EG&G Idaho, Inc.

FORM EG&G-398
(Rev. 11-79)

THIS DOCUMENT CONTAINS
POOR QUALITY PAGES

INTERIM REPORT

Accession No. ______

Report No. EGG-CDAP-5173

Contract Program or Project Title: Fuel Behavior Model Development

Subject of this Document: FRIDA: A Computer Code for the Dynamic Longitudinal Response of Fuel Rods Subjected to Thermal Transients

Type of Document: Interim Report

Author(s): J. N. Singh
M. P. Bohn
G. P. Engelman
S. O. Peck

Date of Document: June 1980

Responsible NRC Individual and NRC Office or Division: G. P. Marino

This document was prepared primarily for preliminary or internal use. It has not received full review and approval. Since there may be substantive changes, this document should not be considered final.

EG&G Idaho, Inc.
Idaho Falls, Idaho 83415

Prepared for the
U.S. Nuclear Regulatory Commission
Washington, D.C.
Under DOE Contract No. DE-AC07-76ID01570
NRC FIN No. A6050

INTERIM REPORT

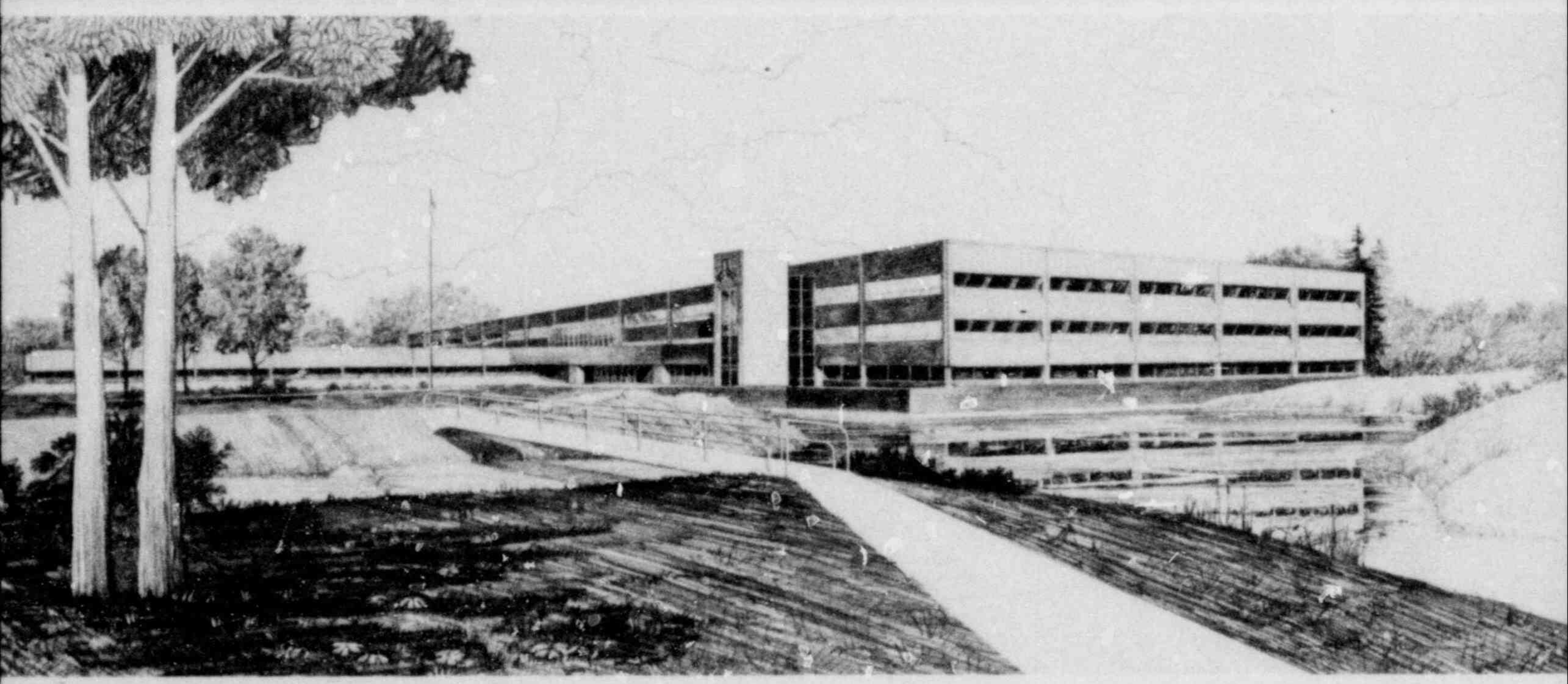
EGG-CDAP-5173
June 1980

FRIDA: A COMPUTER CODE FOR THE DYNAMIC LONGITUDINAL RESPONSE OF FUEL RODS SUBJECTED TO THERMAL TRANSIENTS

J. N. Singh
M. P. Bohn
G. P. Engelman
S. O. Peck

U.S. Department of Energy

Idaho Operations Office • Idaho National Engineering Laboratory



This is an informal report intended for use as a preliminary or working document

Prepared for the U. S. Nuclear Regulatory Commission Under
DOE Contract No. DE-AC07-76ID01570
NRC FIN No. A6050

EG&G Idaho

EG&G Idaho, Inc.

FORM EG&G-398
(Rev. 11-79)

INTERIM REPORT

Accession No. ____________

Report No. EGG-CDAP-5173

Contract Program or Project Title: Fuel Behavior Model Development

Subject of this Document: FRIDA: A Computer Code for the Dynamic Longitudinal Response of Fuel Rods Subjected to Thermal Transients

Type of Document: Interim Report

Author(s): J. N. Singh
M. P. Bohn
G. P. Engelman
S. O. Peck

Date of Document: June 1980

Responsible NRC Individual and NRC Office or Division: G. P. Marino

> This document was prepared primarily for preliminary or internal use. It has not received full review and approval. Since there may be substantive changes, this document should not be considered final.

EG&G Idaho, Inc.
Idaho Falls, Idaho 83415

Prepared for the
U.S. Nuclear Regulatory Commission
Washington, D.C.
Under DOE Contract No. DE-AC07-76ID01570
NRC FIN No. A6050

INTERIM REPORT

FRIDA: A COMPUTER CODE FOR THE DYNAMIC LONGITUDINAL RESPONSE OF FUEL RODS SUBJECTED TO THERMAL TRANSIENTS

J. N. Singh
M. P. Bohn
G. P. Engelman
S. O. Peck

June 1980

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

FRIDA is a computer code that calculates the dynamic longitudinal response of fuel rods subjected to thermal transients. The fuel rod is modeled as a simple beam, fixed at one end and in series with an effective cladding stiffness spring and plenum spring at the other end. The model is discretized through a finite element approach and the resulting system of differential equations solved with a fourth-order Runge-Kutta integration scheme.

I. Introduction

During a hypothetical reactivity insertion accident (RIA), fuel rods will be subjected to thermal loads sufficiently severe that the inertial response of the rods must be considered in a complete mechanical analysis of their behavior. A computer program, FRIDA (Fuel Reactivity Insertion Dynamic Analyses), has been written that computes the dynamic axial loads in fuel rods subjected to such transients. FRIDA compliments the FRAP-T[1] (Fuel Rod Analysis Program - Transient) computer code in that FRAP-T fuel temperature histories are typically used as input to the program, although arbitrary histories are permissible.

FRIDA was originally coded and documented in 1975[2]. Results from the code were sometimes questionable. In particular, the calculated axial loads seemed unreasonably large. Since then, a number of significant improvements have been made, specifically: (a) a new driver program, (b) arbitrary noding, (c) more frequent updating of the stiffness matrix, (d) incorporating the latest MATPRO[3] material properties, (e) an input smoothing function, and (f) calculation of the natural frequencies of the model. The purpose of this report is to demonstrate the effects of these changes on the program calculations.

Section II of this report will discuss the assumed fuel rod model, Section III the solution procedure, and Section IV the general code description. Section V will present the results of a benchmark problem comparison, while Section VI will present the results of an RIA sample case. Finally, Section VII discusses limitations and recommendations for use of FRIDA. Appendix A is a user input manual and Appendix B is a code listing.

II. General Model Description

1. Fuel Rod Model

The fuel rod is modeled as a one-dimensional (longitudinal) beam. One end is assumed to be rigidly fixed while the other end is modeled by two springs in series, one for the plenum spring and one representing an effective cladding stiffness. The stiffness of the beam itself is assumed to be entirely due to the fuel. In other words, no friction is assumed between the fuel and the cladding.

The temperature of the beam model is input by the user as discrete temperature-time histories. The axial temperature distribution and history may be arbitrary although most frequently the histories are derived from FRAP-T analyses. FRAP-T calculates both axial and radial fuel temperatue profiles; however, for FRIDA only one temperature is used at any axial location. Often a volume averaged radial temperature is used to describe the fuel rod temperature.

A simple model for estimating the cladding hoop stress is incorporated in the code. It is assumed that during the high fuel temperatures of an RIA (frequently exceeding the fuel melt temperature) the fuel stress distribution will be very nearly hydrostatic. Thus the axial stress resulting from the calculated axial loads will approximate the fuel-cladding interface pressure. Thin wall tube theory then leads directly to an estimate of the cladding hoop stress.

2. Equation of Motion

The equation of motion for longitudinal vibration of the modeled fuel rod, see Figure 1, may be derived as shown.

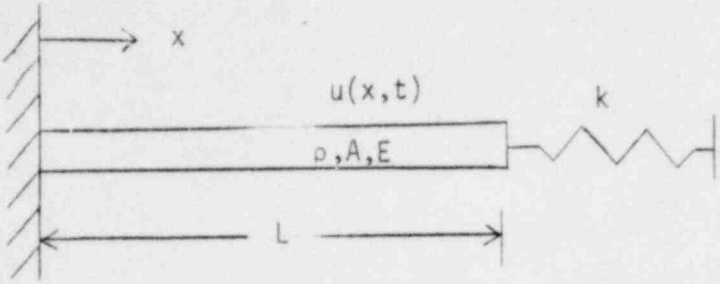


Fig. 1 Beam/spring model of the fuel rod

where

$u(x,t)$	=	longitudinal displacement as a function of distance (x) and time (t)
ρ	=	mass density
A	=	cross sectional area of beam
E	=	Young's modulus of elasticity
L	=	length of the beam
k	=	effective spring constant of the cladding and plenum
ρ,A,E	=	may be functions of temperature, T, and hence of x and t

The thermal heat conduction solution for the beam is assumed to be uncoupled from the mechanical solution. That is, the change in the temperature field arising from work being done on the beam is negligible compared with the change in temperature from the nuclear fuel internal heat generation source. Thus, the thermal solution may be arrived at first and used as input for the mechanical solution without the need for iteration.

The kinetic energy of an element of the beam is proportional to the square of the velocity of the elemental displacement. The kinetic energy of the whole beam is found from the integral over the length, or

$$K_{BEAM} = \int_0^L \frac{1}{2} \rho A \left(\frac{\partial u}{\partial t}\right)^2 dx \quad (1)$$

The work done per unit volume in straining an elastic element to a uniaxial strain ε_x subject to a uniaxial stress σ_x is

$$\frac{\text{work}}{\text{volume}} = \int_0^{\varepsilon_x} \sigma_x d\varepsilon_x \quad (2)$$

Hooke's Law for elastic deformation with thermal loads is given by

$$\varepsilon_x = \frac{1}{E}(\sigma_x - \nu(\sigma_y + \sigma_z)) + \alpha T \quad (3)$$

where

ν = Poisson's ratio

α = linear coefficient of thermal expansion.

The beam is assumed free to contract laterally so that σ_y and σ_z are zero. The work per unit volume can then be solved for by substituting (3) into (2).

$$\frac{\text{work}}{\text{volume}} = \int_0^{\varepsilon_x} (E\,\varepsilon_x - E\alpha T)\, d\varepsilon_x$$

$$= \frac{1}{2} E\,\varepsilon_x^2 - E\alpha T\varepsilon_x$$

Noting that $\varepsilon_x = \frac{\partial u}{\partial x}$ and integrating over the beam, the potential energy associated with the beam is given by

$$V_{BEAM} = \int_0^L \left[\frac{1}{2} AE \left(\frac{\partial u}{\partial x}\right)^2 - AE\alpha T \left(\frac{\partial u}{\partial x}\right)\right] dx \quad (4)$$

The potential energy associated with the spring is given by

$$V_{spring} = \frac{1}{2} K\, u(L,t)^2 \tag{5}$$

The beam is assumed to have a distributed viscous damping, that is, the local damping is proportional to the local velocity of a given element. The dissipative work increment for the whole beam associated with an increment of displacement is thus given by the following, where δ is used to indicate an increment (or as it will be termed later, a variation).

$$\delta W = \int_0^L -b \left(\frac{\partial u}{\partial t}\right) \delta u \, dx \tag{6}$$

where

δW = work increment

b = viscous damping coefficient.

Hamilton's Principle (Reference 4) for continuous systems states that the variational indicator must vanish between times t_1 and t_2, where the variational indicator (VI) is determined as

$$VI = \int_{t_1}^{t_2} (\delta K - \delta V + \delta W)\, dt \tag{7}$$

where K, V, and W refer in general to the kinetic energy, potential energy, and work (force times displacement) defined specifically for this system earlier. The variational indicator is defined to be zero at times t_1 and t_2. Substituting from equations (1), (4), (5), and (6) gives the variational indicator for the fuel rod model.

$$VI = \int_{t_1}^{t_2} \left[\delta \left\{ \int_0^L \left[\frac{1}{2} \rho A \left(\frac{\partial u}{\partial t}\right)^2 - \frac{1}{2} AE \left(\frac{\partial u}{\partial x}\right)^2 + AE\, \alpha T \frac{\partial u}{\partial x}\right] dx - \frac{1}{2} k\, u(L,t)^2 \right\} - \int_0^L b \left(\frac{\partial u}{\partial t}\right) \delta u dx \right] dt \tag{8}$$

Performing the indicated variations, and noting that $\delta\left(\frac{\partial u}{\partial t}\right) = \frac{\partial(\delta u)}{\partial t}$ and $\delta\left(\frac{\partial u}{\partial x}\right) = \frac{\partial(\delta u)}{\partial x}$ the first term of the VI may be integrated by parts timewise and the second and third terms integrated by parts spacewise, Making use of the geometric boundary condition that $u(0,t) = 0$ and $\delta u(0,t) = 0$ and that the VI vanish by definition at t_1 and t_2, the VI becomes

$$VI = \int_{t_1}^{t_2} \left\{ \int_0^L \left[-\rho A \frac{\partial^2 u}{\partial t^2} + \frac{\partial}{\partial x}\left(AE \frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial x}(AE\alpha T) - b\frac{\partial u}{\partial t}\right] \delta u dx + \left[-ku(L,t) - AE \frac{\partial u}{\partial x}(L,t) + AE\alpha T\right] \delta u(L,t) \right\} dt \qquad (9)$$

Also, the mass density, ρ, was assumed to not be a function of time. For the variational indicator to vanish for a geometrically admissible variation $\delta u(x,t)$ implies that

$$-\rho A \frac{\partial^2 u}{\partial t^2} + \frac{\partial}{\partial x}\left(AE \frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial x}(AE\,\alpha T) - b\frac{\partial u}{\partial t} = 0 \quad 0<x<L \qquad (10)$$

and

$$-ku - AE\frac{\partial u}{\partial x} + AE\,\alpha T = 0 \quad x=L \qquad (11)$$

Equation (10) is the equation of motion for the model and Equation (11) is the "natural" boundary condition at x=L.

III. Solution Procedure

1. Finite Element Formulation

The equation of motion (10) contains, in general, partial derivatives of the cross sectional area, Young's modulus, coefficient of thermal expansion, and temperature as well as displacement. For this problem the partial derivative of the mass density with respect to time is assumed to be zero as previously stated. It is also reasonable to assume that time derivatives of the area are zero. However, the other terms vary significantly with temperature. Since temperature is allowed to be an arbitrary function of time and location, these terms cannot be discarded. Thus the solution is analytically intractible and a numerical approach is necessary.

A finite element formulation was chosen to describe the beam behavior. The beam was divided into a number of elements, each described by two nodal locations and an element length. Each element was allowed a linearly distributed mass and linearly distributed displacement function (a constant strain element in static finite element problems). Using kinetic and potential energy arguments for the discretized elements, submatrices were constructed for each element by equilibrating nodal forces with elemental displacements and distributed forces. The submatrices were assembled into an overall matrix representation of the beam, given as

$$[M] \frac{\partial^2}{\partial t^2} \{u\} + [B] \frac{\partial}{\partial t} \{u\} + [K] \{u\} + \{Q\} = 0 \qquad (12)$$

where

[M] = mass matrix

[B] = viscous damping matrix

[K] = stiffness matrix

{Q} = nodal thermal forces

{u} = nodal displacements.

The spacewise variation is handled through this formulation by evaluating material properties and temperatures at discrete locations and the timewise variation through numerical integration over time (dividing the history into load steps). The stiffness matrix is updated at every time step while the mass matrix is assumed constant. The damping matrix is assumed proportional to the mass matrix, but the proportionality factor is assumed to be a function of Young's modulus of elasticity. Hence, the damping matrix is updated every time step as well.

2. Solution Technique

The finite element representation of the beam generally results in a system of second order differential equations to be solved. Equation (12) may be rearranged to solve for the nodal accelerations as functions of the property matrices, nodal velocities and nodal displacements. This initial value problem is then solved given assumed initial displacements and velocities by a fourth-order Runge-Kutta integration scheme. The explicit Runge-Kutta scheme was chosen because of its good convergence qualities and fast running time.

IV. General Code Description

1. Flow Chart

The flow ch. rt for the program is shown in Table 1. Functional activities are described in rectangular boxes and the subroutine names associated with those activities are shown in ovals. Logical decisions are shown in diamond shaped figures.

2. Input Requirements

The input to FRIDA consists of four basic parts: (1) a general physical description of the beam and spring model, (2) a tabular listing of the temperature history as a function of time and location, (3) modeling parameters such as the number of nodes, assumed damping ratio, and so forth, and (4) numerical integration scheme controls. A user input manual giving the specific input details and job control language is given in Appendix A.

3. Output Results

At each axial node FRIDA calculates and plots the displacement, velocity, acceleration, load, stress, strain, strain rate and an estimate of cladding hoop stress. The output is given as a time history. In addition, the natural frequency and modeshape of the model are calculated.

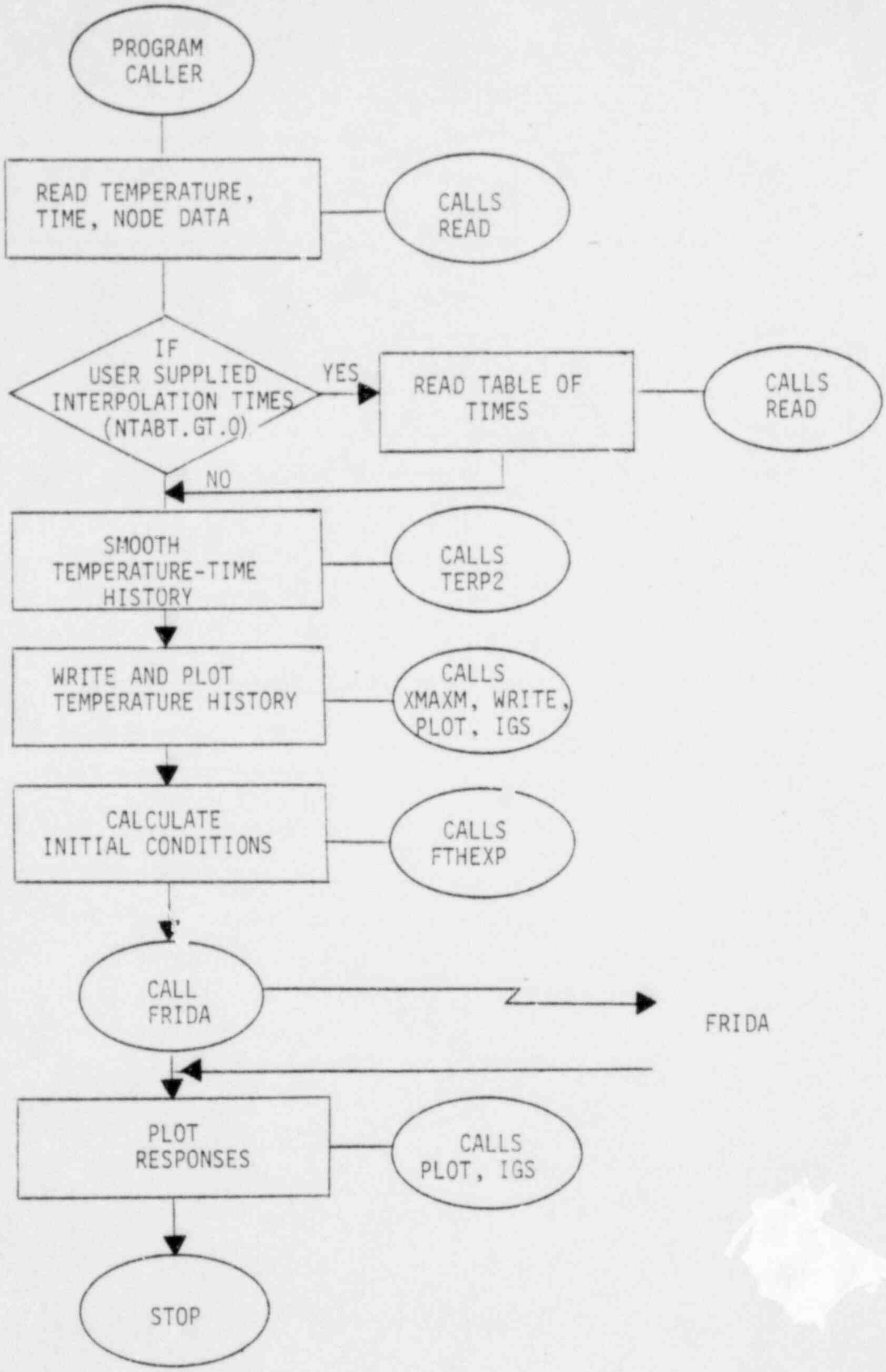


TABLE 1 FRIDA FLOW CHART

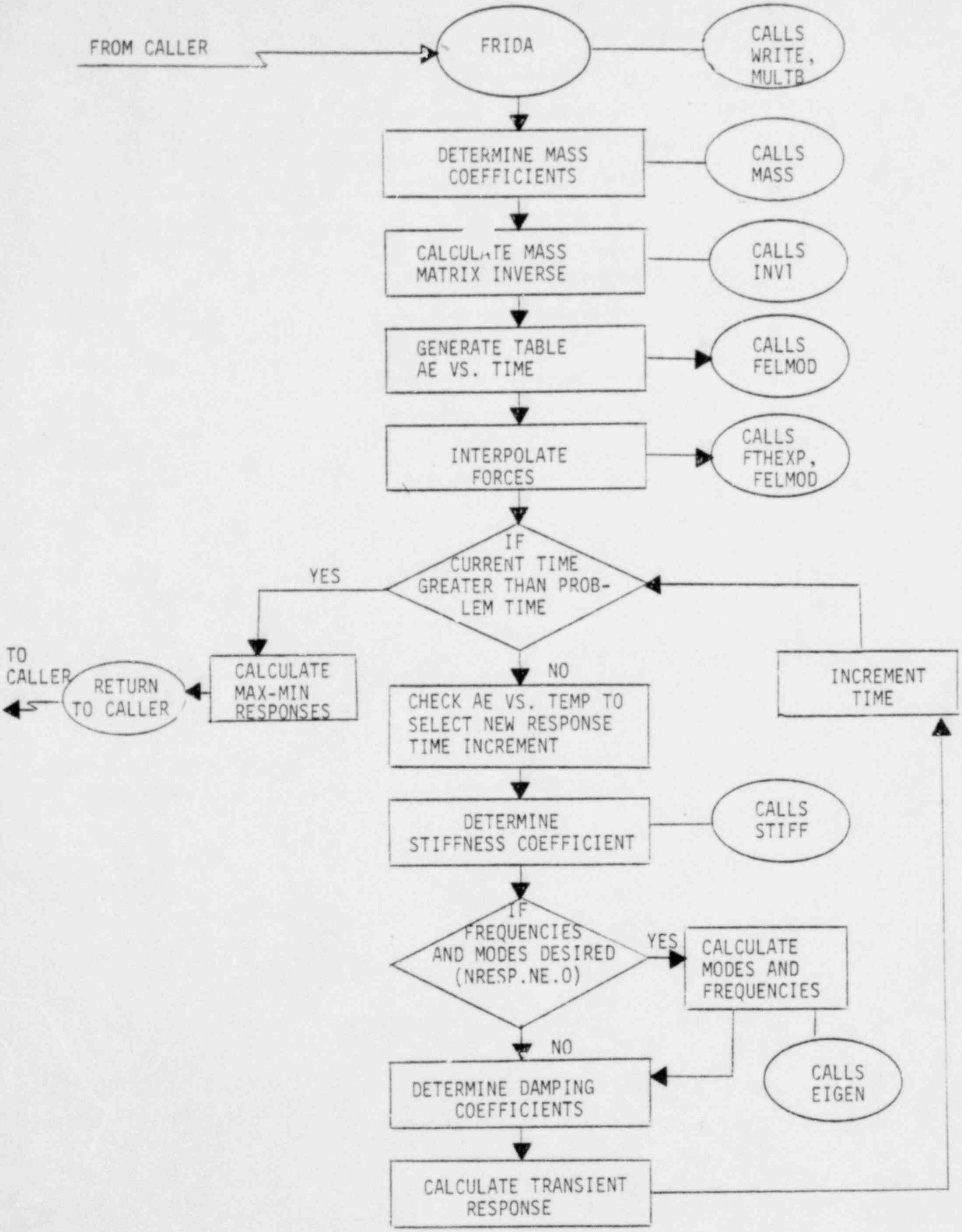


TABLE 1 (Contd.) FRIDA FLOW CHART

V. Benchmark Problem

In order to estimate the accuracy of the numerical approximations in FRIDA, a benchmark problem was devised whereby sufficient simplifying assumptions were made that the equation of motion could be solved analytically. In particular, the beam was assumed to be subjected to an axially uniform step jump in temperature at time zero with no damping and no spring. Equation (10) reduces to

$$\frac{\partial^2 u}{\partial x^2} - \frac{\rho}{E} \frac{\partial^2 u}{\partial t^2} = 0 \qquad (13)$$

under these assumptions. The boundary conditions become

$$u(0,t) = 0$$

$$\frac{\partial u(L,t)}{\partial x} = \alpha T \qquad (14)$$

and the initial conditions become

$$u(x,0) = 0$$

$$\frac{\partial u}{\partial t}(x,0) = 0. \qquad (15)$$

The solution of this problem is given as

$$u(x,t) = \alpha Tx - \frac{8\alpha TL}{\pi^2} \sum_{i=1,3,5...}^{\infty} \frac{(-1)^{\left(\frac{i-1}{2}\right)}}{i^2} \sin \frac{i\pi x}{2L} \cos \frac{i\pi t}{2L}\sqrt{\frac{E}{\rho}} \qquad (16)$$

where T is the change in temperature of the beam.

The analytical solution for the displacement of the midpoint of the beam is shown in Figure 2 and can be compared with the FRIDA solution shown in Figure 3. The particular beam modeled was 3.76 m long, 1.22

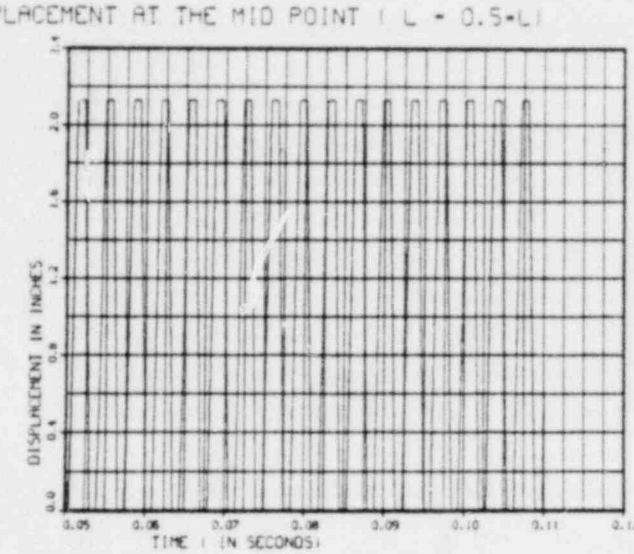


Fig. 2 Analytical solution for the displacement at the midpoint of the benchmark problem.

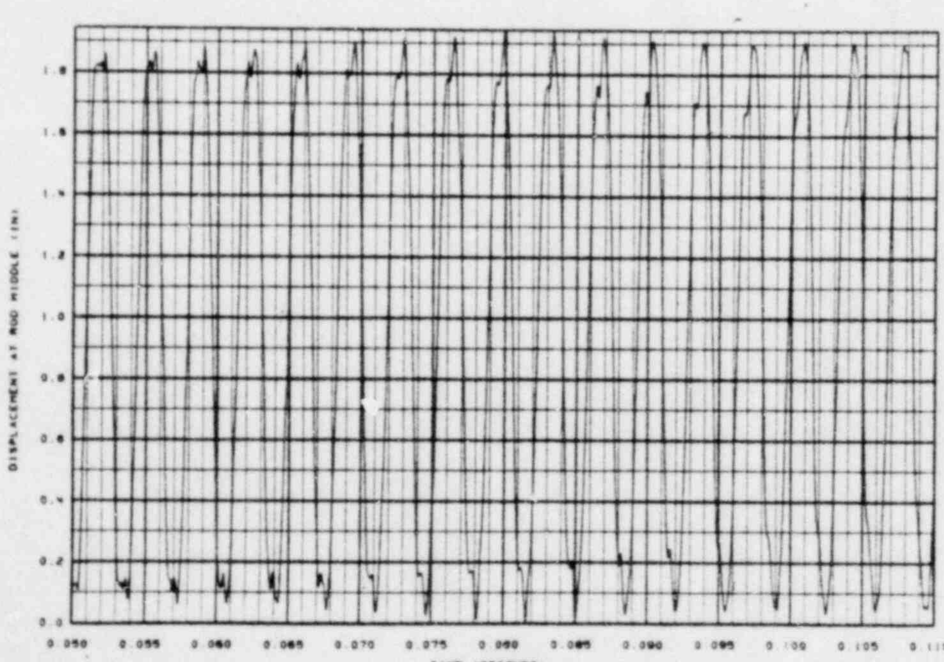


Fig. 3 FRIDA solution for the displacement at the midpoint of the benchmark problem.

cm in diameter, and had a density of 10.42 g/cm^3. The temperature jump was 1000 K from a reference of 572 K. The FRIDA model used 20 nodes. Both solutions predicted a frequency of about 280 Hz, and the analytical solution had about an 8% greater maximum displacement than the FRIDA solution at the midpoint. This is a reasonable agreement between the two solutions.

VI. Sample Problem

RIA 1-2 is the second of five planned tests in the Reactivity Initiated Accident (RIA) Test Series 1. It was performed in the Power Burst Facility (PBF) which is operated by the Thermal Fuels Behavior Program at Idaho National Engineering Laboratory (INEL) as a part of the Nuclear Regulatory Commission's (NRC) Reactor Safety Research Program. The objectives of the RIA Series 1 tests are to determine the thresholds, modes, and consequences of fuel rod failures under RIA conditions as a function of energy deposition, irradiated history, and fuel design. Test RIA 1-2 was comprised of four, individual, pre-irradiated fuel rods, each surrounded by a separate flow shroud. The specific objectives of Test RIA 1-2 were to (a) characterize the response of preirradiated fuel rods during an RIA event conducted at BWR hot-start conditions for an axial peak pellet surface energy of 200 cal/g UO_2, and (b) evaluate the effect of internal rod pressure on preirradiated fuel rod response during an RIA event.

The purpose of this FRIDA sample problem is to investigate the mechanical response of the RIA 1-2 fuel rods. The temperature input to the code was taken from a FRAP-T5 analysis of the RIA. The temperatures were chosen as the hottest calculated fuel temperatures at each axial node. Figure 4 shows the input temperature history as a dotted line and the smoothed input as a solid line. The base data for the analysis are listed in Table 2.

Figure 5 shows the axial acceleration of the rod midpoint as a function of time. The frequency of vibration is about 1300 Hz. The lowest natural frequency of the rod at the initial conditions is 1295 Hz so the rod is primarily vibrating in its first mode. Figure 6 shows the axial displacement and Figure 7 the axial strain at the midpoint as functions of time. The axial load, Figure 8, is seen to have a maximum value of approximately 178 N. However, the maximum axial load occurs at the bottom of the rod and has a value of 215 N occurring at 0.0396

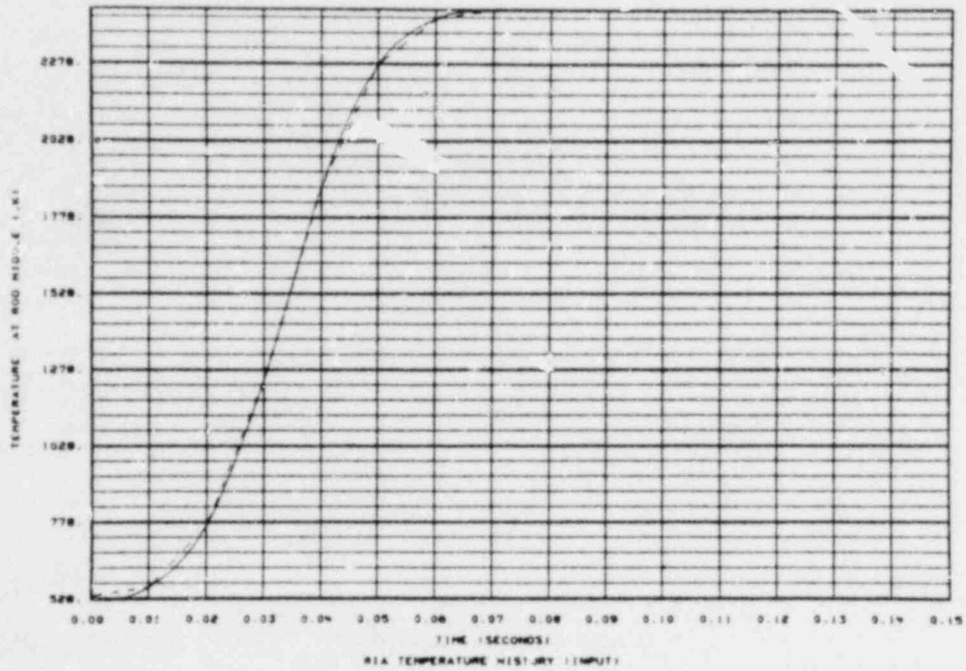


Fig. 4 RIA sample problem input temperature history (dotted line is the actual input, solid line is the smoothed input).

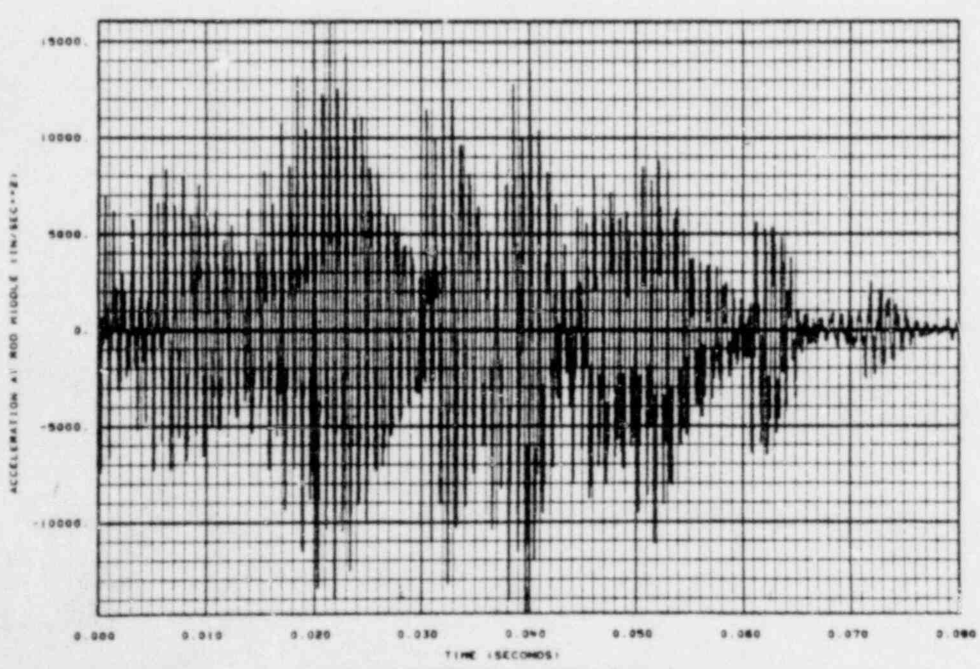


Fig. 5 RIA sample problem axial acceleration at the rod midpoint.

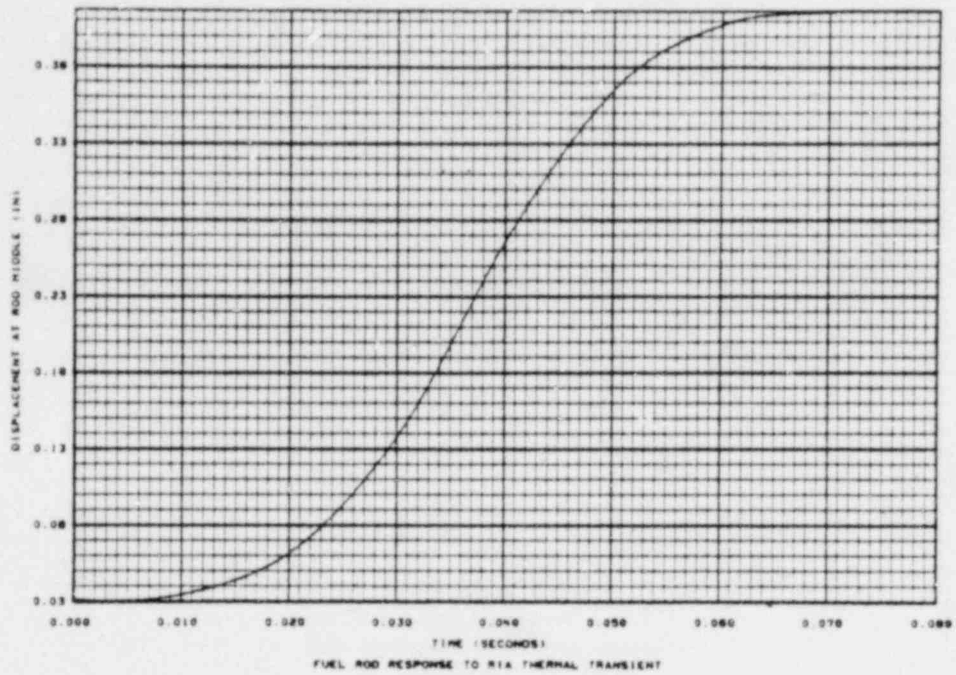


Fig. 6 RIA sample problem axial displacement at the rod midpoint.

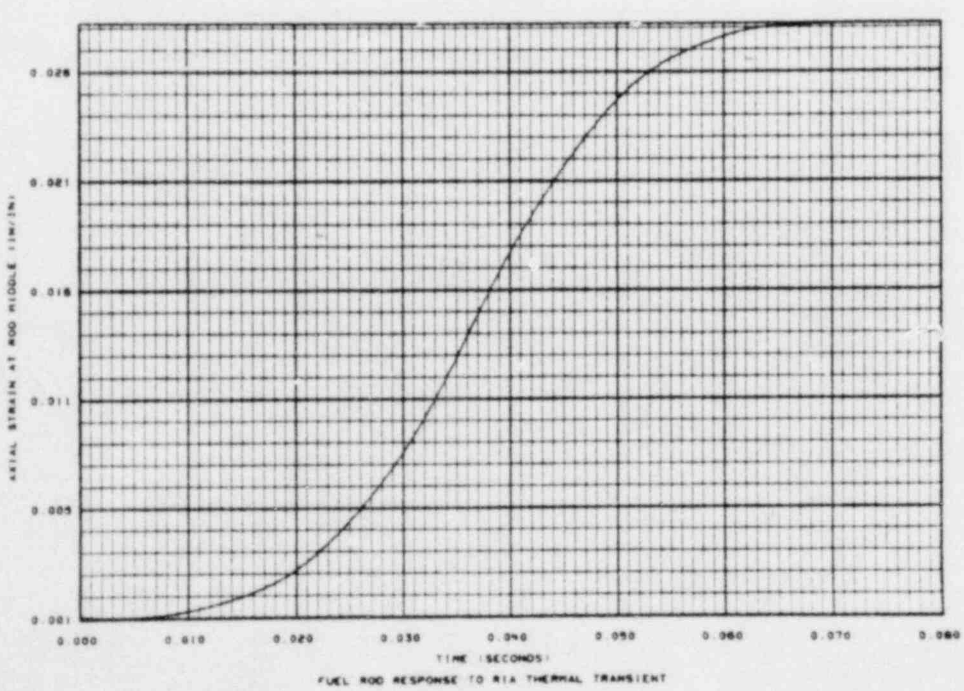


Fig. 7 RIA sample problem axial strain at the rod midpoint.

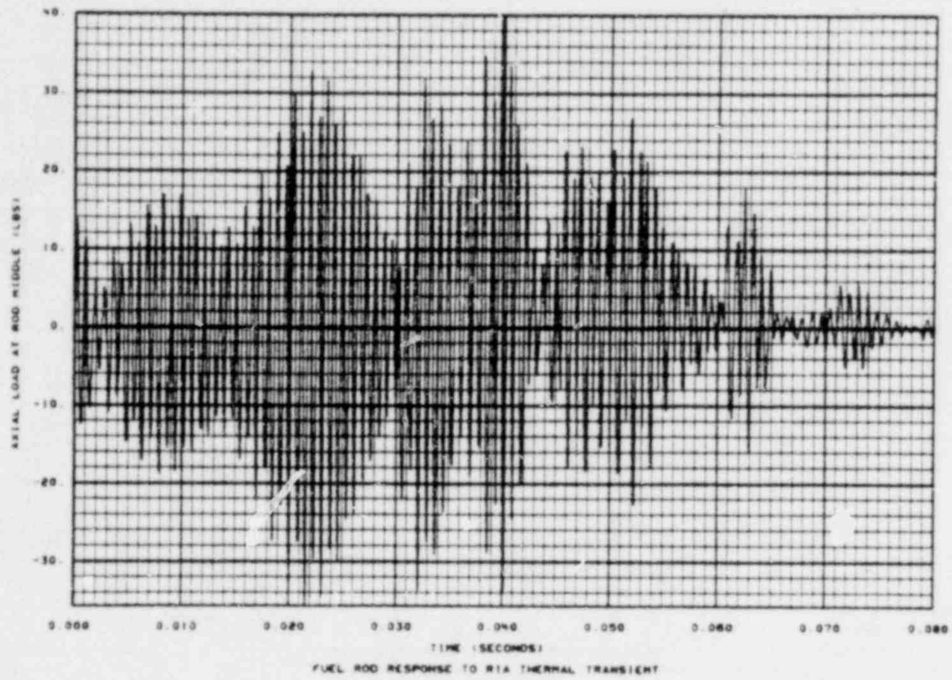


Fig. 8 RIA sample problem axial load at the rod midpoint.

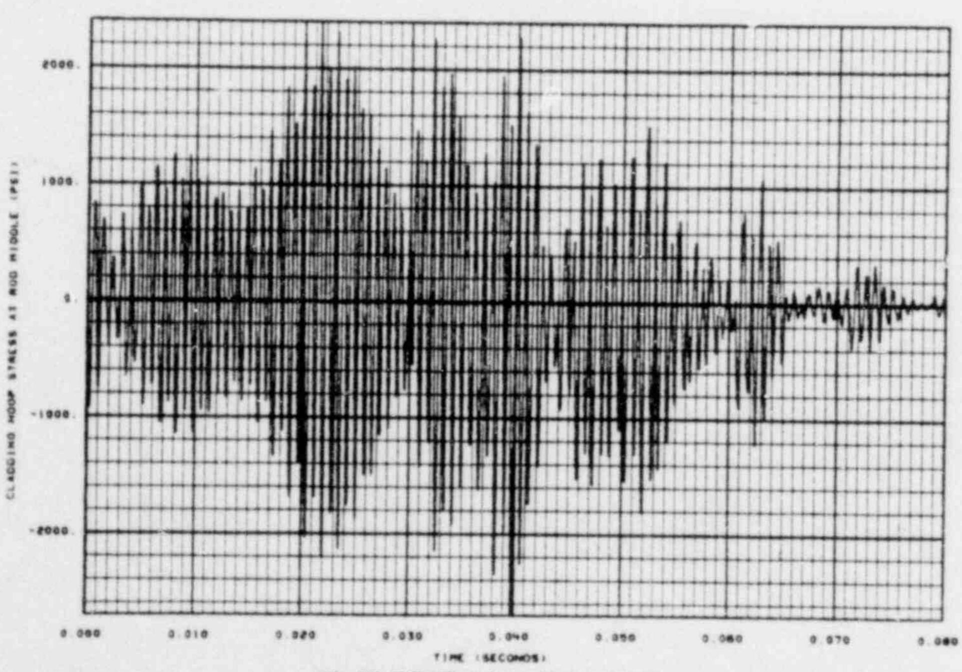


Fig. 9 RIA sample problem estimated cladding hoop stress at the rod midpoint.

TABLE 2

RIA 1-2 FRIDA INPUT

Axial Nodes	10
Fuel Stack Length	86.69 cm
Rod Outside Diameter	1.00 cm
Fuel Density	10.30 g/cm^3
Cladding Thickness	0.08 cm
Cladding Length	95.97 cm
Integration Time Step	1.0 x 10^{-5} seconds
Damping Ratio	0.03

seconds into the transient. Figure 9 shows the estimated cladding hoop stress based on the assumption of a hydrostatic state of stress in the fuel.

The maximum axial load is probably the most interesting result in that it can be used as input to an analysis of the structural integrity of the fuel rod support system. For an 8x8 BWR, 248 fuel rods would possibly experience the dynamic axial loads calculated above, considering only the four adjacent bundles. As many as 750 rods may actually be affected. For the first case, a transient load of around 53,000 N is thus exerted on the fuel rod support system. Such a load is certainly significant and should be considered in a design analysis of the system.

VII. Limitations and Recommendations

Within the framework of the beam model, FRIDA adequately calculates the dynamic structural response of beams subjected to arbitrary temperature histories. The temperature smoothing function, finite element formulation, and Runge-Kutta solution technique all contribute to a numerically reasonable approximation to the continuum. The limitations of the subcode are directly related to the accuracy of the model assumptions and not the programming itself. Specifically, the temperature distribution in a fuel rod during an RIA has a decided radial dependence, sometimes melting at the outer surface. An input single value of temperature at a given axial node may not adequately model the rod conditions. Furthermore, the input of the hottest temperature may not be conservative since the decrease in Young's modulus with increase in temperature may outweigh the increased thermal strain. Second, the effect of the cladding on the dynamic response of the whole rod should probably be included. Although the thermal expansion coefficients of the cladding and fuel are similar for like temperatures, the intense heat generation in the fuel over such a short time (approximating adiabatic heatup) should lead to vastly different actual thermal expansions between the fuel and the cladding. Under such conditions the assumption of no friction between the fuel and cladding does not seem likely. Third, the cladding stress calculation is based on a crude approximation. This approximation is probably only good for the same duration that the no friction assumption is not good. As soon as any appreciable heat transfer has occurred, the cladding will likely melt since the fuel contacting the cladding melted at the onset of the RIA. Fourth, the damping in near molten fuel or highly fragmented fuel is unknown. There is good reason to believe it is nonlinear since fragmented fuel would probably withstand compressive wave fronts but probably not withstand tensile wave fronts such as reflective waves might have. Thus, there is some justification for qualifying the results of the FRIDA code in terms of fuel rod dynamic behavior. However, for general applications where the assumptions are good, the code can and does produce reasonable results.

VIII. References

1. L. J. Siefken et al., FRAP-T5: A Computer Code for the Transient Analysis of Oxide Fuel Rods, NUREG/CR-0840, TREE-1281 (June 1979).

2. R. L. Casperson, "Subroutine FRIDA for Fuel Rod RIA Analysis", Report No CAS-2-75, System Safety Research Division, Aerojet Nuclear Company, (June 1975).

3. D. L. Hagrman et al., MATPRO-Version 11 (Revision 1): A Handbook of Material Properties for Use in the Analysis of Light Water Reactor Fuel Rod Behavior, NUREG/CR-0497, TREE-1280, Revision 1 (February 1980).

4. S. H. Crandall et al., Dynamics of Mechanical and Electromechanical Systems, McGraw-Hill Book Company, 1968.

APPENDIX A

USER INPUT MANUAL

APPENDIX A

INPUT

1. Card 1: (16I5)

Columns	1-5	NX:	Number of nodes,
	6-10	NTEM:	Number of nodes at which temperature history is defined,
	11-15	WRITE:	Multiple of integration steps at which printout is desired
	16-20	IFSI:	Units used (1=S.I., 0=lb., in, sec,)
	21-25	NF1:	Logical file number for accelerations, velocities, displacements and times to be written (eg., 8)
	26-30	NF2:	Logical file number for axial loads, strains, strain-rates, cladding hoop stress and times to be written (eg., 9)
	31-35	IPRNT1:	For intermediate printout, (1=YES, 0=NO)
	36-40	IPRNT2:	For response printout, (1=YES, 0=NO)
	41-45	NTABT:	Interpolation control, (0= Internal generation of interpolation time table). (GT.0=Externally supplied table with NTABT equal to the number of times supplied).

2. Card 2: (8E10.4)

Columns	1-10	RODR:	Fuel outside radius,
	11-20	RHO:	Fuel density,
	21-30	RODL:	Fuel stack length,
	31-40	TS:	Start time of analysis,
	41-50	TD:	Integration time-step size,
	51-60	TE:	End time for analysis,
	61-70	CLADT:	Cladding wall thickness.

3. Card 3: (8E10.4)

Columns	1-10	SPRNG-K:	Spring constant for plenum spring,
	11-20	CLADAE:	Average value of cladding AE,

21-30 CLADL: Length of the cladding tube,

31-40 SFACT: Fraction of change allowed in stiffness before updating.

41-50 DFACT: Fraction of change allowed in damping before updating.

51-60 ZETA: Damping ratio.

4. Temperature input card set

   Temperature is input as a matrix of values by axial location and time. Two matrix indices and four temperatures are input per card.

   First card (A6,I4,I5): Alphanumeric matrix name, total number of axial locations, total number of times.

   Next N cards (2I5,4E16.0): Axial index, time index, four temperatures (K) (to time index +3).

   Last card: Zeros in the first ten card columns.

5. Time history input card set

   Time is input as a vector of values in a matrix format.

   First card (A6,I4,I5): Alphanumeric matrix name, 1, total number of times.

   Next N cards (2I5,4E16.0): 1, time index, four time values in seconds (to time index +3).

   Last card: Zeros in the first ten card columns.

6. Axial locations for temperature input card set

   Axial location is input as a vector of values in a matrix format.

   First card (A6,I4,I5): Alphanumeric matrix name, total number of locations, 1.

   Next N cards (2I5,4E16.0): Axial index, 1, four locations (distances) (to index +3).

   Last card: Zeros in the first ten columns.

7. Interpolation time step card input

   If NTABT is greater than 0, include a card set similar to card set 5 with the desired interpolation time history.

APPENDIX B

LISTING OF THE FRIDA CODE

```
JNS3WC8,STMF6,T500.
ACCOUNT,3520,433MECH00,WBS.
ATTACH,IGS.
LIBRARY,IGS.
FILE,FILMPL,RT=U,BT=C,MRL=320.
FTN(R=3)
REWIND(INPUT)
COPYSBF.
REWIND(INPUT)
COPYSBF.
EXIT.
DMP(0,377000)
REWIND(INPUT)
COPYSBF.
      PROGRAM CALLER (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE8,TAPE9,
     *              FILMPL,TAPE10=FILMPL,TAPE15)
C
      COMMON/LACMOL1/ MAXIDX,EMFLAG(25)
       COMMON  /PHYPRD    / FTMELT,FHEFUS,CTMELT,CHEFUS,CTRANE,
     *                      CTRANE,CTRANZ,FDELTA,BU      ,COMP
      DIMENSION TEMPND(10,1), TEMNOD(10),
     1          EM(10),X(30),XD(30),C(10),EPS(30),ASTRES(30)
      DIMENSION TABTEM(10,100), TABT(10,20), TTRAN(20,10),
     1          TABTR(100,10), VTAT(20), VTABT(100)
      DIMENSION XDD(30),AXLOAD(30),EPSDOT(30),CHS(30),DF(30)
      DIMENSION Z(200), TVAL(13002), XDDVAL(13002), XVAL(13002),
     2          AXLVAL(13002), EPSVAL(13002), CHSVAL(13002)
      DIMENSION PLVEC1(100), PLVEC2(100)
C
      DATA   KX,KTEM, KTF, KTA, NTF/
     1       30,  10, 100,  20, 100/
      DATA EMFLAG/25*3HOFF/
C
C  CALLING PROGRAM FOR SUBROUTINE FRIDA
C
  999 READ(5,1101)NX,NTEM,NWRITE,IFSI,NF1,NF2,IPRNT1,IPRNT2,NTABT
      READ (5,1102)   RODR,RHO,RODL,TS,TD,TE,CLADT
 1000 READ (5,1102) SPRNGK,CLADAE,CLADL,SFACT,DFACT,ZETA
 1101 FORMAT (16I5)
 1102 FORMAT (8E10.4)
C
      FTMELT = 3113.15
      COMP = 0.0
C
      CALL MODESG (Z,C)
C
      DO 10 I=1,KTEM
      DO 10 J=1,KTF
      TABTEM(I,J) = 0.0
   10 TABTR(J,I) = 0.0
      DO 12 I=1,KTEM
      DO 12 J=1,KTA
      TABT(I,J) = 0.0
   12 TTRAN(J,I) = 0.0
      DO 14 I=1,KTA
   14 VTAT(I) = 0.0
      DO 16 I=1,KTF
      PLVEC1(I) = 0.0
      PLVEC2(I) = 0.0
   16 VTABT(I) = 0.0
```

A1

```
C
      CALL READ (TABT,NRTT,NCTT,KTEM,KTA)                            TEM TABL
C
      DO 17 I = 1,NCTT
   17 PLVEC1(I) = TABT(NX/2,I)
C
      DO 18 I=1,NRTT
      DO 18 J=1,NCTT
   18 TTRAN(J,I) = TABT(I,J)
      DO 22 I=1,KTEM
      DO 22 J=1,KTA
   22 TABT(I,J) = 0.0
      CALL READ (TABT,NRT,NCT,KTEM,KTA)                              TIME TA
      DO 24 I=1,NCT
   24 VTAT(I) = TABT(NRT,I)
      CALL READ (TEMPND,NRND,NCND,KTEM,1)                            TEMP.N..
C
      IF ( NTABT .GT. 0) GO TO 27
C
C     INTERPOLATE THE VALUES OF TEMPRATURE FROM KTA TO NTF NUMBER OF
C     TIME POINTS.(NTF=100,FIXED)
C
      ANTF = (NTF-1)
C
      VTABT(1) = VTAT(1)
      VDEL = (VTAT(NCT) - VTAT(1)) / ANTF
      DO 26 I=2,NTF
   26 VTABT(I) = VTABT(I-1) + VDEL
C
   27 CONTINUE
      IF ( NTABT .GT.0) NTF = NTABT
      IF (NTABT .GT. 0) CALL READ (VTABT,NTF1,NRT1,1,KTF)            TIME TA
C
      CALL WRITE (VTAT,NCT,NRT,6HTABTIN,KTA)
      CALL WRITE (VTABT,NTF,NRT,6HTABTOT,KTF)
      CALL WRITE (TTRAN,NCTT,NRTT,6HTBTEMT,KTA)
C
      CALL TERP2 (VTAT,VTABT,TTRAN,TABTR,NCTT,NTF,NRTT,KTA,KTF)
C
      DO 28 I=1,NTF
      DO 28 J=1,NRTT
   28 TABTEM(J,I) = TABTR(I,J)
      CALL WRITE (TABTEM,NRTT,NTF,6HTABTEM,KTEM)
C
C-----------------------------------------------------------------------
C
C-----TEMPERATURE PLOT SECTION.
C
      DO 30 I = 1,NTF
   30 PLVEC2(I) = TABTEM(NX/2,I)
C
      CALL XMAXM (NTF,PLVEC2,YMAX,YMIN,NMAX,AMIN)
      CALL SUBJEG (Z,VTAT(1),YMIN,VTAT(NCTT),YMAX)
      CALL GRAPHG (Z,0,VTAT,PLVEC1,14,14HTIME (SECONDS),
     *                31,31HTEMPERATURE  AT ROD MIDDLE (,K) ,
     *                31,31HRIA TEMPERATURE HISTORY (INPUT) )
      CALL SETSMG (Z,31,2.0)
      CALL LINESG (Z,NCTT,VTAT,PLVEC1)
      CALL SETSMG (Z,31,0.0)
      CALL LINESG (Z,NTF,VTABT,PLVEC2)
```

2

```
      CALL PAGEG (Z,0,0,1)
C-----------------------------------------------------------------------
C
      DO 100 I=1,NTEM
  100 TEMNOD(I) = TEMPND(I,1)
      Q(1) = 0.0
      DO 150 I=2,NTEM
      Q(I) = 0.0
  150 Q(I) = Q(I-1) + (TEMNOD(I) - TEMNOD(I-1)) *
     *       (FTHEXP(TABTEM(I,1),0.0)  +  FTHEXP (TABTEM(I-1,1),0.0))/2.
      NTM1 = NTEM - 1
      SPAN = RODL / NX
      DO 155 I=1,NX
      P = I * SPAN
      DO 153 J=1,NTM1
      IF (P .GE. TEMNOD(J) .AND. P.LT. TEMNOD(J+1)) GO TO 152
      GO TO 153
  152 L = J
      GO TO 154
  153 CONTINUE
  154 FACT = (P-TEMNOD(L))/(TEMNOD(L+1)-TEMNOD(L))
  155 X(I) = Q(L) + FACT * (Q(L+1) - Q(L))
      DO 160 I=1,NX
  160 XD(I) = 0.0
      CALL WRITE (X,NX,1,6HXINITL,KX)
C     DO 300 I=1,NTEM
C 300 RM(I) = 0.333 * RODR
      CALL WRITE (TEMNOD,NTEM,1,6HTEMNOD,KTEM)
      CALL WRITE (VTABT,NTF,NRT,5HVTABT,KTF)
      CALL FRIDA (TABTEM,VTABT,TEMNOD,RM,X,XD,EPS,ASTRES,RODR,RODL,RHC,
     1            SPRNGK,CLADAE,CLADL,CLADT,SFACT,DFACT,ZETA,TS,
     2            TD,TE,NX,NTEM,NTF,IFSI,IPRNT1,IPRNT2,NWRITE,NF1,NF2)
      MID = NX/2
      DO 460 NTIME=1,3
      REWIND NF1
      REWIND NF2
C-----------------------------------------------------------
C
      NOTP = (TE - TS) / TD
      NOP = NOTP / 13000
C
C-----------------------------------------------------------
      M=0
  400 READ (NF1) T,(AXLOAD(I),ASTRES(I),EPS(I),EPSDOT(I),CHS(I),I=1,NX)
C
      IF (EOF(NF1)) 11,9
    9 CONTINUE
C
      READ (NF2) TX,(QF(J),J=1,NX),(XDD(I),XD(I),X(I),I=1,NX)
C-----------------------------------------------------------
C
C-----MODIFICATIONS TO PLOT 13000 POINTS OR LESS OVER THE TOTAL TIME.
C
      IF (NOP .EQ. 0) GO TO 7
C
      DO 5 NI = 1,NOP
      READ (NF1) DUMP1, ((DUMP2,DUMP3,DUMP4,DUMP5,DUMP6), I=1,NX)
C
      IF (EOF(NF1)) 11,5
C
```

```
    5 READ (NF2) DUMP1, ((DUMP2), J=1,NX), ((DUMP3,DUMP4,DUMP5),I=1,NX)
C
    7 CONTINUE
C
C-------------------------------------------------------------------
      IF (T .NE. TX) GO TO 499
      M = M+1
      IF (NTIME .EQ. 1) GO TO 401
      IF (NTIME .EQ. 2) GO TO 402
      IF (NTIME .EQ. 3) GO TO 403
  401 L=1
      GO TO 404
  402 L = MID
      GO TO 404
  403 L = NX
C
  404  TVAL(M) =          T
      XDDVAL(M) =     XDD(L)
        XVAL(M) =       X(L)
      AXLVAL(M) = AXLOAD(L)
      EPSVAL(M) =     EPS(L)
      CHSVAL(M) =     CHS(L)
      IF (T .LT. TE) GO TO 400
   11 CONTINUE
      NO = M
C
C-------------------------------------------------------------------
      WRITE (6,1103) NOP,NO
 1103 FORMAT (1H1 25(/) 30X
     1        *NUMBER OF INTERVAL POINTS                        =*,I5,/
     2    30X  *NO. OF POINTS OVER TOTAL RESPONSE IN EACH PLOT =*,I5)
C-------------------------------------------------------------------
C
      IF (NTIME .EQ. 1) GO TO 411
      IF (NTIME .EQ. 2) GO TO 412
      IF (NTIME .EQ. 3) GO TO 413
      GO TO 460
C
  411 CONTINUE
      CALL XMAXM (NO,XDDVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z,TS,YMIN,TE,YMAX)
      CALL GRAPHG  (Z, 0,TVAL,XDDVAL,14,14HTIME (SECONDS),
     *             38,38HACCELERATION AT ROD BOTTOM (IN/SEC**2),
     *             42,42HFUEL ROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NO,TVAL,XDDVAL)
       CALL PAGEG(Z,0,0,1)
      CALL XMAXM (NO,   XVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z,TS,YMIN,TE,YMAX)
      CALL GRAPHG  (Z, 0,TVAL, XVAL,14,14HTIME (SECONDS),
     *             31,31HDISPLACEMENT AT ROD BOTTOM (IN) ,
     *             42,42HFUEL ROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NO,TVAL,   XVAL)
       CALL PAGEG(Z,0,0,1)
      CALL XMAXM (NO,AXLVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z,TS,YMIN,TE,YMAX)
      CALL GRAPHG  (Z, 0,TVAL,AXLVAL,14,14HTIME (SECONDS) ,
     *             30,30HAXIAL LOAD AT BOTTOM (LBS)     ,
     *             42,42HFUEL ROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NO,TVAL,AXLVAL)
       CALL PAGEG(Z,0,0,1)
```

4

```
      CALL XMAXM (NO,EPSVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z,TS,YMIN,TE,YMAX)
      CALL GRAPHG  (Z, 0,TVAL,EPSVAL,14,14HTIME (SECONDS) ,
     *             34,34HAXIAL STRAIN AT ROD BOTTOM (IN/IN) ,
     *             42,42HFUEL ROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NO,TVAL,EPSVAL)
       CALL PAGEG(Z,0,0,1)
      CALL XMAXM (NO,CHSVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z,TS,YMIN,TE,YMAX)
      CALL GRAPHG   (Z, 0,TVAL,CHSVAL,14,14HTIME (SECONDS),
     *              40,40HCLADDING HOOP STRESS AT ROD BOTTOM (PSI) ,
     *              42,42HFUEL ROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NO,TVAL,CHSVAL)
       CALL PAGEG(Z,0,0,1)
      GO TO 460
  412 CONTINUE
      CALL XMAXM (NO,XDDVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z,TS,YMIN,TE,YMAX)
      CALL GRAPHG  (Z, 0,TVAL,XDDVAL,14,14HTIME (SECONDS) ,
     *             38,38HACCELERATION AT ROD MIDDLE (IN/SEC**2) ,
     *             42,42HFUEL ROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NO,TVAL,XDDVAL)
       CALL PAGEG(Z,0,0,1)
      CALL XMAXM (NO,   XVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z,TS,YMIN,TE,YMAX)
      CALL GRAPHG  (Z, 0,TVAL, XVAL,14,14HTIME (SECONDS) ,
     *             31,31HDISPLACEMENT AT ROD MIDDLE (IN) ,
     *             42,42HFUEL ROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NO,TVAL,   XVAL)
       CALL PAGEG(Z,0,0,1)
      CALL XMAXM (NO,AXLVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z,TS,YMIN,TE,YMAX)
      CALL GRAPHG  (Z, 0,TVAL,AXLVAL,14,14HTIME (SECONDS) ,
     *             30,30HAXIAL LOAD AT ROD MIDDLE (LBS) ,
     *             42,42HFUEL ROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NO,TVAL,AXLVAL)
       CALL PAGEG(Z,0,0,1)
      CALL XMAXM (NO,EPSVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z,TS,YMIN,TE,YMAX)
      CALL GRAPHG   (Z, 0,TVAL,EPSVAL,14,14HTIME (SECONDS) ,
     *              34,34HAXIAL STRAIN AT ROD MIDDLE (IN/IN) ,
     *              42,42HFUEL ROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NO,TVAL,EPSVAL)
       CALL PAGEG(Z,0,0,1)
      CALL XMAXM (NO,CHSVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z,TS,YMIN,TE,YMAX)
      CALL GRAPHG  (Z, 0,TVAL,CHSVAL,14,14HTIME (SECONDS),
     *             40,40HCLADDING HOOP STRESS AT ROD MIDDLE (PSI) ,
     *             42,42HFUEL ROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NO,TVAL,CHSVAL)
       CALL PAGEG(Z,0,0,1)
      GO TO 460
  413 CONTINUE
      CALL XMAXM (NO,XDDVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z,TS,YMIN,TE,YMAX)
      CALL GRAPHG   (Z, 0,TVAL,XDDVAL,14,14HTIME (SECONDS) ,
     *              38,38HACCELERATION AT ROD TOP (IN/SEC**2)    ,
     *              42,42HFUEL ROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NO,TVAL,XDDVAL)
       CALL PAGEG(Z,0,0.1)
```

5

```
      CALL XMAXM (NC,   XVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z, 5,YMIN,TE,YMAX)
      CALL GRAPHG   (Z, 0,TVAL, XVAL,14,14HTIME (SECONDS) ,
     *               31,31HDISPLACEMENT AT ROD TOP (IN)      ,
     *               42,42HFUELROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NC,TVAL,   XVAL)
       CALL PAGEG(Z,0,0,1)
      CALL XMAXM (NC,AXLVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z,TS,YMIN,TE,YMAX)
      CALL GRAPHG   (Z, 0,TVAL,AXLVAL,14,14HTIME (SECONDS) ,
     *               30,30HAXIAL LOAD AT TOP (LBS)           ,
     *               42,42HFUEL ROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NC,TVAL,AXLVAL)
       CALL PAGEG(Z,0,0,1)
      CALL XMAXM (NC,EPSVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z,TS,YMIN,TE,YMAX)
      CALL GRAPHG   (Z, 0,TVAL,EPSVAL,14,14HTIME (SECONDS) ,
     *               34,34HAXIAL STRAIN AT ROD TOP   (IN/IN)    ,
     *               42,42HFUEL ROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NC,TVAL,EPSVAL)
       CALL PAGEG(Z,0,0,1)
      CALL XMAXM (NC,CHSVAL,YMAX,YMIN,NMAX,NMIN)
      CALL SUBJEG (Z,TS,YMIN,TE,YMAX)
      CALL GRAPHG   (Z, 0,TVAL,CHSVAL,14,14HTIME (SECONDS) ,
     *               40,40HCLADDING HOOP STRESS AT ROD TOP   (PSI)     ,
     *               42,42HFUEL ROD RESPONSE TO RIA THERMAL TRANSIENT )
      CALL LINESG (Z,NC,TVAL,CHSVAL)
       CALL PAGEG(Z,0,0,1)
  400 CONTINUE
      GO TO 5000
  499 WRITE (6,4099)
 4099 FORMAT (1H1,///10X,*TX .NE. T  WHEN READING FROM FRIDA TAPES* )
 5000 CONTINUE
      CALL EXITG (Z)
      END
      SUBROUTINE EIGEN (A,VAL,VEC,NIN,FODIN,KR)
      DIMENSION A(KR,1), VAL(1), VEC(KR,1)
C
C-----CALCULATES EIGENVALUES/EIGENVECTORS OF (A)*(VEC) = (VEC) * (VAL).
C-----JACOBI METHOD.
C-----THE MATRIX A SHOULD BE REAL AND SYMMETRIC. UPPER TRIANGULAR HALF
C-----IS USED.
C-----ARGUMENTS.
C----- A      = INPUT   MATRIX TO BE DIAGONALIZED.  *DESTROYED*
C----- VAL    = OUTPUT  VECTOR OF EIGENVALUES. SIZE(N).
C----- VEC    = OUTPUT  MATRIX OF EIGENVECTORS. SIZE(N,N).
C----- NIN    = INPUT   ABS(NIN)=N IS THE SIZE OF MATRICES A,VEC,AND
C                       VECTOR VAL. IF NIN IS NEGATIVE, INITIAL VEC MATRIX IS
C                       ASSUMED TO BE SUPPLIED THROUGH ARGUMENT.
C----- FODIN = INPUT   FINAL OFF DIAGONAL VALUE FOR DIAGONALIZED A.
C                       IF FODIN .LE. 0.0, FOD=TRACE(A)*10**-15 WILL BE USED.
C----- KR     = INPUT   ROW DIMENSION OF A AND VEC IN CALLING PROGRAM.
C
      N = IABS(NIN)
      IF (NIN .LT. 0) GO TO 10
C-----SET INITIAL VEC MATRIX TO UNITY.
      DO 6 I=1,N
      DO 5 J=1,N
    5 VEC(I,J) = 0.0
    6 VEC(I,I) = 1.0
```

6

```
C
   10 IF (N .EQ. 1) GO TO 60
C
C-----LOCATE LARGEST OFFDIAGONAL ELEMENT OF A.
C-----(CALCULATE TRACE OF A FOR COMPARISION).
      TRACE = 0.0
      THRESH = ABS(A(1,2))
      NM1 = N - 1
      DO 15 I=1,NM1
      TRACE = TRACE + A(I,I)
      IP1 = I+1
      DO 15 J=IP1,N
   15 IF (ABS(A(I,J)) .GT. THRESH) THRESH = ABS(A(I,J))
      TRACE = TRACE + A(N,N)
      FOD = FODIN
      IF (FODIN .LE. 0.0) FOD = TRACE*1.0E-15
      IF (THRESH .LE. FOD) GO TO 60
C
C-----LOCATING.
   20 THRESH = THRESH/10.0
      IF (THRESH .LT. FOD) THRESH = FOD
   22 IREDO = 0
      DO 41 IP=1,NM1
      IPM1 = IP-1
      IPP1 = IP+1
      DO 40 JP=IPP1,N
      IF (ABS(A(IP,JP)) .LT. THRESH) GO TO 40
      IREDO = 1
C-----CALCULATE ROTATION VALUES.
      DEL = A(IP,IP) - A(JP,JP)
      RAD = SQRT (DEL**2 + 4.0*A(IP,JP)**2)
      IF (DEL .LT. 0.0) RAD = -RAD
      TN = (2.0 * A(IP,JP))/ (DEL + RAD)
      CS = 1.0 /SQRT (1.0 + TN**2)
      SN = TN * CS
C-----DIAGONALIZE MATRIX (A).
      JPM1 = JP-1
      JPP1 = JP+1
      IF (IP .EQ. 1) GO TO 33
      DO 32 I=1,IPM1
      AIIP = A(I,IP)*CS + A(I,JP)*SN
      A(I,JP) = -A(I,IP)*SN + A(I,JP)*CS
   32 A(I,IP) = AIIP
   33 IF (IPP1 .EQ. JP) GO TO 35
      DO 34 I=IPP1,JPM1
      AIPI = A(IP,I)*CS + A(I,JP)*SN
      A(I,JP) = -A(IP,I)*SN + A(I,JP)*CS
   34 A(IP,I) = AIPI
   35 IF (JP .EQ. N) GO TO 37
      DO 36 I=JPP1,N
      AIPI = A(IP,I)*CS + A(JP,I)*SN
      A(JP,I) = -A(IP,I)*SN + A(JP,I)*CS
   36 A(IP,I) = AIPI
   37 AIPIP = A(IP,IP)
      AJPJP = A(JP,JP)
      CS2 = CS**2
      SN2 = SN**2
      ASC = 2.0 * A(IP,JP)*SN*CS
      A(IP,IP) = AIPIP * CS2 + ASC + AJPJP * SN2
      A(JP,JP) = AIPIP * SN2 - ASC + AJPJP * CS2
```

```
      A(IP,JP) = 0.0
C-----CALCULATE EIGENVECTORS.
      DO 36 I=1,N
      VECIIP = VEC(I,IP)*CS + VEC(I,JP)*SN
      VEC(I,JP) = -VEC(I,IP)*SN + VEC(I,JP)*CS
   36 VEC(I,IP) = VECIIP
   40 CONTINUE
   41 CONTINUE
      IF (IREDO .EQ. 1) GO TO 22
      IF (THRESH .GT. FOD) GO TO 20
C
C-----ASSEMBLE DIAGONAL FROM A INTO VAL(EIGENVALUES).
   60 DO 61 I=1,N
   61 VAL(I) = A(I,I)
C
      RETURN
      END
      SUBROUTINE FRIDA (TABTEM,VTABT,TEMNOD,RM,X,XD,EPS,ASTRES,RODR,
     1  RODL,RHO,SPRNGK,CLADAE,CLADL,CLADT,SFACT,DFACT,ZETA,TS,
     2  TC,TE,NX,NTEM,NTF,IFSI,IPRNT1,IPRNT2,NWRITE,NFILE1,NFILE2)
C
       COMMON  /PHYPRC   / FTMELT,FHEFUS,CTMELT,CHEFUS,CTRANB,
     #                     CTRANE,CTRAN2,FDELTA,BD      ,COMP
C
       DIMENSION  A(30,30), B(30,30), C(30,30), QF(30), PP(30),QQ(30),
     1            TABTEM(10,100), TABT(30,100),TEMP(30,100),VTABT(100),
     2            TABAE(10,100), TABF(30,100), DMASS(1,4), DAE(10,4),
     3            XDD(30),XD(30),X(30),AXLOAD(30),EPS(30),EPSDOT(30),
     4            DAE1(30), DAE2(30), XDO(30), XO(30), ASTRES(30),
     5            XDDMAX(30),XDDMIN(30),XDMAX(30),XDMIN(30),XMAX(30),
     6            XMIN(30),TDDMAX(30),TDDMIN(30),TXDMAX(30),TXDMIN(30),
     7            TXMAX(30),TXMIN(30),AXMAX(30),AXMIN(30),EMAX(30),
     8            EMIN(30),EDMAX(30),EDMIN(30),TAXMAX(30),TAXMIN(30),
     9            TEMAX(30),TEMIN(30),TEDMAX(30),TEDMIN(30), RMELT(30),
     1            ASMAX(30), ASMIN(30), TASMAX(30), TASMIN(30),
     2            CHSMAX(30), CHSMIN(30), TCHMAX(30), TCHMIN(30),
     3            CHS(30), RM(10), TEMNOD(10)
       DATA NIT,NOT / 5,6 /
       DATA NLPP,  KX,KTEM, KTF/
     *      60,  30,  10, 100/
       DATA FCTMTL,FCOMP / 2.0 , 0.0 /
C
C      *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
C
C     *** *** *** *** *** *** ***   FRIDA   *** *** *** *** *** *** ***
C
C
C    DISCRETE MODEL RESPONSE ROUTINE TO SOLVE THE PROBLEM OF FUEL REACTIVITY
C    INSERTION DYNAMIC ANALYSIS (FRIDA)...MASS-SPRING DISCRETE SYSTEM FORCED
C    BY THERMAL FORCES, WHICH, IN TURN, DERIVE FROM INPUT NODAL TEMPERATURE
C    TIME HISTORIES RESULTING FROM A REACTIVITY INSERTION ACCIDENT (RIA).
C
C    THE ANALYSIS IS STRICTLY A LINEAR ONE.
C
C    THIS VERSION OF FRIDA USES ZERO INITIAL DISPLACEMENTS AND CORRESPONDING
C     ZERO INITIAL FORCES.   THE TRANSIENT SOLUTION IS IN TERMS OF RELATIVE
C    DISPLACEMENTS WHICH ARE THEN LINEARLY ADDED TO THE ACTUAL INITIAL
C    DISPLACEMENTS TO GET THE TOTAL DIPLACEMENT - TIME HISTORIES.
C    THE STIFFNESS MATRIX IS MODIFIED AFTER AN INPUT FRACTIONAL CHANGE TAKES
C    PLACE IN STIFFNESS PROPERTIES.
C
```

8

```
C         *****  CODED    BY J. N. SINGH  *****
C         *****      NOV. 1979            ******
C
C
C     *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C
C       FRIDA CALLS THE FOLLOWING SUBROUTINES...INV1, MASS, MULTB,
C                                          STIFF, TR1, TERP2, WRITE
C
C       FRIDA CALLS THE FOLLOWING MATPRO ROUTINES...FTHEXP, FELMOD
C
C     *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C
C
C       *****        NOTE ON INPUT UNITS            *****
C       *****                                       *****
C       ***** VARIABLES ARE INPUT IN SI UNITS       *****
C       *****                 OR                    *****
C       ***** IN UNITS OF LBS-INCHES-SECONDS        *****
C       *****                BUT                    *****
C       ***** TEMPERATURE IS ALWAYS IN KELVIN       *****
C
C                SUBROUTINE ARGUMENTS (INPUT)
C
C     TABT = FIRST--MATRIX OF NODAL TEMPERATURES...EACH COLUMN
C            REPRESENTING NODAL TEMPERATURES FOR A DISCRETE TIME.....EACH
C            ROW REPRESENTING A SERIES OF TEMPERATURES (CHRONOLOGICAL)
C            FOR ONE NODE ON A ROD.
C            SIZE (NTEMP,NCTT(MAX.20)).-------(DEGREES KELVIN)
C            **NOTE** THIS TEMP. SHOULD BE SUCH THAT IT GIVES A FUEL AE
C                     AVERAGED ACROSS THE FUEL CROSS-SECTIONAL AREA.
C          = SECOND--VECTOR OF TIMES CORRESPONDING TO TEMPERATURES IN
C            TABLE  OF TEMPERATURES-----SIZE (1,NCTT).
C   TEMNOD = VECTOR OF NODE LOCATIONS ALONG ROD AXIS AT WHICH TEMPERATURE
C            HISTORIES ARE INPUT.  SIZE (NTEMP).
C            * * * FIRST NODE SHOULD COINCIDE WITH BOTTOM OF ROD * * *
C            * * *  LAST NODE SHOULD COINCIDE WITH TOP OF ROD    * * *
C
C***********************************************************************
C     NOTE= THE SUPPLIED TEMPERATURES (A MAX. OF 20 TIME PTS. PER NODE)
C           IS DIAPARABOLICALLY INTERPOLATED TO 100 TIME PTS. PER NODE
C           TO MAKE IT SMOOTHER.
C***********************************************************************
C
C     RODR = FUEL OUTSIDE RADIUS.
C     RODL = FUEL STACK LENGTH.
C      RHO = FUEL DENSITY.
C   SPRNGK = PLENUM SPRING, SPRING CONSTANT.
C   CLADAE = AVERAGE VALUE OVER LENGTH AND TIME OF CLADDING AE.
C    CLADL = LENGTH OF CLADDING TUBE.
C    CLADT = CLADDING WALL THICKNESS.
C    SFACT = FRACTION OF CHANGE ALLOWED IN ROD AE BEFORE STIFFNESS AND
C            DAMPING ARE MODIFIED AND A NEW INTEGRATION INTERVAL IS STARTED.
C            ( 0.0 .LT. SFACT .LE. 1.0 ).. RECOMMENDED VALUE OF .1 TO .25.
C    DFACT = AMOUNT BY WHICH DAMPING FACTOR IS INCREASED IN EACH INTEGRATION
C            INTERVAL OVER WHICH STIFF. AND DAMPING MATRICES ARE RECALCULATED.
C     ZETA = CRITICAL DAMPING FACTOR USED IN CONSTRUCTING DAMPING MATRIX.
C            THE SAME VALUE IS USED FOR EACH VIBRATORY MODE.
C       TS = STARTING TIME FOR ANALYSIS.
C       TD = INTEGRATION TIME STEP SIZE.
```

9

```
C        = 0.0, IF IT IS TO BE CALCULATED INTERNALLY.
C      TE = FINAL TIME FOR ANALYSIS.
C      NX = NUMBER OF NODES (DOF) FOR RESTRAINED LONGITUDINAL ROD SYSTEM.
C           ** MUST BE .GE. 2 AND LESS THAN OR EQUAL TO 29 **
C    NTEM = NUMBER OF NODES AT WHICH TEMPERATURE HISTORIES ARE DEFINED.
C     NTF = NUMBER OF DISCRETE TIMES IN TEMPERATURE HISTORY MATRICES TABTEM
C           AND TABT.(FIXED AT 100).
C    IFSI = 1, SI UNITS USED.
C         = 0,2,3, ETC, UNITS ARE LB-IN-SEC WITH TEMPERATURE IN DEGREES K.
C  IPRNT1 = 1, IF PRINTOUT OF MATRICES OF STIFFNESS,DAMPING,MASS,TEMPERATURE,
C           FORCES,TIMES,ETC IS DESIRED.
C         = 0, IF NO PRINT OF THIS INFORMATION IS DESIRED.
C  IPRNT2 = 1, IF PRINTOUT OF ROD RESPONSE IS DESIRED EVERY NWRITE * TD.
C         = 0, IF NO PRINT OF THIS INFORMATION IS DESIRED.
C  NWRITE = MULTIPLE OF INTEGRATION STEPS TO PRINT OUT.
C         = 1, PRINT EVERY STEP (0,1,2,...)
C         = 2, PRINT EVERY SECOND STEP (0,2,4,...)
C           ETC, ETC.
C  NF1    = LOGICAL STORAGE UNIT ON WHICH FORCES,ACCELERATIONS,VELOCITIES,
C           DISPLACEMENTS, AND CORRESPONDING TIMES ARE WRITTEN.  THIS UNIT
C           IS WRITTEN IN THE FOLLOWING FASHION...
C         * WRITE (NFILE1) T,(F(J),J=1,NX),(XDD(I),XD(I),X(I),I=1,NX) *
C           AND WHERE T=TIME, AND F(J) = FORCE AT J-TH NODE.
C  NF2    = LOGICAL STORAGE UNIT ON WHICH AXIAL LOADS, STRAINS, STRAIN -RATES,
C           AND CLAD HOOP STRESS, AND THE CORRESPONDING TIMES ARE WRITTEN.
C           THIS UNIT IS WRITTEN IN THE FOLLOWING FASHION ...
C          *WRITE (NFILE2)T,(AXLOAD(I),ASTRES(I),EPS(I),EPSDOT(I),CHS(I),I=1,NX)
C  NTABT  = 0    , IMPLIES 100 EQUAL TIME-POINTS ( BETWEEN START TIME
C                  AND END TIME OF TEMPERATURE TABLE) TO BE USED FOR
C                  INTERPOLATION.
C         = GT.0, IMPLIES NTABT NO. OF TIME POINTS IS TO BE READ IN.
C                  THESE ARE THE TIME POINTS AT WHICH TEMPERATURE IS
C                  TO BE INTERPOLATED. SIZE(1 X NTABT)
C
C
C                 SUBROUTINE ARGUMENTS (OUTPUT, INPUT AND OUTPUT)
C                      ----- ALL INTERNAL -----
C
C       X = INPUT INITIAL DISPLACEMENT AT T = TS.   SIZE (NX).
C         = OUTPUT FINAL DISPLACEMENT AT T = TE.   SIZE (NX).
C      XD = INPUT INITIAL VELOCITY AT T = TS.   SIZE (NX).
C         = OUTPUT FINAL VELOCITY AT T = TE.   SIZE (NX).
C      RM = VECTOR OF MELT RADII CORRESPONDING TO AXIAL NODES.   SIZE (NTEM).
C     EPS = OUTPUT NODAL AXIAL STRAINS AT T = TE.   SIZE (NX).
C  ASTRES = OUTPUT NODAL AXIAL (MOLTEN HYDROSTATIC) STRESS AT T = TE.
C           SIZE (NX).  *** NEGATIVE VALUES ARE COMPRESSIVE STRESS  ***
C
C
C     *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C
C           PROGRAM VARIABLES INTERNAL TO FRIDA...
C
C       A = MASS MATRIX FOR FUEL.
C    AMIN = SMALLEST DIAGONAL ELEMENT OF MASS MATRIX.
C  ASTRES = AXIAL (HYDRO)STRESS, CALCULATED AT EACH INTEG. STEP.   SIZE (NX).
C           ***  NEGATIVE VALUES ARE COMPRESSIVE STRESS  ***
C  AXLOAD = AXIAL INTERNAL LOAD, CALCULATED AT EACH INTEG. STEP.  SIZE (NX).
C           ***  NEGATIVE VALUES ARE COMPRESSIVE LOAD  ***
C       B = DAMPING MATRIX FOR FUEL.
C       C = STIFFNESS MATRIX.
```

10

```
C     CHS = AXIAL NODE CLADDING HOOP STRESS APPROXIMATION, CALCULATED AT EACH
C           INTEGRATION STEP. **** NEGATIVE VALUES W/O MEANING ****
C           SIZE (NX).          **** POSITIVE VALUES ARE TENSILE ****
C    CMAX = LARGEST DIAGONAL ELEMENT OF STIFFNESS MATRIX.
C  CSAREA = CROSS-SECTIONAL AREA OF FUEL.
C   DMASS = DISTRIBUTED MASS PROPERTIES.
C     DAE = DISTRIBUTED STIFFNESS PROPERTIES.
C    EFFK = EFFECTIVE SPRING CONSTANT FOUND AS SERIES ADDITION OF PLENUM
C           SPRING AND CLADDING WALL STIFFNESS.
C    ENDT =    END TIME FOR AN INTEGRATION INTERVAL.
C     EPS = AXIAL STRAINS,          CALCULATED AT EACH INTEG. STEP.  SIZE (NX).
C  EPSDOT = AXIAL STRAIN RATES,  CALCULATED AT EACH INTEG. STEP.  SIZE (NX).
C       F = AXIAL FORCE DUE TO INITIAL DEFLECTION OF PLENUM SPRING.
C           SIZE (NX).
C      KX = DIMENSION SIZE OF ROD PANEL POINTS AND OF MASS AND STIFFNESS
C           MATRICES FOR THE UNRESTRAINED ROD SYSTEM.
C    KTEM = DIMENSION SIZE FOR NUMBER OF INPUT TEMPERATURE NODES.
C     KTF = DIMENSION SIZE FOR NUMBER OF DISCRETE TIMES IN TEMPERATURE -
C           TIME HISTORIES.
C      N1 = NUMBER OF D.O.F. OF UNRESTRAINED ROD SYSTEM.
C         = NX + 1.
C      PP = VECTOR OF NODE (PANEL) POINT LOCATIONS FOR THE UNRESTRAINED ROD.
C      Q0 = VECTOR OF INITIAL ABSOLUTE DISPLACEMENTS.  SIZE (NX).
C      QF = LINEARLY INTERPOLATED NODAL THERMAL FORCES USED IN RUNGE-KUTTA
C           INTEGRATION.  FOUND USING TABF AND TABT IN TRI SUBROUTINE.
C           SIZE (NX).
C   RMELT = VECTOR OF MELT RADII CORRESPONDING TO NTEM AXIAL NODES.
C           IN THIS VERSION THE ROD IS ASSUMED TO MELT ON THE OUTSIDE.
C    SPAN = LENGTH OF EACH ROD SEGMENT (IF NX .NE. NTEM).
C  STARTT = START TIME FOR AN INTEGRATION INTERVAL.
C   TABAE = MATRIX OF NODAL AE VALUES AT VARIOUS TIMES.  SIZE (NTEM,NTF).
C    TABF = MATRIX OF NODAL THERMAL FORCES.  SIZE(NX,NTF).
C    TABT = MATRIX OF TIMES CORRESPONDING TO THE TEMPERATURES IN TABF.
C           SAME SET OF TIMES FOR EACH AXIAL NODE.  SIZE(NX,NTF).
C    TEMP = MATRIX OF INTERPOLATED TEMPERATURES FOUND FROM TABTEM.
C           SIZE (N1,NTF).
C  TT,TTE = TIME VARIABLES USED IN CALCULATING LENGTHS OF INTEGRATION
C           TIME INTERVALS.
C
C-----UPDATE INTERVAL CHECK IS HARD WIRED AT 5.0*TD FOR THE PRESENT.
C
C      X0 = VECTOR OF INITIAL RELATIVE DISPLACEMENTS.  SIZE (NX).
C     XD0 = VECTOR OF INITIAL VELOCITIES.  SIZE (NX).
C       X = AXIAL DISPLACEMENTS, CALCULATED AT EACH INTEG. STEP.  SIZE (NX).
C      XD = AXIAL VELOCITIES,       CALCULATED AT EACH INTEG. STEP.  SIZE (NX).
C     XDD = AXIAL ACCELERATIONS, CALCULATED AT EACH INTEG. STEP.  SIZE (NX).
C
C
C     *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C
C     * * * *  BEGINNING OF PROGRAM  * * * *
C
C
      WRITE (NOT,3001) RODR,RODL,RHO,SPRNGK, CLADAE,CLADL,
     1                 CLADT,SFACT,DFACT,ZETA,TS,TD,TE,NX,NTEM,NTF,IFSI,
     2                 IPRNT1,IPRNT2,NWRITE,NFILE1,NFILE2
 3001 FORMAT (1H1//////////18X*SUBROUTINE FRIDA HAS BEEN CALLED WITH THE
     1 FOLLOWING SCALARS TRANSFERRED THROUGH THE VARIABLE LIST* /////
     2    2X,*  RODR = *,1PE15.8,8X,*  RODL = *,1PE15.8,8X,*   RHO = *,
     3    1PE15.8,8X /// 2X,*SPRNGK = *,1PE15.8,8X,
     4                            *CLADAE = *,1PE15.8,8X,* CLADL = *,
     5    1PE15.8 /// 2X,* CLADT = *,1PE15.8,8X,* SFACT = *,1PE15.8,8X,
```

```
     6    * DFACT = *,1PE15.8,8X,*  ZETA = *,1PE15.8 /// 2X,*     TS = *,
     7    1PE15.8,8X,*     TD = *,1PE15.8,8X,*     TE = *,1PE15.8//////5X,
     8    *     NX = *,I3,13X,*  NTEM = *,I3,13X,*    NTF = *,I3,13X,
     9    *  IFSI = *,I3,13X,*IPRNT1 = *,I3 /// 5X,*IPRNT2 = *,I3,13X,
     *    *NWRITE = *,I3,13X,*NFILE1 = *,I3,13X,*NFILE2 = *,I3)
C
      NRESP = 0
      IF (IFSI .EQ. 1) GO TO 10
      WRITE (NOT,2002)
      GO TO 20
   10 WRITE (NOT,2003)
 2002 FORMAT (1H1,10(/),40X,*THE FOLLOWING INFORMATION IS FRIDA OUTPUT*
     *        ///44X,*ALL OF THE OUTPUT IS IN UNITS OF*
     *            /40X,*LBS - INCHES - SECONDS - DEGREES KELVIN* )
 2003 FORMAT (1H1,10(/),40X,*THE FOLLOWING INFORMATION IS FRIDA OUTPUT*
     *        ///44X,*ALL OF THE OUTPUT IS IN SI UNITS* )
C   SET INITIAL CONDITIONS.
C
   20 N1 = NX + 1
      DO 30 I=1,NX
      XDO(I) = XD(I)
      CO(I) = X(I)
   30 XO(I) = 0.0
      CALL WRITE (CO,NX,1,6HXO  TS,KX)
C
      IFTD = 1
      IF (TD .NE. 0.0) GO TO 40
      IFTD = 0
C
C   FORM NODE POINT LOCATIONS.
C
   40 K1 = KX
      NTEM2 = NTEM/2
      IF (NTEM .EQ. N1) GO TO 100
      SPAN = RODL/NX
      PP(1) = TEMNOD(1)
      DO 50 I=2,N1
   50 PP(I) = PP(I-1) + SPAN
      GO TO 120
  100 DO 110 I=1,N1
  110 PP(I) = TEMNOD(I)
                                                            NERROR = 1
  120 DO 125 I=1,NX
  125 QF(I) = PP(I+1)
      WRITE (NOT,3002) (I,QF(I),I=1,NX)
 3002 FORMAT (1H1///20X,*THE FOLLOWING AXIAL NODE POINT COORDINATES COR*
     1   ,*RESPOND TO THE RESPONSE OUTPUT* ///36X,*NODE NO.*,10X,
     2   *DISTANCE ABOVE ROD BOTTOM* // (36X,I3,17X,1PE16.8))
      IF ((ABS( PP(N1)-PP(1))-RODL) .GE. 1.E-3) GO TO 999
C
C   CALCULATE ONE-SHOT MASS MATRIX FROM UNIFORM MASS DISTRIBUTION.
C
      CSAREA = 3.14159 * RODR * RODR
      DMASS(1,1) = PP(1)
      DMASS(1,2) = PP(N1)
      DMASS(1,3) = RHO * CSAREA
      DMASS(1,4) = RHO * CSAREA
      CALL MASS (PP,DMASS,A,N1,1,1,K1)
      CALL WRITE (A,N1,N1,6HMASS F,K1)
C
```

12

```
C  RESTRAIN MASS MATRIX IN THE FIRST DOF.
C
      DO 130 I=1,NX
      DO 130 J=1,N1
  130 A(I,J) = A(I+1,J)
      DO 140 J=1,NX
      DO 140 I=1,NX
  140 A(I,J) = A(I,J+1)
      CALL WRITE (A,NX,NX,6HMASS R,K1)
C
      AMIN = A(1,1)
       DO 150 I=2,NX
      IF (A(I,I) .GE. AMIN) GO TO 150
      AMIN = A(I,I)
  150  CONTINUE
      REWIND NFILE1
      WRITE (NFILE1)((A(I,J),I=1,NX),J=1,NX)                          A=MASS

      CALL INV1 (A,C,NX,IPRNT1,K1)
      CALL WRITE (C,NX,NX,6HMASSIV,K1)
      WRITE (NFILE1) ((C(I,J),I=1,NX),J=1,NX)
C
C
C  FIND TABLE OF AE .VS. TIME (TABAE).
C
       NX1 = NX - 1
      NTM1 = NTEM - 1
      DO 200 I=1,NTEM
      DO 200 K=1,NTF
  200 TABAE(I,K) = FELMOD(TABTEM(I,K),1.,FOTMTL,FCOMP) * CSAREA
      IF (IFSI .EQ. 1) GO TO 207
      DO 203 I=1,NTEM
      DO 203 K=1,NTF
  203 TABAE(I,K) = .0001450 * TABAE(I,K)
  207 CALL WRITE (TABAE,NTEM,NTF,6HTABAEI,KTEM)
C
C
C  LINEARLY INTERPOLATE TABTEM(NTEM,NTF) TO OBTAIN TABF(NX,NTF).
C
      IF (NTEM .NE. N1) GO TO 220
      DO 210 I=1,NTM1
      DO 210 K=1,NTF
  210 TABF(I,K) = (FELMOD(TABTEM(I,K),1.,FOTMTL,FCOMP) +
     1              FELMOD(TABTEM(I+1,K),1.,FOTMTL,FCOMP))/2.
     1   *((FTHEXP(TABTEM(I,K),0.) + FTHEXP(TABTEM(I+1,K),0.))/2.
     2   - (FTHEXP(TABTEM(I,1),0.) + FTHEXP(TABTEM(I+1,1),0.))/2.)*CSAREA
      GO TO 265
C
C   I INDEX (DO 250) REFERS TO PP NODES.
C
  220 DO 250 I=1,N1
      DO 235 J=1,NTM1
      IF (PP(I).GE.TEMNOD(J) .AND. PP(I).LT.TEMNOD(J+1)) GO TO 230
      GO TO 235
  230 L = J
      GO TO 240
  235 CONTINUE
                                                                 NERROR = 2
      GO TO 999
```

13

```
  240 FACT = (PP(I)-TEMNOD(L)) / (TEMNOD(L+1)-TEMNOD(L))
C     RMELT(I) = RM(L) + FACT * (RM(L+1)-RM(L))
      DO 245 K=1,NTF
  245 TEMP(I,K) = TABTEM(L,K) + FACT * (TABTEM(L+1,K)-TABTEM(L,K))
  250 CONTINUE
C     DO 255 I=1,NX
C 255 RMELT(I) = RMELT(I+1)
C     CALL WRITE (RMELT,NX,1,6HRMELT ,K1)
      CALL WRITE (TEMP,N1,NTF,6HTEMP  ,K1)
C
C   I INDEX (DO 260) REFERS TO I-TH ROD SEGMENT.
C
      DO 260 I=1,NX
      DO 260 K=1,NTF
  260 TABF(I,K) = (FELMOD(TEMP(I,K),1.,FOTMTL,FCOMP) +
     1            FELMOD(TEMP(I+1,K),1.,FOTMTL,FCOMP))/2. *
     1  ((FTHEXP(TEMP(I,K),0.) + FTHEXP(TEMP(I+1,K),0.))/2.
     2  - (FTHEXP(   TEMP(I,1),0.) + FTHEXP(   TEMP(I+1,1),0.))/2.)*CSAREA
C
  265 IF (IFSI .EQ. 1) GO TO 275
      DO 270 I=1,NX
      DO 270 K=1,NTF
  270 TABF(I,K) = .0001450 * TABF(I,K)
  275 CALL WRITE (TABF,NX,NTF,6HAEALFT,K1)
C
C  FORM NODAL THERMAL EXPANSION FORCES FROM ROD ELEMENT TERMS.
C
      DO 277 I=1,NX1
      DO 277 K=1,NTF
  277 TABF(I,K) = TABF(I,K) - TABF(I+1,K)
      CALL WRITE (TABF,NX,NTF,6HTABF-I,K1)
C
C  FORM TIME TABLE CORRESPONDING TO TABF.
C
      DO 280 I=1,NX
      DO 280 J=1,NTF
  280 TABT(I,J) = VTABT(J)
      CALL WRITE (TABT,NX,NTF,6H  TABT,K1)
C
C  FIND EFFECTIVE TOP SPRING STIFFNESS.
C
      EFFK = 1. / (1./SPRNGK + 1./(CLADAE/CLADL))   .
C
C
      STARTT = TS
C
  500 CONTINUE
      IF (STARTT .EQ. TS) GO TO 550
      IF (STARTT .GE. TE) GO TO 800
C
      DO 510 I=1,NX
      XO(I) = X(I)
  510 XDO(I) = XD(I)
C
C  FIND TIME SPAN FOR UNCHANGED STIFFNESS AND DAMPING PROPERTIES.
C
  550 TT = STARTT
C
  560 CONTINUE
C
```

14

```
      DO 620 K=1,NTF
      IF (TT .GE. VTABT(K) .AND. TT .LT. VTABT(K+1)) GO TO 610
      GO TO 620
  610 TTE = VTABT(K+1)
      LL = K
      FACTA = ABS((STARTT-VTABT(LL))/(VTABT(LL+1)-VTABT(LL)))
      AESTAT = TABAE(NTEM2,LL)+FACTA*(TABAE(NTEM2,LL+1)-TABAE(NTEM2,LL))
      GO TO 630
  620 CONTINUE
C
C 630 TT = TT + 5.0 * TD
  630 TT = TT + 1.0 * TD
      IF (TT .LT. TE) GO TO 640
      ENDT = TE
      GO TO 700
C
  640 IF (TT .GT. TTE ) GO TO 560
C
  650 FACTB = ABS((      TT-VTABT(LL))/(VTABT(LL+1)-VTABT(LL)))
        AETT = TABAE(NTEM2,LL)+FACTB*(TABAE(NTEM2,LL+1)-TABAE(NTEM2,LL))
      DELTAE = ABS((AESTAT-AETT)/AESTAT)
      IF (DELTAE .LT. SFACT) GO TO 630
      ENDT = TT
C
C  FIND AVERAGE AE VALUES OVER THE TIME SPAN (STARTT.LE.T.LE.ENDT).
C
  700 FACT1 = (STARTT - VTABT(LL)) / (VTABT(LL+1) - VTABT(LL))
      FACT2 = (   ENDT - VTABT(LL)) / (VTABT(LL+1) - VTABT(LL))
      DO 720 I=1,NTEM
      DAE1(I) = TABAE(I,LL) + FACT1 * (TABAE(I,LL+1)-TABAE(I,LL))
  720 DAE2(I) = TABAE(I,LL) + FACT2 * (TABAE(I,LL+1)-TABAE(I,LL))
      DO 730 I=1,NTM1
      DAE(I,1) = TEMNOD(I)
      DAE(I,2) = TEMNOD(I+1)
      DAE(I,3) = 0.5 * (DAE1(I  ) + DAE2(I  ))
  730 DAE(I,4) = 0.5 * (DAE1(I+1) + DAE2(I+1))
      IF (NRESP .EQ. 0  .AND.  IPRNT1 .NE. 1)
     1CALL WRITE (DAE,NTM1,4,6HIN-DAE,KTEM)
      IF (IPRNT1 .EQ. 1) CALL WRITE (DAE,NTM1,4,6H   DAE,KTEM)
C
C  FIND STIFFNESS MATRIX.
C
      CALL STIFF(PP,DAE,C,N1,NTM1,KTEM,K1)
      IF (NRESP .EQ. 0  .AND.  IPRNT1 .NE. 1)
     1CALL WRITE (C,N1,N1,6HSTIFIF,K1)
      IF (IPRNT1 .EQ. 1)             CALL WRITE (C,N1,N1,6HSTIF F,K1)
C
C  RESTRAIN SYSTEM WITH TOP SPRING AND BOTTOM FIXITY.
C
      DO 731 I=1,NX
      DO 731 J=1,N1
  731 C(I,J) = C(I+1,J)
      DO 732 J=1,NX
      DO 732 I=1,NX
  732 C(I,J) = C(I,J+1)
      C(NX,NX) = C(NX,NX) + EFFK
      IF (NRESP .EQ. 0  .AND.  IPRNT1 .NE. 1)
     1CALL WRITE (C,NX,NX,6HSTIFIK,K1)
      IF (IPRNT1 .EQ. 1)             CALL WRITE (C,NX,NX,6HSTIF K,K1)
      IF (NRESP .EQ. 0  .AND.  IPRNT1 .NE. 1)
```

15

```
      1CALL INV1 (C,B,NX,1,K1)
       IF (IPRNT1 .EQ. 1)       CALL INV1 (C,B,NX,IPRNT1,K1)
       IF (NRESP .EQ. 0   .AND.   IPRNT1 .NE. 1)
      1CALL WRITE (B,NX,NX,6HFLEX-1,K1)
       IF (IPRNT1 .EQ. 1)       CALL WRITE (B,NX,NX,6HFLEX   ,K1)
C
C-----------------------------------------------------------------------
C -----CALCULATE NATURAL FREQ. + MODESHAPES AT INITIAL CONDITIONS.
       IF (NRESP .NE. 0) GO TO 735
       REWIND NFILE1
C
       READ (NFILE1) ((A(I,J),I=1,NX),J=1,NX)
C
       CALL MULTB (B,A,NX,NX,NX,KX,KX)
       CALL EIGEN (A,EPS,B,NX,1.0E-15,KX)
C
       DO 733 I = 1,NX
       A(I,I) = 0.0
  733  EPS(I) = SQRT(1.0/EPS(I)) / 6.2831853
C-----ARRANGE THE FREQ. AND MODESHAPES IN ASCENDING ORDER.
C-----ALSO MAKE FIRST ELEMENT OF EACH MODE POSITIVE.
       IF ( NX .EQ.1 ) GO TO 7340
       NM1 = NX - 1
       DO 7335 J=1,NM1
       EPSMIN = EPS(J)
       NMIN = J
       JP1 = J+1
       DO 7330 I=JP1,NX
       IF ( EPSMIN .LE.EPS(I) ) GO TO 7330
       EPSMIN = EPS(I)
       NMIN = I
 7330  CONTINUE
       IF (NMIN .EQ.J ) GO TO 7335
       EPS(NMIN) = EPS(J)
       EPS(J) = EPSMIN
       DO 7334 K=1,NX
       BKJ = B(K,J)
       B(K,J) = B(K,NMIN)
 7334  B(K,NMIN) = BKJ
 7335  CONTINUE
C
 7340  DO 7345 J=1,NX
       IF ( B(1,J) .GE.0 ) GO TO 7345
       DO 7342 I=1,NX
 7342  B(I,J) = -B(I,J)
 7345  CONTINUE
       CALL WRITE (EPS,NX,1,4HFREQ,KX)
       CALL WRITE (B,NX,NX,4HMODE,KX)
C
       DO 729 I=1,KX
       EPS(I) = 0.0
       DO 729 J=1,KX
       A(I,J)=0.0
  729  B(I,J)=0.0
  735  CONTINUE
C-----------------------------------------------------------------------
C
C  FIND DAMPING MATRIX.
C
       REWIND NFILE1
```

16

```
      READ (NFILE1)((A(I,J),I=1,NX),J=1,NX)                              A=MASS

      IF (STARTT .NE. TS) GO TO 739
      ZZ = 0.
      DO 736 I=1,NTEM
  736 ZZ = ZZ + TABTEM(I,1)
      TEMAVG = ZZ / NTEM
      GAMMA = 3.14 * ZETA * SQRT(FELMOD(TEMAVG,1.,FOTMTL,FCOMP)
     1           / ( RHO * RODL * RODL))
      IF (IFSI .EQ. 1) GO TO 737
      GAMMA = .01204 * GAMMA
  737 DO 738 I=1,NX
      DO 738 J=1,NX
      B(I,J) = 0.0
  738 B(I,J) = GAMMA * A(I,J)                                             B=DAMP
      GO TO 741
  739 GAMMA = (1.0 + DFACT) * GAMMA
      DO 740 I=1,NX
      DO 740 J=1,NX
  740 B(I,J) = GAMMA * A(I,J)                                             B=DAMP
  741 CONTINUE
      IF (NRESP .EQ. 0  .AND.  IPRNT1 .NE. 1)
     1CALL WRITE (B,NX,NX,6HDAMPIN,K1)
      IF (IPRNT1 .EQ. 1)              CALL WRITE (B,NX,NX,6H  DAMP,K1)
C
C
C  DETERMINE INTEGRATION TIME STEP USING 0.1 THE SHORTEST SYSTEM NATURAL
C  PERIOD, WHICH IS, IN TURN, OBTAINED FROM THE LARGEST AND SMALLEST
C  DIAGONAL ELEMENTS, RESP., OF THE STIFFNESS AND MASS MATRICES.
C
      IF (IFTD .NE. 0) GO TO 750
      CMAX = C(1,1)
      DO 744 I=2,NX
      IF (C(I,I) .LE. CMAX) GO TO 744
      CMAX = C(I,I)
  744 CONTINUE
C
       TD =  0.05   * SQRT(AMIN/CMAX) * 6.28319
C
C  PERFORM TRANSIENT RESPONSE ANALYSIS.
C
  750 CALL TRI (A,B,C,TABT,TABF,XDD,XD,XD,X,XDDMAX,XDDMIN,XDMAX,XDMIN,
     1             XMAX,XMIN,STARTT,TD,ENDT,TS,DD,TDDMAX,TDDMIN,TXDMAX,
     2             TXDMIN,TXMAX,TXMIN,IPRNT2,NX,NTF,K1,NFILE1,NFILE2,
     3             IPRNT1)
C
      NRESP = NRESP + 1
      STARTT = ENDT
      GO TO 500
C
C  FIND AND PRINT VECTORS OF AXIAL LOAD,STRAIN,STRAIN RATE, AND
C  CLADDING HOOP STRESS .VS. TIME.
C  PRINT MINIMUMS/MAXIMUMS OF ACCELERATION, VELOCITY, DISPLACEMENT, AND
C  AXIAL LOAD, STRAIN, STRAIN RATE, AND CLADDING HOOP STRESS.
C
  600 REWIND NFILE1
      REWIND NFILE2
      READ (NFILE1) ((A(I,J),I=1,NX),J=1,NX)                              A=MASS
      REWIND NFILE1
```

```
C
      WRITE (NOT,2004) NRESP
 2004 FORMAT (1H1 25(/) 30X
     1         *RESPONSE ROUTINE WAS CALLED --*,I7, *  -- TIMES*)
C
C
      DO 820 I=1,NX
      DO 810 J=1,NX
  810 B(I,J) = 0.0
      M = I
      DO 820 J=M,NX                                                      B=LODSUM
  820 B(I,J) = 1.0
C
      CALL MULTB (B,A,NX,NX,NX,K1,K1)
C
C * NOTE *  LOADS CAN BE FOUND NOW AS, L(T) = -A*XDD(T).
C
C  FIND QUANTITIES AT TIME = TS.
C
      READ (NFILE2) T,(CF(J),J=1,NX),(XDD(I),XD(I),X(I),I=1,NX)
      STARTT = T
      SPAN1 = PP(3) - PP(1)
      SPANX = PP(N1) - PP(NX)
      GO TO 642
C
  640 READ (NFILE2) T,(CF(J),J=1,NX),(XDD(I),XD(I),X(I),I=1,NX)
  642 DO 860 I=1,NX
      AXLOAD(I) = 0.0
      DO 855 J=1,NX
  855 AXLOAD(I) = AXLOAD(I) - A(I,J) * XDD(J)
      ASTRES(I) = AXLOAD(I) / CSAREA
C 860 CHS(I) = - (RMELT(I)/CLACT) * ASTRES(I)                              OLD
  860 CHS(I) = - (  RODR  /CLACT) * ASTRES(I)                              NEW
      EPS(1) = X(2) / SPAN1
      EPS(NX) = (X(NX)-X(NX1))/SPANX
      EPSDOT(1) = XD(2) / SPAN1
      EPSDOT(NX) = (XD(NX)-XD(NX1))/SPANX
      DO 870 I=2,NX1
      SPAN = PP(I+2) - PP(I)
      EPS(I) = (X(I+1)-X(I-1))/SPAN
  870 EPSDOT(I) = (XD(I+1)-XD(I-1))/SPAN
C
      WRITE (NFILE1) T,(AXLOAD(I),ASTRES(I),EPS(I),EPSDOT(I),CHS(I),
     *                I=1,NX)
C
      IF (T .GT. STARTT) GO TO 900
C
C  SET MINIMUM/MAXIMUM VALUES AND CORRESPONDING TIMES OF OCCURRENCE.
C
      DO 880 I=1,NX
       AXMAX(I) = AXLOAD(I)
       AXMIN(I) = AXLOAD(I)
      ASMAX(I) = ASTRES(I)
      ASMIN(I) = ASTRES(I)
        EMAX(I) = EPS(I)
        EMIN(I) = EPS(I)
       EDMAX(I) = EPSDOT(I)
       EDMIN(I) = EPSDOT(I)
      CHSMAX(I) = CHS(I)
      CHSMIN(I) = CHS(I)
```

18

```
      TAXMAX(I) = STARTT
      TAXMIN(I) = STARTT
      TASMAX(I) = STARTT
      TASMIN(I) = STARTT
       TEMAX(I) = STARTT
       TEMIN(I) = STARTT
      TEDMAX(I) = STARTT
      TEDMIN(I) = STARTT
      TCHMAX(I) = STARTT
  880 TCHMIN(I) = STARTT
      GO TO 960
C
  900 DO 950 I=1,NX
      IF (AXLOAD(I) .LE. AXMAX(I)) GO TO 925
       AXMAX(I) = AXLOAD(I)
      TAXMAX(I) = T
  925 IF (AXLOAD(I) .GE. AXMIN(I)) GO TO 927
       AXMIN(I) = AXLOAD(I)
      TAXMIN(I) = T
C
  927 IF (ASTRES(I) .LE. ASMAX(I)) GO TO 928
       ASMAX(I) = ASTRES(I)
      TASMAX(I) = T
  928 IF (ASTRES(I) .GE. ASMIN(I)) GO TO 930
       ASMIN(I) = ASTRES(I)
      TASMIN(I) = T
C
  930 IF (EPS(I) .LE. EMAX(I)) GO TO 935
       EMAX(I) = EPS(I)
      TEMAX(I) = T
  935 IF (EPS(I) .GE. EMIN(I)) GO TO 940
       EMIN(I) = EPS(I)
      TEMIN(I) = T
C
  940 IF (EPSDOT(I) .LE. EDMAX(I)) GO TO 945
       EDMAX(I) = EPSDOT(I)
      TEDMAX(I) = T
  945 IF (EPSDOT(I) .GE. EDMIN(I)) GO TO 947
       EDMIN(I) = EPSDOT(I)
      TEDMIN(I) = T
C
  947 IF (CHS(I) .LE. CHSMAX(I)) GO TO 948
      CHSMAX(I) = CHS(I)
      TCHMAX(I) = T
  948 IF(CHS(I) .GE. CHSMIN(I)) GO TO 950
       CHSMIN(I) = CHS(I)
       TCHMIN(I) = T
  950 CONTINUE
C
  960 IF (T .LT. TE) GO TO 840
C
C  PRINT RESULTS EVERY NWRITE * TD IN TIME.
C
      REWIND NFILE1
      REWIND NFILE2
      IF (IPRNT2 .NE. 1) GO TO 1050
C
  962 NWM1 = NWRITE - 1
      GO TO 965
  963 NWM1 = NWM1 - 1
```

```
965 DC 970 NPR=1,NWM1
      READ (NFILE1) T
      READ (NFILE2) TX
      IF (T .EQ. TS)  GO TO 980
      IF (T .GE. (TE-1.E-6)) GO TO 980
970 CONTINUE
      READ (NFILE1) T,(AXLOAD(I),ASTRES(I),EPS(I),EPSDOT(I),CHS(I),
     *                I=1,NX)
      READ (NFILE2) TX,(QF(J),J=1,NX),(XDD(I),XD(I),X(I),I=1,NX)
                                                          NERROR = 9
      IF (T .NE. TX) GO TO 999
      GO TO 990
980 BACKSPACE NFILE1
      BACKSPACE NFILE2
      READ (NFILE1) T,(AXLOAD(I),ASTRES(I),EPS(I),EPSDOT(I),CHS(I),
     *                I=1,NX)
      READ (NFILE2) TX,(QF(J),J=1,NX),(XDD(I),XD(I),X(I),I=1,NX)
                                                          NERROR = 10
      IF (T .NE. TX) GO TO 999
990 WRITE (NOT,2050) T
      DO 995 I=1,NX
995 WRITE (NOT,2055) I,XDD(I),XD(I),X(I),AXLOAD(I),ASTRES(I),
     *                EPS(I),EPSDOT(I),CHS(I)
      IF (T .EQ. TS) GO TO 963
      IF (T .GE. (TE-1.E-6)) GO TO 1050
      GO TO 962
C
C   PRINT MINIMUM/MAXIMUM VALUES AND THE CORRESPONDING TIMES.
C
1050 DO 1060 I=1,NX
      XMAX(I) = XMAX(I) + QO(I)
1060 XMIN(I) = XMIN(I) + QO(I)
      DO 1110 MM=1,8
      NXE = 0
1100 NXS = NXE + 1
      NXE = NX
      IF ((NXE-NXS) .GT. (NLPP-11)) NXE = NXS + (NLPP-11)
      WRITE (NOT,2001)
      IF (MM .EQ. 1) WRITE (NOT,2010) (I,TDDMAX(I),XDDMAX(I),
     *                        TDDMIN(I),XDDMIN(I),I=NXS,NXE)
      IF (MM .EQ. 2) WRITE (NOT,2020) (I,TXDMAX(I),XDMAX(I),
     *                        TXDMIN(I),XDMIN(I), I=NXS,NXE)
      IF (MM .EQ. 3) WRITE (NOT,2030) (I,TXMAX(I), XMAX(I),
     *                        TXMIN(I), XMIN(I), I=NXS,NXE)
      IF (MM .EQ. 4) WRITE (NOT,2060) (I,TAXMAX(I),AXMAX(I),
     *                        TAXMIN(I),AXMIN(I), I=NXS,NXE)
      IF (MM .EQ. 5) WRITE (NOT,2065) (I,TASMAX(I),ASMAX(I),
     *                        TASMIN(I),ASMIN(I),I=NXS,NXE)
      IF (MM .EQ. 6) WRITE (NOT,2070) (I, TEMAX(I), EMAX(I),
     *                        TEMIN(I), EMIN(I), I=NXS,NXE)
      IF (MM .EQ. 7) WRITE (NOT,2080) (I,TEDMAX(I),EDMAX(I),
     *                        TEDMIN(I),EDMIN(I), I=NXS,NXE)
      IF (MM .EQ. 8) WRITE (NOT,2090) (I,TCHMAX(I),CHSMAX(I),
     *                        TCHMIN(I),CHSMIN(I),I=NXS,NXE)
      IF (NX .GT. NXE) GO TO 1100
1110 CONTINUE
C
      WRITE (NOT,2001)
      RETURN
999 WRITE (NOT,2101) NERROR
```

20

```
      STOP
C
 2001 FORMAT (1H1)
 2050 FORMAT (1H1/40X,*ROD TRANSIENT RESPONSES TO RIA THERMAL LOADING*
     1         ///50X,*TIME = *,F12.5,///2X,*NODE*,3X,*ACCELERATION*,
     2          4X,*VELOCITY*, 6X,*DISPLACEMENT*,4X*AXIAL LOAD*,3X,
     3         *AXIAL STRESS*,3X,*AXIAL STRAIN*,2X,*AX STRAIN RATE*,
     4         2X,*CLAD HOOP STRESS* //)
 2055 FORMAT (3X,I3, 8(2X 1PE13.6))
 2010 FORMAT (// 20X,8H TIME OF,28X,8H TIME OF
     1         79X,4HNODE,6X,10H MAX ACCEL,6X,10H MAX ACCEL,
     2          10X,10H MIN ACCEL,6X,10H MIN ACCEL
     3          // (10X,I3,0PF16.6,1PE18.8,0PF18.6,1PE18.8))
 2020 FORMAT (// 20X,8H TIME OF, 28X,8H TIME OF
     1         79X,4HNODE,7X,8H MAX VEL, 8X,8H MAX VEL,
     2          12X,8H MIN VEL, 8X,8H MIN VEL
     3          // (10X,I3,0PF16.6,1PE18.8,0PF18.6,1PE18.8))
 2030 FORMAT (// 20X,8H TIME OF, 28X,8H TIME OF
     1         79X,4HNODE, 7X,9H MAX DISP, 7X,9H MAX DISP,
     2          10X,9H MIN DISP, 8X,9H MIN DISP
     3          // (10X,I3,0PF16.6,1PE18.8,0PF18.6,1PE18.8))
 2060 FORMAT (//20X,8H TIME OF,28X,8H TIME OF
     1         79X,4HNODE,6X,10HMAX AXLOAD,6X,10HMAX AXLOAD,
     2          10X,10HMIN AXLOAD,6X,10HMIN AXLOAD
     3          // (10X,I3,0PF16.6,1PE18.8,0PF18.6,1PE18.8))
 2065 FORMAT (//21X,8H TIME OF,38X,8H TIME OF
     1         79X,4HNODE,5X,16HMAX AXIAL STRESS,5X,16HMAX AXIAL STRESS,
     2          9X,16HMIN AXIAL STRESS,5X,16HMIN AXIAL STRESS
     3          // (10X,I3,5X,0PF16.6,5X,1PE16.8,9X,0PF16.6,5X,1PE16.8))
 2070 FORMAT (//20X,8H TIME OF,28X,8H TIME OF
     1         79X,4HNODE,6X,10HMAX STRAIN,6X,10HMAX STRAIN,
     2          10X,10HMIN STRAIN,6X,10HMIN STRAIN
     3          // (10X,I3,0PF16.6,1PE18.8,0PF18.6,1PE18.8))
 2080 FORMAT (//22X,8H TIME OF,38X, 8H TIME OF
     1         79X,4HNODE,6X,15HMAX STRAIN RATE,6X,15HMAX STRAIN RATE,
     2          10X,15HMIN STRAIN RATE,6X,15HMIN STRAIN RATE
     3          // (10X,I3,5X,0PF16.6,5X,1PE16.8,9X,0PF16.6,5X,1PE16.8))
 2090 FORMAT (//22X,8H TIME OF,38X, 8H TIME OF
     1         79X,4HNODE,6X,15HMAX CLAD STRESS,6X,15HMAX CLAD STRESS,
     2          10X,15HMIN CLAD STRESS,6X,15HMIN CLAD STRESS
     3          // (10X,I3,5X,0PF16.6,5X,1PE16.8,9X,0PF16.6,5X,1PE16.8))
 2101 FORMAT (1H1,//////20X,*ERROR ENCOUNTERED IN SUBROUTINE FRIDA*
     *                    //30X,*AT NERROR = *,I3)
      END
      FUNCTION FELMOD(FTEMP,FRADEN,FOTMTL,FCOMP)
C
      COMMON  /PHYPRO    / FTMELT,FHEFUS,CTMELT,CHEFUS,CTRANB,
     #                     CTRANE,CTRANZ,FDELTA,BU      ,COMP
C
C     ***************************************************************
C
C FTEMP  = FUEL TEMPERATURE IN DEGREES KELVIN.
C FTEMP  = FUEL TEMPERATURE IN DEGREES KELVIN.
C FRADEN = FRACTIONAL DENSITY OF FUEL MATERIAL.
C FOTMTL = OXYGEN TO METAL RATIO (EG. = 2.0)
C     ***************************************************************
C
C
C
      YS = 23.34E10 * (1. - (2.752*(1. - FRADEN)))*(1. -1.0915E-04*
     #FTEMP)
      UFELMD = 6.0E09
```

```
      IF (FTEMP .LT. 1.6E03) GO TO 10
      UFELMD = UFELMD + YS * (FTEMP - 1.6E03)/6.0526E03
      IF (FTEMP .LT. 3113.15) GO TO 10
      YS = 0.0
      UFELMD = 0.0
      GO TO 60
   10 UNSTOC = ABS(FOTMTL - 2.0)
      IF (UNSTOC .GT. 1.0E-03) GO TO 20
      IF (FCOMP .LT. 1.0E-03) GO TO 80
   20 B = 1.34
      IF (FOTMTL .LT. 2.0) B = 1.75
      Y = YS * EXP(-B*UNSTOC) * (1. + 0.15 * FCOMP)
      UFELMD = (UFELMD**2 + (Y - YS)**2)**0.5
      YS = Y
   80 FELMOD = YS
      RETURN
      END
      FUNCTION  FTHEXP(FTEMP,FACMOT)
C
       COMMON  /PHYPRO   / FTMELT,FHEFUS,CTMELT,CHEFUS,CTRANB,
     #                     CTRANE,CTRANZ,FDELTA,BU     ,COMP
C
C     ****************************************************************
C FCOMP  = FRACTION OF PLUTONIUM IN THE FUEL (EG. = 0.0).
C FACMOT = FRACTION OF THE WAY THROUGH MELTING (EG. = 0.0).
C     ****************************************************************
C
C
      DATA FDELTA / 1. /
      UEX(T) = -4.972E-4 + 7.107E-6*T + 2.581E-9*T*T + 1.140E-13*T**3
      PEX(T) = -3.9735E-4 + 8.4955E-6*T + 2.1513E-9*T*T +
     #         3.7143E-16*(T**3)
      T1 = FTEMP- 273.15
      R=FACMOT
      C1 = COMP/100.
C
      TM=FTMELT - 273.15
      IF (T1 .LT. (TM-1.E-10)) GO TO 10
      IF (T1 .GE. (TM-1.E-10) .AND. T1 .LE. (TM+FDELTA)) GO TO 30
      IF (T1 .GE. (TM+FDELTA)) GO TO 50
   10 IF (COMP .GT. 0.0) GO TO 15
      FTHEXP = UEX(T1)
      GO TO 100
   15 FTHEXP = (1. - C1)*UEX(T1) + C1*PEX(T1)
      GO TO 100
   30 IF (COMP .GT. 0.0) GO TO 35
      FTHEXP = UEX(TM) + R*3.096E-2
      GO TO 100
   35 FTHEXP = (1. - C1)*UEX(TM) + C1*PEX(TM) + R*3.096E-2
      GO TO 100
   50 IF (COMP .GT. 0.0) GO TO 55
      FTHEXP = UEX(TM) + 3.096E-2 + (3.5E-5*(T1 - TM))
      GO TO 100
   55 FTHEXP = PEX(TM) + 3.096E-2 + (3.5E-5*(T1 - TM))
  100 RETURN
      END
      SUBROUTINE INV1 (A,Z,N,IPRINT,KR)
      DIMENSION A(1),Z(1),IX(30),B(30),G(30),DETR(30)
      DATA NIT,NOT 75,6/
C
```

```
C  MATRIX INVERSION (A**-1 = Z).  BORDERING METHOD.
C
C  IF PRINT OPTION IS TURNED ON, THE INVERSION CHECK Z*A AND THE DETERMINANT
C  RATIO DET(I+1)/DET(I) ARE PRINTED.  DET(I) IS THE DETERMINANT OF THE FIRST
C  I BY I SUB-MATRIX OF A.
C
C  MATRICES A,Z MAY SHARE SAME CORE LOCATIONS. (Z*A CHECK IS THEN INVALID).
C
C        SUBROUTINE ARGUMENTS
C      A = INPUT MATRIX TO BE INVERTED.  SIZE (N,N).
C      Z = OUTPUT RESULT MATRIX.  SIZE (N,N).
C      N = INPUT SIZE OF MATRICES A,Z.  MAXIMUM IS DIMENSIONED SIZE.
C IPRINT = INPUT PRINT LOGIC VARIABLE. (=1,PRINT). (=0,2,ETC,NO PRINT).
C     KR = INPUT ROW DIMENSION OF A,Z IN CALLING PROGRAM.
C
 1000 FORMAT (1H1)
 2000 FORMAT (//10X,10(7X,1H(,I2,1H)))
 2001 FORMAT (//10X,*SUBROUTINE INV1 HAS CALCULATED THE DATA BELOW*
     *        ///10X,*THE DETERMINANT RATIOS DET(I+1)/DET(I) ARE*
     *        // (13X,10(1PE11.3)))
 2002 FORMAT (///10X,"THE (A**-1)*(A) INVERSION CHECK GIVES"
     *         ///10X,*THE DIAGONAL ELEMENTS ARE*//(13X,10F1[illegible].8))
 2003 FORMAT (// 10X,*THE MAXIMUM OFF-DIAGONAL ELEMENT IS*,
     *         E11.3,2X,4HAT (I3,1H,I3,1H))
C
      DO 160 I=2,N
  160 IX(I) = I
C
      DO 190 I=1,N
      IF (A(I) .NE. 0.) GO TO 220
  190 CONTINUE
      GO TO 999
C
  220 DETR(1) = A(I)
      Z(I) = 1./A(I)
      IF (N.EQ.1) RETURN
C
      IX(I) = 1
      IX(1) = I
C
      DO 630 L=2,N
      K = L
      L1 = L-1
  250 S = 0.0
        MIXL = KR * (IX(L) - 1)
        LL = IX(L) + MIXL
      DO 450 I=1,L1
        MIXI = KR * (IX(I) - 1)
        LI = IX(L) + MIXI
      B(I) = 0.0
      G(I) = 0.0
      DO 440 J=1,L1
        MIXJ = KR * (IX(J) - 1)
        IJ = IX(I) + MIXJ
        JL = IX(J) + MIXL
      B(I) = B(I) - Z(IJ) * A(JL)
        JI = IX(J) + MIXI
        LJ = IX(L) + MIXJ
  440 G(I) = G(I) - A(LJ)*Z(JI)
  450 S = S + A(LI)*B(I)
```

23

```
      AL = A(LL) + S
      IF (A(LL) .EQ. 0.0) GO TO 480
      ALBAR = ABS(AL / A(LL))
      GO TO 490
  480 ALBAR = ABS(AL)
  490 IF (ALBAR .GE. .1E-6) GO TO 550
C
      K = K+1
      IF (K .GT. N) GO TO 540
      IX L = IX(L)
      IX(L) = IX(K)
      IX(K) = IX L
      GO TO 250
  540 IF (ALBAR .GE. .1E-8) GO TO 550
      GO TO 999
  550 Z(LL) = 1./AL
      DETR(L) = AL
      DO 570 I=1,L1
      IL = IX(I) + MIXL
      LI = IX(L) + KR * (IX(I) - 1)
      Z(IL) = B(I) * Z(LL)
      Z(LI) = G(I) * Z(LL)
      DO 570 J=1,L1
      IJ = IX(I) + KR * (IX(J) - 1)
  570 Z(IJ) = Z(IJ) + G(J) * Z(IL)
  630 CONTINUE
C
      XOFF = 0.0
      DO 720 I=1,N
      DO 710 J=1,N
      X = 0.0
      KJA = KR * (J-1)
      DO 703 K=1,N
      IK = I + KR*(K-1)
      KJ = K + KJA
  703 X = X + Z(IK) * A(KJ)
      IF (I .NE. J) GO TO 705
      G(I) = X
      GO TO 710
  705 IF (ABS(X) .LT. ABS(XOFF)) GO TO 710
      XOFF = X
      IOFF = I
      JOFF = J
  710 CONTINUE
  720 CONTINUE
C
      IF (IPRINT .NE. 1) GO TO 989
      WRITE (NOT,1000)
      WRITE (NOT,2000)  (JC,JC=1,10)
      WRITE (NOT,2001)  (DETR(I),I=1,N)
      WRITE (NOT,2002)  (G(I)   ,I=1,N)
      WRITE (NOT,2003) XOFF,IOFF,JOFF
  989 RETURN
  999 WRITE (NOT,2004)
 2004 FORMAT (1H1,10(/),20X,*ERROR IN SUBROUTINE INV1*)
      STOP
      END
      SUBROUTINE MASS (PP,DMASS,Z,NPP,NDM,KDM,KZ)
      DIMENSION  PP(1),  DMASS(KDM,1), Z(KZ,1)
      DATA NIT,NOT/5,6/
```

24

```
C
C   CALCULATE MASS MATRIX FOR A BEAM.  LINEAR VELOCITY FUNCTION ASSUMED
C   BETWEEN CONSECUTIVE PANEL (NODE) POINTS.
C
C   TRANSLATIONS AT THE PANEL POINTS ARE THE GENERALIZED COORDINATES.
C
C   INPUT IS DISTRIBUTED MASS.  THE DISTRIBUTED DATA MAY NOT EXCEED THE
C   PANEL POINT LIMITS (BEAM ENDS).
C
C         SUBROUTINE ARGUMENTS
C      PP = INPUT VECTOR OF PANEL (NODE) POINTS.  SIZE (NPP).
C   DMASS = INPUT MATRIX OF DISTRIBUTED MASS, STRAIGHT LINE SEGMENT DATA.
C           SIZE (NDM,4).
C                COL 1 = X COORD. AT SEGMENT END 1.
C                COL 2 = X COORD. AT SEGMENT END 2.
C                COL 3 = MASS/UNIT DISTANCE AT SEGMENT END 1.
C                COL 4 = MASS/UNIT DISTANCE AT SEGMENT END 2.
C       Z = OUTPUT TRI-DIAGONAL MASS MATRIX.  SIZE(NPP,NPP).
C     NPP = INPUT NUMBER OF PANEL POINTS.  SIZE OF VECTOR PP, MATRIX Z.
C     NDM = INPUT NUMBER OF SEGMENTS (ROWS) IN DMASS.
C     KDM = INPUT ROW DIMENSION OF DMASS IN CALLING PROGRAM.
C      KZ = INPUT ROW DIMENSION OF Z IN CALLING PROGRAM.
C
      DO 10 I=1,NPP
      DO 10 J=1,NPP
   10 Z(I,J) = 0.0
      NBAYS = NPP-1
      DO 90 I=1,NDM
      X1 = DMASS(I,1)
      X2 = DMASS(I,2)
      V1 = DMASS(I,3)
      V2 = DMASS(I,4)
                                                          NERROR = 1
      IF (X1.LT.PP(1) .OR. X2.GT.PP(NPP) .OR. X1.GE.X2) GO TO 999
      DO 32 K=1,NBAYS
      IF (X1 .LT. PP(K+1)) GO TO 34
   32 CONTINUE
   34 XP = X1
      VP = V1
   36 IF (X2 .LE. PP(K+1)) GO TO 38
      XQ = PP(K+1)
      VQ = V1 + (XQ-X1)*(V2-V1)/(X2-X1)
      GO TO 39
   38 XQ = X2
      VQ = V2
   39 BAYL = PP(K+1) - PP(K)
      SEGL = XQ - XP
      HP = (XP-PP(K)) / BAYL
      HQ = (XQ-PP(K)) / BAYL
      VPVQ = VP + VQ
      F1 = SEGL * VPVQ/2.
      F2 = SEGL * (VPVQ*(HP+HQ) + VP*HP + VQ*HQ) / 6.
      F3 = SEGL * (VPVQ*(HP+HQ)**2 + 2.*(VP*HP**2 + VQ*HQ**2))/12.
      L = K+1
      Z(K,K) = Z(K,K) + F1 - 2.*F2 + F3
      Z(K,L) = Z(K,L) + F2 - F3
      Z(L,L) = Z(L,L) + F3
      IF (X2 .LE. PP(K+1)) GO TO 90
      K = K+1
      XP = XQ
```

```
      VP = VC
      GO TO 36
   90 CONTINUE
      DO 110 K=1,NBAYS
  110 Z(K+1,K) = Z(K,K+1)
C
C TOTAL MASS PROPERTIES...FOR CHECKOUT PURPOSES.
C
      TM = 0.
      TP = 0.
      TI = 0.
      DO 100 I=1,NPP
      DO 100 J=1,NPP
      TM = TM + Z(I,J)
      TP = TP + Z(I,J) * PP(J)
  100 TI = TI + PP(I)*Z(I,J)*PP(J)
      CG = TP/TM
      TI = TI - TM*CG**2
      WRITE (NOT,2001) TM,CG,TI
 2001 FORMAT (1H1,10(/),48X,*SUBROUTINE MASS* //
     *        41X,    *COMPUTES THE TOTAL PROPERTIES* ///
     *    43X,*M   = *,E15.8,//43X,*XCG = *,E15.8,//43X,*ICG = *,E15.8)
      RETURN
  999 WRITE (NOT,2002) NERROR
 2002 FORMAT (1H1,20X,*ERROR ENCOUNTERED IN SUBROUTINE MASS*
     *        ///30X,*AT NERROR = *,I3)
      END
      SUBROUTINE MULTB (A,BZ,NRA,NRB,NCB,KA,KBZ)
      DIMENSION A(KA,1),BZ(KBZ,1), W(30)
C
C MATRIX MULTIPLICATION.  A * B = Z.
C
C USES TWO WORK SPACES.  RESULT Z IS PLACED IN B.
C BZ MUST BE DIMENSIONED LARGE ENOUGH IN CALLING PROGRAM TO CONTAIN THE
C LARGER OF B OR Z.
C
C       SUBROUTINE ARGUMENTS
C    A = INPUT MATRIX.  SIZE (NRA,NRB).
C   BZ = INPUT MATRIX.  SIZE (NRB,NCB).
C      = OUTPUT RESULT MATRIX.  SIZE (NRA,NCB).  MAX.=DIMENSION SIZE OF W.
C  NRA = INPUT NUMBER OF ROWS OF MATRICES A,Z.
C  NRB = INPUT NUMBER OF ROWS OF MATRIX B, COLUMNS OF MATRIX A.
C  NCB = INPUT NUMBER OF COLUMNS OF MATRICES B,Z.
C   KA = INPUT ROW DIMENSION OF A  IN CALLING PROGRAM.
C  KBZ = INPUT ROW DIMENSION OF BZ IN CALLING PROGRAM.
C
      DO 40 J=1,NCB
      DO 30 I=1,NRA
      W(I) = 0.0
      DO 30 K=1,NRB
   30 W(I) = W(I) + A(I,K) * BZ(K,J)
      DO 40 I=1,NRA
   40 BZ(I,J) = W(I)
      RETURN
      END
      SUBROUTINE READ (A,NR,NC,KR,KC)
      DIMENSION A(KR,1),X(4)
      DATA NIT,NOT/5,6/
C
 1001 FORMAT (A6,I4,I5)
```

26

```
 1002 FORMAT (2I5,4E15.0)
 2001 FORMAT (1H1//1X,*CARD INPUT MATRIX*,2X,A6,2X,1H(I4,2H X I4,2H ) )
 2002 FORMAT (//19H CARD INPUT MATRIX,2X,A6, 2X 1H( I4,2H X I4,2H )
     *           3X 9HCONTINUED //)
 2004 FORMAT (1X 2I5,4E17.8)
 2005 FORMAT (13H0END OF READ.)
 2006 FORMAT (1H1)
 2007 FORMAT (1H1,////10X,*ERROR ENCOUNTERED IN SUBROUTINE READ*)
      READ (NIT,1001) ANAME,NR,NC
      WRITE (NOT,2001) ANAME,NR,NC
      IF (NR.GT.KR .OR. NC.GT.KC) GO TO 999
      NLINE = 0
      DO 105 I=1,NR
      DO 105 J=1,NC
  105 A(I,J) = 0.0
  110 READ (NIT,1002) I,JS,X
      IF (I.EQ.0 .AND. JS.EQ.0) GO TO 300
      IF (I.LE.0 .OR. I.GT.NR .OR. JS.LE.0 .OR. JS.GT.NC) GO TO 999
      JE = JS+3
      IF (JE.LE.NC) GO TO 115
      JX = NC-JS+2
      DO 112 J=JX,4
      IF (X(J) .NE. 0.) GO TO 999
  112 CONTINUE
      JE = NC
  115 N = 0
      DO 120 J=JS,JE
      N = N+1
  120 A(I,J) = X(N)
      NLINE = NLINE+1
      IF (NLINE.LE.47) GO TO 125
      WRITE (NOT,2006)
      WRITE (NOT,2002) ANAME,NR,NC
      NLINE = 1
  125 WRITE (NOT,2004) I,JS,(A(I,J),J=JS,JE)
      GO TO 110
  300 WRITE (NOT,2005)
      RETURN
  999 WRITE (NOT,2007)
      STOP
      END
      SUBROUTINE STIFF (PP,DAE,Z,NPP,NDAE,KDAE,KZ)
      DIMENSION PP(1),DAE(KDAE,1),Z(KZ,1)
      DATA NIT,NOT/5,6/
C
C     CALCULATE STIFFNESS MATRIX (FREE-FREE) FOR A LONGITUDINAL ROD.
C     CONSTANT FORCE ASSUMED BETWEEN CONSECUTIVE PANEL (NODE) POINTS.
C     TRANSLATIONS AT THE PANEL POINTS ARE THE GENERALIZED COORDINATES.
C
C     INPUT IS DISTRIBUTED STIFFNESS (AE).  SUBROUTINE APPLICABLE ALSO TO
C     THE TORSION PROBLEM.  THE AE DATA MUST START AND END AT THE ENDS OF
C     THE ROD.
C
C         SUBROUTINE ARGUMENTS
C      PP = INPUT VECTOR OF PANEL (NODE) POINTS.  SIZE (NPP).
C     DAE = INPUT MATRIX OF DISTRIBUTED STIFFNESS, STRAIGHT LINE SEGMENT DATA.
C           SIZE (NDAE,4).
C                COL 1 = X COORD. AT SEGMENT END 1.
C                COL 2 = X COORD. AT SEGMENT END 2.
C                COL 3 = STIFFNESS (AE) AT SEGMENT END 1.
```

```
C             COL 4 = STIFFNESS (AE) AT SEGMENT END 2.
C        Z = OUTPUT STIFFNESS MATRIX.  SIZE (NPP,NPP).
C      NPP = INPUT NUMBER OF PANEL POINTS.  SIZE OF VECTOR PP, MATRIX Z.
C     NDAE = INPUT NUMBER OF SEGMENTS (ROWS) IN DAE.
C     KDAE = INPUT ROW DIMENSION OF DAE IN CALLING PROGRAM.
C       KZ = INPUT ROW DIMENSION OF  Z  IN CALLING PROGRAM.
C
      DO 10 I=1,NPP
      DO 10 J=1,NPP
   10 Z(I,J) = 0.0
      NBAYS = NPP-1
      DO 90 I=1,NDAE
      X1 = DAE(I,1)
      X2 = DAE(I,2)
      V1 = DAE(I,3)
      V2 = DAE(I,4)
                                                          NERROR = 1
      PP1 = PP(1) - .0001
      PPNP = PP(NPP) + .0001
      IF (X1.LT.PP1    .OR. X2.GT.PPNP     .OR. X1.GE.X2) GO TO 999
      DO 32 K=1,NBAYS
      IF (X1 .LT. PP(K+1)) GO TO 34
   32 CONTINUE
   34 XP = X1
      VP = V1
   36 IF (X2 .LE. PP(K+1)) GO TO 38
      XQ = PP(K+1)
      VQ = V1 + (XQ-X1)*(V2-V1)/(X2-X1)
      GO TO 39
   38 XQ = X2
      VQ = V2
   39 B = (VQ-VP)/(XQ-XP)
      IF (B .EQ. 0.) GO TO 55
      Z(K,K) = Z(K,K) + ALOG(VQ/VP) / B
      GO TO 70
   55 Z(K,K) = Z(K,K) + (XQ-XP)/VP
C
   70 IF (X2 .LE. (PP(K+1)+.0001)) GO TO 90
      K = K+1
      XP = XQ
      VP = VQ
      GO TO 36
   90 CONTINUE
      STOR2 = Z(1,1)
      Z(1,1) = 0.0
      DO 120 K=1,NBAYS
      L = K+1
      STOR1 = 1./STOR2
      STOR2 = Z(L,L)
      Z(K,K) = Z(K,K) + STOR1
      Z(K,L) = - STOR1
      Z(L,K) = -STOR1
  120 Z(L,L) = STOR1
      RETURN
  999 WRITE (NOT,2002) NERROR
 2002 FORMAT (1H1,20X,*ERROR ENCOUNTERED IN SUBROUTINE STIFF*
     *            ///30X,*AT NERROR = *,I3)
       STOP
      END
      SUBROUTINE TRI (A,B,C,TABT,TABF,XDD,X0,XD,X,XDDMAX,XDDMIN,XDMAX,
```

28

```
     1              XDMIN,XMAX,XMIN,STARTT,DELTAT,ENDT,TS,CO,TDDMAX,
     2              TDDMIN,TXDMAX,TXDMIN,TXMAX,TXMIN,IPRNT2,NX,NTF,
     3              KA,NTAPE1,NTAPE2,IPRNT1)
      DIMENSION  A(KA,1),B(KA,1),C(KA,1),TABT(KA,1),TABF(KA,1),
     1          XDO(KA),XO(KA),XDD(30),XD(KA),X(KA),F(30),O:(30),Q(30),
     2           XDDMAX(KA), XDMAX(KA), XMAX(KA), P(4), OO(KA), XTOT(30),
     3           XDDMIN(KA),  XDMIN(KA),  XMIN(KA),
     4           TDDMAX(KA), TXDMAX(KA), TXMAX(KA),
     5           TDDMIN(KA), TXDMIN(KA), TXMIN(KA)
      DATA NIT,NOT/5,6/
C
C     RESPONSE ROUTINE TO SOLVE THE SECOND ORDER DIFFERENTIAL EQUATION
C       (A)XDD + (B)XD + (C)X = F    FOR XDD,XD,X.
C     FOURTH-ORDER RUNGE-KUTTA (GILL MODIFICATION) NUMERICAL INTEGRATION USED.
C
C     VECTOR F IS OBTAINED BY LINEAR INTERPOLATION USING TABT,TABF.
C     MATRICES A,B,C SHOULD NOT SHARE SAME CORE LOCATION (DUE TO MULTB).
C
C     THE RESULTS (T,F,XDD,XD,X) WILL BE WRITTEN ON NTAPE2 EVERY DELTAT.
C
C     CALLS FOLLOWING SUBROUTINES...MULTB.
C
C     MAXIMUM NUMBER OF EQUATIONS SOLVED IS GIVEN BY DIMENSION SIZE OF XDD OR F.
C
C           SUBROUTINE ARGUMENTS (INPUT)
C         A = MATRIX COEFFICIENT OF XDD.  SIZE(NX,NX).
C             ** NOT NEEDED IN VARIABLE LIST FOR THIS VERSION, BUT RETAINED **
C             ** FOR VARIABLE DIMENSIONING.                                  **
C         B = MATRIX COEFFICIENT OF XD.   SIZE(NX,NX).  ** DESTROYED **
C         C = MATRIX COEFFICIENT OF X.    SIZE(NX,NX).  ** DESTROYED **
C      TABT = TABLE OF TIMES FOR FORCE IN TABF.  SIZE(NX,NTF).
C      TABF = TABLE OF FORCES.  SIZE(NX,NTF).
C       XDO = VECTOR OF INITIAL VELOCITIES.      SIZE (NX).
C        XO = VECTOR OF INITIAL RELATIVE DISPLACEMENTS.  SIZE (NX).
C    STARTT = INTEGRATION STEP STARTING TIME.
C    DELTAT = INTEGRATION STEP SIZE.
C      ENDT = INTEGRATION STEP END TIME.
C        TS = STARTING TIME OF TOTAL INTEGRATION PROCEDURE IN CALLING PROGRAM.
C        CO = INITIAL ABSOLUTE DISPLACEMENTS.  SIZE (NX).
C        NX = SIZE OF MATRICES A,B,C (SQUARE), ROW SIZE OF TABT,TABF.
C       NTF = NUMBER OF COLS IN TABT,TABF.
C        KA = ROW DIMENSION OF A,B,C,TABF,TABT IN CALLING PROGRAM.
C    NTAPE1 = LOGICAL UNIT NUMBER OF FILE CONTAINING A(1ST), A**-1(2ND).
C    NTAPE2 = LOGICAL UNIT NUMBER OF FILE CONTAINING RESPONSE OUTPUT.
C
C           SUBROUTINE ARGUMENTS (OUTPUT)
C        XD = VECTOR OF VELOCITIES AT ENDT.      SIZE (NX).
C         X = VECTOR OF RELATIVE DISPLACEMENTS AT ENDT.  SIZE (NX).
C    XDDMAX = VECTOR OF MAXIMUM ACCELERATIONS FOR (STARTT .LE. T .LE. ENDT). (NX).

C    XDDMIN = VECTOR OF MINIMUM ACCELERATIONS FOR (STARTT .LE. T .LE. ENDT). (NX).

C     XDMAX = VECTOR OF MAXIMUM  VELOCITIES    FOR (STARTT .LE. T .LE. ENDT). (NX).
C     XDMIN = VECTOR OF MINIMUM  VELOCITIES    FOR (STARTT .LE. T .LE. ENDT). (NX).
C      XMAX = VECTOR OF MAXIMUM DISPLACEMENTS FOR (STARTT .LE. T .LE. ENDT). (NX).

C      XMIN = VECTOR OF MINIMUM DISPLACEMENTS FOR (STARTT .LE. T .LE. ENDT). (NX).
```

29

```
C   TDDMAX = VECTOR OF TIMES CORRESPONDING TO XDDMAX.
C   TDDMIN = VECTOR OF TIMES CORRESPONDING TO XDDMIN.
C   TXDMAX = VECTOR OF TIMES CORRESPONDING TO XDMAX.
C   TXDMIN = VECTOR OF TIMES CORRESPONDING TO XDMIN.
C    TXMAX = VECTOR OF TIMES CORRESPONDING TO XMAX.
C    TXMIN = VECTOR OF TIMES CORRESPONDING TO XMIN.
C
C
      IF (STARTT .EQ. ENDT ) RETURN
                                                         NERROR = 1
      IF (NX .GT. KA) GO TO 999
C
      IF (IPRNT1 .EQ. 1) WRITE (NOT,2001) STARTT,DELTAT,ENDT
 2001 FORMAT (      3(/),16X,*THE INPUT SCALARS TO SUBROUTINE TRI ARE*,
     1                   / 23X,  10H STARTT = F10.6,
     2                   / 23X,  10H DELTAT = F10.6,
     3                   / 23X,  10H ENDT   = F10.6)
C
      DO 18 I=1,NX
                                                         NERROR = 2
      IF (STARTT .LT. TABT(I,1)) GO TO 999
      DO 12 J=2,NTF
      IF (TABT(I,J-1) .GE. TABT(I,J)) GO TO 14
   12 CONTINUE
      J = NTF+1
   14 IF (ENDT .LE. TABT(I,J-1)) GO TO 18
                                                         NERROR = 3
      GO TO 999
   18 CONTINUE
C
      REWIND NTAPE1
      READ (NTAPE1)
      READ (NTAPE1)((A(I,J),I=1,NX),J=1,NX)                      A=AI
      CALL MULTB (A, B, NX, NX, NX, KA, KA)                      B=AIB
      CALL MULTB (A, C, NX, NX, NX, KA, KA)                      C=AIC
C
      NSTEPS = 0
      T = STARTT
      DO 30 I=1,NX
      DD(I) = 0.0
      C (I) = 0.0
      XD(I) = XD0(I)
   30 X (I) = X0 (I)
      DO 36 I=1,NX
      DO 34 J=1,NTF
      IF (T .LE. TABT(I,J+1) .OR. (J+1) .EQ. NTF)  GO TO 36
   34 CONTINUE
   36 F(I) = TABF(I,J) + (T-TABT(I,J)) * (TABF(I,J+1)-TABF(I,J)) /
     *                                   (TABT(I,J+1)-TABT(I,J))
      DO 38 I=1,NX
      XDD(I) = 0.0
      DO 37 J=1,NX
   37 XDD(I) = XDD(I) + A(I,J)*F(J)- B(I,J)*XD(J) - C(I,J)*X(J)
   38 CONTINUE
      IF (T .GT. TS) GO TO 50
      DO 40 I=1,NX
      XDDMAX(I) = XDD(I)
      XDDMIN(I) = XDD(I)
      TDDMAX(I) = STARTT
```

30

```
      TCDMIN(I) = STARTT
       XDMAX(I) = XD(I)
       XDMIN(I) = XD(I)
      TXDMAX(I) = STARTT
      TXDMIN(I) = STARTT
        XMAX(I) = X(I)
        XMIN(I) = X(I)
       TXMAX(I) = STARTT
   40  TXMIN(I) = STARTT
C
   50 P(1) = .5
      P(2) = 1. - SQRT(.5)
      P(3) = 1. + SQRT(.5)
      P(4) = .5
C
      GO TO 340
C
C   INTEGRATION LOOP....(J=1,HALF STEP), (J=2,HALF STEP AGAIN),
C                        (J=3,FULL STEP), (J=4,END OF STEP).
C   GILL FACTOR = 0.5
C
  100 DO 150 J=1,4
      DO 110 I=1,NX
      Z = XD(I) * DELTAT
      ZD = XDD(I) * DELTAT
      IF (J .EQ. 4) GO TO 105
      R = P(J) * (Z - Q(I))
      RD = P(J) * (ZD - QD(I))
      GO TO 107
  105 R  =  (Z  - 2.* Q(I))/6.
      RD =  (ZD - 2.*QD(I))/6.
  107 X (I) = X (I) + R
      XD(I) = XD(I) + RD
      Q (I) = Q (I) + 3.*R  -P(J)*Z
  110 QD(I) = QD(I) + 3.*RD -P(J)*ZD
      IF (J .NE. 1) GO TO 115
      T = T + .5*DELTAT
      GO TO 130
  115 IF (J .NE. 3) GO TO 140
      NSTEPS = NSTEPS + 1
      T = STARTT + FLOAT(NSTEPS)*DELTAT
  130 DO 136 I=1,NX
      DO 134 K=1,NTF
      IF (T .LE. TABT(I,K+1) .OR. (K+1) .EQ. NTF) GO TO 136
  134 CONTINUE
  136 F(I) = TABF(I,K) + (T-TABT(I,K)) * (TABF(I,K+1) - TABF(I,K)) /
     *                                   (TABT(I,K+1) - TABT(I,K))
  140 DO 150 I=1,NX
      XDD(I) = 0.0
      DO 145 K=1,NX
  145 XDD(I) = XDD(I) + A(I,K)*F(K) -B(I,K)*XD(K) - C(I,K)*X(K)
  150 CONTINUE
C
C   MAXIMUMS AND MINIMUMS.
C
      DO 330 I=1,NX
      IF (XDD(I) .LE. XDDMAX(I)) GO TO 305
      XDDMAX(I) = XDD(I)
      TDDMAX(I) = T
  305 IF (XDD(I) .GE. XDDMIN(I)) GO TO 310
```

31

```
      XDDMIN(I) = XDD(I)
      TDDMIN(I) = T
  310 IF (XD(I) .LE. XDMAX(I)) GO TO 315
       XDMAX(I) = XD(I)
      TXDMAX(I) = T
  315 IF (XD(I) .GE. XDMIN(I)) GO TO 320
       XDMIN(I) = XD(I)
      TXDMIN(I) = T
  320 IF (X(I) .LE. XMAX(I)) GO TO 325
       XMAX(I) = X(I)
      TXMAX(I) = T
  325 IF (X(I) .GE. XMIN(I)) GO TO 330
       XMIN(I) = X(I)
      TXMIN(I) = T
  330 CONTINUE
C
  340 IF (T.EQ.STARTT .AND. STARTT.NE.TS) GO TO 341
      DO 3400 I=1,NX
 3400 XTOT(I) = GO(I) + X(I)
      WRITE (NTAPE2)T, (F(J),J=1,NX), (XDD(I),XD(I),XTOT(I) I=1,NX)
C
C  DIVERGENCE CHECK.
C
  341 DO 350 I=1,NX                                          NERROR = 4
      IF (ABS(X(I)) .GT. 1.E+35) GO TO 999
  350 CONTINUE
C
      IF (T .LT. ENDT) GO TO 100
C
      ENDT = T
C
      RETURN
C
  999 WRITE (NOT,1002) NERROR
 1002 FORMAT (1H1,10(/),30X,*ERROR IN SUBROUTINE TR1* /
     *            30X, * AT NERROR = *, I3 )
      STOP
      END
      SUBROUTINE TERP2 (XA,XZ,YA,YZ,NXA,NXZ,NCA,KA,KZ)
      DIMENSION XA(1),XZ(1),YA(KA,1),YZ(KZ,1)
C
C------DIPARABOLIC INTERPOLATION.
C------(PARABOLIC INTERPOLATION IN FIRST,LAST BAYS AND OUTSIDE XA).
C------VALUES OF XZ MAY BE OUTSIDE OF XA. (EXTRAPOLATION).
C
C------SUBROUTINE ARGUMENTS.
C     XA  = INPUT   VECTOR OF X-COORDINATES FOR ROWS OF YA. MUST BE IN
C                   INCREASING ORDER. SIZE(NXA).
C     XZ  = INPUT   VECTOR OF X-COORDINATES FOR INTERPOLATED/EXTRAPOLATED
C                   VALUES. SIZE(NXZ).
C     YA  = INPUT   MATRIX OF Y-COORDINATES TO BE INTERPOLATED.
C                   SIZE(NXA,NCA).
C     YZ  = OUTPUT  MATRIX OF INTERPOLATED Y-COORDINATES. SIZE(NXZ,NCA).
C                   EACH COLUMN OF YZ HAS INTERPOLATED VALUES OF THE
C                   RESPECTIVE COLUMN OF YA.
C     NXA = INPUT   NUMBER OF XA STATIONS, ROWS OF MATRIX YA.
C     NXZ = INPUT   NUMBER OF XZ STATIONS, ROWS OF MATRIX YZ.
C     NCA = INPUT   NUMBER OF COLUMN VECTORS IN MATRICES YA,YZ.
C     KA  = INPUT   ROW DIMENSION OF YA IN CALLING PROGRAM.
```

```
C      KZ  = INPUT  ROW DIMENSION OF YZ IN CALLING PROGRAM.
C
       NERROR = 1
       IF (NXA .LT. 3) GO TO 999
C
       DO 400 K=1,NXZ
       IF (XZ(K) .LE. XA(2)) GO TO 100
       IF (XZ(K) .GE. XA(NXA-1)) GO TO 300
       DO 50 I=3,NXA
       IF(XZ(K) .LE. XA(I)) GO TO 200
   50  CONTINUE
C
C      FIRST BAY OR LEFT EXTRAPOLATION.
  100  BAYL = XA(2) - XA(1)
       H = (XZ(K) - XA(1)) / BAYL
       D = (XA(3) - XA(1)) / BAYL
       DO 102 J=1,NCA
  102  YZ(K,J) = YA(1,J)*(H**2-H*(1.0+D)+D) / D
     1            + YA(2,J)*(H**2-H*D) / (1.0-D)
     2            + YA(3,J)*(-H**2+H) / (D-D**2)
       GO TO 400
C
C      INTERIOR BAY.
  200  BAYL = XA(I)-XA(I-1)
       H = (XZ(K)   -XA(I-1))/BAYL
       C = (XA(I-2)-XA(I-1))/BAYL
       D = (XA(I+1)-XA(I-1))/BAYL
       DO 202 J= 1,NCA
  202  YZ(K,J)= YA(I-2,J)*(H**3-2.0*H**2+H)/(C-C**2)
     1           + YA(I-1,J)*(H**3*(C-D)+H**2*(2.0*D-C)-H*(D+C*D)+C*D)/(C*D)
     2           + YA(I   ,J)*(H**3*(D-C)+H**2*(1.0-2.0*D+C)-H*C*(1.0-D))/
     3                                                 ((1.0-C)*(1.0-D))
     4           + YA(I+1,J)*(-H**3+H**2)/(D-D**2)
       GO TO 400
C
C      LAST BAY OR RIGHT EXTRAPOLATION.
  300  BAYL = XA(NXA)-XA(NXA-1)
       H = (XZ(K)      -XA(NXA-1))/BAYL
       C = (XA(NXA-2)-XA(NXA-1))/BAYL
       DO 302 J=1,NCA
  302  YZ(K,J)=YA(NXA-2,J)*(-H**2+H)/(C-C**2)
     1           +YA(NXA-1,J)*(H**2-H*(1.0+C)+C)/C
     2           +YA(NXA   ,J)*(H**2-H*C)/(1.0-C)
C
  400  CONTINUE
       RETURN
C
  999  WRITE (6,1002) NERROR
 1002  FORMAT (1H1,10(/),30X,*ERROR IN SUBROUTINE TERP2* /
     1         30X, * AT NERROR = *, I3)
       STOP
       END
       SUBROUTINE WRITE (A,NR,NC,ANAME,KR)
       DIMENSION A(KR,1)
 2000  FORMAT (1H1)
 2010  FORMAT (//15H OUTPUT MATRIX ,A6,2X,1H(I4,2H X,I4,2H ) //
     *         10X,10(7X,1H( I2,1H))/)
 2020  FORMAT (//15H OUTPUT MATRIX ,A6,2X,1H(I4,2H X,I4,2H )
     *         3X, 9HCONTINUED //10X,10(7X,1H( I2,1H))/)
 2030  FORMAT (1X,2I5,2X,10(1PE11.3))
```

33

```
2040 FORMAT (14HOEND OF WRITE.)
     WRITE (6,2000)
     WRITE (6,2010) ANAME,NR,NC,(L,L=1,10)
     NLINE = 0
     DO 60 I=1,NR
     NZERO = 0
     JS = 1
  10 JE = JS + 9
     IF (JE .GT. NC) JE = NC
     DO 20 J=JS,JE
     IF (A(I,J) .NE. 0.) GO TO 30
  20 CONTINUE
     GO TO 40
  30 NLINE = NLINE + 1
     IF (NLINE .LE. 50) GO TO 35
     WRITE (6,2000)
     WRITE (6,2020) ANAME,NR,NC,(L,L=1,10)
     NLINE = 1
  35 WRITE (6,2030) I,JS,(A(I,J), J=JS,JE)
     NZERO = 1
  40 IF (JE .EQ. NC) GO TO 50
     JS = JS + 10
     GO TO 10
  50 IF (NC.LE.10 .OR. NZERO.EQ.0 .OR. I.EQ.NR) GO TO 60
     NLINE = NLINE + 1
     WRITE (6,2030)
  60 CONTINUE
     WRITE (6,2040)
     RETURN
     END
      SUBROUTINE XMAXM (N,X,XMAX,XMIN,NMAX,NMIN)
      DIMENSION X(1)
      XMAX=X(1)
      XMIN=X(1)
     NMAX=1
     NMIN=1
      DO 1 J=1,N
      IF(X(J).LT.XMIN) NMIN=J
      IF(X(J).LT.XMIN) XMIN=X(J)
      IF(X(J).GT.XMAX) NMAX=J
      IF(X(J).GT.XMAX) XMAX=X(J)
   1  CONTINUE
      RETURN
      END
```

34