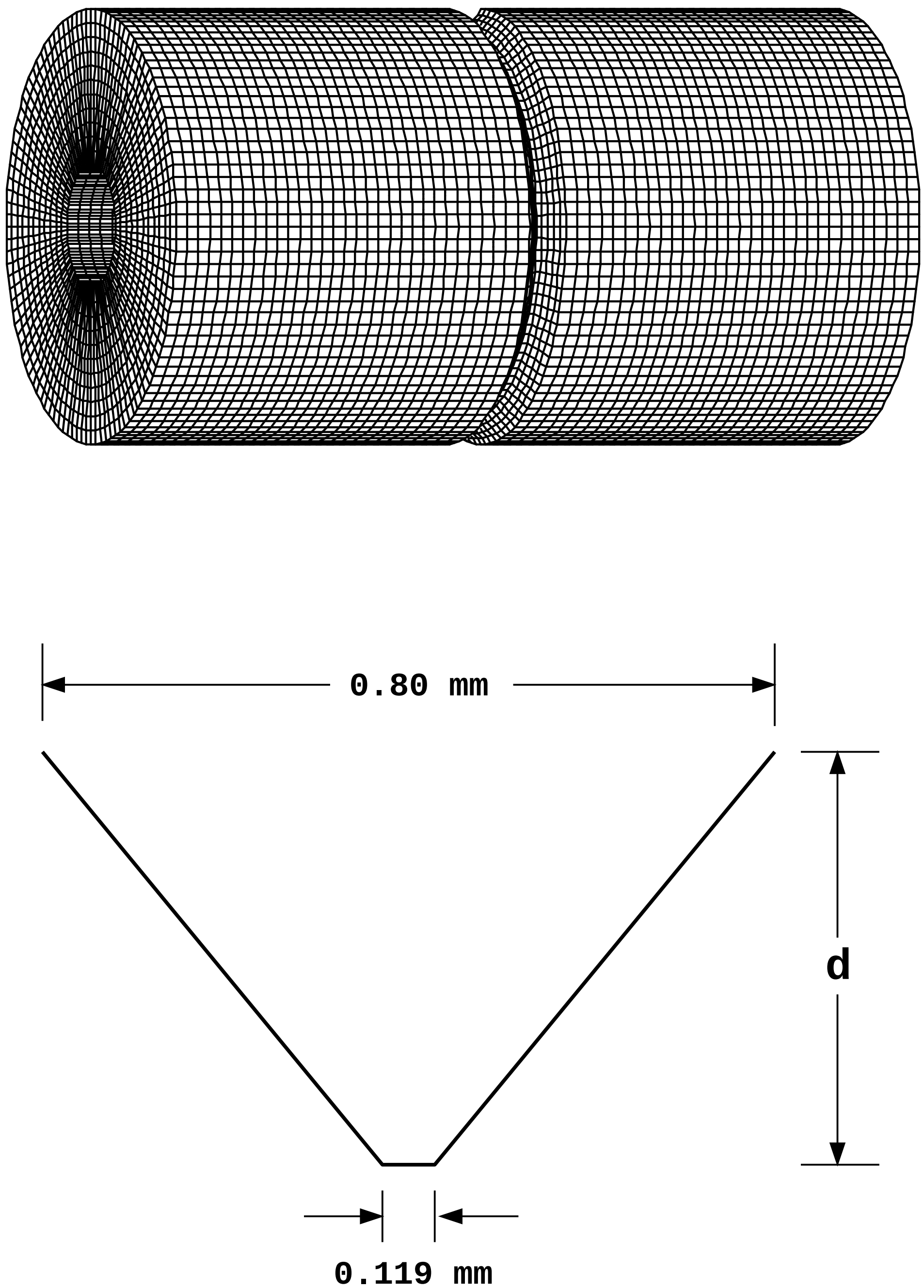


Figure 1: (a) Finite element discretization of the tube, and (b) Details of the notch.

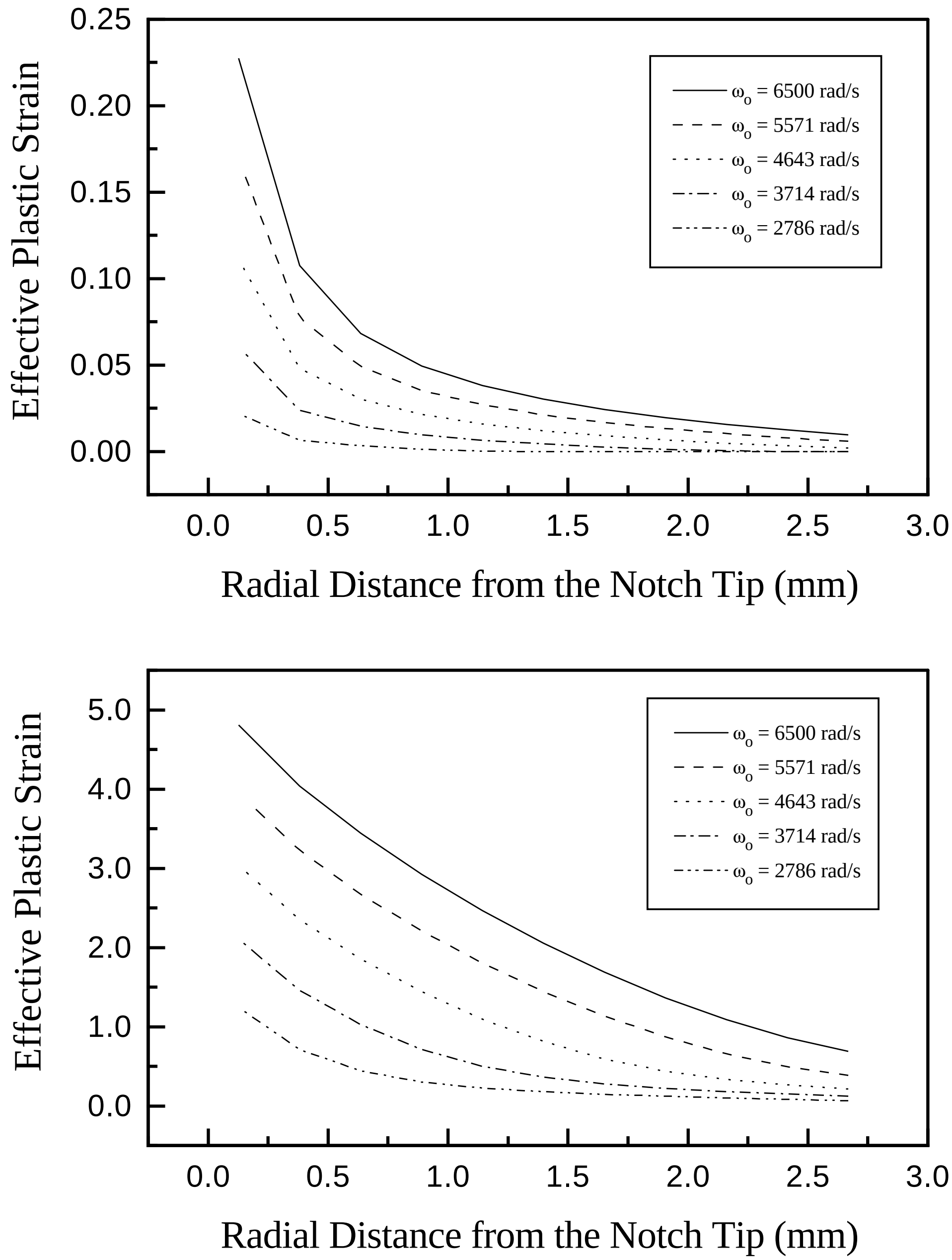


Figure 2: Distribution of the effective plastic strain on a radial line for notch 4 with different values of the maximum applied angular speed; (a) $t = 20$ $\mu$s, (b) $t = 70$ $\mu$s.

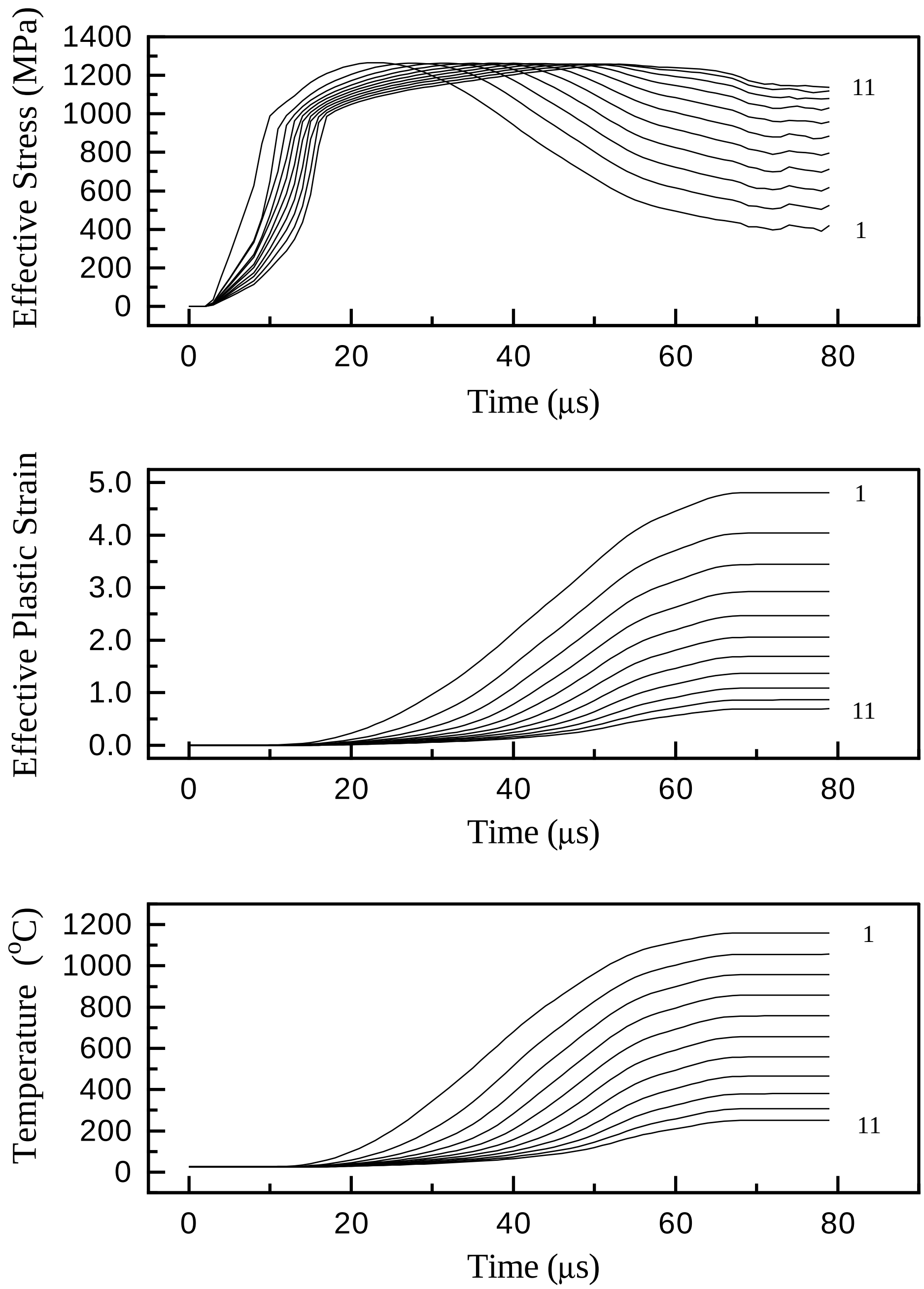


Figure 3: Time history of (a) the effective stress, (b) the effective plastic strain, and (c) the temperature at the centroids of eleven elements on a radial line through the notch tip, for notch 4 with $\omega_o = 6500$ rad/s.

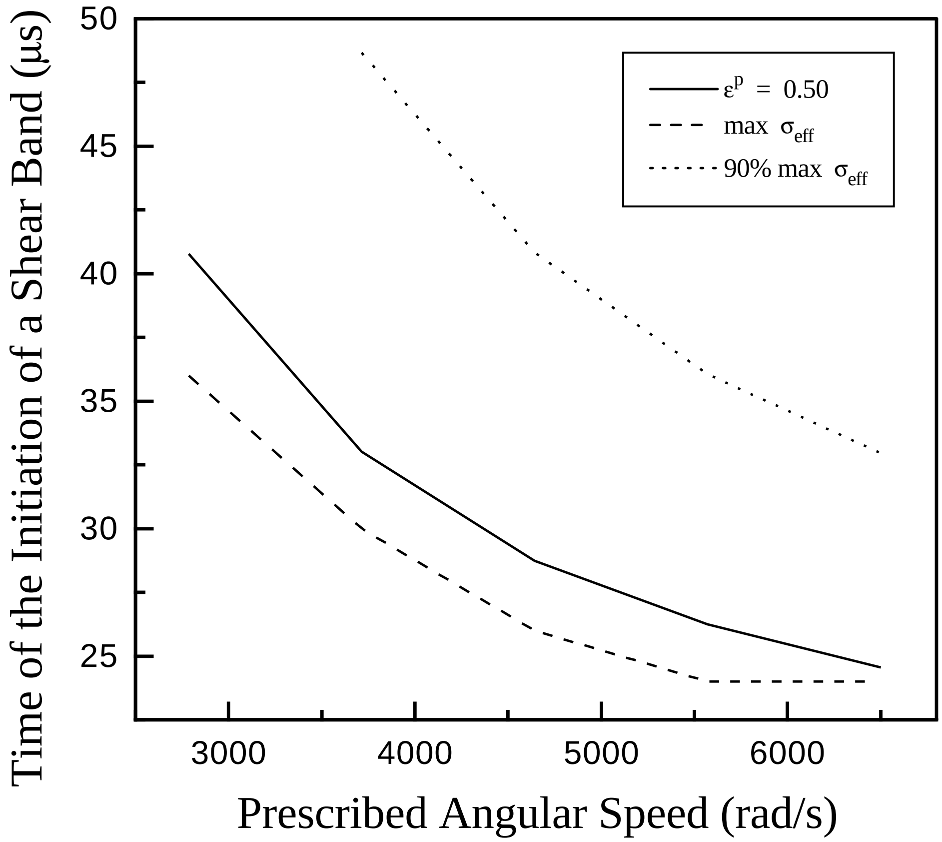


Figure 4: Dependence of the initiation of a shear band on the prescribed angular speed, calculated by three different criteria, for notch 4 and a load duration of 70 $\mu$s.

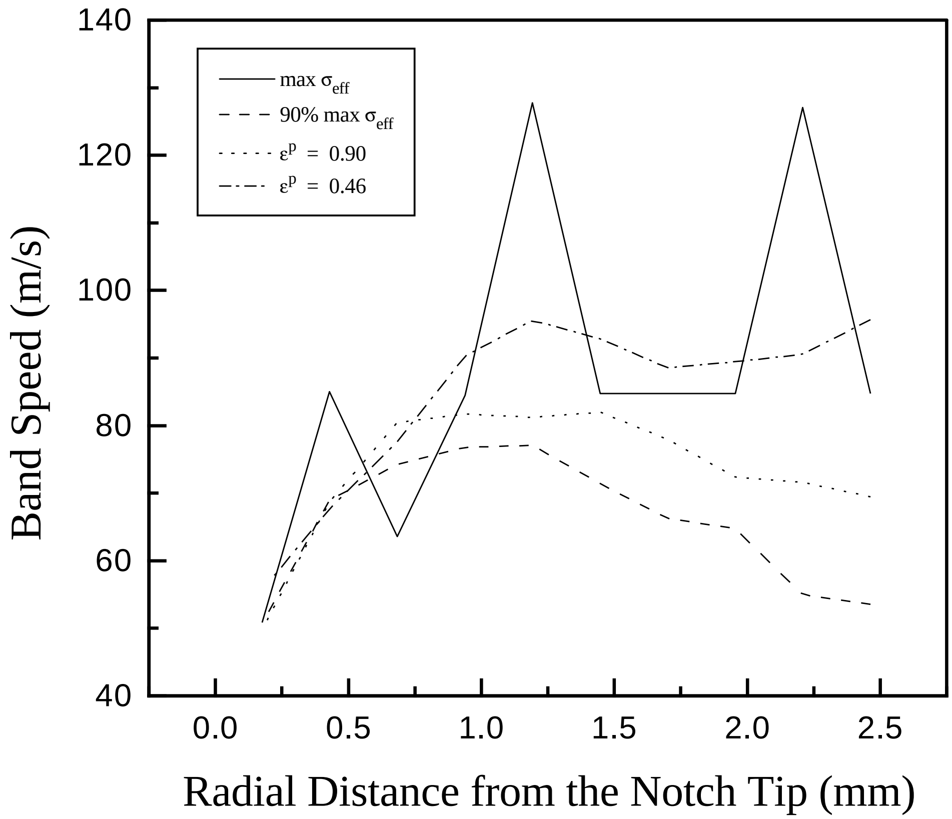


Figure 5: Band speed according to different initiation criteria vs. the radial distance from the notch tip; The band initiates when (a) the effective stress attains its maximum value (b) the effective stress has dropped to 90% of its maximum value; (c) and (d) the effective plastic strain equals 0.46 and 0.90 respectively. Results are for notch 4 with $\omega_o = 6500$ rad/s.

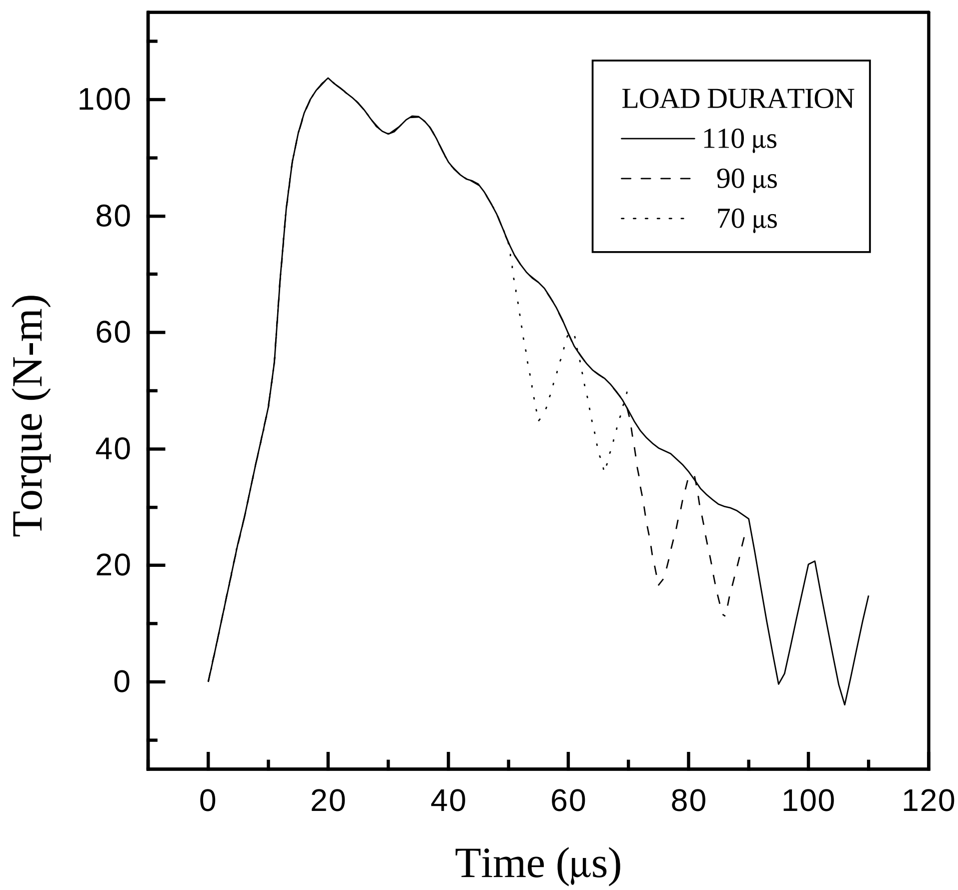


Figure 6: Time history of the torque required to deform the tube with notch 4 for three different durations of the applied angular speed, with $\omega_o = 6500$ rad/s.

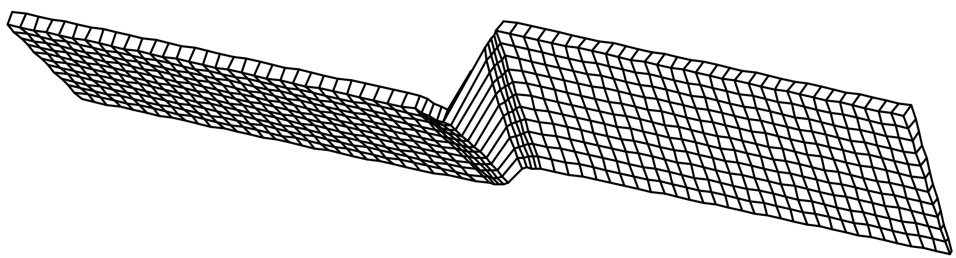

Figure 7: Deformed configuration of a plane passing through the tube's axis at $t =$ 100 $\mu$s for notch 4 with $\omega_o = 6500$ rad/s and a load duration of 110 $\mu$s.

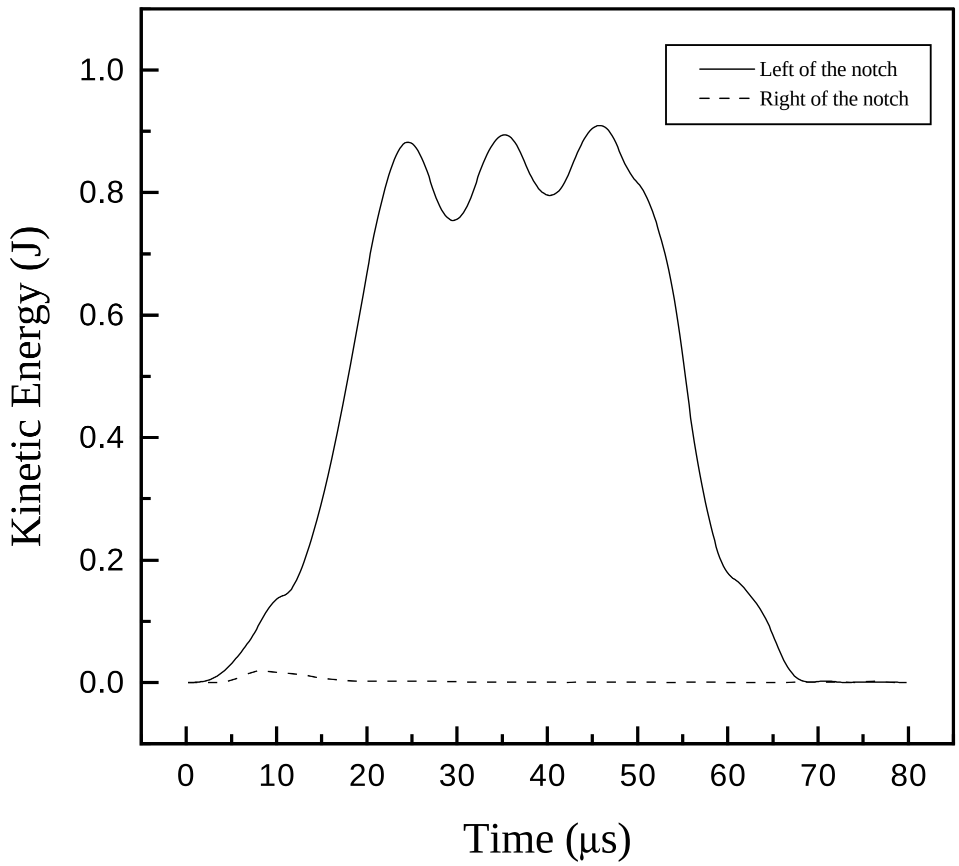


Figure 8: Time history of the kinetic energy of the material to the left and right of the notch for notch 5 with $\omega_o = 6500$ rad/s.

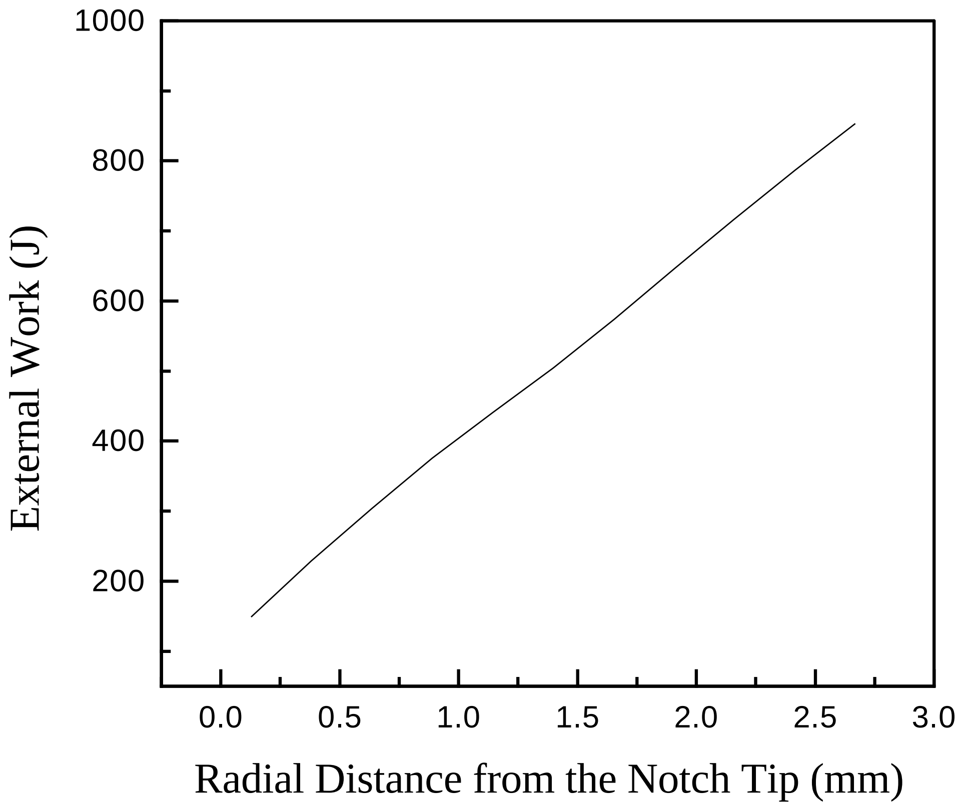


Figure 9: External work done vs. the radial distance through which a shear band (a contour of effective plastic strain of 0.50) has propagated for notch 4 with $\omega_o = 6500$ rad/s.

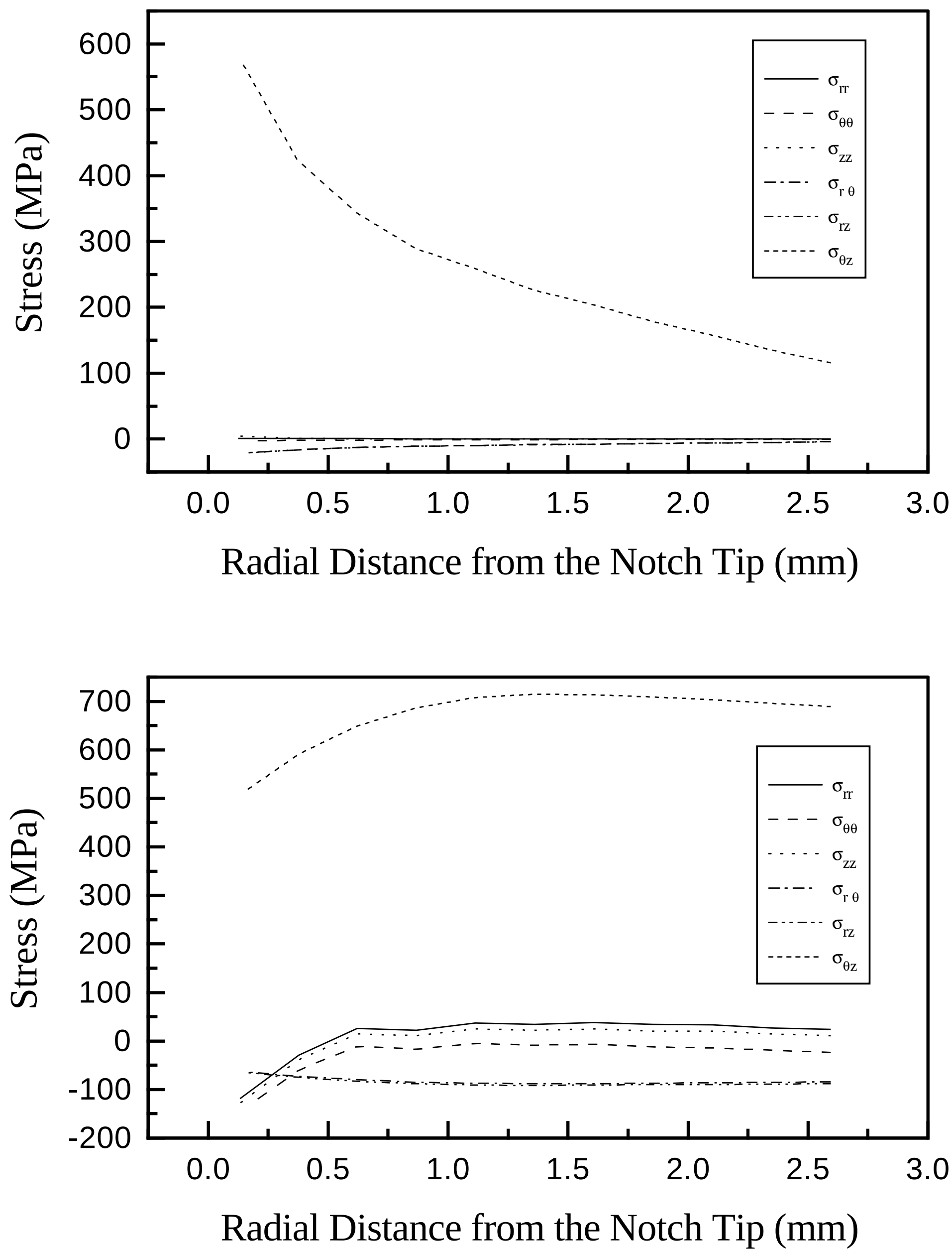


Figure 10: Stress distribution on a radial line for notch 5 with $\omega_o = 6500$ rad/s at (a) $t = 10$ $\mu$s, and (b) $t = 40$ $\mu$s.

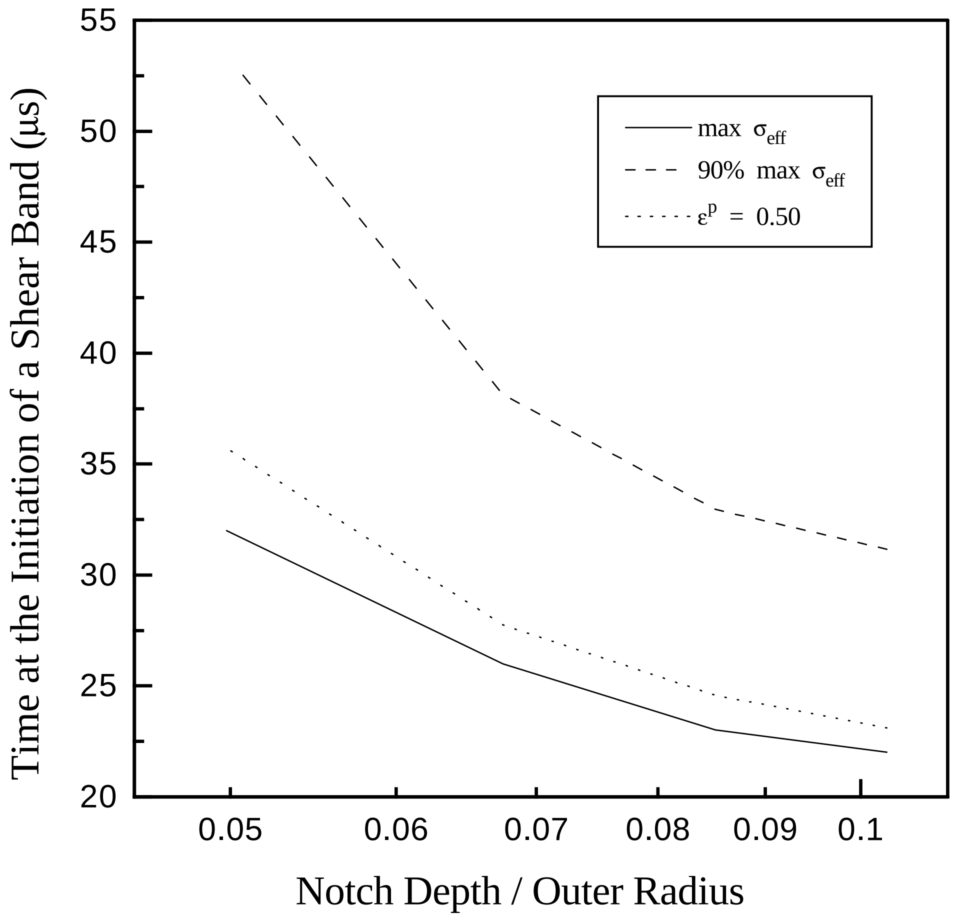


Figure 11: Dependence of the time of initiation of a shear band upon the defect size as computed by three different criteria, for the case of $\omega_o = 6500$ rad/s.

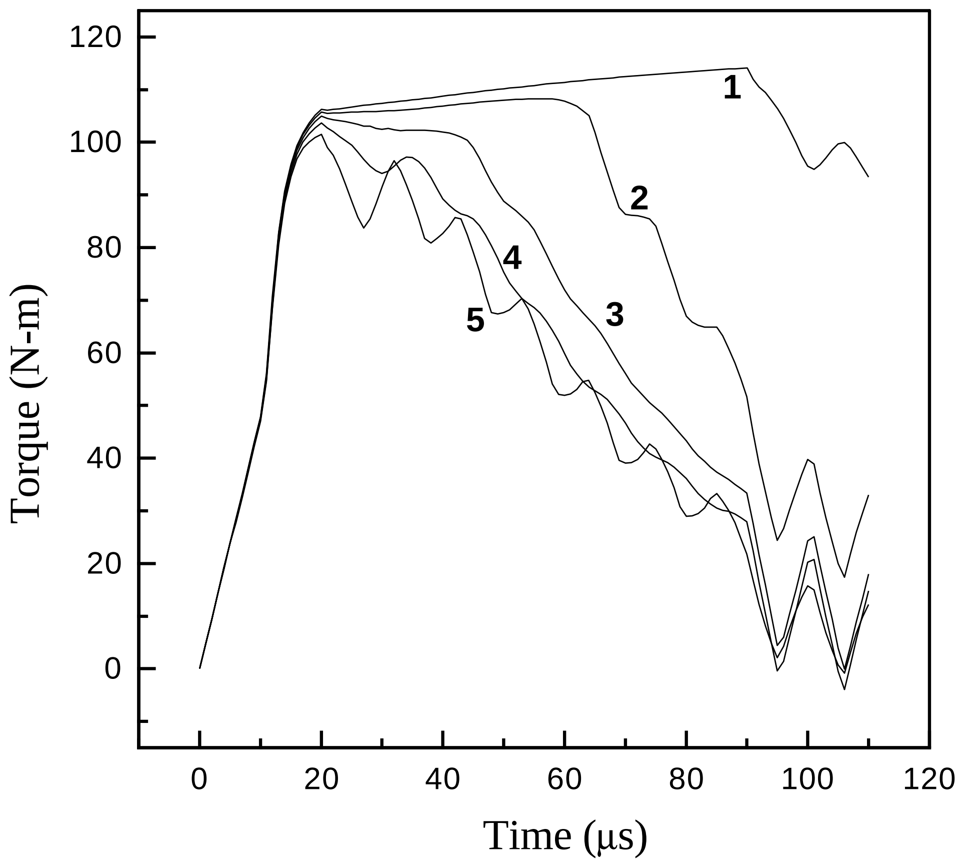


Figure 12: Time history of the torque required to deform the tube for five different notches with $\omega_o = 6500$ rad/s.

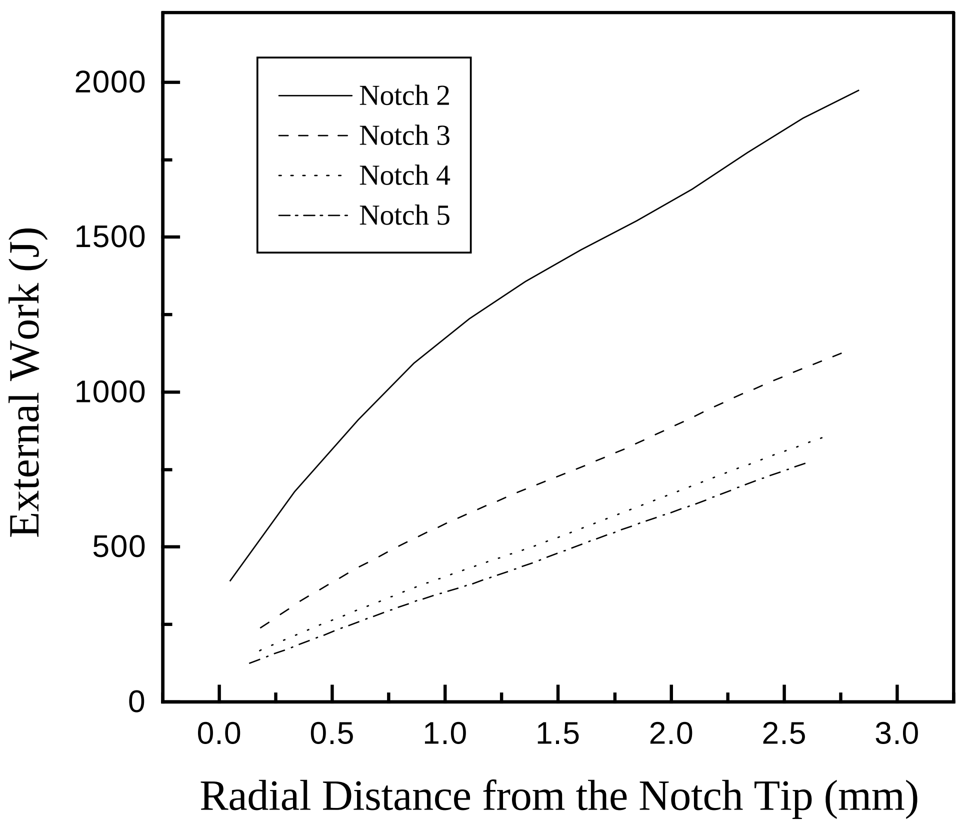


Figure 13: External work done vs. the radial distance through which a shear band (a contour of effective plastic strain of 0.5) has propagated for notches 2, 3, 4 and 5 with $\omega_o = 6500$ rad/s.

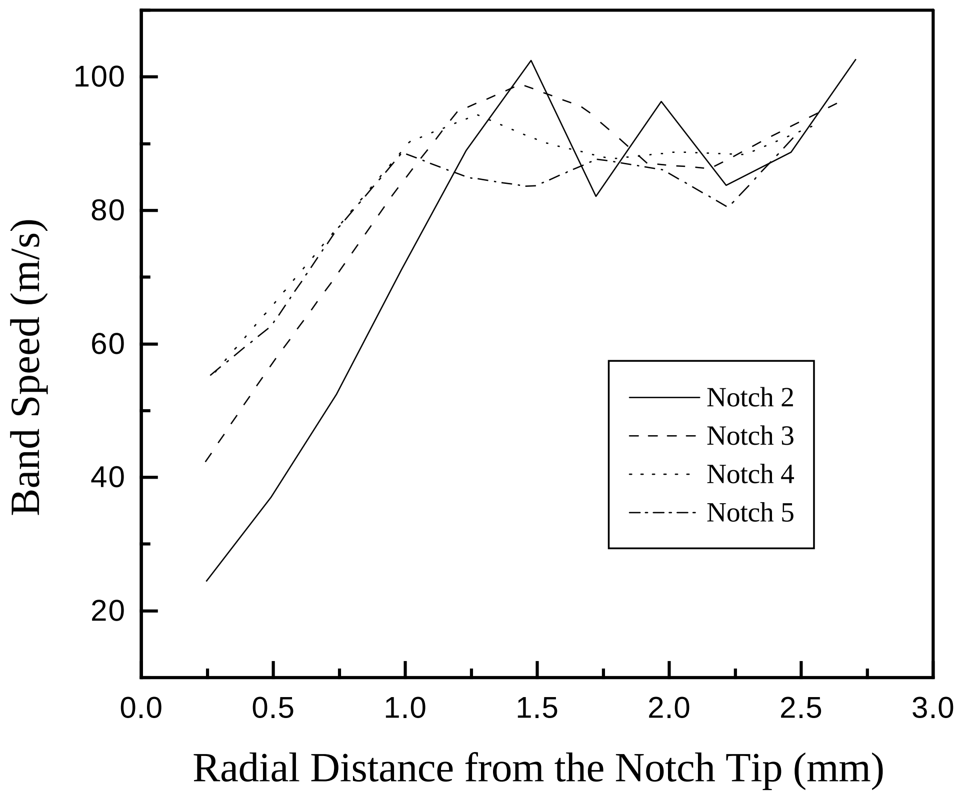


Figure 14: Band speed vs. the radial distance through which a shear band (a contour of effective plastic strain of 0.5) has propagated for notches 2, 3, 4 and 5 with $\omega_o = 6500$ rad/s.

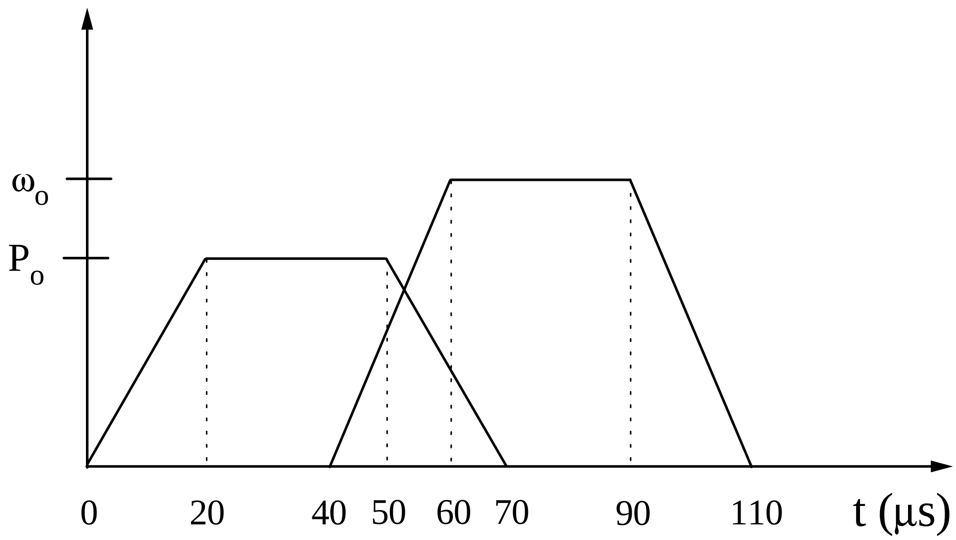


Figure 15: Loading history for combined pressure and torsional loading.

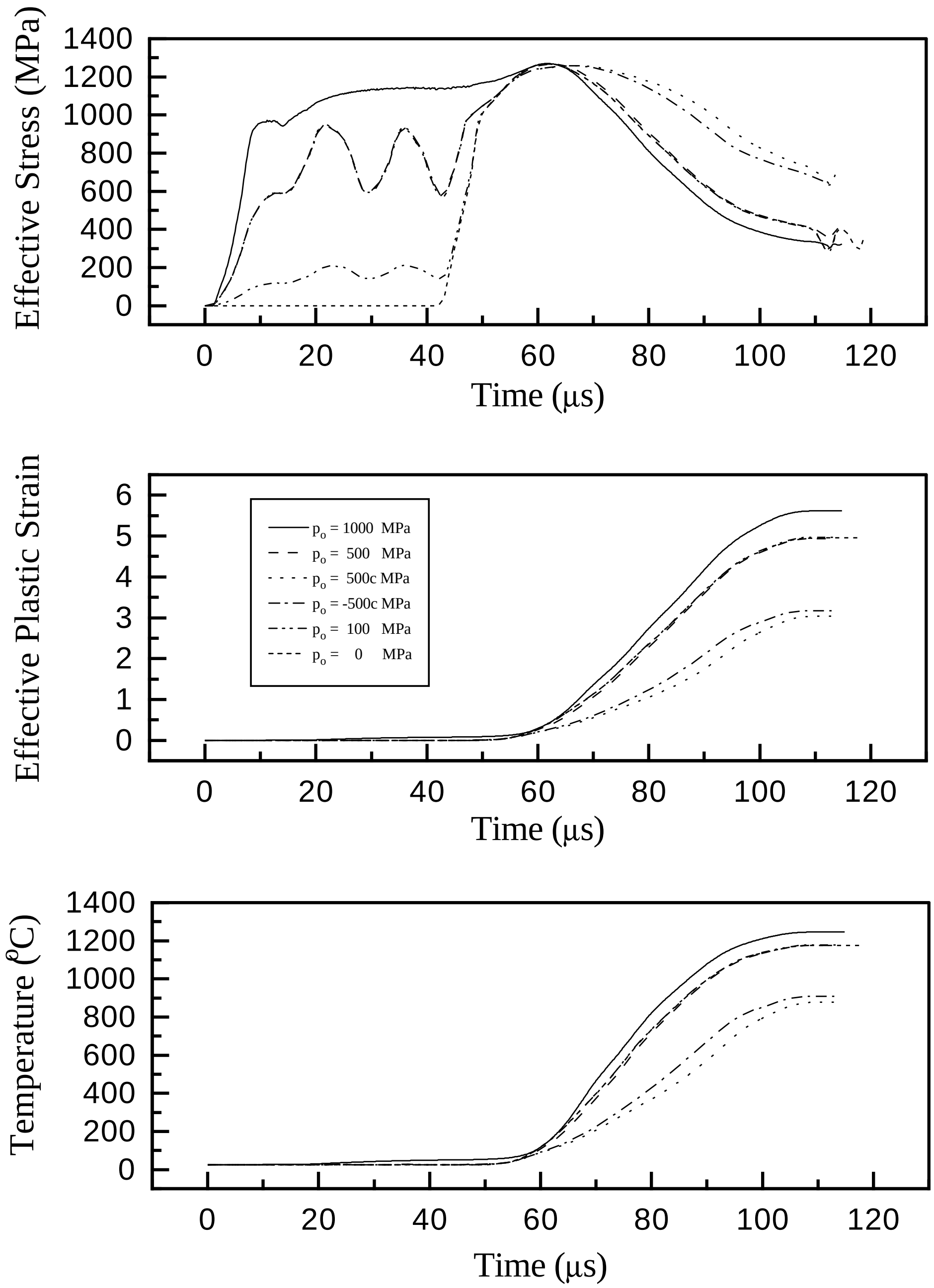


Figure 16: Time history of (a) the effective stress, (b) the effective plastic strain and (c) the temperature at the centroid of the element below the notch tip for notch 5 with combined axial and torsional loading.

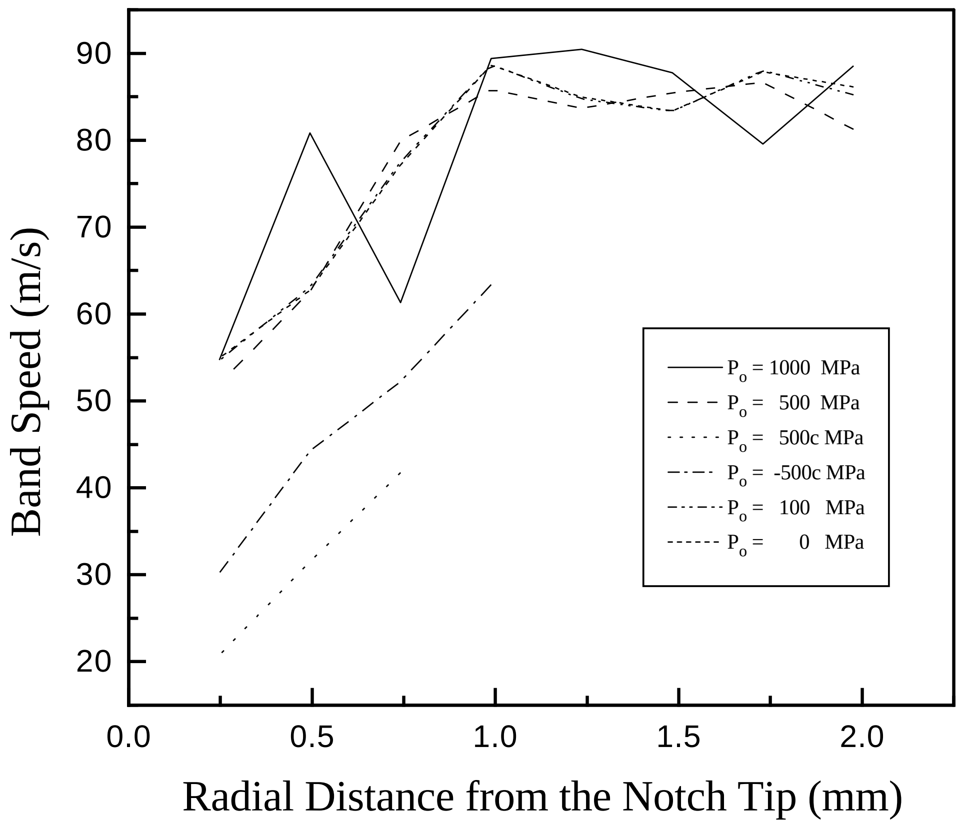


Figure 17: Band speed in the radial direction for different axial loads vs. the radial distance through which a shear band (a contour of effective plastic strain of 0.5) has propagated for notch 5 with combined axial and torsional loading.

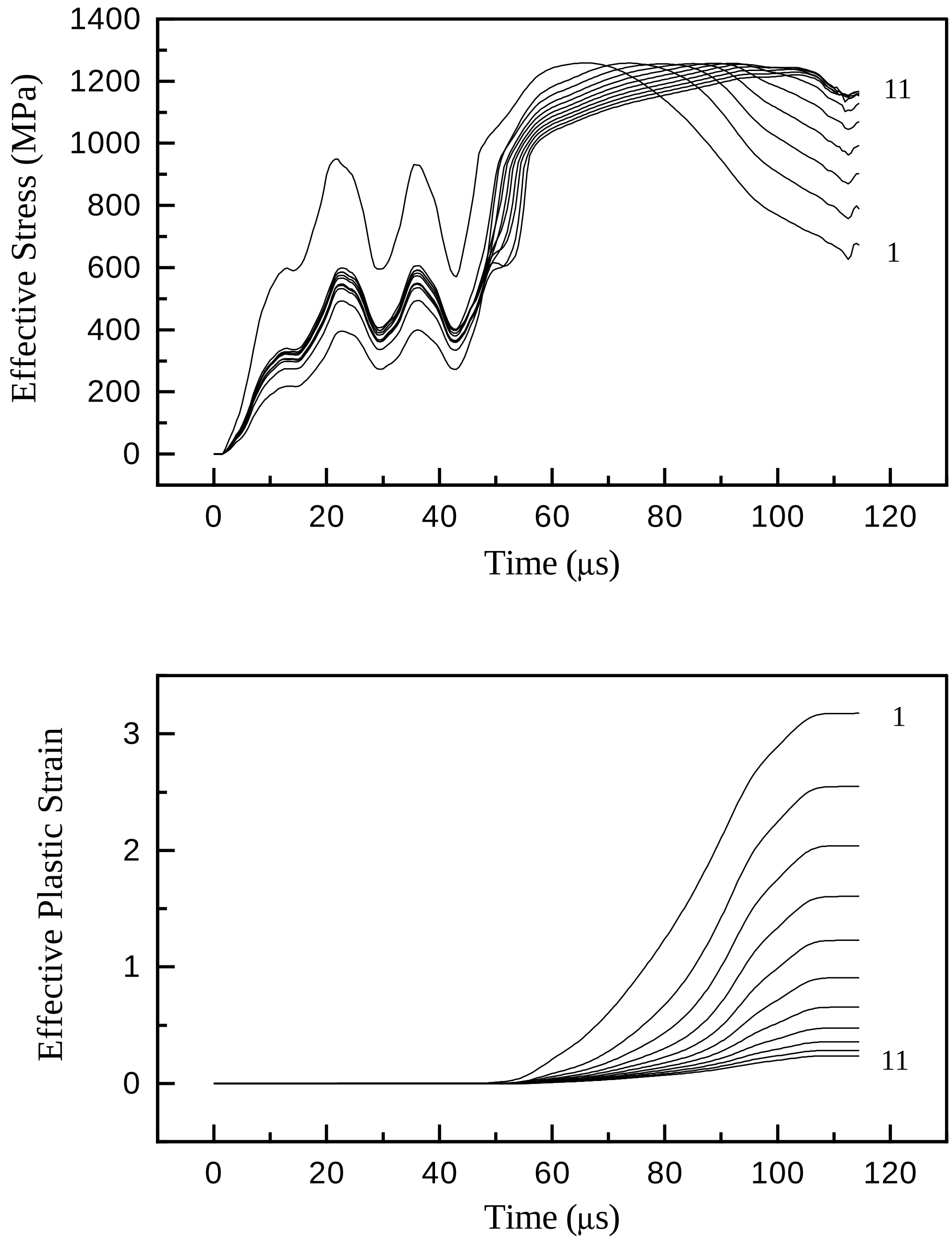


Figure 18: Time history of (a) the effective stress and (b) the effective plastic strain at the centroids of eleven elements on a radial line through the notch tip for notch 5 for the case of combined tensile axial and torsional loading with $P_o = 500$ MPa.

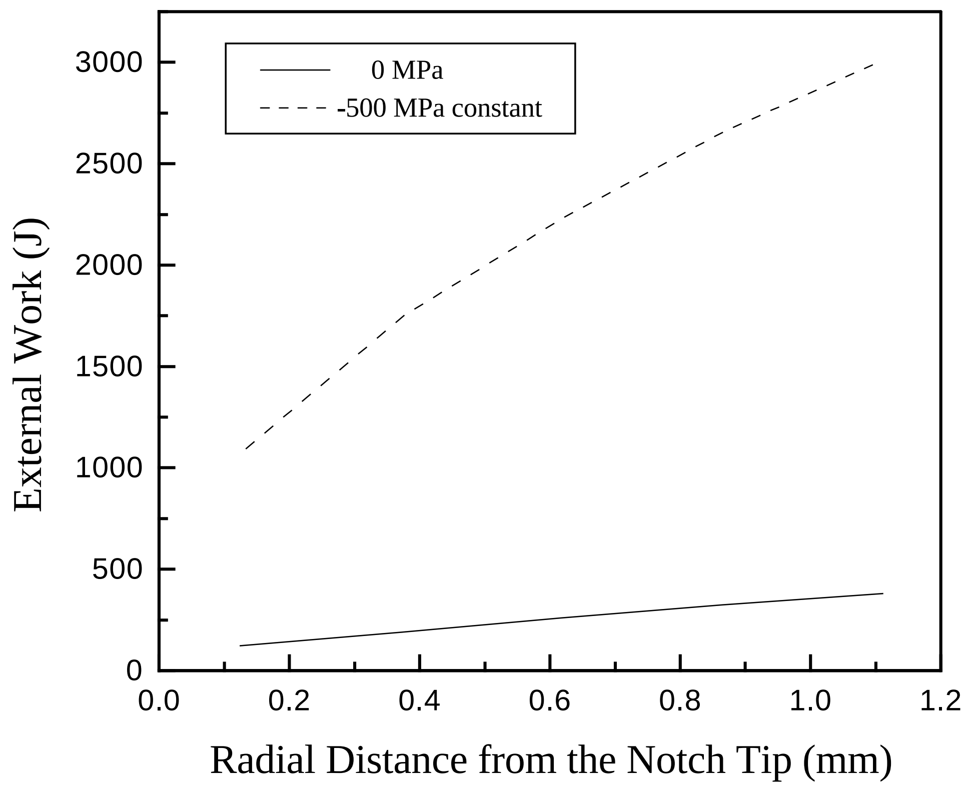


Figure 19: External work done vs. the radial distance through which a shear band has propagated for notch 5 for the cases of pure torsional loading, and constant tensile axial loading combined with torsional loading.

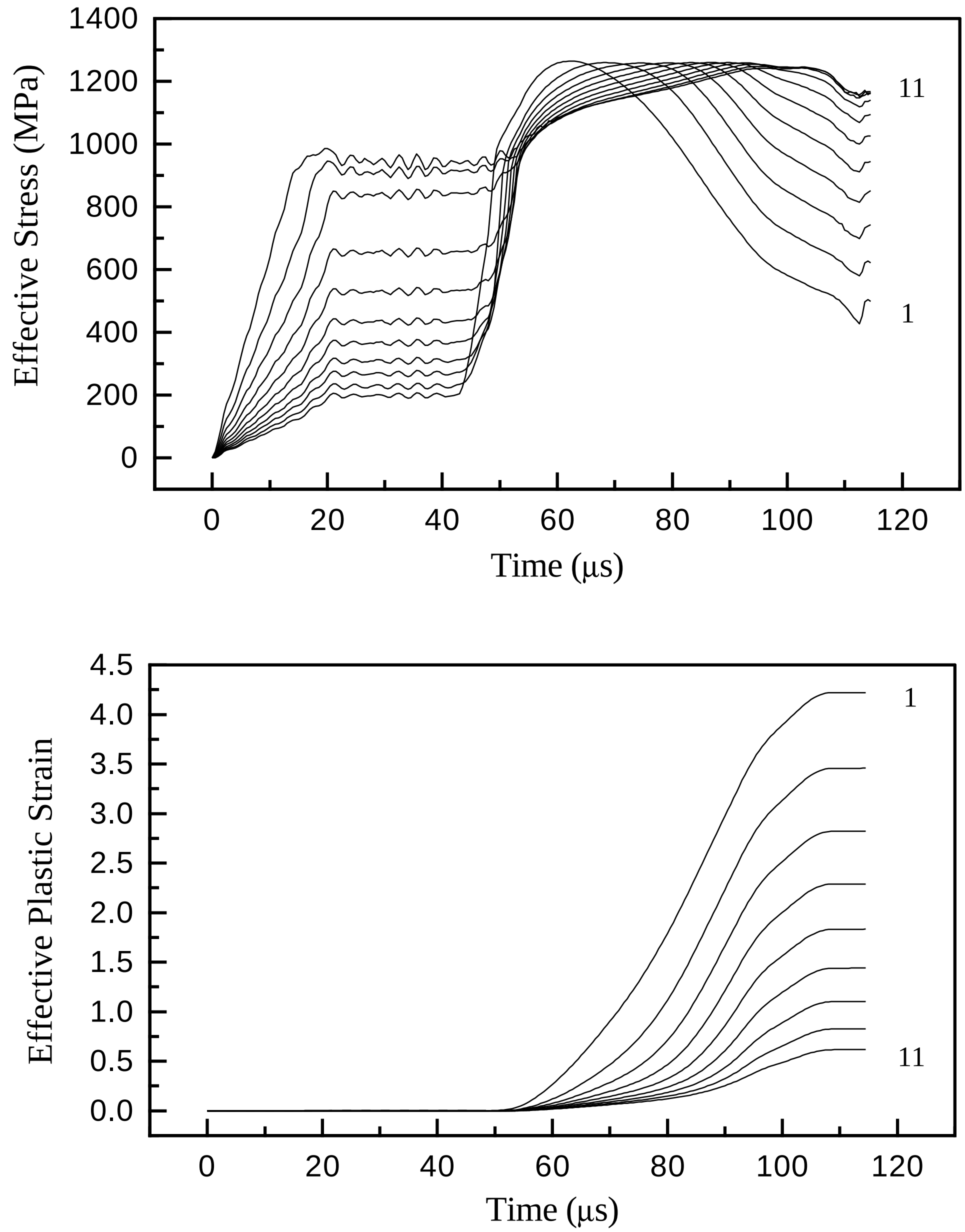


Figure 20: Time history of (a) the effective stress and (b) the effective plastic strain at the centroids of eleven elements on a radial line through the notch tip for notch 5 with a constant internal pressure of 800 MPa and a torsional pulse starting at $t = 40\ \mu s$.

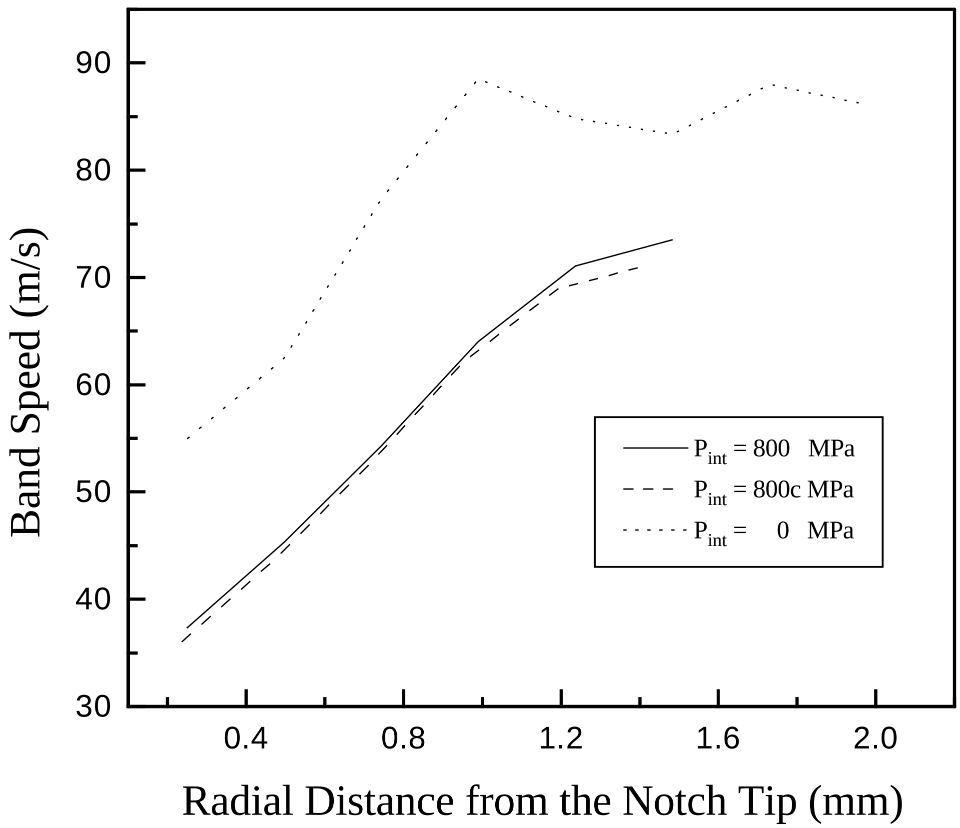


Figure 21: Variation of shear band (a contour of effective plastic strain of 0.5) speed with the radial distance from the notch tip for notch 5 with an applied internal pressure of 800 MPa followed by torsional loading.

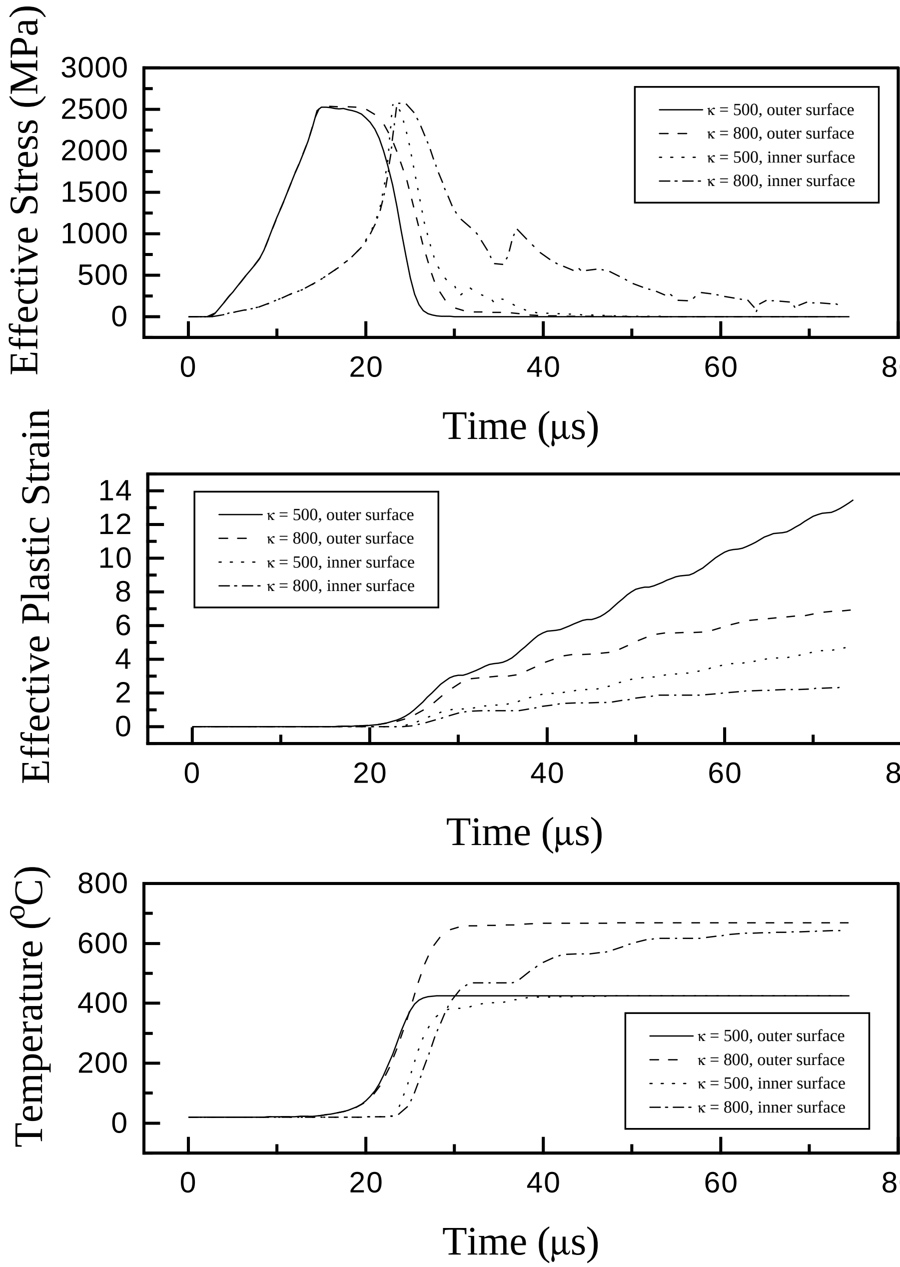


Figure 22: Time history of (a) the effective stress, (b) the effective plastic strain and (c) the temperature at two elements - one just below the notch tip and the other adjoining the inner surface of a CR-300 steel thick-walled tube with notch 5 deformed in torsion.

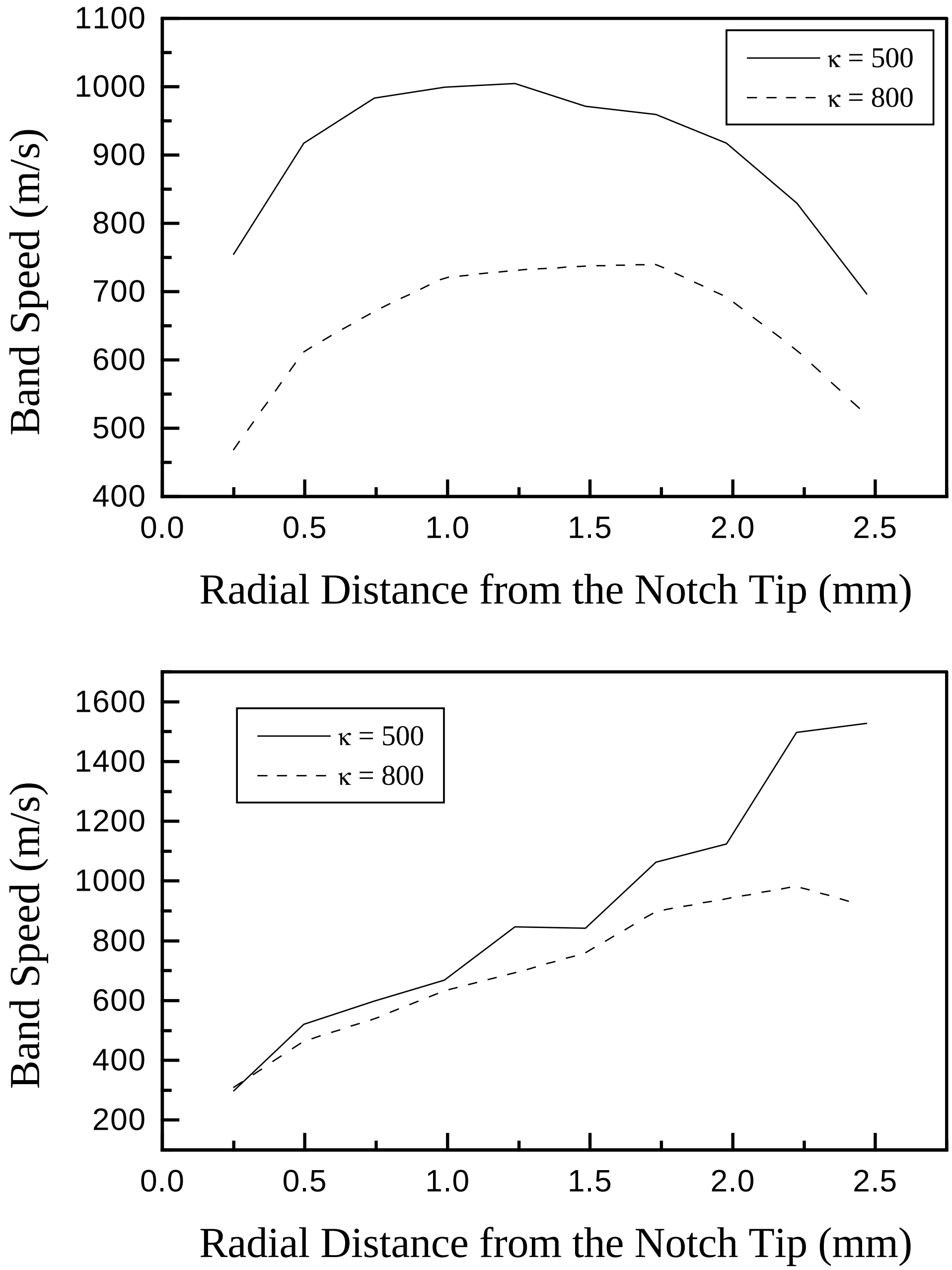


Figure 23: Band speed vs. the radial distance from the notch tip of a shear band in a CR-300 steel thick-walled tube with notch 5 deformed in torsion, as computed by the criteria (a) a contour of effective plastic strain of 0.50, and (b) a drop to 90% of the maximum effective stress.

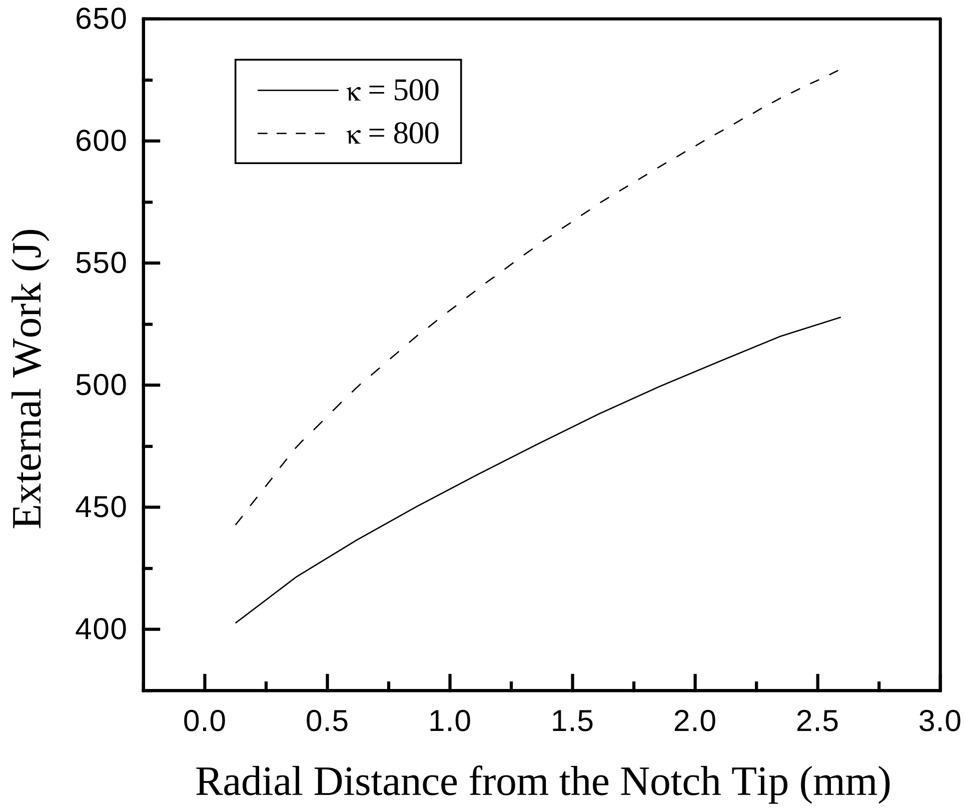


Figure 24: External work done vs. the radial distance through which a shear band (a contour of effective plastic strain of 0.5) has propagated for torsional deformations of the CR-300 steel thick-walled tube with notch 5.

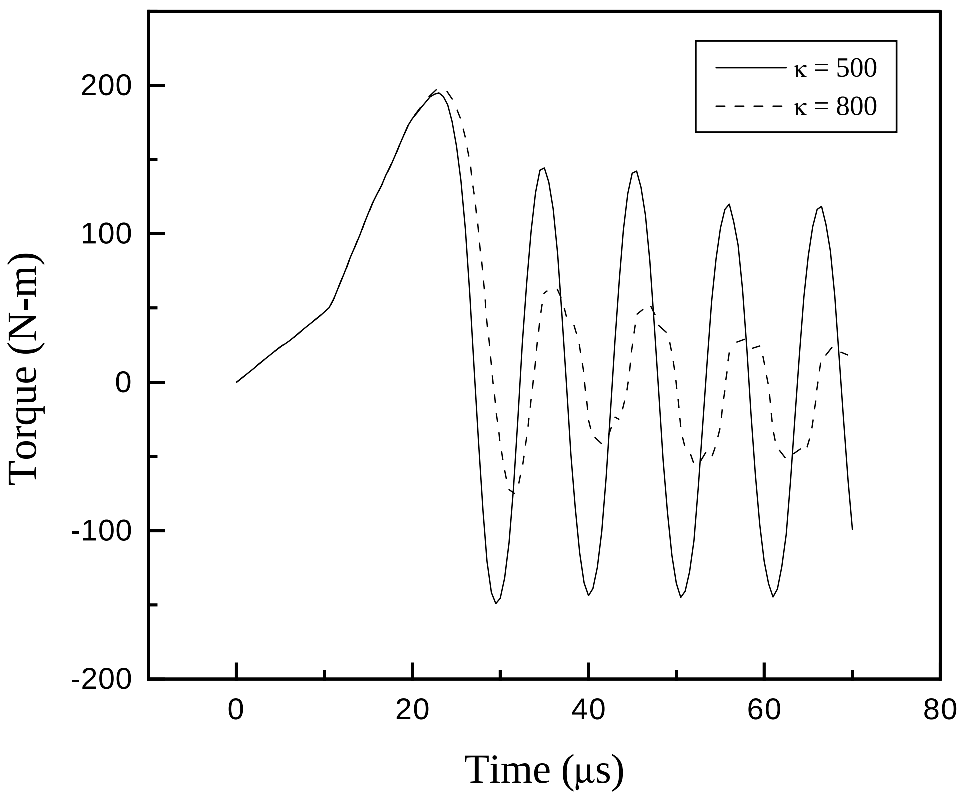


Figure 25: Time history of the torque required to deform the CR-300 steel thick-walled tube with notch 5.

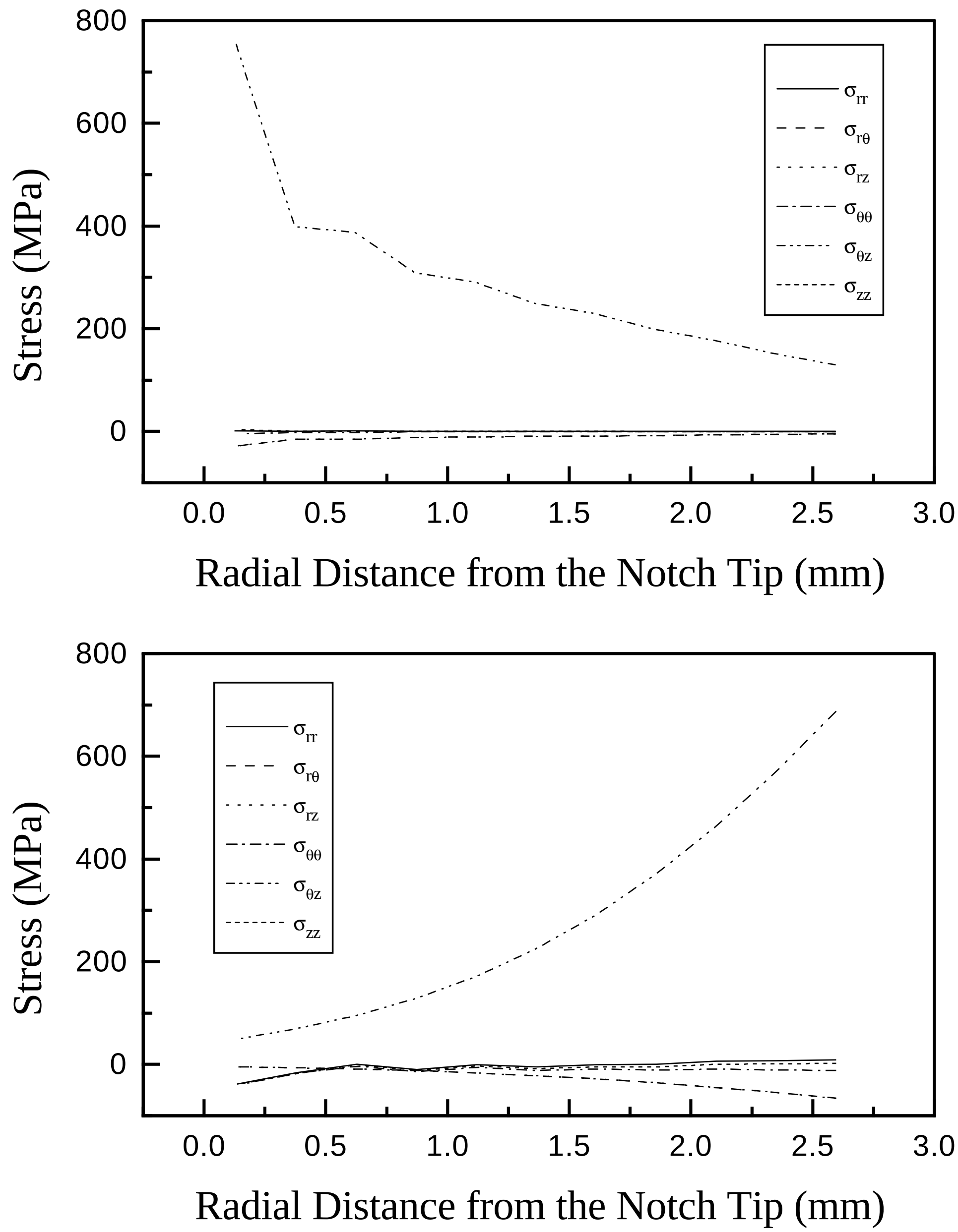


Figure 26: Distribution of the different components of stress on a radial line through the notch tip for torsional deformations of the CR-300 steel thick-walled tube with notch 5 for $\kappa = 800$ K, at (a) $t = 10.5$ $\mu s$ and (b) $t = 30.5$ $\mu s$

# A FORTRAN Converter Programs

## A.1 Program to Convert a PATRAN Neutral File to a DYNA3D Input File

```
c************************************************************
c**** This program converts a PATRAN neutral file into a DYNA3D
c**** input file.  It includes material models for 4340 Steel and
c**** alpha titanium.  There is a choice of axial loading, torsional
c**** loading, internal pressure, or a combination of the same loading
c**** conditions.
c**** version: patdyn3.f     Last modified 8/16/96
c**** written by: Dean Rattazzi
c************************************************************

      implicit double precision (a-h,o-z)

      character netfile*25, dynfile*25, title*72, check*2,ccard*80,
     & yn1flag*1,yn2flag*2,tbnfile*25,etfile*25,pstrain*1

      parameter (max=400000,nbig=25000)

      dimension nodnum(max),nelnum(max),ccard(64),npress1(nbig),
     & npress2(nbig),ndisp1(nbig),ndisp2(nbig),xcord(max),ycord(max),
     & zcord(max),node1(max),node2(max),node3(max),node4(max),
     & node5(max),node6(max),node7(max),node8(max),nthnumn(nbig),
     & nthnume(nbig),knblkn1(nbig),knblkn2(nbig),keblkn1(nbig),
     & keblkn2(nbig)

      write(*,*)
      write(*,*) 'This Program converts a PATRAN neutral file into a'
      write(*,*) 'DYNA3D input file'
      write(*,*)
      write(*,*) 'Enter the name of the PATRAN neutral file'
      read(*,10) netfile
      write(*,*)
      write(*,*) 'Enter the name you want for the DYNA3D input file'
      read(*,10) dynfile
10    format(a)
      write(*,*)
      write(*,*) 'Enter the title for the DYNA3D input file'
      read(*,10) title
      write(*,*)
      write(*,*) 'Enter a 1 for 4340 STEEL'
      write(*,*) 'Enter a 2 for ALPHA TITANIUM'
      read(*,*) matpick

C***** Pick the type of load and don't worry about the direction
C***** because it can be reversed in the load curve.  Pressures
C***** act opposite to the normal of the element surface which is
C***** defined by four nodes and the right hand rule.

       write(*,*)
       write(*,*) 'Choose a number for the type of load:'
       write(*,*) '------------------------------------'
       write(*,*)
       write(*,*) '1 = PRESSURE applied to the left end only.'
       write(*,*) '2 = PRESSURE applied to both ends.'
       write(*,*) '3 = AXIAL VELOCITY in the X direction applied to'
```

```
       write(*,*) '    the left end.'
       write(*,*) '4 = ANGULAR VELOCITY about the X axis applied to'
       write(*,*) '    the left end.'
       write(*,*) '5 = ANGULAR VELOCITY about the X axis applied to'
       write(*,*) '    both ends.'
       write(*,*) '6 = PRESSURE and ANGULAR VELOCITY about the X axis'
       write(*,*) '    applied to the left end.'
       write(*,*) '7 = PRESSURE and ANGULAR VELOCITY about the X axis'
       write(*,*) '    applied to both ends.'
       write(*,*) '8 = INTERNAL PRESSURE only.'
       write(*,*) '9 = INTERNAL PRESSURE and ANGULAR VELOCITY about'
       write(*,*) '    the X axis applied to the left end.'
       write(*,*)

      read(*,*) npvflag

      if ((npvflag.eq.8).or.(npvflag.eq.9)) then
        write(*,*)
        write(*,*) 'Do you want plane strain?'
        write(*,*) 'Enter y or n'
        read(*,14) pstrain
14      format(a)
      endif

C****** Open the neutral file and skip the first 2 title lines

      open (1,file=netfile,status='old')

      do 20 i = 1,2
        read(1,*)
20    continue

C****** Read the number of nodes and elements from the third line
C****** of the neutral file, then read the node coordinates and the
C****** connectivity.

      read(1,*) nada, nada, nada, nada, numnod, numel
      read(1,*)

      do 30 i = 1, numnod
        read(1,*) nada, nodnum(i)
        read(1,*) xcord(i), ycord(i), zcord(i)
        read(1,*)
30    continue

      do 60 i = 1, numel
        read(1,*) nada, nelnum(i)
        read(1,*)
        read(1,*) node1(i),node2(i),node3(i),node4(i),
     &            node5(i),node6(i),node7(i),node8(i)
60    continue

C***** Read the element numbers in the pressure load sets that
C***** were defined in PATRAN.  These element numbers will be
C***** used to apply the axial and internal pressures.
C***** we will assign the element numbers to arrays based on the
C***** pressure set number. (i.e. I made it set 3 and 4 in PATRAN)
```

```
      nump1 = 0
      nump2 = 0
      ninc1 = 1
      ninc2 = 1

C***** Read through the pressures and search for the 6 flag in the
C***** left column, when the 8 flag is reached jump out and start
C***** reading displacements which are used for velocity b.c.'s
C***** and constraint b.c.'s.

      do 80 i = 1, max
        read(1,75) check, ntemp, nset
        if (check.eq.' 6') then
           if (nset.eq.3) then
             npress1(ninc1) = ntemp
             nump1 = nump1 + 1
             ninc1 = ninc1 + 1
           elseif (nset.eq.4) then
             npress2(ninc2) = ntemp
             nump2 = nump2 + 1
             ninc2 = ninc2 + 1
           endif
          read(1,*)
          read(1,*)
          else
            goto 90
        endif

80    continue
90    continue
75    format(a2,2i8)

C***** we read the first node of the displacement set so assign
C***** that first before continuing.
C***** Note that PATRAN puts the node numbers in ascending order
C***** even if the displacements were not assigned in order, and
C***** the node numbers from different sets will be mixed together.

C***** we will assign the node numbers to the arrays based on the
C***** displacement set number.  (i.e. I made it set 1 and 2)

      ninc1 = 1
      ninc2 = 1

      if (nset.eq.1) then
        ndisp1(1) = ntemp
        numd1 = 1
        ninc1 = ninc1 + 1
      elseif (nset.eq.2) then
        ndisp2(1) = ntemp
        numd2 = 1
        ninc2 = ninc2 + 1
      endif
      read(1,*)
      read(1,*)

C***** Now read the rest of the nodes by searching for the 8 flag,
C***** if a number different from 8 is read then stop reading.
```

```
      do 100 i = 2, max
        read(1,75) check, ntemp, nset
        if (check.eq.' 8') then
           if (nset.eq.1) then
             ndisp1(ninc1) = ntemp
             numd1 = numd1 + 1
             ninc1 = ninc1 + 1
           elseif (nset.eq.2) then
             ndisp2(ninc2) = ntemp
             numd2 = numd2 + 1
             ninc2 = ninc2 + 1
           endif
          read(1,*)
          read(1,*)
        else
          goto 110
      endif

100   continue
110   continue

      close(1)


C***** Now we have all the nodes and their coordinates, all the
C***** elements and their connectivities, all the elements for
C***** pressure loads, and all the nodes for applying velocity
C***** or for constraining.
C***** Now we can begin creating the DYNA3D input file.


C***** Get the templates for the control cards to print out in the
C***** input file.

       open(14,file='/serv7/rattazzi/bin/comment_cards',status='old')

       do 120 i = 1, 64
         read(14,125) ccard(i)
120     continue

       close(14)
125    format(a)

      open (1,file=dynfile,status='unknown')

C***** Write the 1st control card (title).

      write(1,130) title, '88 large'
130   format(a72,a8)

C***** Write the 2nd control card.

      do 140 i = 1, 10
        write(1,145) ccard(i)
140   continue
145   format(a)

      write(1,150) '1',numnod,numel,'0','0','0','0','0','0'

150   format(4x,a1,2i10,5(9x,a1),4x,a1)
```

```
C***** We need the number of node and element time history blocks
C***** for control card 3.

C***** We are assuming that the numbering of nodes and elements in
C***** the files will be in ascending order.  If not then use the
C***** program nodeth to make them in order.

      write(*,*)
      write(*,*) 'Are there any NODE TIME HISTORY BLOCKS?'
      write(*,*) 'y or n'
      read(*,155) yn1flag
155   format(a)

      if (yn1flag.eq.'y') then
      write(*,*)
      write(*,*) 'Enter the file name of node block numbers.'
      read(*,155) tbnfile
      open(15, file=tbnfile,status='old')

      read(15,*) nmnthb

      do 160 i = 1, nmnthb
        read(15,*) nthnumn(i)
160   continue

      jnblks = 1
      knblkn1(jnblks) = nthnumn(1)

      do 170 i = 2, nmnthb
        if ((nthnumn(i)-nthnumn(i-1)).eq.1) then
          goto 170
          else
             knblkn2(jnblks) = nthnumn(i-1)
             jnblks = jnblks + 1
             knblkn1(jnblks) = nthnumn(i)
        endif
170   continue
      knblkn2(jnblks) = nthnumn(nmnthb)

      if (mod(jnblks,5).eq.0) then
       nnrows = jnblks/5
      else
       nnrows = jnblks/5 + 1
      endif

      close(15)
      endif

C***** End of if statement asking y or n for node blocks.

      write(*,*)
      write(*,*) 'Are there any ELEMENT TIME HISTORY BLOCKS?'
      write(*,*) 'y or n'
      read(*,175) yn2flag

175   format(a)

      if (yn2flag.eq.'y') then
```

```
      write(*,*)
      write(*,*) 'Enter the file name of element block numbers.'
      read(*,175) etfile

      open(15,file=etfile,status='old')
      read(15,*) nmethb

      do 180 i = 1, nmethb
        read(15,*) nthnume(i)
180   continue

      jeblks = 1
      keblkn1(jeblks) = nthnume(1)

      do 190 i = 2, nmethb
        if ((nthnume(i)-nthnume(i-1)).eq.1) then
          goto 190
          else
             keblkn2(jeblks) = nthnume(i-1)
             jeblks = jeblks + 1
             keblkn1(jeblks) = nthnume(i)
        endif
190   continue

      keblkn2(jeblks) = nthnume(nmethb)

      if (mod(jeblks,5).eq.0) then
       nerows = jeblks/5
      else
       nerows = jeblks/5 + 1
      endif

      close(15)

      endif

C***** End of if statement asking y or n for element blocks.


C***** Write the 3rd control card.

     do 200 i = 11, 15
       write(1,205) ccard(i)
200  continue

205  format(a)


     write(1,210) jnblks,jeblks,'0','0','0','0','0','0','0'

210  format(2i5,7(4x,a1))

C***** Write the 4th control card.

      do 220 i = 16, 22
         write(1,215) ccard(i)
220   continue

215   format(a)
```

```
      write(1,225) ' ','0','0','0','0','0','e20.9','0','0','0','0'

225   format(6(4x,a1),a5,4x,a1,9x,a1,2(4x,a1))

C***** Figure out the number of load curves and the appropriate number
C***** of pressure or velocity cards for control card 5.
C***** Write the 5th control card.

      do 230 i = 23, 31
         write(1,235) ccard(i)
230   continue

235   format(a)

      if (npvflag.eq.1) then

      write(1,240)  '1','0',nump1,'0','0','0','0','0','0','0',
     &   '0','0','0','0','0','0'

240   format(2(4x,a1),i5,13(4x,a1))

      elseif (npvflag.eq.2) then

      write(1,240) '2','0',(nump1+nump2),'0','0','0','0','0','0',
     &   '0','0','0','0','0','0','0'

      elseif (npvflag.eq.3) then

      write(1,241) '1','0','0',numd1,'0','0','0','0','0','0',
     &   '0','0','0','0','0','0'

241   format(3(4x,a1),i5,12(4x,a1))

      elseif (npvflag.eq.4) then

      write(1,241) '1','0','0',numd1,'0','0','0','0','0','0',
     &   '0','0','0','0','0','0'

      elseif (npvflag.eq.5) then

      write(1,241) '2','0','0',(numd1+numd2),'0','0','0','0','0','0',
     &   '0','0','0','0','0','0'

      elseif (npvflag.eq.6) then

      write(1,242) '2','0',nump1,numd1,'0','0','0','0','0','0',
     &   '0','0','0','0','0','0'

242   format(2(4x,a1),2i5,12(4x,a1))

      elseif (npvflag.eq.7) then

      write(1,242) '4','0',(nump1+nump2),(numd1+numd2),'0','0','0',
     &  '0','0','0','0','0','0','0','0','0'

      elseif (npvflag.eq.8) then

      write(1,240) '1','0',nump2,'0','0','0','0','0','0','0',
     &   '0','0','0','0','0','0'

      elseif (npvflag.eq.9) then
```

```
      write(1,242) '2','0',nump2,numd1,'0','0','0','0','0','0',
     &   '0','0','0','0','0','0'

      endif


C***** Write the 6th control card.

      do 250 i = 32, 38
         write(1,255) ccard(i)
250   continue

255   format(a)


      write(*,*)
      write(*,*) 'Enter the termination time.'
      read(*,*) ttime
      write(*,*)
      write(*,*) 'Enter delta t between time history dumps.'
      read(*,*) tthd
      write(*,*)
      write(*,*) 'Enter delta t between state plot dumps.'
      read(*,*) tspd

      write(1,260) ttime,tthd,tspd,'0','0','0','0','0','0','0'

260   format(3e10.4,2(4x,a1),2(9x,a1),2(4x,a1),9x,a1)


C***** Write the 7th control card.

      do 270 i = 39, 48
         write(1,275) ccard(i)
270   continue

275   format(a)

      write(1,280) '0','0','0','0','0','0','0','0','0','0','0',
     &   '0','0'

280   format(13(4x,a1))


C***** Write the 8th control card.

      do 290 i = 49, 57
         write(1,295) ccard(i)
290   continue

295   format(a)

      write(1,300) '0','0',' 0.000E+00','0','0','0','0','  250',
     &   ' 1.000E-04',' 9.950E-01',' 0.000E+00','0'

300   format(2(4x,a1),a10,4(4x,a1),a5,3a10,4x,a1)


C***** Write the 9th control card.
```

```
      do 310 i = 58, 61
         write(1,315) ccard(i)
310   continue

315   format(a)

      write(1,320) '1','0','0','0',' 0.000E+00','0','0','0'

320   format(4(4x,a1),a10,3(4x,a1))


C***** Write the material card.
C***** Note: These units are in centimeters, grams, microseconds.

      do 330 i = 62, 64
         write(1,335) ccard(i)
330   continue

335   format(a)

      if (matpick.eq.1) then

      write(1,340) '1','15',' 7.840E+00','1','0',' 1.000E-01',
     &   '0',' 1.500E+00',' 6.000E-02','0','0','0'
      write(1,341) ' Material Type # 15 (Johnson/Cook)'
      write(1,342) ' .7600E+00',' .7922E-02',' .5095E-02',
     &   ' .2600E+00',' .1400E-01',' .1030E+01',' .1793E+04',
     &   ' .2980E+03'
      write(1,343) ' .1000E-05',' .4770E-05',' .0000E+00',
     &   ' .0000E+00',' .0000E+00'
      write(1,344) ' .2000E+20',' .0000E+00',' .0000E+00',
     &   ' .0000E+00',' .0000E+00'

      elseif (matpick.eq.2) then

      write(1,340) '1','15',' 4.507E+00','1','0',' 1.000E-01',
     &   '0',' 1.500E+00',' 6.000E-02','0','0','0'
      write(1,341) ' Material Type # 15 (Johnson/Cook)'
      write(1,342) ' .4300E+00',' .3090E-02',' .9290E-02',
     &   ' .7300E+00',' .3340E-01',' .3500E+00',' .1943E+04',
     &   ' .2980E+03'
      write(1,343) ' .1000E-10',' .5230E-05',' .0000E+00',
     &   ' .0000E+00',' .0000E+00'
      write(1,344) ' .2000E+20',' .0000E+00',' .0000E+00',
     &   ' .0000E+00',' .0000E+00'


      endif

340   format(4x,a1,3x,a2,a10,2(4x,a1),a10,4x,a1,2a10,3(4x,a1))
341   format(a62)
342   format(8a10)
343   format(5a10)
344   format(5a10)

      write(1,*)
      write(1,*)
      write(1,*)
```

```
        if (matpick.eq.1) then

        write(1,345) ' Equation-of-State Form 1 (Linear Polynomial)'
        write(1,346) ' .0000E+00',' .1570E+01',' .0000E+00',
     &   ' .0000E+00',' .0000E+00',' .0000E+00',' .0000E+00',
     &   ' .0000E+00'
        write(1,347) ' .1000E+01'

        elseif (matpick.eq.2) then

        write(1,345) ' Equation-of-State Form 1 (Linear Polynomial)'
        write(1,346) ' .0000E+00',' .1070E+01',' .0000E+00',
     &   ' .0000E+00',' .0000E+00',' .0000E+00',' .0000E+00',
     &   ' .0000E+00'
        write(1,347) ' .1000E+01'

        endif

345     format(a45)
346     format(8a10)
347     format(a10)


C***** Write the node cards and assign the degree of freedom code
C***** based on the type of loading chosen previously.

C***** No constraint
       ncode0 = 0
C***** Constrained in X
       ncode1 = 1
C***** Constrained in X,Y,Z
       ncode7 = 7

        write(1,350) '*------------------------- NODE DEFINITIONS ',
     &  '--------------------'

350     format(a45,a20)

365    format(i8,i5,3e20.13,i5)

       j = 1
       jj = 1

       if ((npvflag.eq.1).or.(npvflag.eq.3)) then

         do 360 i = 1, numnod

           if (nodnum(i).eq.ndisp2(j)) then

           write(1,365) nodnum(i),ncode1,xcord(i),ycord(i),zcord(i),
     &      ncode0
           j = j + 1

           else

           write(1,365) nodnum(i),ncode0,xcord(i),ycord(i),zcord(i),
     &      ncode0

           endif
```

```
360    continue

       elseif (((npvflag.eq.2).or.(npvflag.eq.5)).or.(npvflag.eq.7))
     &    then

         do 370 i = 1, numnod

           write(1,365) nodnum(i),ncode0,xcord(i),ycord(i),zcord(i),
     &     ncode0

370    continue

       elseif ((npvflag.eq.4).or.(npvflag.eq.6)) then

         do 380 i = 1, numnod

           if (nodnum(i).eq.ndisp2(j)) then

           write(1,365) nodnum(i),ncode7,xcord(i),ycord(i),zcord(i),
     &     ncode0
           j = j + 1

           else

           write(1,365) nodnum(i),ncode0,xcord(i),ycord(i),zcord(i),
     &     ncode0

           endif

380    continue

       elseif (npvflag.eq.8) then

         if (pstrain.eq.'y') then

         do 382 i = 1, numnod
           write(1,365) nodnum(i),ncode1,xcord(i),ycord(i),zcord(i),
     &     ncode0
382      continue

        else

         do 383 i = 1, numnod

         if (nodnum(i).eq.ndisp1(j)) then
           write(1,365) nodnum(i),ncode1,xcord(i),ycord(i),zcord(i),
     &     ncode0
           j = j + 1
         elseif (nodnum(i).eq.ndisp2(jj)) then
           write(1,365) nodnum(i),ncode1,xcord(i),ycord(i),zcord(i),
     &     ncode0
           jj = jj + 1
         else
           write(1,365) nodnum(i),ncode0,xcord(i),ycord(i),zcord(i),
     &     ncode0
         endif

383      continue

        endif  ! If for plane strain.
```

```
      elseif (npvflag.eq.9) then

        if (pstrain.eq.'y') then

        do 384 i = 1, numnod
          if (nodnum(i).eq.ndisp2(j)) then

          write(1,365) nodnum(i),ncode7,xcord(i),ycord(i),zcord(i),
     &     ncode0
          j = j + 1

          else

          write(1,365) nodnum(i),ncode1,xcord(i),ycord(i),zcord(i),
     &     ncode0

         endif
384     continue

       else   !not plane strain

         do 385 i = 1, numnod

          if (nodnum(i).eq.ndisp2(j)) then

          write(1,365) nodnum(i),ncode7,xcord(i),ycord(i),zcord(i),
     &     ncode0
          j = j + 1

          else

          write(1,365) nodnum(i),ncode0,xcord(i),ycord(i),zcord(i),
     &     ncode0

          endif

385     continue

       endif  !end of if for plane strain choice.

      endif  !end of if for all the loading choices.


C***** Write the solid element cards.

       write(1,390) '*------------------ SOLID ELEMENT DEFINITIONS',
     &   '------------------'

390   format(a45,a19)

        mat = 1

        do 400 i = 1, numel

          write(1,405) nelnum(i),mat,node1(i),node2(i),node3(i),
     &          node4(i),node5(i),node6(i),node7(i),node8(i)

400     continue

405   format(i8,i5,8i8)
```

```
C***** Write node time history blocks if they were requested.

       if (yn1flag.eq.'y') then

        write(1,410) '*------------------- Node Time History Blocks ',
     &    '-----------------------------'

410     format(a46,a30)

         mstart = 1
         mend = 5
         do 420 i = 1, nnrows
           if ((i.eq.nnrows).and.(mod(jnblks,5).gt.0)) then
             mend = mend - 5
             mend = mend + mod(jnblks,5)
           endif
           write(1,425) (knblkn1(m),knblkn2(m), m=mstart,mend)
           mstart = mstart + 5
           if (i.ne.nnrows) mend = mend + 5

420      continue
425      format(10i8)

      endif


C***** Write element time history blocks if they were requested.

      if (yn2flag.eq.'y') then

       write(1,430) '*----------------- Element Time History Blocks ',
     &    '-----------------------------'

430      format(a46,a30)

         mstart = 1
         mend = 5
         do 440 i = 1, nerows
           if ((i.eq.nerows).and.(mod(jeblks,5).gt.0)) then
             mend = mend - 5
             mend = mend + mod(jeblks,5)
           endif
           write(1,445) (keblkn1(m),keblkn2(m), m=mstart,mend)
           mstart = mstart + 5
           if (i.ne.nerows) mend = mend + 5

440      continue
445      format(10i8)

      endif


C***** Write the load curves based on the type of loading chosen.


      if ((npvflag.eq.1).or.(npvflag.eq.8)) then

       write(1,450) '*----------------- Load Curve #1 For Pressure ',
     &  '----------------------'
```

```
450   format(a45,a22)

        write(*,*)
        write(*,*) 'How many points in the PRESSURE load curve?'
        read(*,*) npoints
        write(1,455) '1',npoints,'0'
455     format(4x,a1,i5,4x,a1)
        write(*,*)

        do 460 i = 1, npoints
          write(*,465) 'Enter TIME point number',i
          read(*,*) time
          write(*,466) 'Enter PRESSURE point number',i
          read(*,*) zload
          write(1,470) time, zload
460     continue

465     format(a23,i3)
466     format(a27,i3)
470     format(e10.5,e10.4)


      elseif (npvflag.eq.2) then

        write(1,480) '*---------------- Load Curve #1 For Pressure 1',
     &  ' ----------------------'

480   format(a46,a23)

        write(*,*)
        write(*,*) 'How many points in the 1st PRESSURE load curve?'
        read(*,*) npoints
        write(1,485) '1',npoints,'0'
485     format(4x,a1,i5,4x,a1)
        write(*,*)

        do 490 i = 1, npoints
          write(*,495) 'Enter TIME point number',i
          read(*,*) time
          write(*,496) 'Enter PRESSURE point number',i
          read(*,*) zload
          write(1,500) time, zload
490     continue

495     format(a23,i3)
496     format(a27,i3)
500     format(e10.5,e10.4)


        write(1,510) '*---------------- Load Curve #2 For Pressure 2',
     &  ' ----------------------'

510   format(a46,a23)

        write(*,*)
        write(*,*) 'How many points in the 2nd PRESSURE load curve?'
```

```
        read(*,*) npoints
        write(1,515) '2',npoints,'0'
515     format(4x,a1,i5,4x,a1)
        write(*,*)

        do 520 i = 1, npoints
          write(*,525) 'Enter TIME point number',i
          read(*,*) time
          write(*,526) 'Enter PRESSURE point number',i
          read(*,*) zload
          write(1,530) time, zload
520     continue

525     format(a23,i3)
526     format(a27,i3)
530     format(e10.5,e10.4)


      elseif ((npvflag.eq.3).or.(npvflag.eq.4)) then

       write(1,540) '*---------------- Load Curve #1 For Velocity ',
     &  '----------------------'

540    format(a45,a22)

        write(*,*)
        write(*,*) 'How many points in the VELOCITY load curve?'
        read(*,*) npoints
        write(1,545) '1',npoints,'0'
545     format(4x,a1,i5,4x,a1)
        write(*,*)

        do 550 i = 1, npoints
          write(*,555) 'Enter TIME point number',i
          read(*,*) time
          write(*,556) 'Enter VELOCITY point number',i
          read(*,*) zload
          write(1,560) time, zload
550     continue

555     format(a23,i3)
556     format(a27,i3)
560     format(e10.5,e10.4)


      elseif (npvflag.eq.5) then

       write(1,570) '*---------------- Load Curve #1 For Velocity 1',
     &  ' ----------------------'

570    format(a46,a23)

        write(*,*)
        write(*,*) 'How many points in the 1st VELOCITY load curve?'
        read(*,*) npoints
        write(1,575) '1',npoints,'0'
575     format(4x,a1,i5,4x,a1)
```

```
        write(*,*)

        do 580 i = 1, npoints
          write(*,585) 'Enter TIME point number',i
          read(*,*) time
          write(*,586) 'Enter VELOCITY point number',i
          read(*,*) zload
          write(1,590) time, zload
580     continue

585     format(a23,i3)
586     format(a27,i3)
590     format(e10.5,e10.4)


      write(1,600) '*---------------- Load Curve #2 For Velocity 2',
     &  ' ----------------------'

600   format(a46,a23)

        write(*,*)
        write(*,*) 'How many points in the 2nd VELOCITY load curve?'
        read(*,*) npoints
        write(1,605) '2',npoints,'0'
605     format(4x,a1,i5,4x,a1)
        write(*,*)

        do 610 i = 1, npoints
          write(*,615) 'Enter TIME point number',i
          read(*,*) time
          write(*,616) 'Enter VELOCITY point number',i
          read(*,*) zload
          write(1,620) time, zload
610     continue

615     format(a23,i3)
616     format(a27,i3)
620     format(e10.5,e10.4)


     elseif ((npvflag.eq.6).or.(npvflag.eq.9)) then

      write(1,630) '*---------------- Load Curve #1 For Pressure ',
     &  '----------------------'

630   format(a45,a22)

        write(*,*)
        write(*,*) 'How many points in the PRESSURE load curve?'
        read(*,*) npoints
        write(1,635) '1',npoints,'0'
635     format(4x,a1,i5,4x,a1)
        write(*,*)

        do 640 i = 1, npoints
          write(*,645) 'Enter TIME point number',i
          read(*,*) time
          write(*,646) 'Enter PRESSURE point number',i
```

```
          read(*,*) zload
          write(1,650) time, zload
640     continue

645     format(a23,i3)
646     format(a27,i3)
650     format(e10.5,e10.4)


      write(1,660) '*---------------- Load Curve #2 For Velocity ',
     &  '----------------------'

660   format(a45,a22)

        write(*,*)
        write(*,*) 'How many points in the VELOCITY load curve?'
        read(*,*) npoints
        write(1,665) '2',npoints,'0'
665     format(4x,a1,i5,4x,a1)
        write(*,*)

        do 670 i = 1, npoints
          write(*,675) 'Enter TIME point number',i
          read(*,*) time
          write(*,676) 'Enter VELOCITY point number',i
          read(*,*) zload
          write(1,680) time, zload
670     continue

675     format(a23,i3)
676     format(a27,i3)
680     format(e10.5,e10.4)


     elseif (npvflag.eq.7) then

      write(1,690) '*---------------- Load Curve #1 For Pressure 1',
     &  ' ----------------------'

690   format(a46,a23)

        write(*,*)
        write(*,*) 'How many points in the 1st PRESSURE load curve?'
        read(*,*) npoints
        write(1,695) '1',npoints,'0'
695     format(4x,a1,i5,4x,a1)
        write(*,*)

        do 700 i = 1, npoints
          write(*,705) 'Enter TIME point number',i
          read(*,*) time
          write(*,706) 'Enter PRESSURE point number',i
          read(*,*) zload
          write(1,710) time, zload
700     continue

705     format(a23,i3)
706     format(a27,i3)
```

```
710       format(e10.5,e10.4)


        write(1,720) '*---------------- Load Curve #2 For Pressure 2',
     &  ' ----------------------'

720     format(a46,a23)

          write(*,*)
          write(*,*) 'How many points in the 2nd PRESSURE load curve?'
          read(*,*) npoints
          write(1,725) '2',npoints,'0'
725       format(4x,a1,i5,4x,a1)
          write(*,*)

          do 730 i = 1, npoints
            write(*,735) 'Enter TIME point number',i
            read(*,*) time
            write(*,736) 'Enter PRESSURE point number',i
            read(*,*) zload
            write(1,740) time, zload
730       continue

735       format(a23,i3)
736       format(a27,i3)
740       format(e10.5,e10.4)


        write(1,750) '*---------------- Load Curve #3 For Velocity 1',
     &  ' ----------------------'

750     format(a46,a23)

          write(*,*)
          write(*,*) 'How many points in the 1st VELOCITY load curve?'
          read(*,*) npoints
          write(1,755) '3',npoints,'0'
755       format(4x,a1,i5,4x,a1)
          write(*,*)

          do 760 i = 1, npoints
            write(*,765) 'Enter TIME point number',i
            read(*,*) time
            write(*,766) 'Enter VELOCITY point number',i
            read(*,*) zload
            write(1,770) time, zload
760       continue

765       format(a23,i3)
766       format(a27,i3)
770       format(e10.5,e10.4)


        write(1,780) '*---------------- Load Curve #2 For Velocity 2',
     &  ' ----------------------'

780     format(a46,a23)
```

```
         write(*,*)
         write(*,*) 'How many points in the 2nd VELOCITY load curve?'
         read(*,*) npoints
         write(1,785) '4',npoints,'0'
785      format(4x,a1,i5,4x,a1)
         write(*,*)

         do 790 i = 1, npoints
           write(*,795) 'Enter TIME point number',i
           read(*,*) time
           write(*,796) 'Enter VELOCITY point number',i
           read(*,*) zload
           write(1,800) time, zload
790      continue

795      format(a23,i3)
796      format(a27,i3)
800      format(e10.5,e10.4)


      endif


C***** Write the prescribed pressure cards and the prescribed velocity
C***** cards based on the type of loading chosen.


      scale = 1.0
      vect = 0.

      if (npvflag.eq.1) then

       write(1,810) '*---------------------- PRESCRIBED PRESSURE',
     &    ' ------------------'

810    format(a47,a19)

       do 820 i = 1, nump1

       write(1,825) '1', node4(npress1(i)),node3(npress1(i)),
     &   node2(npress1(i)),node1(npress1(i)),scale,scale,scale,scale

820 continue

825     format(4x,a1,4i8,4e10.4)


      elseif (npvflag.eq.2) then

       write(1,830) '*---------------------- PRESCRIBED PRESSURE',
     &    ' ------------------'

830    format(a47,a19)

       do 840 i = 1, nump1

        write(1,845) '1', node4(npress1(i)),node3(npress1(i)),
```

```
     &   node2(npress1(i)),node1(npress1(i)),scale,scale,scale,scale
840 continue
845     format(4x,a1,4i8,4e10.4)

      do 850 i = 1, nump2
       write(1,845) '2', node4(npress2(i)),node3(npress2(i)),
     &   node2(npress2(i)),node1(npress2(i)),scale,scale,scale,scale
850 continue

      elseif (npvflag.eq.3) then

       write(1,860) '*---------------------- PRESCRIBED VELOCITY',
     &    ' ------------------'
860    format(a47,a19)

      do 870 i = 1, numd1
        write(1,875) ndisp1(i),'1','1',scale,vect,vect,vect,'0'
870 continue
875     format(i8,4x,a1,4x,a1,4e10.4,4x,a1)

      elseif (npvflag.eq.4) then
       write(1,880) '*---------------------- PRESCRIBED VELOCITY',
     &    ' ------------------'
880    format(a47,a19)

      do 890 i = 1, numd1
        write(1,895) ndisp1(i),'1','9',scale,vect,vect,vect,'0'
890 continue
895     format(i8,4x,a1,4x,a1,4e10.4,4x,a1)

      elseif (npvflag.eq.5) then

       write(1,900) '*---------------------- PRESCRIBED VELOCITY',
     &    ' ------------------'
900    format(a47,a19)

      do 910 i = 1, numd1
        write(1,915) ndisp1(i),'1','9',scale,vect,vect,vect,'0'
910 continue
915     format(i8,4x,a1,4x,a1,4e10.4,4x,a1)

      do 920 i = 1, numd2
        write(1,915) ndisp2(i),'2','9',scale,vect,vect,vect,'0'
920 continue
```

```
        elseif (npvflag.eq.6) then

         write(1,930) '*---------------------- PRESCRIBED PRESSURE',
     &      ' -----------------'
930      format(a47,a19)
         do 940 i = 1, nump1
           write(1,945) '1', node4(npress1(i)),node3(npress1(i)),
     &     node2(npress1(i)),node1(npress1(i)),scale,scale,scale,scale
940 continue
945       format(4x,a1,4i8,4e10.4)

         write(1,950) '*---------------------- PRESCRIBED VELOCITY',
     &      ' -----------------'
950      format(a47,a19)
         do 960 i = 1, numd1
           write(1,965) ndisp1(i),'2','9',scale,vect,vect,vect,'0'
960 continue
965       format(i8,4x,a1,4x,a1,4e10.4,4x,a1)
        elseif (npvflag.eq.7) then
         write(1,970) '*---------------------- PRESCRIBED PRESSURE',
     &      ' -----------------'
970      format(a47,a19)
        do 980 i = 1, nump1
          write(1,985) '1', node4(npress1(i)),node3(npress1(i)),
     &     node2(npress1(i)),node1(npress1(i)),scale,scale,scale,scale
980 continue
985       format(4x,a1,4i8,4e10.4)

        do 990 i = 1, nump2
         write(1,985) '2', node4(npress2(i)),node3(npress2(i)),
     &     node2(npress2(i)),node1(npress2(i)),scale,scale,scale,scale
990 continue

         write(1,1000) '*---------------------- PRESCRIBED VELOCITY',
     &      ' -----------------'
1000      format(a47,a19)
```

```
      do 1010 i = 1, numd1
        write(1,1015) ndisp1(i),'3','9',scale,vect,vect,vect,'0'
1010 continue

1015     format(i8,4x,a1,4x,a1,4e10.4,4x,a1)

      do 1020 i = 1, numd2
        write(1,1015) ndisp2(i),'4','9',scale,vect,vect,vect,'0'
1020 continue

      elseif (npvflag.eq.8) then

       write(1,1028) '*---------------------- PRESCRIBED PRESSURE',
     &    ' ------------------'

1028    format(a47,a19)

       do 1030 i = 1, nump2

       write(1,1035) '1', node1(npress2(i)),node5(npress2(i)),
     &   node8(npress2(i)),node4(npress2(i)),scale,scale,scale,scale

1030   continue

1035     format(4x,a1,4i8,4e10.4)

       elseif (npvflag.eq.9) then

       write(1,1040) '*---------------------- PRESCRIBED PRESSURE',
     &    ' ------------------'

1040    format(a47,a19)

       do 1050 i = 1, nump2

         write(1,1055) '1', node1(npress2(i)),node5(npress2(i)),
     &   node8(npress2(i)),node4(npress2(i)),scale,scale,scale,scale

1050   continue

1055     format(4x,a1,4i8,4e10.4)

       write(1,1060) '*----------------------- PRESCRIBED VELOCITY',
     &    ' ------------------'

1060    format(a47,a19)

       do 1070 i = 1, numd1
         write(1,1065) ndisp1(i),'2','9',scale,vect,vect,vect,'0'
1070   continue

1065     format(i8,4x,a1,4x,a1,4e10.4,4x,a1)

      endif  !end of all load definitions.

      stop
      end
```

## A.2 Program to Convert Modified DYNA3D Output Files to PATRAN Results Files

```
c*****************************************************************
c*********  program: dynapat.f
c*********  Program to create Patran elem/node results files for each
c*********  individual time available from the binary files created
c*********  by the modified version of DYNA3D. You can choose the
c*********  number of time states you want to include, and the time
c*********  interval.  You supply the files of element numbers and
c*********  node numbers you want to extract data for.
c*********  Program last modified 4/13/96
c*****************************************************************

      implicit double precision (a-h,o-z)
      parameter (max=400000)
      character star*1,filen*12,filen2*12,fnum*2,title*16,
     &           file1*25,filen3*12
      dimension filen(100), fnum(100), nelnum(max), time(max),
     & sxx(max),syy(max),szz(max),sxy(max),szx(max),syz(max),
     & menum(max),nodnum(max),filen2(100),defmax(max),ndmax(max),
     & mnnum(max),dz(max),dx(max),dy(max),seff(max),filen3(100),
     & ps(max),temp(max)

      open(1,file='tmpfile',status='unknown')

20    format(a,i1)

      do 25 i = 1, 9
        write(1,20) '0',i
25    continue

30    format(i2)

      do 40 i = 10,99
        write(1,30) i
40    continue

      close(1)

      open(1,file='tmpfile',status='old')

50    format(a2)
      do 60 i = 1, 99
        read(1,50) fnum(i)
        filen(i) = 'stress'//fnum(i)
        filen2(i) = 'ndisp'//fnum(i)
        filen3(i) = 'tps'//fnum(i)
60    continue

      close(1)

62    format(a)

c***** Open the files with the requested element and node numbers,
c***** and read them into arrays.

      write(*,*) 'Enter the file name of element numbers'
      read(*,62) file1
```

```
      open(1,file=file1,status='old')

      read(1,*) numel

      do 70 i = 1, numel
        read(1,*) nelnum(i)
70    continue

      close(1)

      write(*,*)
      write(*,*) 'Enter the file name of node numbers'
      read(*,62) file1

      open(1,file=file1,status='old')

      read(1,*) numnod

      maxnod = 0
      do 80 i = 1, numnod
        read(1,*) nodnum(i)
        if (nodnum(i).gt.maxnod) then
        maxnod = nodnum(i)
        endif
80    continue

      close(1)

c***** Open the binary files of time history data for stress and
c***** effective plastic strain / temperature.

      open(1,file='stress_th',status='old',form='unformatted')
      open(3,file='ps_temp',status='old',form='unformatted')

84       format(3e16.9)
85       format(a10,i8,a7)

         read(1) numeb,nstates
         read(3) numeb,nstates
         write(*,*)
         write(*,85) 'There are ',nstates,' states'
         write(*,*) 'How many states do you want to include?'
         read(*,*) nrstate
         write(*,*)
         write(*,*) 'What state interval do you want'
         read(*,*) nint

         mm = 1

         do 90 i = 1, nstates
           read(1) time(i)
           read(3) time(i)

           do 100 j = 1, numeb
             read(1) menum(j),sxx(menum(j)),syy(menum(j)),
     &       szz(menum(j)),sxy(menum(j)),syz(menum(j)),
     &       szx(menum(j)),seff(menum(j))
```

```
c***** PATRAN doesn't fill elements with color that have a
c***** quantity with a value of zero.

              if (sxx(menum(j)).eq.0.) sxx(menum(j))=1.e-20
              if (syy(menum(j)).eq.0.) syy(menum(j))=1.e-20
              if (szz(menum(j)).eq.0.) szz(menum(j))=1.e-20
              if (sxy(menum(j)).eq.0.) sxy(menum(j))=1.e-20
              if (syz(menum(j)).eq.0.) syz(menum(j))=1.e-20
              if (szx(menum(j)).eq.0.) szx(menum(j))=1.e-20
              if (seff(menum(j)).eq.0.) seff(menum(j))=1.e-20

              read(3) mrep,ps(menum(j)),temp(menum(j))

              if (ps(menum(j)).eq.0.) ps(menum(j))=1.e-20
              if (temp(menum(j)).eq.0.) temp(menum(j))=1.e-20

100        continue !end of the loop over the elements.


c***** This if statement will cause the program to
c***** skip to the requested time interval, and end at
c***** the requested time state.

       if ((i.le.nrstate).and.((mod(i,nint)).eq.0)) then

         open(2,file=filen(mm),status='unknown')
         open(4,file=filen3(mm),status='unknown')

115      format(i8,7x,'8')
116      format(a7,e13.7,a5)
119      format(6e13.7)
118      format(1e13.7)
         write(2,116) 'time = ', time(i), ' usec'
         write(2,*) '7'
         write(2,*) 'Element Stress'
         write(2,*)

         write(4,116) 'time = ', time(i), ' usec'
         write(4,*) '2'
         write(4,*) 'PS or temp.'
         write(4,*)


         do 110 kk = 1, numel
           write(2,115) nelnum(kk)
           write(2,119) sxx(nelnum(kk)),syy(nelnum(kk)),
     &        szz(nelnum(kk)),sxy(nelnum(kk)),syz(nelnum(kk)),
     &        szx(nelnum(kk))
           write(2,118) seff(nelnum(kk))

           write(4,115) nelnum(kk)
           write(4,119) ps(nelnum(kk)),temp(nelnum(kk))

110      continue

         mm = mm + 1

         close(2)
```

```
        close(4)

        endif  ! End the if statement for skipping time states.

90      continue  !end of the loop over the time states.

        close(1)
        close(3)

C********* End of element data **********************************


C********* Start of node data ************************************


         do 125 i = 1, max
           defmax(i) = 0.
125      continue

         mm = 1

c**** Open up the binary file with the time history data for
c**** displacements.

      open(1,file='disp_th',status='old',form='unformatted')
         read(1) numnb,nstates

         do 130 i = 1, nstates
           read(1) time(i)

         do 140 j = 1, numnb
          read(1) mnnum(j),dx(mnnum(j)),dy(mnnum(j)),
     &           dz(mnnum(j))

140      continue

230   format(i8,3e13.7)
232   format(a7,e13.7,a5)
233   format(2i9,e15.6,2i9)
234   format(6e13.7)

      nwidth = 3

c***** This if statement will cause the program to
c***** skip to the requested time interval, and end at
c***** the requested time state.

        if ((i.le.nrstate).and.((mod(i,nint)).eq.0)) then

c***** The PATRAN results file needs the maximum deformation.

           do 245 km = 1,numnod

             if (dabs(dx(nodnum(km))).gt.defmax(i)) then
               defmax(i) = dx(nodnum(km))
               ndmax(i) = nodnum(km)
             endif

             if (dabs(dy(nodnum(km))).gt.defmax(i)) then
               defmax(i) = dy(nodnum(km))
```

```
              ndmax(i) = nodnum(km)
            endif

            if (dabs(dz(nodnum(km))).gt.defmax(i)) then
              defmax(i) = dz(nodnum(km))
              ndmax(i) = nodnum(km)
            endif

245       continue


          open(2,file=filen2(mm),status='unknown')

          write(2,232) 'time = ', time(i), ' usec'
          write(2,233) numnod,maxnod,defmax(i),ndmax(i),nwidth
          write(2,*)
          write(2,*) 'Nodal Displacement'

          do 250 j = 1, numnod
            write(2,230) nodnum(j), dx(nodnum(j)),
     &      dy(nodnum(j)),dz(nodnum(j))
250       continue


          mm = mm + 1

          close(2)

           endif

130   continue !End the loop over the number of time states.

      close(1)

      write(*,*)
      write(*,*) 'PROGRAM COMPLETED'
      write(*,*)
      stop
      end
```

# B FORTRAN Utility Programs

## B.1 Program to Calculate Torque on the Tube

```
C************************************************************
C*****  Program torq  (version torq5.f)
C*****  Program reads in the mesh, reads in time history data, then
C*****  reads the element numbers that will be used to calculate
C*****  torque.  It finds the 4 surface nodes for each element,
C*****  then updates the coordinates, calculates centroids, then
C*****  calculates surface area, and finally calculates the
C*****  torque.
C*****  Program last modified 05/12/96
C************************************************************

      implicit double precision (a-h,o-z)
      parameter (max=400000)
      dimension nodnum(max),xcord(max),ycord(max),zcord(max),
     &  nelnum(max),node1(max),node2(max),node3(max),node4(max),
     &  node5(max),node6(max),node7(max),node8(max),
     &  time(500),s12(max),s13(max),xdisp(max),
     &  ydisp(max),zdisp(max),nhelem(max),nhnode(max),
     &  nspnnum(max),nspenum(max),dtime(2)

      character file1a*25,file1b*25,netfile*25,file2*25,file3*25,
     &  com*1,star*1


10    format(a)
      write(*,*) 'Enter the file name containing the element'
      write(*,*) 'numbers'
      read(*,10) file2
      write(*,*)
      write(*,*) 'Enter the PATRAN neutral file name'
      read(*,10) netfile
      write(*,*)

C***** Read the element numbers in the special element file

      open (1,file=file2,status='old')
      read(1,*) numspe

45    format(i8)

      do 50 i = 1, numspe
       read(1,*) nspenum(i)
50    continue

      close(1)

C****** check the number of states and delta t ******************

      open (1,file='stress_th',status='old',form='unformatted')


      read(1) nte,nst

      do 400 jj = 1, 2
```

```
        read(1) dtime(jj)

        do  390 k = 1, nte
         read(1) nh,var1,var2,var3,var4,
     &              var5,var6,var7

 390    continue

 400    continue

        delt = dtime(2) - dtime(1)

 410    format(a10,i5,a7)
 420    format(a11,e14.7)
        close(1)

        write(*,*)
        write(*,410) 'There are ',nst,' states'
        write(*,*)
        write(*,420) 'Delta t is ',delt
        write(*,*)
        write(*,*) 'How many states do you want to include'
        read(*,*) nrstate

C*****  read in all the node numbers, their coordinates, the element
C*****  numbers and the connectivities.

      open(1,file=netfile,status='old')

      do 20 i = 1,2
        read(1,*)
20    continue

      read(1,*) nada,nada,nada,nada,numnod,numel
      read(1,*)

      do 30 i = 1, numnod
        read(1,*) nada,nodnum(i)
        read(1,*) xcord(i),ycord(i),zcord(i)
        read(1,*)
30    continue

      do 40 i = 1, numel
        read(1,*) nada, nelnum(i)
        read(1,*)
        read(1,*) node1(i),node2(i),node3(i),node4(i),
     &            node5(i),node6(i),node7(i),node8(i)
40    continue

      close(1)

C*********************************************************************


C***** Read the time history data  ********************************

      open (1,file='stress_th',status='old',form='unformatted')
      open (2,file='disp_th',status='old',form='unformatted')
      open (3,file='torq.dat',status='unknown')
```

```
      read(1) ntote,nstates
      read(2) ntotn,nstates


      do 90 jj = 1, nrstate

        read(1) time(jj)
        read(2) time(jj)

        do 100 k = 1, ntote
         read(1) nhelem(k),s11,s22,s33,s12(nhelem(k)),
     &               s23,s13(nhelem(k)),seff

 100    continue

        do 110 k = 1, ntotn
         read(2) nhnode(k),xdisp(nhnode(k)),
     &            ydisp(nhnode(k)),zdisp(nhnode(k))

 110    continue


C**********************************************************


         torque = 0.d0
         tsum = 0.d0
         do 200 j = 1, numspe

C***** Update the coordinates of the 4 surface nodes with
C***** the nodal displacements for each time step

           xcord1 = xcord(node1(nspenum(j)))
     &      + xdisp(node1(nspenum(j)))
           xcord2 = xcord(node2(nspenum(j)))
     &      + xdisp(node2(nspenum(j)))
           xcord3 = xcord(node3(nspenum(j)))
     &      + xdisp(node3(nspenum(j)))
           xcord4 = xcord(node4(nspenum(j)))
     &      + xdisp(node4(nspenum(j)))

           ycord1 = ycord(node1(nspenum(j)))
     &      + ydisp(node1(nspenum(j)))
           ycord2 = ycord(node2(nspenum(j)))
     &      + ydisp(node2(nspenum(j)))
           ycord3 = ycord(node3(nspenum(j)))
     &      + ydisp(node3(nspenum(j)))
           ycord4 = ycord(node4(nspenum(j)))
     &      + ydisp(node4(nspenum(j)))

           zcord1 = zcord(node1(nspenum(j)))
     &      + zdisp(node1(nspenum(j)))
           zcord2 = zcord(node2(nspenum(j)))
     &      + zdisp(node2(nspenum(j)))
```

```
            zcord3 = zcord(node3(nspenum(j)))
     &       + zdisp(node3(nspenum(j)))
            zcord4 = zcord(node4(nspenum(j)))
     &       + zdisp(node4(nspenum(j)))


C***** Calculate the surface area using four nodes

            xx1=(-xcord1+xcord2+xcord3-xcord4)/2.d0
            xx2=(-ycord1+ycord2+ycord3-ycord4)/2.d0
            xx3=(-zcord1+zcord2+zcord3-zcord4)/2.d0
            xy1=(-xcord1-xcord2+xcord3+xcord4)/2.d0
            xy2=(-ycord1-ycord2+ycord3+ycord4)/2.d0
            xy3=(-zcord1-zcord2+zcord3+zcord4)/2.d0
            ee = xx1*xx1 + xx2*xx2 + xx3*xx3
            ff = xx1*xy1 + xx2*xy2 + xx3*xy3
            gg = xy1*xy1 + xy2*xy2 + xy3*xy3

            area=dsqrt(ee*gg-ff*ff)

C***** Calculate the centroid of the element
C**** xcent is not really correct because you would need all
C**** 8 nodes and divide by 8

            xcent = (xcord1 + xcord2 + xcord3 + xcord4)/4.d0
            ycent = (ycord1 + ycord2 + ycord3 + ycord4)/4.d0
            zcent = (zcord1 + zcord2 + zcord3 + zcord4)/4.d0


C***** Calculate torque

      if ((zcent.lt.0.).and.(s12(nspenum(j)).gt.0.))
     &  then
        sign2 = 1.d0
      elseif
     &  ((zcent.lt.0.).and.(s12(nspenum(j)).lt.0.))
     &  then
        sign2 = -1.d0
      elseif
     &  ((zcent.gt.0.).and.(s12(nspenum(j)).gt.0.))
     &  then
        sign2 = -1.d0
      elseif
     &  ((zcent.gt.0.).and.(s12(nspenum(j)).lt.0.))
     &  then
        sign2 = 1.d0
      endif

      if ((ycent.lt.0.).and.(s13(nspenum(j)).gt.0.))

     &  then
        sign3 = -1.d0
      elseif
     &  ((ycent.lt.0.).and.(s13(nspenum(j)).lt.0.))
     &  then
        sign3 = 1.d0
      elseif
```

```
     &  ((ycent.gt.0.).and.(s13(nspenum(j)).gt.0.))
     &  then
          sign3 = 1.d0
       elseif
     & ((ycent.gt.0.).and.(s13(nspenum(j)).lt.0.))
     &  then
          sign3 = -1.d0
       endif

          torque = torque
     &         + (sign3*dabs(ycent)
     &         *dabs(s13(nspenum(j)))
     &         + sign2*dabs(zcent)
     &         *dabs(s12(nspenum(j))))*area


200       continue


185       format(2e14.7)
          write(3,185) time(jj), torque

90     continue

        close(1)
        close(2)
        close(3)


        write(*,*) 'Program Completed'
        end
```

## B.2 Program to Calculate External Work Done by Torque

```
C************************************************************
C*****  Program for External Work (version work.f)
C*****  This program calculates the external work done by the
C*****  torque using the file of torque data (created by the
C*****  program torque) and the load curve data.
C*****  Program last modified 07/14/96
C************************************************************

      implicit double precision (a-h,o-z)
      parameter (max=400000)
      character file1*25
      dimension time(max),tpoint(100),wpoint(100),womega(max),
     &   torq(max),atorq(max), work(max)

10    format(a)

      write(*,*) 'Enter the name of the torque data file'
      read(*,10) file1
      write(*,*)
      write(*,*) 'Enter the number of time states spanned by'
      write(*,*) 'the load curve'
      read(*,*) nstates
      write(*,*)
      write(*,*) 'Load curve data for angular velocity'
      write(*,*)
      write(*,*) 'Enter the number of points needed to define'
      write(*,*) 'the load curve'
      read(*,*) npoint
      write(*,*)

25    format(a17,i3)

      do 30 i = 1, npoint
        write(*,*)
        write(*,25) 'Enter time point ',i
        read(*,*) tpoint(i)
        write(*,*)
        write(*,25) 'Enter load point ',i
        read(*,*) wpoint(i)
30    continue

      open (1,file=file1,status='old')
      do 35 i = 1, nstates
        read(1,*) time(i), vtorq
        torq(i) = dabs(vtorq)
 35   continue

      close(1)


      do 40 i = 1, nstates
        if (time(i).lt.tpoint(2)) then
        womega(i) = ((wpoint(2)-wpoint(1))/(tpoint(2)-tpoint(1)))
     &  *(time(i)-tpoint(2)) + wpoint(2)
        elseif ((time(i).gt.tpoint(2)).and.(time(i).lt.tpoint(3)))
```

```
     &    then
        womega(i) = wpoint(3)
        elseif (time(i).gt.tpoint(3)) then
        womega(i) = ((wpoint(4)-wpoint(3))/(tpoint(4)-tpoint(3)))
     &  *(time(i)-tpoint(4)) + wpoint(4)
        endif

40    continue

       womega(nstates) = wpoint(4)

C         open(2,file='loadc.dat',status='unknown')

C        do 50 i = 1, nstates
C         write(2,45) time(i), womega(i)
C50      continue

45      format(2e16.7)

C        close(2)

      atorq(1) = (torq(2)+torq(1))/2.
      atorq(nstates) = (torq(nstates)+torq(nstates-1))/2.

      do 60 i = 2, (nstates-1)
        atorq(i) = (torq(i+1)+torq(i)+torq(i-1))/3.
60    continue

      open (3,file='work.dat',status='unknown')

65    format(2e16.7)

      wsum = 0.

      do 70 i = 1, nstates
        wsum = wsum + atorq(i)*womega(i)*time(i)
        work(i) = wsum
        write(3,65) time(i), work(i)
70    continue

        stop
        end
```

## B.3 Program to Calculate External Work Done by Axial Pressure

```
C*************************************************************
C*****  Program exwp (version exwp1.f)
C*****  Program reads in the mesh, reads in time history data, then
C*****  reads the element numbers that will be used to calculate
C*****  external work.  It finds the 4 surface nodes for each element,
C*****  then updates the coordinates, calculates centroids, then
C*****  calculates surface area, and finally calculates the
C*****  external work done by the pressure load.  It needs a file
C*****  with the load curve for pressure called load_curve.
C*****  Program last modified 07/14/96
C*************************************************************


      implicit double precision (a-h,o-z)
      parameter (max=400000)
      dimension nodnum(max),xcord(max),ycord(max),zcord(max),
     &  nelnum(max),node1(max),node2(max),node3(max),node4(max),
     &  node5(max),node6(max),node7(max),node8(max),
     &  time(500),xdisp(max),ptime(5000),pload(5000),
     &  ydisp(max),zdisp(max),nhelem(max),nhnode(max),
     &  nspnnum(max),nspenum(max),dtime(2)

      character file1a*25,file1b*25,netfile*25,file2*25,file3*25,
     &  com*1,star*1


10    format(a)
      write(*,*) 'Enter the file name containing the element'
      write(*,*) 'numbers'
      read(*,10) file2
      write(*,*)
      write(*,*) 'Enter the pressure amplitude'
      read(*,*) pamp
      write(*,*)
      write(*,*) 'Enter the PATRAN neutral file name'
      read(*,10) file1c
      write(*,*)

C***** Read the element numbers in the special element file

      open (1,file=file2,status='old')
      read(1,*) numspe

45    format(i8)

      do 50 i = 1, numspe
       read(1,*) nspenum(i)
50    continue

      close(1)

C***** Read in the pressure load curve data and scale it with the
C***** given amplitude.

       open(1,file='load_curve',status='old')
       read(1,*) numpres
       do 1000 i = 1, numpres
```

```
          read(1,*) ptime(i),press
          pload(i) = press*pamp
1000    continue
        close(1)
        pload(numpres+1) = 0.d0

C****** check the number of states and delta t *******************

      open (1,file='disp_th',status='old',form='unformatted')


      read(1) nte,nst

      do 400 jj = 1, 2

        read(1) dtime(jj)

        do  390 k = 1, nte
         read(1) nh,var1,var2,var3

 390    continue

 400    continue

        delt = dtime(2) - dtime(1)

 410    format(a10,i5,a7)
 420    format(a11,e14.7)
        close(1)

        write(*,*)
        write(*,410) 'There are ',nst,' states'
        write(*,*)
        write(*,420) 'Delta t is ',delt
        write(*,*)
        write(*,*) 'How many states do you want to include'
        read(*,*) nrstate

C*****  read in all the node numbers, their coordinates, the element
C*****  numbers and the connectivities from the PATRAN neutral file.


      open(1,file=netfile,status='old')

      do 20 i = 1,2
        read(1,*)
20    continue

      read(1,*) nada,nada,nada,nada,numnod,numel
      read(1,*)

      do 30 i = 1, numnod
        read(1,*) nada,nodnum(i)
        read(1,*) xcord(i),ycord(i),zcord(i)
        read(1,*)
30    continue

      do 40 i = 1, numel
        read(1,*) nada, nelnum(i)
        read(1,*)
```

```
        read(1,*) node1(i),node2(i),node3(i),node4(i),
     &            node5(i),node6(i),node7(i),node8(i)
40    continue

      close(1)

C******************************************************


C***** Read the time history data  *****************

      open (2,file='disp_th',status='old',form='unformatted')
      open (3,file='exwp.dat',status='unknown')


      read(2) ntotn,nstates

      do 90 jj = 1, nrstate

        read(2) time(jj)

        do 110 k = 1, ntotn
         read(2) nhnode(k),xdisp(nhnode(k)),
     &           ydisp(nhnode(k)),zdisp(nhnode(k))

 110    continue


         exwsum = 0.d0
         do 200 j = 1, numspe

C***** Update the coordinates of the 4 surface nodes with
C***** the nodal displacements for each time step

           xcord1 = xcord(node1(nspenum(j)))
     &      + xdisp(node1(nspenum(j)))
           xcord2 = xcord(node2(nspenum(j)))
     &      + xdisp(node2(nspenum(j)))
           xcord3 = xcord(node3(nspenum(j)))
     &      + xdisp(node3(nspenum(j)))
           xcord4 = xcord(node4(nspenum(j)))
     &      + xdisp(node4(nspenum(j)))

           ycord1 = ycord(node1(nspenum(j)))
     &      + ydisp(node1(nspenum(j)))
           ycord2 = ycord(node2(nspenum(j)))
     &      + ydisp(node2(nspenum(j)))
           ycord3 = ycord(node3(nspenum(j)))
     &      + ydisp(node3(nspenum(j)))
           ycord4 = ycord(node4(nspenum(j)))
     &      + ydisp(node4(nspenum(j)))

           zcord1 = zcord(node1(nspenum(j)))
     &      + zdisp(node1(nspenum(j)))
           zcord2 = zcord(node2(nspenum(j)))
     &      + zdisp(node2(nspenum(j)))
```

```
          zcord3 = zcord(node3(nspenum(j)))
     &     + zdisp(node3(nspenum(j)))
          zcord4 = zcord(node4(nspenum(j)))
     &     + zdisp(node4(nspenum(j)))


C***** Calculate the surface area using four nodes

          xx1=(-xcord1+xcord2+xcord3-xcord4)/2.d0
          xx2=(-ycord1+ycord2+ycord3-ycord4)/2.d0
          xx3=(-zcord1+zcord2+zcord3-zcord4)/2.d0
          xy1=(-xcord1-xcord2+xcord3+xcord4)/2.d0
          xy2=(-ycord1-ycord2+ycord3+ycord4)/2.d0
          xy3=(-zcord1-zcord2+zcord3+zcord4)/2.d0
          ee = xx1*xx1 + xx2*xx2 + xx3*xx3
          ff = xx1*xy1 + xx2*xy2 + xx3*xy3
          gg = xy1*xy1 + xy2*xy2 + xy3*xy3

          area=dsqrt(ee*gg-ff*ff)

C***** Calculate the centroid of the element
C**** xcent is not really correct because you would need all
C**** 8 nodes and divide by 8

          xcent = (xcord1 + xcord2 + xcord3 + xcord4)/4.d0
          ycent = (ycord1 + ycord2 + ycord3 + ycord4)/4.d0
          zcent = (zcord1 + zcord2 + zcord3 + zcord4)/4.d0

C**** For each element multiply the pressure times the element area,
C**** then put 1/4 at the nodes. Then multiply this by the nodal
C**** displacements and sum over all the nodes on the face.

      force = ((pload(jj-1)+pload(jj)+pload(jj+1))/3.d0)
     &        *area/4.d0

      exwsum = exwsum
     &    + force*xdisp(node1(nspenum(j)))
     &    + force*xdisp(node2(nspenum(j)))
     &    + force*xdisp(node3(nspenum(j)))
     &    + force*xdisp(node4(nspenum(j)))

200      continue


185      format(2e14.7)
         write(3,185) time(jj), exwsum

90    continue

      close(2)
      close(3)


      write(*,*) 'Program Completed'
      stop
      end
```

## B.4 Program to Calculate Shear Band Speed

```
C**************************************************************
C*****  PROGRAM:  bandv1.f
C*****  This program calculates band speed for notch elements
C*****  using the modified binary DYNA3D files.
C*****  last modified  07/23/96
C**************************************************************

      implicit double precision (a-h,o-z)
      parameter (max=400000,nm=25)
      dimension nodnum(max),xcord(max),ycord(max),zcord(max),
     &  nelnum(max),node1(max),node2(max),node3(max),node4(max),
     &  node5(max),node6(max),node7(max),node8(max),
     &  time(1000),s11(max),s22(max),s33(max),s12(max),s13(max),
     &  s23(max),xdisp(max),ydisp(max),zdisp(max),
     &  nhelem(max),nhnode(max),nspenum(max),nspnnum(max),
     &  xcent(max),ycent(max),zcent(max),
     &  ps(max),temp(max),seff(max),ndone(nm),pbelow(nm),pabove(nm),
     &  pbtime(nm),patime(nm),prad(nm),ptime(nm),pcoord(nm),
     &  pbandv(nm),smax(nm),smtime(nm),s9time(nm),s90(nm),slast(nm),
     &  tlast(nm),sabove(nm),satime(nm),s9rad(nm),sbelow(nm),
     &  sbtime(nm),smrad(nm),smbandv(nm),s9bandv(nm),ndone2(nm),
     &  smcoord(nm),s9coord(nm),nhit(nm)

      character file1a*25,file1b*25,netfile*25,file2*25,file3*25,
     &  com*1,star*1,dstar*2,filen*25


10    format(a)
      write(*,*)
      write(*,*) 'This program calculates band speed.'
      write(*,*)
      write(*,*) 'Enter the file name containing the element'
      write(*,*) 'numbers'
      read(*,10) file2
      write(*,*)
      write(*,*) 'Enter the file name of node numbers'
      read(*,10) filen
      write(*,*)
      write(*,*) 'Enter the PATRAN neutral file name'
      read(*,10) netfile
      write(*,*)
      write(*,*) 'Enter the critical plastic strain value'
      read(*,*) pscrit
      write(*,*)
      write(*,*) 'Enter the offset for the number of states'
      write(*,*) 'i.e. -2 -1 0 +1 +2 etc.'
      read(*,*) noffset

C*****  read in all the node numbers, their coordinates, the element
C*****  numbers and the connectivities from the PATRAN neutral file.

      open(1,file=netfile,status='old')

      do 20 i = 1,2
        read(1,*)
20    continue
```

```
      read(1,*) nada,nada,nada,nada,numnod,numel
      read(1,*)

      do 30 i = 1, numnod
        read(1,*) nada,nodnum(i)
        read(1,*) xcord(i),ycord(i),zcord(i)
        read(1,*)
30    continue

      do 40 i = 1, numel
        read(1,*) nada, nelnum(i)
        read(1,*)
        read(1,*) node1(i),node2(i),node3(i),node4(i),
     &            node5(i),node6(i),node7(i),node8(i)
40    continue

      close(1)

C***********************************************************

C***** Read the element numbers in the special element file
C***** and read the node numbers in the special node file

      open (1,file=file2,status='old')
      read(1,*) numspe

45    format(i8)

      do 50 i = 1, numspe
       read(1,*) nspenum(i)
50    continue

      close(1)

      open(1,file=filen,status='old')
      read(1,*) numspn

      do 55 i = 1, numspn
        read(1,*) nspnnum(i)
55    continue

      close(1)


C***** Read the time history data corresponding to the element
C***** numbers in the special element

      open(1,file='stress_th',status='old',form='unformatted')
c      open(2,file='disp_th',status='old',form='unformatted')
      open(3,file='ps_temp',status='old',form='unformatted')

444   format(a7,25i10)


      read(1) ntote,nstates
c      read(2) ntotn,nstates
      read(3) ntote,nstates
```

```
      do 90 jall = 1, (nstates+noffset)

        read(1) time(jall)
c        read(2) time(jall)
        read(3) time(jall)

        do 100 k = 1, ntote
         read(1) nhelem(k),s11(nhelem(k)),s22(nhelem(k)),
     &   s33(nhelem(k)),s12(nhelem(k)),s23(nhelem(k)),
     &   s13(nhelem(k)),seff(nhelem(k))

         read(3) nhyep, ps(nhelem(k)),temp(nhelem(k))

100     continue

        do 110 k = 1, ntotn

         read(2) nhnode(k),xdisp(nhnode(k)),
     &   ydisp(nhnode(k)),zdisp(nhnode(k))

110     continue


        do 120 j = 1, numspe


C***** Update the coordinates of the 8 nodes with
C***** the nodal displacements for each time step

           xcord1 = xcord(node1(nspenum(j)))
     &      + xdisp(node1(nspenum(j)))
           xcord2 = xcord(node2(nspenum(j)))
     &      + xdisp(node2(nspenum(j)))
           xcord3 = xcord(node3(nspenum(j)))
     &      + xdisp(node3(nspenum(j)))
           xcord4 = xcord(node4(nspenum(j)))
     &      + xdisp(node4(nspenum(j)))
           xcord5 = xcord(node5(nspenum(j)))
     &      + xdisp(node5(nspenum(j)))
           xcord6 = xcord(node6(nspenum(j)))
     &      + xdisp(node6(nspenum(j)))
           xcord7 = xcord(node7(nspenum(j)))
     &      + xdisp(node7(nspenum(j)))
           xcord8 = xcord(node8(nspenum(j)))
     &      + xdisp(node8(nspenum(j)))


           ycord1 = ycord(node1(nspenum(j)))
     &      + ydisp(node1(nspenum(j)))
           ycord2 = ycord(node2(nspenum(j)))
     &      + ydisp(node2(nspenum(j)))
           ycord3 = ycord(node3(nspenum(j)))
     &      + ydisp(node3(nspenum(j)))
           ycord4 = ycord(node4(nspenum(j)))
     &      + ydisp(node4(nspenum(j)))
```

```
         ycord5 = ycord(node5(nspenum(j)))
     &     + ydisp(node5(nspenum(j)))
         ycord6 = ycord(node6(nspenum(j)))
     &     + ydisp(node6(nspenum(j)))
         ycord7 = ycord(node7(nspenum(j)))
     &     + ydisp(node7(nspenum(j)))
         ycord8 = ycord(node8(nspenum(j)))
     &     + ydisp(node8(nspenum(j)))

         zcord1 = zcord(node1(nspenum(j)))
     &     + zdisp(node1(nspenum(j)))
         zcord2 = zcord(node2(nspenum(j)))
     &     + zdisp(node2(nspenum(j)))
         zcord3 = zcord(node3(nspenum(j)))
     &     + zdisp(node3(nspenum(j)))
         zcord4 = zcord(node4(nspenum(j)))
     &     + zdisp(node4(nspenum(j)))
         zcord5 = zcord(node5(nspenum(j)))
     &     + zdisp(node5(nspenum(j)))
         zcord6 = zcord(node6(nspenum(j)))
     &     + zdisp(node6(nspenum(j)))
         zcord7 = zcord(node7(nspenum(j)))
     &     + zdisp(node7(nspenum(j)))
         zcord8 = zcord(node8(nspenum(j)))
     &     + zdisp(node8(nspenum(j)))


C***** Calculate the centroid of the element

         xcent(nspenum(j))
     &          = (xcord1 + xcord2 + xcord3 + xcord4
     &          + xcord5 + xcord6 + xcord7 + xcord8)/8.d0
         ycent(nspenum(j))
     &          = (ycord1 + ycord2 + ycord3 + ycord4
     &          + ycord5 + ycord6 + ycord7 + ycord8)/8.d0
         zcent(nspenum(j))
     &          = (zcord1 + zcord2 + zcord3 + zcord4
     &          + zcord5 + zcord6 + zcord7 + zcord8)/8.d0


120       continue


c*****  Find the plastic strains and the corresponding times above
c*****  and below the critical value, for each of the specified
c*****  elements.  They will be used for interpolation later on.

      do 1000 m2 = 1, numspe
       if (ndone(m2).eq.1) goto 1000
         if (ps(nspenum(m2)).lt.pscrit) then
            pbelow(m2) = ps(nspenum(m2))
            pbtime(m2) = time(jall)
```

```
          prad(m2) = dsqrt(ycent(nspenum(m2))**2
     &                     + zcent(nspenum(m2))**2)
         else
          ndone(m2) = 1
          pabove(m2) = ps(nspenum(m2))
          patime(m2) = time(jall)
        endif

1000  continue


      do 3010 m2 = 1, numspe
        diff = dabs(seff(nspenum(m2))-smax(m2))
        if ((seff(nspenum(m2)).ge.smax(m2)).or.
     &               (diff.lt.(1.d-10))) then
          smax(m2) = seff(nspenum(m2))
          smtime(m2) = time(jall)
          smrad(m2) = dsqrt(ycent(nspenum(m2))**2
     &                     + zcent(nspenum(m2))**2)
        endif


3010  continue


      do 3020 m2 = 1, numspe
        if ((time(jall).gt.smtime(m2)).and.(nhit(m2).eq.0)) then
          s90(m2) = .9d0*smax(m2)
          nhit(m2) = 1
        endif
3020  continue


      do 3040 m2 = 1, numspe
        if (nhit(m2).eq.1) then
          if (ndone2(m2).eq.1) goto 3040
          if (seff(nspenum(m2)).lt.s90(m2)) goto 3050
          slast(m2) = seff(nspenum(m2))
          tlast(m2) = time(jall)
3050      continue
          if (seff(nspenum(m2)).lt.s90(m2)) then
            sabove(m2) = seff(nspenum(m2))
            satime(m2) = time(jall)
            s9rad(m2) = dsqrt(ycent(nspenum(m2))**2
     &                     + zcent(nspenum(m2))**2)
            ndone2(m2) = 1
            sbelow(m2) = slast(m2)
            sbtime(m2) = tlast(m2)

          endif
         endif

3040  continue


90    continue
```

```
C***** end the loop over time states. *********************

c***** We checked all time states and have the values needed for
c***** interpolation for the plastic strain, and a drop to 90%
c***** maximun effective stress, so interpolate now.


        do 1020 i = 1, numspe
          ptime(i) = ((pscrit-pabove(i))*pbtime(i)
     &      + (pbelow(i)-pscrit)*patime(i)) /
     &        ((pbelow(i)-pscrit) + (pscrit-pabove(i)))
           s9time(i) = ((s90(i)-sabove(i))*sbtime(i)
     &      + (sbelow(i)-s90(i))*satime(i)) /
     &        ((sbelow(i)-s90(i)) + (s90(i)-sabove(i)))

1020    continue

c**** Now we have the time that ps critical, maximum effective stress,
c**** and 90% maximun effective stress reaches each of the
c**** specified elements, so we can now calculate band speed.

      do 1080 i = 1, (numspe-1)
        pbandv(numspe-i) = (prad(numspe+1-i)-prad(numspe-i))
     &             / (ptime(numspe-i)-ptime(numspe+1-i))
        pcoord(numspe-i) = (prad(numspe+1-i)-prad(numspe-i))/2.d0
     &                   + prad(numspe-i)
        smbandv(numspe-i) = (smrad(numspe+1-i)-smrad(numspe-i))
     &            / (smtime(numspe-i)-smtime(numspe+1-i))
        smcoord(numspe-i) = (smrad(numspe+1-i)-smrad(numspe-i))/2.d0
     &                   + smrad(numspe-i)
        s9bandv(numspe-i) = (s9rad(numspe+1-i)-s9rad(numspe-i))
     &            / (s9time(numspe-i)-s9time(numspe+1-i))
        s9coord(numspe-i) = (s9rad(numspe+1-i)-s9rad(numspe-i))/2.d0
     &                   + s9rad(numspe-i)

1080  continue

c***** Now print out the band speed and the radial coordinate between
c***** the two elements used for calculating the speed.

      open(7,file='psbv.dat',status='unknown')
      open(8,file='meffsbv.dat',status='unknown')
      open(9,file='effs90bv.dat',status='unknown')


c***** multiply by speed by 10^4 to get into m/s, and distance by
c***** 10^1 to get it in millimeters.

      do 2040 i = 1, (numspe-1)

         write(7,2045) (pcoord(numspe-i)*1.d1),
     &             (pbandv(numspe-i)*1.d4), ptime(numspe+1-i)
         write(8,2045) (smcoord(numspe-i)*1.d1),
     &             (smbandv(numspe-i)*1.d4), smtime(numspe+1-i)
         write(9,2045) (s9coord(numspe-i)*1.d1),
     &             (s9bandv(numspe-i)*1.d4), s9time(numspe+1-i)
```

```
2040  continue

       write(7,2046) ptime(1)
       write(8,2046) smtime(1)
       write(9,2046) s9time(1)

2045  format(3e16.9)
2046  format(32x,e16.9)

135     format(50e14.7)

       close(1)
c       close(2)
       close(3)
       close(7)
       close(8)
       close(9)

       write(*,*) 'Program Completed'
       stop
       end
```

## B.5 Program to Extract Data from the Modified DYNA3D Output Files

```
C*******************************************************************
C
C    Program postx.f     Last modified 4/8/96
C
C*******************************************************************
C***** Program to extract time history data from the binary files
C***** output by modified DYNA3D.  It prints out the data in columns
C***** with time in the first column and the data for each element or
C***** node in successive columns.  It prints temperature for the
C***** requested elements to the file temp.dat, plastic strain to
C***** ps.dat, effective stress to effs.dat, and displacements for the
C***** requested nodes to xdisp.dat, ydisp.dat, and zdisp.dat.
C*******************************************************************

      implicit double precision(a-h,o-z)

      write(*,*)
      write(*,*) 'This program extracts time history data from the'
      write(*,*) 'binary files written by modified DYNA3D.'
      write(*,*)
 10   write(*,*) 'Enter a 1 if you want nodal results'
      write(*,*) 'Enter a 2 if you want element results'
      read(*,*) nne
      write(*,*)

      if (nne.eq.1) then
 20     write(*,*) 'Enter a 1 to read node numbers from a file'
        write(*,*) 'Enter a 2 to input node numbers manually'
        read(*,*) nfm
        if ((nfm.eq.1).or.(nfm.eq.2)) then
          call readwri(nne,nfm)
            else
              goto 20
        endif

      elseif (nne.eq.2) then
 30     write(*,*) 'Enter a 1 to read element numbers from a file'
        write(*,*) 'Enter a 2 to input element numbers manually'
        read(*,*) nfm

        if ((nfm.eq.1).or.(nfm.eq.2)) then
           call readwri(nne,nfm)
           else
            goto 30
        endif
      else
         goto 10

      endif

        stop
        end !End main program.

      subroutine readwri(nne,nfm)
      implicit double precision (a-h,o-z)
      parameter (max=400000)
      character filein*25,nfile*25,com*1,dstar*2
```

```
      dimension ps(max),temp(max),seff(max),time(1000),nelnum(max),
     &          nreq(max),dx(max),dy(max),dz(max),nodnum(max),
     &          mnode(max),xcord(max),ycord(max),zcord(max)


      if (nfm.eq.2) then

      if (nne.eq.1) then
      write(*,*) 'How many nodes do you want data for'
      else
      write(*,*) 'How many elements do you want data for'
      endif
      read(*,*) ndata
      write(*,*)

2     format(a31,i3)
      do 5 i = 1, ndata
        if (nne.eq.1) then
        write(*,2) 'Enter requested node number ',i
        else
        write(*,2) 'Enter requested element number ',i
        endif
        read(*,*) nreq(i)
5     continue

      endif

444   format(a)

      if (nfm.eq.1) then

      if (nne.eq.1) then
      write(*,*) 'Enter the file name containing the node numbers'
      read(*,444) filein
      elseif (nne.eq.2) then
      write(*,*) 'Enter the file name containing the element numbers'
      read(*,444) filein
      endif

      open(1,file=filein,status='old')

      read(1,*) ndata

      do 7 i = 1, ndata
        read(1,*) nreq(i)
 7    continue

      close(1)

      endif


 8    format(a10,500i14)

      if (nne.eq.1) then

      open(1,file='disp_th',status='old',form='unformatted')
      open(2,file='xdisp.dat',status='unknown')
      open(3,file='ydisp.dat',status='unknown')
```

```
      open(4,file='zdisp.dat',status='unknown')
      open(7,file='ydmat.dat',status='unknown')

      write(2,8) 'time      ',(nreq(i),i=1,ndata)
      write(3,8) 'time      ',(nreq(i),i=1,ndata)
      write(4,8) 'time      ',(nreq(i),i=1,ndata)

      read(1) numnb,nstates
      do 10 i = 1, nstates
        read(1) time(i)

        do 20 j = 1, numnb
          read(1) nodnum(j),dx(j),dy(j),dz(j)
 20     continue

 25   format(e10.4,500e14.7)
 26   format(3e14.7)

         write(2,25) time(i), (dx(nreq(k)),k=1,ndata)
         write(3,25) time(i), (dy(nreq(k)),k=1,ndata)
         write(4,25) time(i), (dz(nreq(k)),k=1,ndata)

         do 27 k = 1, ndata
           write(7,26) xcord(nreq(k)),time(i),dy(nreq(k))
27       continue

10    continue

      elseif (nne.eq.2) then

      open(1,file='stress_th',status='old',form='unformatted')
      open(2,file='ps_temp',status='old',form='unformatted')
      open(3,file='ps.dat',status='unknown')
      open(4,file='temp.dat',status='unknown')
      open(7,file='effs.dat',status='unknown')

      write(3,8) 'time      ',(nreq(i),i=1,ndata)
      write(4,8) 'time      ',(nreq(i),i=1,ndata)
      write(7,8) 'time      ',(nreq(i),i=1,ndata)

      read(1) numeb,nstates
      read(2) numeb,nstates
      do 30 i = 1, nstates
        read(1) time(i)
        read(2) time(i)

        do 40 j = 1, numeb
          read(1) nelnum(j),sxx,syy,szz,sxy,syz,szx,seff(j)
          read(2) nelnum(j),ps(j),temp(j)
 40     continue


         write(3,35) time(i), (ps(nreq(k)),k=1,ndata)
         write(4,35) time(i), (temp(nreq(k)),k=1,ndata)
         write(7,35) time(i), (seff(nreq(k)),k=1,ndata)


 30   continue
```

```
35    format(e10.4,500e14.7)
      endif

      return
      end !End of subroutine readwri.
```

## B.6 Program to Convert Modified DYNA3D Binary Output Files to Ascii

```
c******************************************************************
c***** Program rbin (version rbin2.f)
c***** This program reads the binary files created by the modified
c***** DYNA3D program, and writes out the ascii equivalents.
c***** There is a choice to increase or decrease the number of states
c***** because sometimes there is an off by 1 error in the files, or
c***** DYNA3D crashed and the files are incomplete.
c***** Last Modified 6/30/96
c*******************************************************************

      implicit double precision(a-h,o-z)
      character yornd*1,yorns*1,yornp*1

      write(*,*)
      write(*,*) 'This program converts the DYNA3D binary files'
      write(*,*) 'to their ascii equivalents'
      write(*,*)
      write(*,*) 'Do you want to read disp_th ?'
      write(*,*) 'Enter y or n'
      read(*,5) yornd
      write(*,*)
      write(*,*) 'Do you want to read stress_th ?'
      write(*,*) 'Enter y or n'
      read(*,5) yorns
      write(*,*)
      write(*,*) 'Do you want to read ps_temp ?'
      write(*,*) 'Enter y or n'
      read(*,5) yornp
      write(*,*)
5     format(a)

      if (yorns.eq.'y') then
      open(1,file='stress_th',status='old',form='unformatted')
      open(4,file='stress_read',status='unknown')
      read(1) numeb,nstates
      endif

      if (yornd.eq.'y') then
      open(2,file='disp_th',status='old',form='unformatted')
      open(7,file='disp_read',status='unknown')
      read(2) numnb,nstates
      endif

      if (yornp.eq.'y') then
      open(3,file='ps_temp',status='old',form='unformatted')
      open(8,file='pst_read',status='unknown')
      read(3) numeb,nstates
      endif

      write(*,*) 'There are ',nstates,' states'
      write(*,*) 'Do you want to adjust the number?'
      write(*,*) 'Enter -2 -1 0 1 2 etc'
      read(*,*) nextra

      if (yorns.eq.'y') write(4,*) numeb,(nstates+nextra)
```

```
      if (yornd.eq.'y') write(7,*) numnb,(nstates+nextra)
      if (yornp.eq.'y') write(8,*) numeb,(nstates+nextra)

      do 10 i = 1, (nstates+nextra)

      if (yorns.eq.'y') then
        read(1) time
        write(4,*) time
      endif

      if (yornd.eq.'y') then
        read(2) time
        write(7,*) time
      endif

      if (yornp.eq.'y') then
       read(3) time
       write(8,*) time
      endif

        if (yorns.eq.'y') then

        do 20 j = 1, numeb
          read(1) nelnum,sxx,syy,szz,sxy,syz,szx,seff
          write(4,400) nelnum,sxx,syy,szz,sxy,syz,szx,seff
20      continue

        endif


        if (yornp.eq.'y') then

         do 25 j = 1, numeb
          read(3) nelnum,ps,temp
          write(8,410) nelnum,ps,temp
 25     continue

        endif

        if (yornd.eq.'y') then
        do 40 jj = 1,numnb
            read(2) nodenum,d1,d2,d3
            write(7,420) nodenum,d1,d2,d3
 40     continue
        endif

 10   continue

400   format(i8,7e13.6)
410   format(i8,2e14.7)
420   format(i8,3e14.7)

 30   format(i5,7e12.4)
      stop
      end
```

## B.7 Program to Convert Stress and Displacement to Cylindrical Coordinates

```
C***************************************************************
C*****  PROGRAM: xxtorr.f
C*****  This program converts x,y,z, stresses to r,theta and z
C*****  stresses in cylindrical coordinates
C*****  last modified 4/17/96
C***************************************************************

      implicit double precision (a-h,o-z)
      parameter (max=400000)
      dimension nodnum(max),xcord(max),ycord(max),zcord(max),
     &  nelnum(max),node1(max),node2(max),node3(max),node4(max),
     &  node5(max),node6(max),node7(max),node8(max),
     &  time(500),s11(max),s22(max),s33(max),s12(max),s13(max),
     &  s23(max),srr(max),srt(max),srz(max),stt(max),stz(max),
     &  szz(max),xdisp(max),ydisp(max),zdisp(max),
     &  nhelem(max),nhnode(max),nspenum(max),nspnnum(max),
     &  ur(max),ut(max),uz(max),xcent(max),ycent(max),zcent(max)

      character file1a*25,file1b*25,netfile*25,file2*25,file3*25,
     &  com*1,star*1,dstar*2,filen*25


10    format(a)
      write(*,*) 'Enter the file name containing the element'
      write(*,*) 'numbers'
      read(*,10) file2
      write(*,*)
      write(*,*) 'Enter the file name of node numbers for'
      write(*,*) 'conversion of displacement and velocity'
      read(*,10) filen
      write(*,*) 'Enter the PATRAN neutral file name'
      read(*,10) netfile
      write(*,*)


C*****  read in all the node numbers, their coordinates, the element
C*****  numbers and the connectivities.

      do 20 i = 1,2
        read(1,*)
20    continue

      read(1,*) nada,nada,nada,nada,numnod,numel
      read(1,*)

      do 30 i = 1, numnod
        read(1,*) nada,nodnum(i)
        read(1,*) xcord(i),ycord(i),zcord(i)
        read(1,*)
30    continue

      do 40 i = 1, numel
        read(1,*) nada, nelnum(i)
        read(1,*)
        read(1,*) node1(i),node2(i),node3(i),node4(i),
     &            node5(i),node6(i),node7(i),node8(i)
40    continue
```

```
      close(1)

C**********************************************************

C***** Read the element numbers in the special element file
C***** and read the node numbers in the special node file

      open (1,file=file2,status='old')
      read(1,*) numspe

45    format(i8)

      do 50 i = 1, numspe
       read(1,*) nspenum(i)
50    continue

      close(1)

      open(1,file=filen,status='old')
      read(1,*) numspn

      do 55 i = 1, numspn
        read(1,*) nspnnum(i)
55    continue

      close(1)


C***** Read the time history data corresponding to the element
C***** numbers in the special element file - sigma13

      open(1,file='stress_th',status='old',form='unformatted')
      open(2,file='disp_th',status='old',form='unformatted')
      open(11,file='stress_rr',status='unknown')
      open(12,file='stress_tt',status='unknown')
      open(13,file='stress_zz',status='unknown')
      open(14,file='stress_rt',status='unknown')
      open(15,file='stress_rz',status='unknown')
      open(16,file='stress_tz',status='unknown')
      open(17,file='rad_dist',status='unknown')
      open(4,file='stress_xyz',status='unknown')
      open(7,file='disp_rtz',status='unknown')
      open(8,file='disp_xyz',status='unknown')

      write(11,444) 'time  ',(nspenum(m),m=1,numspe)
      write(12,444) 'time  ',(nspenum(m),m=1,numspe)
      write(13,444) 'time  ',(nspenum(m),m=1,numspe)
      write(14,444) 'time  ',(nspenum(m),m=1,numspe)
      write(15,444) 'time  ',(nspenum(m),m=1,numspe)
      write(16,444) 'time  ',(nspenum(m),m=1,numspe)

444   format(a7,25i10)


      read(1) ntote,nstates
      read(2) ntotn,nstates
```

```
      write(*,*)
      write(*,*) 'There are ',nstates,' states'
      write(*,*) 'How many states do you want'
      read(*,*) nrstate
      write(*,*) 'What state interval do you want'
      read(*,*) nsint

      do 90 jall = 1, nrstate

        read(1) time(jall)
        read(2) time(jall)

        do 100 k = 1, ntote
         read(1) nhelem(k),s11(nhelem(k)),s22(nhelem(k)),
     &   s33(nhelem(k)),s12(nhelem(k)),s23(nhelem(k)),
     &   s13(nhelem(k)),seff

         if (dabs(s11(nhelem(k))).lt.(1.e-30))
     &     s11(nhelem(k)) = 0.
         if (dabs(s22(nhelem(k))).lt.(1.e-30))
     &     s11(nhelem(k)) = 0.
         if (dabs(s33(nhelem(k))).lt.(1.e-30))
     &     s11(nhelem(k)) = 0.
         if (dabs(s12(nhelem(k))).lt.(1.e-30))
     &     s11(nhelem(k)) = 0.
         if (dabs(s13(nhelem(k))).lt.(1.e-30))
     &     s11(nhelem(k)) = 0.
         if (dabs(s23(nhelem(k))).lt.(1.e-30))
     &     s11(nhelem(k)) = 0.


100     continue

        do 110 k = 1, ntotn

         read(2) nhnode(k),xdisp(nhnode(k)),
     &   ydisp(nhnode(k)),zdisp(nhnode(k))

         if (dabs(xdisp(nhnode(k))).lt.(1.e-30))
     &     xdisp(nhnode(k)) = 0.

         if (dabs(ydisp(nhnode(k))).lt.(1.e-30))
     &     ydisp(nhnode(k)) = 0.

         if (dabs(zdisp(nhnode(k))).lt.(1.e-30))
     &     zdisp(nhnode(k)) = 0.

110     continue


        if (mod(jall,nsint).ne.0) goto 90

        do 120 j = 1, numspe


C***** Update the coordinates of the 8 nodes with
C***** the nodal displacements for each time step
```

```
      xcord1 = xcord(node1(nspenum(j)))
&      + xdisp(node1(nspenum(j)))
      xcord2 = xcord(node2(nspenum(j)))
&      + xdisp(node2(nspenum(j)))
      xcord3 = xcord(node3(nspenum(j)))
&      + xdisp(node3(nspenum(j)))
      xcord4 = xcord(node4(nspenum(j)))
&      + xdisp(node4(nspenum(j)))
      xcord5 = xcord(node5(nspenum(j)))
&      + xdisp(node5(nspenum(j)))
      xcord6 = xcord(node6(nspenum(j)))
&      + xdisp(node6(nspenum(j)))
      xcord7 = xcord(node7(nspenum(j)))
&      + xdisp(node7(nspenum(j)))
      xcord8 = xcord(node8(nspenum(j)))
&      + xdisp(node8(nspenum(j)))

      ycord1 = ycord(node1(nspenum(j)))
&      + ydisp(node1(nspenum(j)))
      ycord2 = ycord(node2(nspenum(j)))
&      + ydisp(node2(nspenum(j)))
      ycord3 = ycord(node3(nspenum(j)))
&      + ydisp(node3(nspenum(j)))
      ycord4 = ycord(node4(nspenum(j)))
&      + ydisp(node4(nspenum(j)))
      ycord5 = ycord(node5(nspenum(j)))
&      + ydisp(node5(nspenum(j)))
      ycord6 = ycord(node6(nspenum(j)))
&      + ydisp(node6(nspenum(j)))
      ycord7 = ycord(node7(nspenum(j)))
&      + ydisp(node7(nspenum(j)))
      ycord8 = ycord(node8(nspenum(j)))
&      + ydisp(node8(nspenum(j)))

      zcord1 = zcord(node1(nspenum(j)))
&      + zdisp(node1(nspenum(j)))
      zcord2 = zcord(node2(nspenum(j)))
&      + zdisp(node2(nspenum(j)))
      zcord3 = zcord(node3(nspenum(j)))
&      + zdisp(node3(nspenum(j)))
      zcord4 = zcord(node4(nspenum(j)))
&      + zdisp(node4(nspenum(j)))
      zcord5 = zcord(node5(nspenum(j)))
&      + zdisp(node5(nspenum(j)))
      zcord6 = zcord(node6(nspenum(j)))
&      + zdisp(node6(nspenum(j)))
      zcord7 = zcord(node7(nspenum(j)))
&      + zdisp(node7(nspenum(j)))
      zcord8 = zcord(node8(nspenum(j)))
&      + zdisp(node8(nspenum(j)))
```

```
C***** Calculate the centroid of the element

           xcent(nspenum(j))
     &            = (xcord1 + xcord2 + xcord3 + xcord4
     &            + xcord5 + xcord6 + xcord7 + xcord8)/8.
           ycent(nspenum(j))
     &            = (ycord1 + ycord2 + ycord3 + ycord4
     &            + ycord5 + ycord6 + ycord7 + ycord8)/8.
           zcent(nspenum(j))
     &            = (zcord1 + zcord2 + zcord3 + zcord4
     &            + zcord5 + zcord6 + zcord7 + zcord8)/8.


C***** Calculate cos and sin using element centroids ********

      cth = zcent(nspenum(j))/dsqrt(ycent(nspenum(j))**2
     &  +zcent(nspenum(j))**2)
      sth = ycent(nspenum(j))/dsqrt(ycent(nspenum(j))**2
     &  +zcent(nspenum(j))**2)

      if ((ycent(nspenum(j)).eq.0.).and.
     &    (zcent(nspenum(j)).eq.0.)) then
           cth = 0.
           sth = 0.
        endif

C**********************************************************

      srr(nspenum(j)) = s11(nspenum(j))*cth*cth
     &          + s22(nspenum(j))*sth*sth
     &          + 2.*s12(nspenum(j))*sth*cth

      srt(nspenum(j)) = (s22(nspenum(j))-s11(nspenum(j)))*sth*cth
     &  + s12(nspenum(j))*cth*cth - s12(nspenum(j))*sth*sth

      srz(nspenum(j)) = s13(nspenum(j))*cth + s23(nspenum(j))*sth

      stt(nspenum(j)) = s11(nspenum(j))*sth*sth
     &  + s22(nspenum(j))*cth*cth - 2.*s12(nspenum(j))*sth*cth

      stz(nspenum(j)) = s23(nspenum(j))*cth - s13(nspenum(j))*sth

      szz(nspenum(j)) = s33(nspenum(j))

      write(4,125) time(jall),
     & s11(nspenum(j)),s22(nspenum(j)),s33(nspenum(j)),
     & s12(nspenum(j)),s13(nspenum(j)),s23(nspenum(j))

125    format(25e15.7)

120      continue

      write(17,205) time(jall),(dsqrt(ycent(nspenum(kz))**2
     &  + zcent(nspenum(kz))**2),kz=1,numspe)
      write(11,200) time(jall),(srr(nspenum(kz)),kz=1,numspe)
      write(12,200) time(jall),(stt(nspenum(kz)),kz=1,numspe)
```

```
      write(13,200) time(jall),(szz(nspenum(kz)),kz=1,numspe)
      write(14,200) time(jall),(srt(nspenum(kz)),kz=1,numspe)
      write(15,200) time(jall),(srz(nspenum(kz)),kz=1,numspe)
      write(16,200) time(jall),(stz(nspenum(kz)),kz=1,numspe)

200   format(e14.7,25e14.7)
205   format(e14.7,25e14.7)

      do 130 m = 1, numspn


C***** Calculate cos and sin using element centroids *******

      cth = zcord(nspnnum(m))/dsqrt(ycord(nspnnum(m))
     &  *ycord(nspnnum(m))+zcord(nspnnum(m))*zcord(nspnnum(m)))
      sth = ycord(nspnnum(m))/dsqrt(ycord(nspnnum(m))
     &  *ycord(nspnnum(m))+zcord(nspnnum(m))*zcord(nspnnum(m)))

C*********************************************************

      if ((ycord(nspnnum(m)).eq.0.).and.(zcord(nspnnum(m)).eq.0.))
     &  then
           cth = 0.
           sth = 0.
        endif

        ur(nspnnum(m)) = xdisp(nspnnum(m))*cth +
     &    ydisp(nspnnum(m))*sth

        ut(nspnnum(m)) = -1.*xdisp(nspnnum(m))*sth +
     &    ydisp(nspnnum(m))*cth

        uz(nspnnum(m)) = zdisp(nspnnum(m))

        write(7,135) time(jall),ycord(nspnnum(m)),
     &   zcord(nspnnum(m)),
     &   ur(nspnnum(m)),ut(nspnnum(m)),uz(nspnnum(m))

        write(8,135) time(jall),ycord(nspnnum(m)),
     &   zcord(nspnnum(m)),
     &   xdisp(nspnnum(m)),ydisp(nspnnum(m)),zdisp(nspnnum(m))

130    continue

90    continue

135     format(6e14.7)

       close(1)
       close(2)
       close(3)
       close(4)

       write(*,*) 'Program Completed'
       stop
       end
```