A SIMULATION PROGRAM FOR THE DYNAMICS OF A RADIALLY ROTATING BEAM WITH IMPACT

Technical Report No. UM-MEAM-88/01

by
Yigit, A., Ulsoy, A.G., and Scott, R.A.

Department of Mechanical Engineering and Applied Mechanics
The University of Michigan
Ann Arbor, MI 48109

May 6, 1988

1 Introduction

The purpose of this report is to provide the necessary information for running the programs IMPACTC.FTN and IMPACTB.FTN for the dynamics of a radially rotating beam with impact. In particular the report outlines the structure of the software and how to compile, run and modify the program. All programs are written in standard FORTRAN 77 and run on APOLLO DN3000 or DN560 workstations.

2 Problem Definition

The problem considered is the dynamics, with impact, of a radially rotating flexible beam attached to a rigid shaft. The rigid shaft is given a torque profile and the beam impacts, at a prescribed point, on a rigid surface (see Fig.1).

Longitudinal deformations are neglected and it is assumed that Euler- Bernoulli beam theory is adequate to describe the flexural motions.

To describe the kinematics, a frame moving with the shaft is introduced. This frame rotates with the beam as if the beam were rigid and is so oriented that one of its axes coincides with the elastic axis of the undeformed beam. The general problem consists of calculating the motion, both of the rigid shaft and the flexible beam, when a prescribed torque is applied to the shaft.

First the kinetic and potential energy are written after which Hamilton's Principle is used to derive the equations of motion. Galerkin's method is used to suppress the spatial dependence. The details of the derivation of the equations of motion and the impact modeling can be found in [1].

3 Software Structure

The simulation package consists of two separate types of programs.

1 - Preparation programs for the main simulation package

In order to save computer time roots of eigenvalue equation and the evaluation of modal integrals are performed separately and the results are inputed to the main program.

2 - Main simulation package

This includes the subroutines for inputting the system variables, forming the equations of motion, integration, forming the momentum balance equations, etc.

3.1 Preparation Programs

ROOTM

Purpose: Solves the eigenvalue equation for a nonrotating cantilever beam
Method: Newton's Iteration Method

Source: NAAS Library [2]
Input variables:
NU : Number of roots desired

The details of the program can be found in [2].
INTEG

Purpose : Evaluates the normalized modal integrals for a nonrotating cantilever beam
Method: Romberg method
Source : NAAS Library [2]
Input variables: None
Link Information: Should be linked with program QCRP from NAAS library.
The details for program QCRP can be found in [2].

3.2 Main Simulation Package

IMPACTC

Purpose : Solves for the dynamics of a radially rotating beam with impact
Method : Backward Difference method for integration, uses momentum balance method for impact modeling
Link Information: Should be linked with program DEPISODE from NAAS library [2].

The details for the program DEPISODE can be found in [3].
Input variables:

RO : mass per unit length of beam $[kg/m]$
FLMOD : flexural rigidity of the beam $[Nm^2]$
L : length of the beam $[m]$
A : length of the root (or radius of the hub) $[m]$
IR : ratio of rigid shaft and flexible beam inertia
TORQ: magnitude of the applied torque pulse $[Nm]$
TTORQ: duration of each pulse $[s]$
T1: fraction of zero pulse $((T_2 - T_1)/TTORQ)$
R1 : radius of the circular cross-section beam $[m]$
NU : number of modes to be used in simulation
XP, YP : location of the impact surface $[m]$
X : location of the beam at which the output is desired $[m]$
TINITL : beginning time for simulation $[s]$
TFINAL : final time in the simulation $[s]$
STEP : step size for simulation (except for integration) $[s]$
DAMP : joint damping coefficient $[Nms]$

SDAMP : modal damping ratio of the beam [ ]

Simulation variables:

H0 : Initial time step for the integration [s]
EPS : Error limit for the integrator
EPSI : Width of the clearance zone for impact [m]
EPSV : velocity tolerance for impact [m/s]

<u>IMPACTB</u>

Purpose : For the dynamics of a radially rotating beam with impact
Method : Backward Difference method for integration, uses a spring-dashpotmodel for impact modeling
Link Information: Should be linked with program DEPISODE from NAAS library [2].

Input variables:

Same as program IMPACTC with the following additions :

CRES : spring coefficient for the impact surface pair [N/m]
CDAMP : damping coefficient for dashpot [Ns/m]

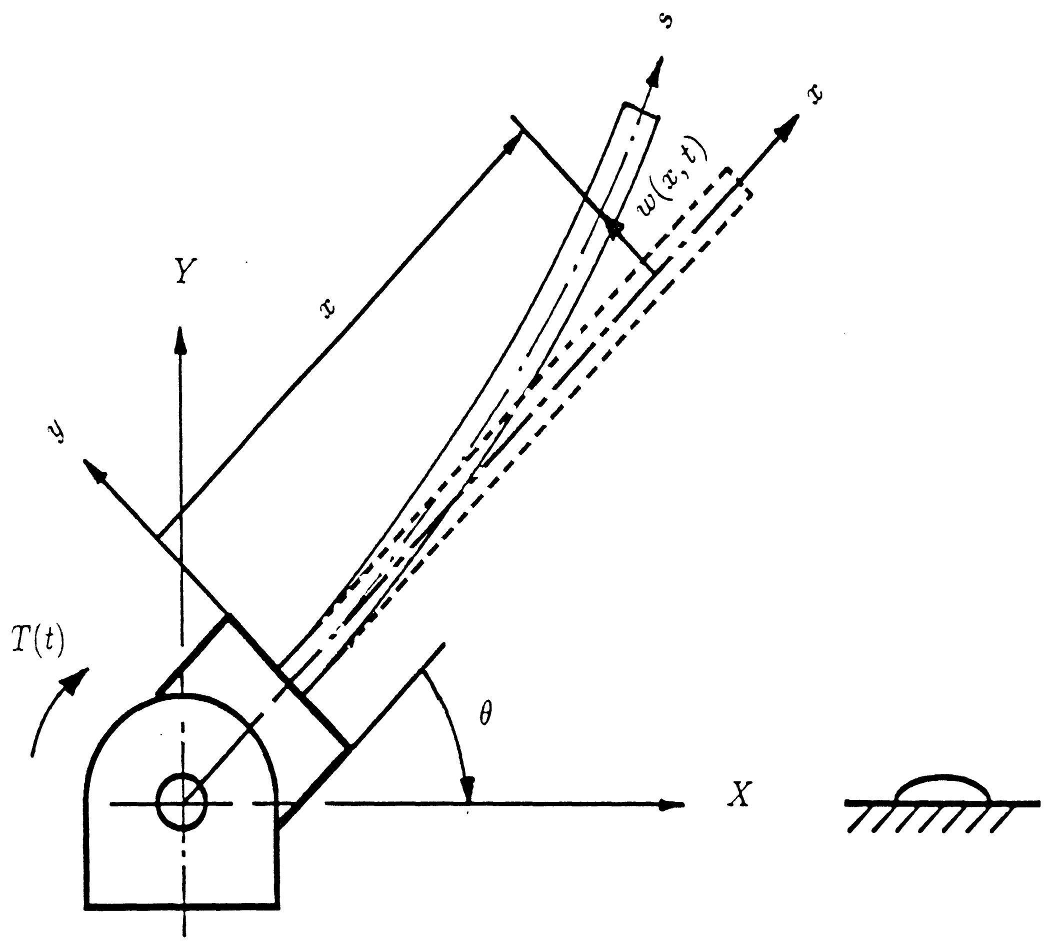

Fig. 1 The sketch of the Rotating Beam

4 Sample Runs

Programs IMPACTC and IMPACTB were compiled, linked and run on an Apollo DN3000 workstation for a particular set of parameter values. The screen copies from the workstation for these cases are presented below. The results of these simulations are presented in [1] as Figs. 6.14 and 6.28 respectively.

4.1 Sample Run with IMPACTC

```
$ ftn impactc.ftn
no errors, no warnings in $MAIN, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
no errors, no warnings in INPUT, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
no errors, no warnings in NUMRT, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
no errors, no warnings in IMPACT, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
no errors, no warnings in NMODE, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
no errors, no warnings in MODSHAPE, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
no errors, no warnings in STRAIN, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
no errors, no warnings in POSIT, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
no errors, no warnings in AROOT, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
no errors, no warnings in TRESP, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
no errors, no warnings in NTIMES, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
no errors, no warnings in AIC, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
no errors, no warnings in EQNS, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
no errors, no warnings in DDIFUN, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
no errors, no warnings in DPDERV, Fortran version 9.04 1988/04/27 12:36:25 EDT (Wed)
$ bind impactc.bin depisode.bin -b impactc
All Globals are resolved.

$ impactc

 PARAMATER INPUTS ;

   ENTER MASS PER UNIT LENGTH OF BEAM (KG/M) : .0855

   ENTER FLEXURAL RIGIDITY OF BEAM (NM**2) :5.50
   ENTER LENGTH OF THE BEAM (M) :.530
   ENTER THE LENGTH OF THE ROOT (M) :0.
   ENTER RATIO OF INERTIAS (IR/JBEAM):.002
   ENTER APPLIED JOINT TORQUE (N-M) ;-.18
   ENTER DURATION OF PULSE (SECS) :.45
   ENTER FRACTION OF ZERO PULSE :10.
   ENTER RADIUS OF BEAM FOR STRAIN CALC.(M):.003175
FILE NAME FOR DYNAMIC RESULTS:ft.dat
FILE NAME FOR INPUT DATA AND OTHER RESULTS:ft.msg
   ENTER THE NUMBER OF MODES YOU WANT :3

 ENTER INITIAL TIME,FINAL TIME AND INCREMENT :0.,1.,.0005
   ENTER X AND Y COORDINATES OF THE STOP :.515,0.
   ENTER COEFFICIENT OF RESTITUTION : .4
0ENTER QDOT(0),TETA(0),TETDOT(0):0.,3.3,0.
        0.0855        5.5000        0.5300       -0.1800        0.4500   3

0ENTER BEAM POSITION IN METERS FOR RESPONSE :.270
0ENTER DAMPING AND MODAL DAMPING :.00537,0.

---  MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER.  ---


 KFLAG = -1 FROM INTEGRATOR AT T =      0.44999905E+00
  ERROR TEST FAILED WITH ABS(H) = HMIN =      0.10000000E-05

  H HAS BEEN REDUCED TO     0.10000000E-06  AND STEP WILL BE RETRIED


---  MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER.  ---


 KFLAG = -1 FROM INTEGRATOR AT T =      0.44999996E+00
  ERROR TEST FAILED WITH ABS(H) = HMIN =      0.10000000E-06
```

```
 H HAS BEEN REDUCED TO     0.10000000E-07  AND STEP WILL BE RETRIED

0.7809999999999694 5.0000000000000000E-05 5.0000000000000000E-04

 TIME=    0.781050TETA=    0.00071839  WP=    0.00013615WPP=   -3.99515583

 TIME=    0.781100TETA=    0.00032893  WP=   -0.00006363WPP=   -3.99606792

NO. OF IMPACT=  1TH*** IMPACT ***
 VELOCITIES : ***     -3.996068      1.417128      0.354631

--- MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER.  ---


KFLAG = -1 FROM INTEGRATOR AT T =      0.78110000E+00
 ERROR TEST FAILED WITH ABS(H) = HMIN =     0.20347164E-03

 H HAS BEEN REDUCED TO     0.20347164E-04  AND STEP WILL BE RETRIED


--- MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER.  ---


KFLAG = -1 FROM INTEGRATOR AT T =      0.78110000E+00
 ERROR TEST FAILED WITH ABS(H) = HMIN =     0.20347164E-04

 H HAS BEEN REDUCED TO     0.20347164E-05  AND STEP WILL BE RETRIED


--- MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER.  ---


KFLAG = -1 FROM INTEGRATOR AT T =      0.78110000E+00
 ERROR TEST FAILED WITH ABS(H) = HMIN =     0.20347164E-05

 H HAS BEEN REDUCED TO     0.20347164E-06  AND STEP WILL BE RETRIED

0.7822499999999692 5.0000000000000000E-05 5.0000000000000000E-04

 TIME=    0.782300TETA=    0.00022422  WP=    0.00019111WPP=   -1.28061739

 TIME=    0.782350TETA=    0.00025930  WP=    0.00012498WPP=   -1.36530146

 TIME=    0.782400TETA=    0.00011285  WP=    0.00005476WPP=   -1.44306472

NO. OF IMPACT=  2TH*** IMPACT ***
 VELOCITIES : ***     -1.443065      0.660159      0.457470

--- MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER.  ---


KFLAG = -1 FROM INTEGRATOR AT T =      0.78240000E+00
 ERROR TEST FAILED WITH ABS(H) = HMIN =     0.64737495E-04

 H HAS BEEN REDUCED TO     0.64737495E-05  AND STEP WILL BE RETRIED
```

```
--- MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER. ---


 KFLAG = -1 FROM INTEGRATOR AT T =      0.78240000E+00
  ERROR TEST FAILED WITH ABS(H) = HMIN =     0.64737495E-05

  H HAS BEEN REDUCED TO     0.64737495E-06  AND STEP WILL BE RETRIED


 0.7829999999999692 5.0000000000000000E-05 5.0000000000000000E-04

  TIME=     0.783050TETA=    -0.01962500   WP=     0.00010940WPP=    -0.60763160

  TIME=     0.783100TETA=    -0.02164167   WP=     0.00007580WPP=    -0.73212068

 NO. OF IMPACT=  3TH*** IMPACT ***
  VELOCITIES : ***      -0.732121      0.396845      0.542049

--- MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER. ---


 KFLAG = -1 FROM INTEGRATOR AT T =      0.78310000E+00
  ERROR TEST FAILED WITH ABS(H) = HMIN =     0.70173763E-04

  H HAS BEEN REDUCED TO     0.70173763E-05  AND STEP WILL BE RETRIED


--- MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER. ---


 KFLAG = -1 FROM INTEGRATOR AT T =      0.78310000E+00
  ERROR TEST FAILED WITH ABS(H) = HMIN =     0.70173763E-05

  H HAS BEEN REDUCED TO     0.70173763E-06  AND STEP WILL BE RETRIED


--- MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER. ---


 KFLAG = -1 FROM INTEGRATOR AT T =      0.78310000E+00
  ERROR TEST FAILED WITH ABS(H) = HMIN =     0.70173763E-06

  H HAS BEEN REDUCED TO     0.70173763E-07  AND STEP WILL BE RETRIED

 0.7831999999999691 5.0000000000000000E-05 5.0000000000000000E-04

  TIME=     0.783250TETA=    -0.02711332   WP=     0.00009137WPP=     0.02825443

  TIME=     0.783300TETA=    -0.02929578   WP=     0.00009354WPP=    -0.12132989

 NO. OF IMPACT=  4TH*** IMPACT ***
  VELOCITIES : ***      -0.121330      0.100279      0.826497

--- MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER. ---


 KFLAG = -1 FROM INTEGRATOR AT T =      0.78330000E+00
```

```
 ERROR TEST FAILED WITH ABS(H) = HMIN =    0.13174384E-04

 H HAS BEEN REDUCED TO     0.13174384E-05  AND STEP WILL BE RETRIED


---  MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER.  ---


KFLAG = -1 FROM INTEGRATOR AT T =     0.78330000E+00
 ERROR TEST FAILED WITH ABS(H) = HMIN =    0.13174384E-05

 H HAS BEEN REDUCED TO     0.13174384E-06  AND STEP WILL BE RETRIED

0.7832999999999691 5.0000000000000000E-05 5.0000000000000000E-04

 TIME=    0.783350TETA=   -0.03115114  WP=    0.00009516WPP=   -0.03567990

NO. OF IMPACT=  4TH*** IMPACT ***
 VELOCITIES : ***     -0.035680      0.033578      0.941099

---  MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER.  ---


KFLAG = -1 FROM INTEGRATOR AT T =     0.78335000E+00
 ERROR TEST FAILED WITH ABS(H) = HMIN =    0.10932111E-04

 H HAS BEEN REDUCED TO     0.10932111E-05  AND STEP WILL BE RETRIED


---  MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER.  ---


KFLAG = -1 FROM INTEGRATOR AT T =     0.78335000E+00
 ERROR TEST FAILED WITH ABS(H) = HMIN =    0.10932111E-05

 H HAS BEEN REDUCED TO     0.10932111E-06  AND STEP WILL BE RETRIED

0.7833499999999691 5.0000000000000000E-05 5.0000000000000000E-04

 TIME=    0.783400TETA=   -0.03289159  WP=    0.00009343WPP=   -0.10274866

NO. OF IMPACT=  4TH*** IMPACT ***
 VELOCITIES : ***     -0.102749      0.087146      0.848146

.....


---  MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER.  ---


KFLAG = -1 FROM INTEGRATOR AT T =     0.78980000E+00
 ERROR TEST FAILED WITH ABS(H) = HMIN =    0.74971074E-05

 H HAS BEEN REDUCED TO     0.74971074E-06  AND STEP WILL BE RETRIED

0.7897999999999684 5.0000000000000000E-05 5.0000000000000000E-04
```

```
 TIME=     0.789850TETA=   -0.03654348  WP=     0.00008535WPP=   -0.00432877

 TIME=     0.789900TETA=   -0.03621787  WP=     0.00008496WPP=   -0.01069614

NO. OF IMPACT=  8TH*** IMPACT ***
 VELOCITIES : ***     -0.010696      0.010500       0.981654

--- MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER.  ---


KFLAG = -1 FROM INTEGRATOR AT T =     0.78990000E+00
 ERROR TEST FAILED WITH ABS(H) = HMIN =    0.40973880E-04

 H HAS BEEN REDUCED TO     0.40973880E-05  AND STEP WILL BE RETRIED


--- MESSAGE FROM SUBROUTINE DRIVE IN EPISODE, THE O.D.E. SOLVER.  ---


KFLAG = -1 FROM INTEGRATOR AT T =     0.78990000E+00
 ERROR TEST FAILED WITH ABS(H) = HMIN =    0.40973880E-05

 H HAS BEEN REDUCED TO     0.40973880E-06  AND STEP WILL BE RETRIED

    -5.209827
     1.032053
    -0.151943
    -0.002021
     0.175919
    -0.000538
     0.000050
     0.000000

Fortran STOP
```

4.2 Sample Run with IMPACTB

```
$ ftn impactb
no errors, no warnings in $MAIN, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in INPUT, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in NUMRT, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in IMPACT, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in NMODE, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in MODSHAPE, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in MODSHAPE2, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri
no errors, no warnings in STRAIN, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in POSIT, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in AROOT, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in TRESP, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in NTIMES, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in AIC, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in RESOUT, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in EQNS, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in DDIFUN, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
no errors, no warnings in DPDERV, Fortran version 9.38 1988/04/29 11:18:59 EDT (Fri)
$ bind impactb.bin depisode.bin -b impactb
All Globals are resolved.

$ impactb

 PARAMATER INPUTS ;

   ENTER MASS PER UNIT LENGTH OF BEAM (KG/M) : .0855

   ENTER FLEXURAL RIGIDITY OF BEAM (NM**2) :5.50
   ENTER LENGTH OF THE BEAM (M) :.530
   ENTER THE LENGTH OF THE ROOT (M) :0.
   ENTER RATIO OF INERTIAS (IR/JBEAM):.002
   ENTER APPLIED JOINT TORQUE (N-M) ;-.45
   ENTER DURATION OF PULSE (SECS) :?(sh) "./impactb" - process quit (OS/fault handler)
In routine "PROCESS_DEFERRED_FAULTS" line 723.
$ impactb

 PARAMATER INPUTS ;

   ENTER MASS PER UNIT LENGTH OF BEAM (KG/M) : .0855

   ENTER FLEXURAL RIGIDITY OF BEAM (NM**2) :5.50
   ENTER LENGTH OF THE BEAM (M) :.53
   ENTER THE LENGTH OF THE ROOT (M) :0.
   ENTER RATIO OF INERTIAS (IR/JBEAM):.002
   ENTER APPLIED JOINT TORQUE (N-M) ;-.18
   ENTER DURATION OF PULSE (SECS) :.45
   ENTER FRACTION OF ZERO PULSE :10.
   ENTER RADIUS OF BEAM FOR STRAIN CALC.(M):.003175
FILE NAME FOR RESULTS:ft1.dat
FILE NAME FOR INPUT DATA AND OTHER RESULTS:ft1.msg
   ENTER THE NUMBER OF MODES YOU WANT :3

 ENTER INITIAL TIME,FINAL TIME AND INCREMENT :0.,1.,.0005
   ENTER X AND Y COORDINATES OF THE STOP :.515,0.
   ENTER SPRING COEFFICIENT : 3.2e9
   ENTER DAMPING COEFFICIENT : .006
0ENTER WDOT(L,0),TETA(0),TETDOT(0),W(L,0):0.,3.3,0.,0.
        0.0855        5.5000        0.5300       -0.1800        0.4500   3

0ENTER BEAM POSITION IN METERS FOR RESPONSE :.270
0ENTER DAMPING AND MODAL DAMPING :.00537,0.

  TSTR=    0.78109025V1=    3.995910
 TIME=    0.781085FORCE=  243.331638WP=    0.000017WPP=    -3.995724* IMPACT *
```

```
TIME=     0.781090FORCE= 1316.332172WP=   -0.000003WPP=    -3.977030* IMPACT *
TIME=     0.781095FORCE= 1410.879164WP=   -0.000022WPP=    -3.137011* IMPACT *
TIME=     0.781100FORCE=  939.403621WP=   -0.000034WPP=    -1.654186* IMPACT *
TIME=     0.781105FORCE=  502.990294WP=   -0.000039WPP=    -0.395933* IMPACT *
TIME=     0.781110FORCE=  235.196393WP=   -0.000038WPP=     0.382752* IMPACT *
TIME=     0.781115FORCE=  107.380003WP=   -0.000035WPP=     0.791656* IMPACT *
TIME=     0.781120FORCE=   19.031648WP=   -0.000031WPP=     0.980241* IMPACT *

 TSTP=     0.78112358V2=    1.056325
TIME=     0.782205FORCE=   50.874430WP=    0.000004WPP=    -1.287060* IMPACT *
TIME=     0.782210FORCE=  230.638635WP=   -0.000002WPP=    -1.288969* IMPACT *
TIME=     0.782215FORCE=  341.623295WP=   -0.000008WPP=    -1.162200* IMPACT *
TIME=     0.782220FORCE=  351.259975WP=   -0.000013WPP=    -0.885052* IMPACT *
TIME=     0.782225FORCE=  291.769510WP=   -0.000017WPP=    -0.535437* IMPACT *
TIME=     0.782230FORCE=  240.992086WP=   -0.000019WPP=    -0.196499* IMPACT *
TIME=     0.782235FORCE=  166.647740WP=   -0.000019WPP=     0.082089* IMPACT *
TIME=     0.782240FORCE=  105.408315WP=   -0.000018WPP=     0.285928* IMPACT *
TIME=     0.782245FORCE=   59.871007WP=   -0.000016WPP=     0.421437* IMPACT *
TIME=     0.782250FORCE=   25.228059WP=   -0.000014WPP=     0.502544* IMPACT *
TIME=     0.782255FORCE=   10.932691WP=   -0.000011WPP=     0.543599* IMPACT *
TIME=     0.782835FORCE=   25.880216WP=    0.000001WPP=    -0.551815* IMPACT *
TIME=     0.782840FORCE=   66.027195WP=   -0.000002WPP=    -0.556058* IMPACT *
TIME=     0.782845FORCE=  114.501244WP=   -0.000004WPP=    -0.523393* IMPACT *
TIME=     0.782850FORCE=  133.060162WP=   -0.000007WPP=    -0.447967* IMPACT *
TIME=     0.782855FORCE=  135.741347WP=   -0.000009WPP=    -0.338975* IMPACT *
TIME=     0.782860FORCE=  125.767910WP=   -0.000010WPP=    -0.212510* IMPACT *
TIME=     0.782865FORCE=  108.028503WP=   -0.000011WPP=    -0.085256* IMPACT *
TIME=     0.782870FORCE=   92.416283WP=   -0.000011WPP=     0.029945* IMPACT *
TIME=     0.782875FORCE=   70.807720WP=   -0.000011WPP=     0.125628* IMPACT *
TIME=     0.782880FORCE=   50.961878WP=   -0.000010WPP=     0.199107* IMPACT *
TIME=     0.782885FORCE=   34.050708WP=   -0.000009WPP=     0.251036* IMPACT *
TIME=     0.782890FORCE=   20.469016WP=   -0.000007WPP=     0.283885* IMPACT *
TIME=     0.782895FORCE=   10.218050WP=   -0.000006WPP=     0.300913* IMPACT *
TIME=     0.782900FORCE=    1.940127WP=   -0.000004WPP=     0.305568* IMPACT *
TIME=     0.783145FORCE=    1.163887WP=    0.000001WPP=    -0.282525* IMPACT *
TIME=     0.783150FORCE=   15.235171WP=    0.000000WPP=    -0.295440* IMPACT *
TIME=     0.783155FORCE=   31.404109WP=   -0.000002WPP=    -0.301217* IMPACT *
TIME=     0.783160FORCE=   49.827683WP=   -0.000003WPP=    -0.291124* IMPACT *
TIME=     0.783165FORCE=   64.921931WP=   -0.000005WPP=    -0.263217* IMPACT *
TIME=     0.783170FORCE=   73.267502WP=   -0.000006WPP=    -0.219027* IMPACT *
TIME=     0.783175FORCE=   74.805273WP=   -0.000007WPP=    -0.162807* IMPACT *
TIME=     0.783180FORCE=   72.201101WP=   -0.000007WPP=    -0.100261* IMPACT *
TIME=     0.783185FORCE=   66.440837WP=   -0.000008WPP=    -0.037173* IMPACT *
TIME=     0.783190FORCE=   56.486298WP=   -0.000008WPP=     0.021616* IMPACT *
TIME=     0.783195FORCE=   46.684842WP=   -0.000008WPP=     0.072697* IMPACT *
TIME=     0.783200FORCE=   33.214123WP=   -0.000007WPP=     0.114155* IMPACT *
TIME=     0.783205FORCE=   27.049260WP=   -0.000006WPP=     0.145356* IMPACT *
TIME=     0.783210FORCE=   19.069494WP=   -0.000006WPP=     0.166606* IMPACT *
TIME=     0.783215FORCE=   12.429070WP=   -0.000005WPP=     0.178820* IMPACT *
TIME=     0.783220FORCE=    7.181039WP=   -0.000004WPP=     0.183231* IMPACT *
TIME=     0.783225FORCE=    3.316028WP=   -0.000003WPP=     0.181208* IMPACT *
TIME=     0.783230FORCE=    0.827967WP=   -0.000002WPP=     0.174156* IMPACT *
TIME=     0.783345FORCE=    0.865942WP=    0.000000WPP=    -0.134641* IMPACT *
TIME=     0.783350FORCE=    5.786933WP=    0.000000WPP=    -0.148117* IMPACT *
TIME=     0.783355FORCE=   15.437929WP=   -0.000001WPP=    -0.158616* IMPACT *
TIME=     0.783360FORCE=   22.613794WP=   -0.000002WPP=    -0.163267* IMPACT *
TIME=     0.783365FORCE=   32.223459WP=   -0.000003WPP=    -0.160557* IMPACT *
TIME=     0.783370FORCE=   39.575093WP=   -0.000003WPP=    -0.149857* IMPACT *
TIME=     0.783375FORCE=   39.932296WP=   -0.000004WPP=    -0.131469* IMPACT *
TIME=     0.783380FORCE=   43.434381WP=   -0.000005WPP=    -0.106579* IMPACT *
TIME=     0.783385FORCE=   44.153739WP=   -0.000005WPP=    -0.077102* IMPACT *
TIME=     0.783390FORCE=   43.146737WP=   -0.000005WPP=    -0.045226* IMPACT *
TIME=     0.783395FORCE=   39.839860WP=   -0.000006WPP=    -0.013152* IMPACT *
TIME=     0.783400FORCE=   35.314398WP=   -0.000006WPP=     0.017141* IMPACT *
TIME=     0.783405FORCE=   31.818694WP=   -0.000005WPP=     0.044110* IMPACT *
TIME=     0.783410FORCE=   24.561191WP=   -0.000005WPP=     0.066714* IMPACT *
```

```
TIME=     0.783415FORCE=   21.045255WP=   -0.000005WPP=     0.084382* IMPACT *
TIME=     0.783420FORCE=   17.723581WP=   -0.000004WPP=     0.096970* IMPACT *
TIME=     0.783425FORCE=   11.877637WP=   -0.000004WPP=     0.104649* IMPACT *
TIME=     0.783430FORCE=    8.650876WP=   -0.000003WPP=     0.107801* IMPACT *
TIME=     0.783435FORCE=    5.996693WP=   -0.000003WPP=     0.106944* IMPACT *
TIME=     0.783440FORCE=    3.045791WP=   -0.000002WPP=     0.102671* IMPACT *
TIME=     0.783445FORCE=    3.045791WP=   -0.000002WPP=     0.095591* IMPACT *
TIME=     0.783450FORCE=    1.194841WP=   -0.000001WPP=     0.086298* IMPACT *
TIME=     0.783455FORCE=    0.242744WP=   -0.000001WPP=     0.075345* IMPACT *
TIME=     0.783490FORCE=    0.186939WP=    0.000000WPP=    -0.015979* IMPACT *
TIME=     0.783495FORCE=    1.029429WP=    0.000000WPP=    -0.029193* IMPACT *
TIME=     0.783500FORCE=    1.029429WP=    0.000000WPP=    -0.042287* IMPACT *
TIME=     0.783505FORCE=    2.785131WP=    0.000000WPP=    -0.054878* IMPACT *
TIME=     0.783510FORCE=    6.095074WP=   -0.000001WPP=    -0.066313* IMPACT *
TIME=     0.783515FORCE=    8.111096WP=   -0.000001WPP=    -0.075944* IMPACT *
TIME=     0.783520FORCE=   12.761837WP=   -0.000001WPP=    -0.083096* IMPACT *
TIME=     0.783525FORCE=   16.510686WP=   -0.000002WPP=    -0.087148* IMPACT *
TIME=     0.783530FORCE=   20.156324WP=   -0.000002WPP=    -0.087635* IMPACT *
TIME=     0.783535FORCE=   23.402099WP=   -0.000003WPP=    -0.084317* IMPACT *
TIME=     0.783540FORCE=   25.981281WP=   -0.000003WPP=    -0.077237* IMPACT *
TIME=     0.783545FORCE=   27.711589WP=   -0.000003WPP=    -0.066743* IMPACT *
TIME=     0.783550FORCE=   28.506715WP=   -0.000004WPP=    -0.053444* IMPACT *
TIME=     0.783555FORCE=   28.384493WP=   -0.000004WPP=    -0.038151* IMPACT *
TIME=     0.783560FORCE=   27.426015WP=   -0.000004WPP=    -0.021780* IMPACT *
TIME=     0.783565FORCE=   25.786178WP=   -0.000004WPP=    -0.005259* IMPACT *
TIME=     0.783570FORCE=   23.653136WP=   -0.000004WPP=     0.010559* IMPACT *
TIME=     0.783575FORCE=   21.206501WP=   -0.000004WPP=     0.024953* IMPACT *
TIME=     0.783580FORCE=   18.616352WP=   -0.000004WPP=     0.037373* IMPACT *
TIME=     0.783585FORCE=   16.027202WP=   -0.000004WPP=     0.047453* IMPACT *
TIME=     0.783590FORCE=   13.551548WP=   -0.000004WPP=     0.054996* IMPACT *
TIME=     0.783595FORCE=   11.274526WP=   -0.000003WPP=     0.059956* IMPACT *
TIME=     0.783600FORCE=    8.243353WP=   -0.000003WPP=     0.062408* IMPACT *
TIME=     0.783605FORCE=    8.243353WP=   -0.000003WPP=     0.062522* IMPACT *
TIME=     0.783610FORCE=    5.220974WP=   -0.000002WPP=     0.060530* IMPACT *
TIME=     0.783615FORCE=    5.220974WP=   -0.000002WPP=     0.056701* IMPACT *
TIME=     0.783620FORCE=    3.379458WP=   -0.000002WPP=     0.051326* IMPACT *
TIME=     0.783625FORCE=    3.379458WP=   -0.000001WPP=     0.044690* IMPACT *
TIME=     0.783630FORCE=    3.222435WP=   -0.000001WPP=     0.037077* IMPACT *
TIME=     0.783635FORCE=    2.767999WP=   -0.000001WPP=     0.028729* IMPACT *
TIME=     0.783640FORCE=    2.543186WP=   -0.000001WPP=     0.019878* IMPACT *
TIME=     0.783645FORCE=    2.643948WP=   -0.000001WPP=     0.010753* IMPACT *
TIME=     0.783650FORCE=    3.222038WP=   -0.000001WPP=     0.001564* IMPACT *
TIME=     0.783655FORCE=    4.098967WP=   -0.000001WPP=    -0.007483* IMPACT *
TIME=     0.783660FORCE=    4.098967WP=   -0.000001WPP=    -0.016172* IMPACT *
TIME=     0.783665FORCE=    5.417407WP=   -0.000001WPP=    -0.024271* IMPACT *
TIME=     0.783670FORCE=    7.165840WP=   -0.000001WPP=    -0.031531* IMPACT *
TIME=     0.783675FORCE=    9.279061WP=   -0.000001WPP=    -0.037689* IMPACT *
TIME=     0.783680FORCE=    9.279061WP=   -0.000002WPP=    -0.042485* IMPACT *
TIME=     0.783685FORCE=   11.609590WP=   -0.000002WPP=    -0.045688* IMPACT *
TIME=     0.783690FORCE=   13.950194WP=   -0.000002WPP=    -0.047109* IMPACT *
TIME=     0.783695FORCE=   16.064707WP=   -0.000002WPP=    -0.046639* IMPACT *
TIME=     0.783700FORCE=   16.064707WP=   -0.000002WPP=    -0.044266* IMPACT *
TIME=     0.783705FORCE=   17.732018WP=   -0.000003WPP=    -0.040085* IMPACT *
TIME=     0.783710FORCE=   18.787842WP=   -0.000003WPP=    -0.034293* IMPACT *
TIME=     0.783715FORCE=   19.151894WP=   -0.000003WPP=    -0.027185* IMPACT *
TIME=     0.783720FORCE=   19.151894WP=   -0.000003WPP=    -0.019127* IMPACT *
TIME=     0.783725FORCE=   18.996576WP=   -0.000003WPP=    -0.010523* IMPACT *
TIME=     0.783730FORCE=   17.802978WP=   -0.000003WPP=    -0.001797* IMPACT *
TIME=     0.783735FORCE=   16.858849WP=   -0.000003WPP=     0.006657* IMPACT *
TIME=     0.783740FORCE=   15.750791WP=   -0.000003WPP=     0.014492* IMPACT *
TIME=     0.783745FORCE=   14.132435WP=   -0.000003WPP=     0.021423* IMPACT *
TIME=     0.783750FORCE=   11.453086WP=   -0.000003WPP=     0.027235* IMPACT *
TIME=     0.783755FORCE=   11.453086WP=   -0.000003WPP=     0.031785* IMPACT *
TIME=     0.783760FORCE=    9.248661WP=   -0.000003WPP=     0.034997* IMPACT *
TIME=     0.783765FORCE=    9.248661WP=   -0.000002WPP=     0.036860* IMPACT *
TIME=     0.783770FORCE=    7.856686WP=   -0.000002WPP=     0.037409* IMPACT *
```

```
TIME=     0.783775FORCE=     6.674999WP=    -0.000002WPP=     0.036737* IMPACT *
TIME=     0.783780FORCE=     6.674999WP=    -0.000002WPP=     0.034957* IMPACT *
TIME=     0.783785FORCE=     5.720257WP=    -0.000002WPP=     0.032206* IMPACT *
TIME=     0.783790FORCE=     4.995831WP=    -0.000002WPP=     0.028634* IMPACT *
TIME=     0.783795FORCE=     4.497535WP=    -0.000001WPP=     0.024396* IMPACT *
TIME=     0.783800FORCE=     4.155915WP=    -0.000001WPP=     0.019644* IMPACT *
TIME=     0.783805FORCE=     4.155915WP=    -0.000001WPP=     0.014530* IMPACT *
TIME=     0.783810FORCE=     4.286550WP=    -0.000001WPP=     0.009198* IMPACT *
TIME=     0.783815FORCE=     4.286550WP=    -0.000001WPP=     0.003794* IMPACT *
TIME=     0.783820FORCE=     4.882254WP=    -0.000001WPP=    -0.001542* IMPACT *
TIME=     0.783825FORCE=     4.882254WP=    -0.000001WPP=    -0.006668* IMPACT *
TIME=     0.783830FORCE=     5.922040WP=    -0.000001WPP=    -0.011447* IMPACT *
TIME=     0.783835FORCE=     5.922040WP=    -0.000001WPP=    -0.015744* IMPACT *
TIME=     0.783840FORCE=     6.756497WP=    -0.000001WPP=    -0.019426* IMPACT *
TIME=     0.783845FORCE=     7.700468WP=    -0.000002WPP=    -0.022370* IMPACT *
TIME=     0.783850FORCE=     8.707324WP=    -0.000002WPP=    -0.024467* IMPACT *
TIME=     0.783855FORCE=     9.719526WP=    -0.000002WPP=    -0.025634* IMPACT *
TIME=     0.783860FORCE=    10.674072WP=    -0.000002WPP=    -0.025821* IMPACT *
TIME=     0.783865FORCE=    11.244925WP=    -0.000002WPP=    -0.025015* IMPACT *
TIME=     0.783870FORCE=    11.745584WP=    -0.000002WPP=    -0.023248* IMPACT *
TIME=     0.783875FORCE=    12.161169WP=    -0.000002WPP=    -0.020590* IMPACT *
TIME=     0.783880FORCE=    12.691834WP=    -0.000002WPP=    -0.017155* IMPACT *
TIME=     0.783885FORCE=    12.796683WP=    -0.000002WPP=    -0.013097* IMPACT *
TIME=     0.783890FORCE=    12.764539WP=    -0.000002WPP=    -0.008594* IMPACT *
TIME=     0.783895FORCE=    12.385368WP=    -0.000003WPP=    -0.003837* IMPACT *
TIME=     0.783900FORCE=    12.385368WP=    -0.000003WPP=     0.000981* IMPACT *
TIME=     0.783905FORCE=    11.652597WP=    -0.000002WPP=     0.005674* IMPACT *
TIME=     0.783910FORCE=    11.077887WP=    -0.000002WPP=     0.010079* IMPACT *
TIME=     0.783915FORCE=    10.431984WP=    -0.000002WPP=     0.014040* IMPACT *
TIME=     0.783920FORCE=     9.738954WP=    -0.000002WPP=     0.017438* IMPACT *
TIME=     0.783925FORCE=     8.718269WP=    -0.000002WPP=     0.020186* IMPACT *
TIME=     0.783930FORCE=     7.709230WP=    -0.000002WPP=     0.022229* IMPACT *
TIME=     0.783935FORCE=     6.310388WP=    -0.000002WPP=     0.023544* IMPACT *
TIME=     0.783940FORCE=     6.310388WP=    -0.000002WPP=     0.024137* IMPACT *
TIME=     0.783945FORCE=     5.150181WP=    -0.000002WPP=     0.024034* IMPACT *
TIME=     0.783950FORCE=     5.150181WP=    -0.000002WPP=     0.023286* IMPACT *
TIME=     0.783955FORCE=     4.282820WP=    -0.000002WPP=     0.021951* IMPACT *
TIME=     0.783960FORCE=     4.282820WP=    -0.000001WPP=     0.020109* IMPACT *
TIME=     0.783965FORCE=     3.724222WP=    -0.000001WPP=     0.017830* IMPACT *
TIME=     0.783970FORCE=     3.724222WP=    -0.000001WPP=     0.015208* IMPACT *
TIME=     0.783975FORCE=     3.468018WP=    -0.000001WPP=     0.012320* IMPACT *
TIME=     0.783980FORCE=     3.468018WP=    -0.000001WPP=     0.009258* IMPACT *
TIME=     0.783985FORCE=     3.494207WP=    -0.000001WPP=     0.006099* IMPACT *
TIME=     0.783990FORCE=     3.494207WP=    -0.000001WPP=     0.002928* IMPACT *
TIME=     0.783995FORCE=     3.782039WP=    -0.000001WPP=    -0.000182* IMPACT *
TIME=     0.784000FORCE=     3.782039WP=    -0.000001WPP=    -0.003153* IMPACT *
TIME=     0.784005FORCE=     4.305496WP=    -0.000001WPP=    -0.005912* IMPACT *
TIME=     0.784010FORCE=     4.305496WP=    -0.000001WPP=    -0.008391* IMPACT *
TIME=     0.784015FORCE=     4.750490WP=    -0.000001WPP=    -0.010525* IMPACT *
TIME=     0.784020FORCE=     5.253700WP=    -0.000001WPP=    -0.012251* IMPACT *
TIME=     0.784025FORCE=     5.787046WP=    -0.000001WPP=    -0.013519* IMPACT *
TIME=     0.784030FORCE=     5.787046WP=    -0.000001WPP=    -0.014290* IMPACT *
TIME=     0.784035FORCE=     6.317708WP=    -0.000001WPP=    -0.014540* IMPACT *
TIME=     0.784040FORCE=     6.810610WP=    -0.000002WPP=    -0.014261* IMPACT *
TIME=     0.784045FORCE=     7.231452WP=    -0.000002WPP=    -0.013461* IMPACT *
TIME=     0.784050FORCE=     7.550039WP=    -0.000002WPP=    -0.012169* IMPACT *
TIME=     0.784055FORCE=     7.550039WP=    -0.000002WPP=    -0.010431* IMPACT *
TIME=     0.784060FORCE=     7.743287WP=    -0.000002WPP=    -0.008310* IMPACT *
TIME=     0.784065FORCE=     7.797409WP=    -0.000002WPP=    -0.005885* IMPACT *
TIME=     0.784070FORCE=     7.708944WP=    -0.000002WPP=    -0.003240* IMPACT *
TIME=     0.784075FORCE=     7.477401WP=    -0.000002WPP=    -0.000467* IMPACT *
TIME=     0.784080FORCE=     7.477401WP=    -0.000002WPP=     0.002337* IMPACT *
TIME=     0.784085FORCE=     6.929915WP=    -0.000002WPP=     0.005080* IMPACT *
TIME=     0.784090FORCE=     6.469668WP=    -0.000002WPP=     0.007684* IMPACT *
TIME=     0.784095FORCE=     6.469668WP=    -0.000002WPP=     0.010065* IMPACT *
TIME=     0.784100FORCE=     5.954324WP=    -0.000002WPP=     0.012165* IMPACT *
```

```
TIME=     0.784105FORCE=     5.407725WP=    -0.000002WPP=      0.013933* IMPACT *
TIME=     0.784110FORCE=     4.647200WP=    -0.000002WPP=      0.015332* IMPACT *
TIME=     0.784115FORCE=     4.647200WP=    -0.000001WPP=      0.016340* IMPACT *
TIME=     0.784120FORCE=     3.923130WP=    -0.000001WPP=      0.016954* IMPACT *
TIME=     0.784125FORCE=     2.974401WP=    -0.000001WPP=      0.017178* IMPACT *
TIME=     0.784130FORCE=     2.974401WP=    -0.000001WPP=      0.017030* IMPACT *
TIME=     0.784135FORCE=     2.974401WP=    -0.000001WPP=      0.016535* IMPACT *
TIME=     0.784140FORCE=     2.258152WP=    -0.000001WPP=      0.015729* IMPACT *
TIME=     0.784145FORCE=     2.258152WP=    -0.000001WPP=      0.014646* IMPACT *
TIME=     0.784150FORCE=     1.900956WP=    -0.000001WPP=      0.013328* IMPACT *
TIME=     0.784155FORCE=     1.900956WP=    -0.000001WPP=      0.011821* IMPACT *
TIME=     0.784160FORCE=     1.663983WP=    -0.000001WPP=      0.010164* IMPACT *
TIME=     0.784165FORCE=     1.663983WP=    -0.000001WPP=      0.008402* IMPACT *
TIME=     0.784170FORCE=     1.538660WP=    -0.000001WPP=      0.006574* IMPACT *
TIME=     0.784175FORCE=     1.515207WP=    -0.000001WPP=      0.004718* IMPACT *
TIME=     0.784180FORCE=     1.515207WP=    -0.000001WPP=      0.002873* IMPACT *
TIME=     0.784185FORCE=     1.655222WP=    -0.000001WPP=      0.001071* IMPACT *
TIME=     0.784190FORCE=     1.655222WP=    -0.000001WPP=     -0.000654* IMPACT *
TIME=     0.784195FORCE=     1.655222WP=    -0.000001WPP=     -0.002270* IMPACT *
TIME=     0.784200FORCE=     2.049521WP=    -0.000001WPP=     -0.003747* IMPACT *
TIME=     0.784205FORCE=     2.049521WP=    -0.000001WPP=     -0.005059* IMPACT *
TIME=     0.784210FORCE=     2.049521WP=    -0.000001WPP=     -0.006177* IMPACT *
TIME=     0.784215FORCE=     2.161929WP=    -0.000001WPP=     -0.007079* IMPACT *
TIME=     0.784220FORCE=     2.374573WP=    -0.000001WPP=     -0.007741* IMPACT *
TIME=     0.784225FORCE=     2.538435WP=    -0.000001WPP=     -0.008148* IMPACT *
TIME=     0.784230FORCE=     2.784538WP=    -0.000001WPP=     -0.008290* IMPACT *
TIME=     0.784235FORCE=     3.010748WP=    -0.000001WPP=     -0.008164* IMPACT *
TIME=     0.784240FORCE=     3.010748WP=    -0.000001WPP=     -0.007772* IMPACT *
TIME=     0.784245FORCE=     3.334314WP=    -0.000001WPP=     -0.007125* IMPACT *
TIME=     0.784250FORCE=     3.334314WP=    -0.000001WPP=     -0.006238* IMPACT *
TIME=     0.784255FORCE=     3.334314WP=    -0.000001WPP=     -0.005134* IMPACT *
TIME=     0.784260FORCE=     3.355861WP=    -0.000001WPP=     -0.003845* IMPACT *
TIME=     0.784265FORCE=     3.355861WP=    -0.000001WPP=     -0.002401* IMPACT *
TIME=     0.784270FORCE=     3.355861WP=    -0.000001WPP=     -0.000837* IMPACT *
TIME=     0.784275FORCE=     3.355861WP=    -0.000001WPP=      0.000802* IMPACT *
TIME=     0.784280FORCE=     3.304697WP=    -0.000001WPP=      0.002467* IMPACT *
TIME=     0.784285FORCE=     2.967870WP=    -0.000001WPP=      0.004108* IMPACT *
TIME=     0.784290FORCE=     2.787887WP=    -0.000001WPP=      0.005688* IMPACT *
TIME=     0.784295FORCE=     2.478741WP=    -0.000001WPP=      0.007175* IMPACT *
TIME=     0.784300FORCE=     2.478741WP=    -0.000001WPP=      0.008540* IMPACT *
TIME=     0.784305FORCE=     2.106057WP=    -0.000001WPP=      0.009758* IMPACT *
TIME=     0.784310FORCE=     1.617902WP=    -0.000001WPP=      0.010812* IMPACT *
TIME=     0.784315FORCE=     1.617902WP=    -0.000001WPP=      0.011684* IMPACT *
TIME=     0.784320FORCE=     1.306203WP=    -0.000001WPP=      0.012375* IMPACT *
TIME=     0.784325FORCE=     1.306203WP=    -0.000001WPP=      0.012873* IMPACT *
TIME=     0.784330FORCE=     1.015832WP=    -0.000001WPP=      0.013186* IMPACT *
TIME=     0.784335FORCE=     0.754190WP=    -0.000001WPP=      0.013320* IMPACT *
TIME=     0.784340FORCE=     0.526510WP=     0.000000WPP=      0.013288* IMPACT *
TIME=     0.784345FORCE=     0.526510WP=     0.000000WPP=      0.013108* IMPACT *
TIME=     0.784350FORCE=     0.336785WP=     0.000000WPP=      0.012799* IMPACT *
TIME=     0.784355FORCE=     0.143876WP=     0.000000WPP=      0.012381* IMPACT *
TIME=     0.784360FORCE=     0.143876WP=     0.000000WPP=      0.011876* IMPACT *
TIME=     0.784365FORCE=     0.003779WP=     0.000000WPP=      0.011308* IMPACT *
TIME=     0.784370FORCE=     0.003779WP=     0.000000WPP=      0.010701* IMPACT *

.....

TIME=     0.789950FORCE=     0.134728WP=     0.000000WPP=      0.004089* IMPACT *
TIME=     0.789955FORCE=     0.134728WP=     0.000000WPP=      0.004219* IMPACT *
TIME=     0.789960FORCE=     0.077345WP=     0.000000WPP=      0.004353* IMPACT *
TIME=     0.789965FORCE=     0.056035WP=     0.000000WPP=      0.004494* IMPACT *
TIME=     0.789970FORCE=     0.017660WP=     0.000000WPP=      0.004644* IMPACT *
TIME=     0.789975FORCE=     0.017660WP=     0.000000WPP=      0.004808* IMPACT *
    -5.200794
     1.036907
    -0.152827
```

```
  -0.001918
   0.181022
  -0.000448
   0.000037
   0.000000

Fortran STOP
```

Program Listings

```
N     N    A          A    U     U
NN    N   A A        A A   U     U
N N   N  A   A      A   A  U     U
N  N  N A     A    A     A U     U
N   N N AAAAAAA    AAAAAAA U     U
N    NN A     A    A     A U     U
N     N A     A    A     A  UUUUU
```

```
rrrrr    oooo    oooo   ttttt  m     m         ffffff  ttttt  n    n
r    r  o    o  o    o    t    mm   mm         f         t    nn   n
r    r  o    o  o    o    t    m mm m          fffff     t    n n  n
rrrrr   o    o  o    o    t    m     m   ...   f         t    n  n n
r  r    o    o  o    o    t    m     m   ...   f         t    n   nn
r    r   oooo    oooo     t    m     m   ...   f         t    n    n
```

//meam/users/naau/soft.dir/rootm.ftn

LAST MODIFIED ON: 88/04/28 11:17 AM
FILE PRINTED: 88/04/28 11:21 AM

```
C
C     PROGRAM FOR SOLVING THE EIGENVALUE EQN. FOR THE
C     NONROTATING CANTILEVER BEAM
C
C
        EXTERNAL FCT
        DOUBLE PRECISION X,F,DERF,XST(100),XST1,XST2
C
C       ASK FOR NO. OF ROOTS TO BE FOUND
C
        WRITE(6,10)
10      FORMAT(3X,'ENTER NUMBER OF ROOTS :',$)
        READ(5,*) NU
C
C       INITIAL ESTIMATES FOR THE ROOTS
C
        XST(1)=1.875
        XST(2)=4.693
        XST(3)=7.855
        XST(4)=10.966
        XST(5)=14.137
        PI=3.1415927
        DO 100 I=6,100
100     XST(I)=((2*I-1)*PI)/2.D0
        DO 110 I=1,NU
        XST1=XST(I)
        EPS=1.E-8
        IEND=100000
C
C       SOLVE THE ASSOCIATED NONLINEAR EQUATION
C
         CALL DRTNI(X,F,DERF,FCT,XST1,EPS,IEND,IER)
110      WRITE(6,1) X,F,IEND,IER

1       FORMAT(2X,2(F16.8,1X),I6,2X,I2)
         STOP
         END
C
C       SUBROUTINE FOR DEFINING THE EIGENVALUE EQUATION
C
        SUBROUTINE FCT(X,F,DERF)
        DOUBLE PRECISION X,F,DERF
        F=DCOS(X)*DCOSH(X)+1.D0
        DERF=-DSIN(X)*DCOSH(X)+DSINH(X)*DCOS(X)
        RETURN
        END


C
C     .....................................  .......................
C
C        SUBROUTINE DRTNI
C
C        PURPOSE
C           TO SOLVE GENERAL NONLINEAR EQUATIONS OF THE FORM F(X)=0
C           BY MEANS OF NEWTON-S ITERATION METHOD.
C
C        USAGE
C           CALL DRTNI (X,F,DERF,FCT,XST,EPS,IEND,IER)
C           PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT.
C
C        DESCRIPTION OF PARAMETERS
C           X      - DOUBLE PRECISION RESULTANT ROOT OF EQUATION F(X)=0.
C           F      - DOUBLE PRECISION RESULTANT FUNCTION VALUE AT
C                    ROOT X.
C           DERF   - DOUBLE PRECISION RESULTANT VALUE OF DERIVATIVE
```

```
C                    AT ROOT X.
C           FCT    - NAME OF THE EXTERNAL SUBROUTINE USED. IT COMPUTES
C                    TO GIVEN ARGUMENT X FUNCTION VALUE F AND DERIVATIVE
C                    DERF. ITS PARAMETER LIST MUST BE X,F,DERF, WHERE
C                    ALL PARAMETERS ARE DOUBLE PRECISION.
C           XST    - DOUBLE PRECISION INPUT VALUE WHICH SPECIFIES THE
C                    INITIAL GUESS OF THE ROOT X.
C           EPS    - SINGLE PRECISION INPUT VALUE WHICH SPECIFIES THE
C                    UPPER BOUND OF THE ERROR OF RESULT X.
C           IEND   - MAXIMUM NUMBER OF ITERATION STEPS SPECIFIED.
C           IER    - RESULTANT ERROR PARAMETER CODED AS FOLLOWS
C                     IER=0 - NO ERROR,
C                     IER=1 - NO CONVERGENCE AFTER IEND ITERATION STEPS,
C                     IER=2 - AT ANY ITERATION STEP DERIVATIVE DERF WAS
C                             EQUAL TO ZERO.
C
C        REMARKS
C           THE PROCEDURE IS BYPASSED AND GIVES THE ERROR MESSAGE IER=2
C           IF AT ANY ITERATION STEP DERIVATIVE OF F(X) IS EQUAL TO 0.
C           POSSIBLY THE PROCEDURE WOULD BE SUCCESSFUL IF IT IS STARTED
C           ONCE MORE WITH ANOTHER INITIAL GUESS XST.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           THE EXTERNAL SUBROUTINE FCT(X,F,DERF) MUST BE FURNISHED
C           BY THE USER.
C
C        METHOD
C           SOLUTION OF EQUATION F(X)=0 IS DONE BY MEANS OF NEWTON-S
C           ITERATION METHOD, WHICH STARTS AT THE INITIAL GUESS XST OF
C           A ROOT X. CONVERGENCE IS QUADRATIC IF THE DERIVATIVE OF
C           F(X) AT ROOT X IS NOT EQUAL TO ZERO. ONE ITERATION STEP
C           REQUIRES ONE EVALUATION OF F(X) AND ONE EVALUATION OF THE
C           DERIVATIVE OF F(X). FOR TEST ON SATISFACTORY ACCURACY SEE
C           FORMULAE (2) OF MATHEMATICAL DESCRIPTION.
C           FOR REFERENCE, SEE R. ZURMUEHL, PRAKTISCHE MATHEMATIK FUER
C           INGENIEURE UND PHYSIKER, SPRINGER, BERLIN/GOETTINGEN/
C           HEIDELBERG, 1963, PP.12-17.
C
C     ..................................................................
C
      SUBROUTINE DRTNI(X,F,DERF,FCT,XST,EPS,IEND,IER)
C
C
      DOUBLE PRECISION X,F,DERF,XST,TOL,TOLF,DX,A
C
C     PREPARE ITERATION
      IER=0
      X=XST
      TOL=X
      CALL FCT(TOL,F,DERF)
      TOLF=100.*EPS
C
C
C     START ITERATION LOOP
      DO 6 I=1,IEND
      IF(F)1,7,1
C
C     EQUATION IS NOT SATISFIED BY X
    1 IF(DERF)2,8,2
C
C     ITERATION IS POSSIBLE
    2 DX=F/DERF
      X=X-DX
      TOL=X
      CALL FCT(TOL,F,DERF)
C
```

```
C     TEST ON SATISFACTORY ACCURACY
      TOL=EPS
      A=DABS(X)
      IF(A-1.D0)4,4,3
    3 TOL=TOL*A
    4 IF(DABS(DX)-TOL)5,5,6
    5 IF(DABS(F)-TOLF)7,7,6
    6 CONTINUE
C     END OF ITERATION LOOP
C
C
C     NO CONVERGENCE AFTER IEND ITERATION STEPS. ERROR RETURN.
      IER=1
    7 RETURN
C
C     ERROR RETURN IN CASE OF ZERO DIVISOR
    8 IER=2
      RETURN
      END
```

```
N     N    A       A    U     U
NN    N   A A     A A   U     U
N N   N  A   A   A   A  U     U
N  N  N A     A A     A U     U
N   N N AAAAAAA AAAAAAA U     U
N    NN A     A A     A U     U
N     N A     A A     A  UUUUU
```

```
i  n    n  ttttt  eeeeee   gggg          ffffff  ttttt  n    n
i  nn   n    t    e       g    g         f         t    nn   n
i  n n  n    t    eeeee   g              fffff     t    n n  n
i  n  n n    t    e       g  ggg   ...   f         t    n  n n
i  n   nn    t    e       g    g   ...   f         t    n   nn
i  n    n    t    eeeeee   gggg    ...   f         t    n    n
```

//meam/users/naau/soft.dir/integ.ftn

LAST MODIFIED ON: 88/04/28 11:12 AM
FILE PRINTED: 88/04/28 11:16 AM

```
C
C     PROGRAM FOR EVALUATING THE MODAL INTEGRALS
C
      IMPLICIT DOUBLE PRECISION (A-Z)
      INTEGER I,N,J,IY
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ
      COMMON/EQN/ALAMDA,NU
      COMMON/A/IY
      DIMENSION ALAMDA(100)

      COMMON N
      DIMENSION ATEMP(12)
      EXTERNAL SRMAT,AMASS,CENT,CENT1,SRMAT1,CENT2,CENT3,CENT4,CENT5
      L=1.
      ALAMDA(1)=1.87510407
      ALAMDA(2)= 4.69409113
      ALAMDA(3)= 7.85475744
      ALAMDA(4)= 10.99554073
      ALAMDA(5)= 14.13716839
      ALAMDA(6)= 17.27875953
      ALAMDA(7)= 20.42035225
      ALAMDA(8)= 23.56194490
      ALAMDA(9)= 26.70353756
      ALAMDA(10)= 29.84513021
C
C     LIMITS FOR THE INTEGRALS
C     TAKE L=1 FOR NORMALIZED MODAL INTEGRALS
C
      A=0.
      B=L
      EPS=1.E-05
C
C     ERROR BOUNDS FOR INTEGRATION
C
      AERR=1.E-14
      RERR=1.E-10
      DO 10 IY=1,10
C      ANS=QCRP(AMASS,A,B,AERR,RERR,ERROR,IER,NSI)
      BANS=QCRP(SRMAT,A,B,AERR,RERR,ERROR,IER,NSI)
      CANS=QCRP(CENT2,A,B,AERR,RERR,ERROR,IER,NSI)
      CANS=CANS-L
      WRITE(6,1) IY,ANS,BANS,CANS
      BANS1=QCRP(SRMAT1,A,B,AERR,RERR,ERROR,IER,NSI)
      CANS1=QCRP(CENT3,A,B,AERR,RERR,ERROR,IER,NSI)
      CANS1=CANS1-L
      WRITE(6,1) IY,ANS,BANS1,CANS1
      CANS=QCRP(CENT4,A,B,AERR,RERR,ERROR,IER,NSI)
      CANS=CANS-L
      WRITE(6,1) IY,ANS,BANS,CANS
      CANS=QCRP(CENT5,A,B,AERR,RERR,ERROR,IER,NSI)
      CANS=CANS-L
   10 WRITE(6,1) IY,ANS,BANS1,CANS

    1 FORMAT(2X,I2,2X,'M=',F12.6,2X,'SR=',F12.6,2X,' CEN=',F12.6)
      STOP
      END

C
C     SUBROUTINES FOR ENTERING THE MODAL INTEGRALS
C
      FUNCTION SRMAT(X)
      IMPLICIT DOUBLE PRECISION (A-Z)
      INTEGER II,NU
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ
      COMMON/EQN/ALAMDA,NU
      DIMENSION ALAMDA(100)
```

```
      COMMON/A/II

      LAM=ALAMDA(II)
      A1=0.50*(1.D0+DEXP(-2.D0*LAM))+DCOS(LAM)*DEXP(-LAM)
      A2=0.50*(1.D0-DEXP(-2.D0*LAM))-DSIN(LAM)*DEXP(-LAM)
      A3=-1.D0*A2/A1
      ARGU=LAM*X/L
      DENOM=DEXP(-ARGU)
      F=(0.50*(1.D0+DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DCOS(ARGU)+
     >A3*(0.50*(1.D0-DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DSIN(ARGU)))
     >/(DENOM)
      SRMAT=X*F
      RETURN
      END

C
      FUNCTION CENT(X)
      IMPLICIT DOUBLE PRECISION (A-Z)
      INTEGER II,NU,JJ
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ
      COMMON/EQN/ALAMDA,NU
      COMMON/A/II
      DIMENSION ALAMDA(100)
C       LAM2=ALAMDA(2)
      LAM=ALAMDA(II)
      LAM2=ALAMDA(II)
      A1=0.50*(1.D0+EXP(-2.D0*LAM))+DCOS(LAM)*DEXP(-LAM)
      A2=0.50*(1.D0-DEXP(-2.D0*LAM))+DSIN(LAM)*DEXP(-LAM)
      A3=-1.D0*A1/A2
      A12=0.50*(1.D0+DEXP(-2.D0*LAM2))+DCOS(LAM2)*DEXP(-LAM2)
      A22=0.50*(1.D0-DEXP(-2.D0*LAM2))+DSIN(LAM2)*DEXP(-LAM2)
      A32=-1.D0*A12/A22

      ARGU=LAM*X/L
      ARGUM=LAM2*X/L
      DENOM=DEXP(-ARGU)
      F=(0.50*(1.D0+DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DCOS(ARGU)+
     >A3*(0.50*(1.D0-DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DSIN(ARGU)))
     >/(DENOM)
      DF1=DSINH(ARGUM)+DSIN(ARGUM)+A32*(DCOSH(ARGUM)-DCOS(ARGUM))
      DF1=DF1*LAM
      DF=SINH(ARGU)+SIN(ARGU)+A3*(COSH(ARGU)-COS(ARGU))
      DF=DF*LAM
      CENT=0.5*(L*L-X*X)*DF1*DF1/L**2+1.D0
      RETURN
      END
C
      FUNCTION SRMAT1(X)
      IMPLICIT DOUBLE PRECISION (A-Z)
      INTEGER II,NU
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ
      COMMON/EQN/ALAMDA,NU
      DIMENSION ALAMDA(100)
      COMMON/A/II

      LAM=ALAMDA(II)
      A1=0.50*(1.D0+DEXP(-2.D0*LAM))+DCOS(LAM)*DEXP(-LAM)
      A2=0.50*(1.D0-DEXP(-2.D0*LAM))-DSIN(LAM)*DEXP(-LAM)
      A3=-1.D0*A2/A1
      ARGU=LAM*X/L
      DENOM=DEXP(-ARGU)
      F=(0.50*(1.D0+DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DCOS(ARGU)+
     >A3*(0.50*(1.D0-DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DSIN(ARGU)))
     >/(DENOM)
      SRMAT1=F
      RETURN
```

```
      END

C
      FUNCTION CENT1(X)
      IMPLICIT DOUBLE PRECISION (A-Z)
      INTEGER II,NU,JJ
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ
      COMMON/EQN/ALAMDA,NU
      COMMON/A/II
      DIMENSION ALAMDA(100)
C       LAM2=ALAMDA(2)
      LAM=ALAMDA(II)
      LAM2=ALAMDA(II)
      A1=0.50*(1.D0+EXP(-2.D0*LAM))+DCOS(LAM)*DEXP(-LAM)
      A2=0.50*(1.D0-DEXP(-2.D0*LAM))+DSIN(LAM)*DEXP(-LAM)
      A3=-1.D0*A1/A2
      A12=0.50*(1.D0+DEXP(-2.D0*LAM2))+DCOS(LAM2)*DEXP(-LAM2)
      A22=0.50*(1.D0-DEXP(-2.D0*LAM2))+DSIN(LAM2)*DEXP(-LAM2)
      A32=-1.D0*A12/A22

      ARGU=LAM*X/L
      ARGUM=LAM2*X/L
      DENOM=DEXP(-ARGU)
      F=(0.50*(1.D0+DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DCOS(ARGU)+
     >A3*(0.50*(1.D0-DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DSIN(ARGU)))
     >/(DENOM)
      DF1=DSINH(ARGUM)+DSIN(ARGUM)+A32*(DCOSH(ARGUM)-DCOS(ARGUM))
      DF1=DF1*LAM
      DF=SINH(ARGU)+SIN(ARGU)+A3*(COSH(ARGU)-COS(ARGU))
      DF=DF*LAM
      CENT1=(L-X)*DF1*DF1/L**2+1.D0
      RETURN
      END
C
C
      FUNCTION CENT2(X)
      IMPLICIT DOUBLE PRECISION (A-Z)
      INTEGER II,NU,JJ
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ
      COMMON/EQN/ALAMDA,NU
      COMMON/A/II
      DIMENSION ALAMDA(100)
C       LAM2=ALAMDA(2)
      LAM=ALAMDA(II)
      LAM2=ALAMDA(II)
      A1=0.50*(1.D0+EXP(-2.D0*LAM))+DCOS(LAM)*DEXP(-LAM)
      A2=0.50*(1.D0-DEXP(-2.D0*LAM))+DSIN(LAM)*DEXP(-LAM)
      A3=-1.D0*A1/A2
      A12=0.50*(1.D0+DEXP(-2.D0*LAM2))+DCOS(LAM2)*DEXP(-LAM2)
      A22=0.50*(1.D0-DEXP(-2.D0*LAM2))+DSIN(LAM2)*DEXP(-LAM2)
      A32=-1.D0*A12/A22

      ARGU=LAM*X/L
      ARGUM=LAM2*X/L
      DENOM=DEXP(-ARGU)
      F=(0.50*(1.D0+DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DCOS(ARGU)+
     >A3*(0.50*(1.D0-DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DSIN(ARGU)))
     >/(DENOM)
      DF1=DSINH(ARGUM)+DSIN(ARGUM)+A32*(DCOSH(ARGUM)-DCOS(ARGUM))
      DF1=DF1*LAM
      DF=SINH(ARGU)+SIN(ARGU)+A3*(COSH(ARGU)-COS(ARGU))
      DF=DF*LAM
      DF2=DCOSH(ARGUM)+DCOS(ARGUM)+A32*(DSINH(ARGUM)+DSIN(ARGUM))
      DF2=DF2*LAM*LAM
      CENT2=0.5*(L*L-X*X)*F*DF2/L**2+1.D0
      RETURN
```

```
      END
C
      FUNCTION CENT3(X)
      IMPLICIT DOUBLE PRECISION (A-Z)
      INTEGER II,NU,JJ
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ
      COMMON/EQN/ALAMDA,NU
      COMMON/A/II
      DIMENSION ALAMDA(100)
C       LAM2=ALAMDA(2)
      LAM=ALAMDA(II)
      LAM2=ALAMDA(II)
      A1=0.50*(1.D0+EXP(-2.D0*LAM))+DCOS(LAM)*DEXP(-LAM)
      A2=0.50*(1.D0-DEXP(-2.D0*LAM))+DSIN(LAM)*DEXP(-LAM)
      A3=-1.D0*A1/A2
      A12=0.50*(1.D0+DEXP(-2.D0*LAM2))+DCOS(LAM2)*DEXP(-LAM2)
      A22=0.50*(1.D0-DEXP(-2.D0*LAM2))+DSIN(LAM2)*DEXP(-LAM2)
      A32=-1.D0*A12/A22

      ARGU=LAM*X/L
      ARGUM=LAM2*X/L
      DENOM=DEXP(-ARGU)
      F=(0.50*(1.D0+DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DCOS(ARGU)+
     >A3*(0.50*(1.D0-DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DSIN(ARGU)))
     >/(DENOM)
      DF1=DSINH(ARGUM)+DSIN(ARGUM)+A32*(DCOSH(ARGUM)-DCOS(ARGUM))
      DF1=DF1*LAM
      DF=SINH(ARGU)+SIN(ARGU)+A3*(COSH(ARGU)-COS(ARGU))
      DF=DF*LAM
      DF2=DCOSH(ARGUM)+DCOS(ARGUM)+A32*(DSINH(ARGUM)+DSIN(ARGUM))
      DF2=DF2*LAM*LAM
C
      CENT3=(L-X)*F*DF2/L**2+1.D0
      RETURN
      END
C
      FUNCTION CENT4(X)
      IMPLICIT DOUBLE PRECISION (A-Z)
      INTEGER II,NU,JJ
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ
      COMMON/EQN/ALAMDA,NU
      COMMON/A/II
      DIMENSION ALAMDA(100)
C       LAM2=ALAMDA(2)
      LAM=ALAMDA(II)
      LAM2=ALAMDA(II)
      A1=0.50*(1.D0+EXP(-2.D0*LAM))+DCOS(LAM)*DEXP(-LAM)
      A2=0.50*(1.D0-DEXP(-2.D0*LAM))+DSIN(LAM)*DEXP(-LAM)
      A3=-1.D0*A1/A2
      A12=0.50*(1.D0+DEXP(-2.D0*LAM2))+DCOS(LAM2)*DEXP(-LAM2)
      A22=0.50*(1.D0-DEXP(-2.D0*LAM2))+DSIN(LAM2)*DEXP(-LAM2)
      A32=-1.D0*A12/A22

      ARGU=LAM*X/L
      ARGUM=LAM2*X/L
      DENOM=DEXP(-ARGU)
      F=(0.50*(1.D0+DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DCOS(ARGU)+
     >A3*(0.50*(1.D0-DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DSIN(ARGU)))
     >/(DENOM)
      DF1=DSINH(ARGUM)+DSIN(ARGUM)+A32*(DCOSH(ARGUM)-DCOS(ARGUM))
      DF1=DF1*LAM
      DF=SINH(ARGU)+SIN(ARGU)+A3*(COSH(ARGU)-COS(ARGU))
      DF=DF*LAM
      DF2=DCOSH(ARGUM)+DCOS(ARGUM)+A32*(DSINH(ARGUM)+DSIN(ARGUM))
      DF2=DF2*LAM*LAM
      CENT4=X*F*DF1/L**2+1.D0
```

```
      RETURN
      END
C
      FUNCTION CENT5(X)
      IMPLICIT DOUBLE PRECISION (A-Z)
      INTEGER II,NU,JJ
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ
      COMMON/EQN/ALAMDA,NU
      COMMON/A/II
      DIMENSION ALAMDA(100)
C       LAM2=ALAMDA(2)
      LAM=ALAMDA(II)
      LAM2=ALAMDA(II)
      A1=0.50*(1.D0+EXP(-2.D0*LAM))+DCOS(LAM)*DEXP(-LAM)
      A2=0.50*(1.D0-DEXP(-2.D0*LAM))+DSIN(LAM)*DEXP(-LAM)
      A3=-1.D0*A1/A2
      A12=0.50*(1.D0+DEXP(-2.D0*LAM2))+DCOS(LAM2)*DEXP(-LAM2)
      A22=0.50*(1.D0-DEXP(-2.D0*LAM2))+DSIN(LAM2)*DEXP(-LAM2)
      A32=-1.D0*A12/A22

      ARGU=LAM*X/L
      ARGUM=LAM2*X/L
      DENOM=DEXP(-ARGU)
      F=(0.50*(1.D0+DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DCOS(ARGU)+
     >A3*(0.50*(1.D0-DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DSIN(ARGU)))
     >/(DENOM)
      DF1=DSINH(ARGUM)+DSIN(ARGUM)+A32*(DCOSH(ARGUM)-DCOS(ARGUM))
      DF1=DF1*LAM
      DF=SINH(ARGU)+SIN(ARGU)+A3*(COSH(ARGU)-COS(ARGU))
      DF=DF*LAM
      DF2=DCOSH(ARGUM)+DCOS(ARGUM)+A32*(DSINH(ARGUM)+DSIN(ARGUM))
      DF2=DF2*LAM*LAM
      CENT5=F*DF1
      RETURN
      END
```

a p o l l o   d o m a i n

CAEN/Apollo

```
N     N    A        A    U     U
NN    N   A A      A A   U     U
N N   N  A   A    A   A  U     U
N  N  N A     A  A     A U     U
N   N N AAAAAAA AAAAAAA  U     U
N    NN A     A A     A  U     U
N     N A     A A     A   UUUUU
```

```
i   m     m  ppppp     aa     cccc    ttttt   cccc          ffffff  ttttt  n    n
i   mm   mm  p    p   a  a   c    c     t    c    c         f         t    nn   n
i   m mm m   p    p  a    a  c          t    c              fffff     t    n n  n
i   m     m  ppppp   aaaaaa  c          t    c          ... f         t    n  n n
i   m     m  p       a    a  c    c     t    c    c     ... f         t    n   nn
i   m     m  p       a    a   cccc      t     cccc      ... f         t    n    n
```

//meam/users/naau/soft.dir/impactc.ftn

LAST MODIFIED ON: 88/04/22 4:02 PM
FILE PRINTED: 88/04/28 11:02 AM

```
c
C     test program for report
C     USES GEAR ALGORITHM TO INTEGRATE , APPROPRIATE FOR DIFFERENT LENGTHS
C     MAIN PROGRAM, INTEGRALS ARE FROM OUTSIDE, INPUT: STEP ZERO STEP
C     INCLUDES ROOT LENGTH A
C     ALLOWS MULTIPLE IMPACTS
C     linearized with respect to elastic motion
c     strains are calculated along with deflections.
c     INCLUDES GRAVITY
C     DAMPING INCLUDED JOINT DAMPING (C) AND MODAL DAMP. ZETA
C     INPUT PARAMETERS ARE RECORDED ALONG WITH THE RESULTS
C     total energy is computed
c
c     contact algorithm developed
c     records no. of impacts and impact velocities.
C     MODIFIED TO TAKE THE TOLERANCE INTO ACCOUNT
c     DO  IMPACT WITHIN THE TOLERANCE
C     PRINTS ONLY AT STEP INTERVALS
C     fully variable e as a function of velocity
c

C
C     MAIN PROGRAM
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION L
      DOUBLE PRECISION MA,IR,R1
      INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM,NU
      INTEGER CHOICE
      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ,T1,A,IR,R1
      DATA LU1,LU2,LU3,IPLT,IERR,ISUM/6,5,7,8,0,0/
      CALL INPUT
      CALL TRESP(CHOICE)
      STOP
      END
C
C
C     SUBROUTINE FOR INPUT PARAMATERS
C
      SUBROUTINE INPUT
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      DOUBLE PRECISION L
      DOUBLE PRECISION MA,IR
      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
      INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ,T1,A,IR,R1
      WRITE(LU1,1)
    1 FORMAT(/1X,'PARAMATER INPUTS ;')
      WRITE(LU1,2)
    2 FORMAT(/3X,'ENTER MASS PER UNIT LENGTH OF BEAM (KG/M)',
     >      ' : ',$)
      READ(LU2,*) RO
      WRITE(LU1,3)
    3 FORMAT(/3X,'ENTER FLEXURAL RIGIDITY OF BEAM (NM**2) :',$)
      READ(LU2,*) FLMOD
      WRITE(LU1,4)
    4 FORMAT(3X,'ENTER LENGTH OF THE BEAM (M) :',$)
      READ(LU2,*) L
      WRITE(LU1,8)
    8 FORMAT(3X,'ENTER THE LENGTH OF THE ROOT (M) :',$)
      READ(LU2,*) A
      WRITE(LU1,9)
    9 FORMAT(3X,'ENTER RATIO OF INERTIAS (IR/JBEAM):',$)
      READ(LU2,*) IR
```

```
      WRITE(LU1,5)
    5 FORMAT(3X,'ENTER APPLIED JOINT TORQUE (N-M) ;',$)
      READ(LU2,*)TORQ
      WRITE(LU1,6)
    6 FORMAT(3X,'ENTER DURATION OF PULSE (SECS) :',$)
      READ(LU2,*)TTORQ
      WRITE(LU1,7)
    7 FORMAT(3X,'ENTER FRACTION OF ZERO PULSE :',$)
      READ(LU2,*)T1
      WRITE(LU1,10)
   10 FORMAT(3X,'ENTER RADIUS OF BEAM FOR STRAIN CALC.(M):',$)
      READ(LU2,*)R1
C
      RETURN
      END
C
C
C     SUBROUTINE FOR DETERMINING THE NUMBER OF MODES TO BE INCLUDED
C
      SUBROUTINE NUMRT(NU)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
      INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM,NU
      WRITE(LU1,1)
    1 FORMAT(3X,'ENTER THE NUMBER OF ROOTS YOU WANT FOR EIGENVLS :',$)
      READ(LU2,2) NU
    2 FORMAT(I1)
      RETURN
      END
C
C
C     SUBROUTINE FOR IMPACT PARAMETERS
C
      SUBROUTINE IMPACT(XP,YP,CRES)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
      INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM
      WRITE(LU1,1)
    1 FORMAT(3X,'ENTER X AND Y COORDINATES OF THE STOP :',$)
      READ(LU2,*) XP,YP
      WRITE(LU1,2)
    2 FORMAT(3X,'ENTER COEFFICIENT OF RESTITUTION : ',$)
      READ(LU2,*) CRES
      RETURN
      END
C
C
C     SUBROUTINE FOR ENTERING THE NUMBER OF MODES
C
C
      SUBROUTINE NMODE(NU)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
      INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM,NU
      WRITE(LU1,1)
    1 FORMAT(3X,'ENTER THE NUMBER OF MODES YOU WANT :',$)
      READ(LU2,2)NU
    2 FORMAT(I1)
      RETURN
      END


C
C     SUBROUTINE FOR CALCULATING THE EIGEN FUNCTIONS
```

```
C
C
      SUBROUTINE MODSHAPE(LAM,X,F)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION LAM,L
      DOUBLE PRECISION MA,IR
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ,T1,A,IR,R1
      A1=0.5D0*(1.D0+DEXP(-2.D0*LAM))+DCOS(LAM)*DEXP(-LAM)
      A2=0.5D0*(1.D0-DEXP(-2.D0*LAM))+DSIN(LAM)*DEXP(-LAM)
      A3=-1.D0*A1/A2
      ARGU=LAM*X/L
      DENOM=DEXP(-ARGU)
      F=(0.5D0*(1.D0+DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DCOS(ARGU)+
     >A3*(0.5D0*(1.D0-DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DSIN(ARGU)))
     >/(DENOM)
      RETURN
      END
C
C     SUBROUTINE FOR COMPUTING DERIVATIVES OF EIGENFUNCTIONS
C
C
      SUBROUTINE STRAIN(LAM,X,F)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      DOUBLE PRECISION LAM,L
      DOUBLE PRECISION MA,IR
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ,T1,A,IR,R1
      A1=0.5D0*(1.D0+DEXP(-2.D0*LAM))+DCOS(LAM)*DEXP(-LAM)
      A2=0.5D0*(1.D0-DEXP(-2.D0*LAM))+DSIN(LAM)*DEXP(-LAM)
      A3=-1.D0*A1/A2
      ARGU=LAM*X/L
      DENOM=DEXP(-ARGU)
      A1=0.5D0*(1.D0+DEXP(-2.D0*ARGU))+DCOS(ARGU)*DEXP(-ARGU)
      A2=0.5D0*(1.D0-DEXP(-2.D0*ARGU))+DSIN(ARGU)*DEXP(-ARGU)
      F=(A1+A3*A2)/DENOM
C       F=DCOSH(ARGU)+DCOS(ARGU)+A3*(DSINH(ARGU)+DSIN(ARGU))
      F=F*LAM*LAM/(L*L)

C       F=(0.5D0*(1.D0+DEXP(-2.D0*ARGU))+DEXP(-ARGU)*DCOS(ARGU)+
C     >A3*(0.5D0*(1.D0-DEXP(-2.D0*ARGU))+DEXP(-ARGU)*DSIN(ARGU)))
C     >/(DENOM)
      RETURN
      END

C
C     SUBROUTINE FOR ENTERING THE POSITION ON THE BEAM
C     AT WHICH THE OUTPUT DESIRED
C
C
      SUBROUTINE POSIT(X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
      INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM
      WRITE(LU1,1)
    1 FORMAT(/,'0ENTER BEAM POSITION IN METERS FOR RESPONSE :',$)
      READ(LU2,*) X
      RETURN
      END
C
C     ************
C
C     SUBROUTINE FOR ENTERING THE ROOTS OF EIGENVL. EQN.
C
C
      SUBROUTINE AROOT(ALAMDA,NU)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
      INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM
      DIMENSION ALAMDA(10)
      ALAMDA(1)=1.87510407
      ALAMDA(2)= 4.69409113
      ALAMDA(3)= 7.85475744
      ALAMDA(4)= 10.99554073
      ALAMDA(5)= 14.13716839
      ALAMDA(6)= 17.27875953
      ALAMDA(7)= 20.42035225
      ALAMDA(8)= 23.56194490
      ALAMDA(9)= 26.70353756
      ALAMDA(10)= 29.84513021
      RETURN
      END
C
C
C     SUBROUTINE FOR TRANSIENT RESPONSE AND IMPACT
C
C
      SUBROUTINE TRESP(CHOICE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      DOUBLE PRECISION LAM
      DOUBLE PRECISION MA,IR
      DOUBLE PRECISION L

      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
      COMMON/DAMPIN/DAMP,SDAMP
      INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ,T1,A,IR,R1
      COMMON/DIFLST/TIME,STEP
      COMMON/IC/ A1,A2,A3
      DIMENSION Q(100),ALAMDA(100),QDOT(100),V(100),E(100),UDT2(100)
      DIMENSION SM(11,11),SK(11,11),SF(100),SR(100),VDOT(100)
      DIMENSION EP(100),AA(12,12),B(12),C(12),VTEMP(100)
      INTEGER NEQ,NU,I,J,NCUTS,NCUTSP,NSTEPS,CHOICE,NUM
      INTEGER JK,INDEX,IERROR,MF,N2,NUN,NSTEPS1
      INTEGER IJ,IK,JJ,NUM2,NUMM
      INTEGER IMPAC
      COMMON/EQN/ALAMDA,SR,NU
      COMMON/EQ/SM,SK,SF,V
      COMMON/EQNN/NUN
      DIMENSION ABSCIS(3001),ORD(3001),ENERGY(3001),ORDT(3001)
      DIMENSION ORDTT(3001),ORRT(3001),ORRDT(3001),ORRVT(3001)
      DIMENSION IP(100),POT(3001),AKIN(3001),VTEMP1(100)

      LOGICAL STPSZ

C
C     OPEN THE FILE FOR RESULTS
C
      OPEN(UNIT=3,FILE='*FILE NAME FOR DYNAMIC RESULTS: ')
C
C     OPEN THE FILE FOR WRITING THE INPUT DATA
C
      OPEN(UNIT=8,FILE='*FILE NAME FOR INPUT DATA AND OTHER RESULTS: ')

C
C     ASK FOR NO. OF MODES
C
      CALL NMODE(NU)
C
C     ASK FOR BEGINNIG, AND END OF SIMULATION AND THE STEP SIZE
C
```

```
      CALL NTIMES(TINITL,TFINAL,STEP)
C
C     ASK FOR IMPACT PARAMETERS
C
      CALL IMPACT(XP,YP,RES)

C
C     PARAMETERS FOR INTEGRATION AND IMPACT
C

      NUM=NU+1
      NEQ=2*NUM
      INX=0
      T0=TINITL
      TOUT=STEP
      H0=1.E-6
      EPS=1.E-5
      EPSI=1.E-4
      EPSV=5.E-3
      IERROR=1
      MF=22
      INDEX=1

      NSTEPS=IDINT((TFINAL-TINITL)/STEP+0.1)+1

C
C     INITIAL COND'S
C
      IF(NU.GT.0) CALL AROOT(ALAMDA,NU)
 1003 CALL AIC(A1,A2,A3)
      NUN=NU
      V(1)=A3
      V(2+NU)=A2
      DO 100 I=1,NU
      V(I+1)=0.0
  100 V(2+NU+I)=A1
      DO 1200 I=1,2*NU+2
 1200 UDT2(I)=0.0
C
C
      WRITE(LU1,4000) RO,FLMOD,L,TORQ,TTORQ,NU
 4000 FORMAT(2X,5(F12.4,2X),I2)
C
      DO 74 I=1,NU
      LAM=ALAMDA(I)
      CALL MODSHAPE(LAM,XP,F)
74    EP(I)=F
      CALL POSIT(X)
      WRITE(LU1,111)
  111 FORMAT('0ENTER DAMPING AND MODAL DAMPING :',$)
      READ(LU2,*)DAMP,SDAMP


      TRES=RES
C
C
      WRITE(8,4500)RO,FLMOD,L,A,IR,TORQ,TTORQ,T1,R1,NU,XP,YP,TRES,
     >DAMP,SDAMP,A2,A3
4500  FORMAT(1X,'RO=',F12.6/1X,'EI=',F12.6/1X,'L=',F12.6/1X,'A=',
     >F12.6/1X,'IR/IB =',F12.6/1X,'TORQUE=',F12.6/1X,'TTORQ=',
     >F12.6/1X,'ZERO PULSE =',F12.6/1X,'RADIUS=',F12.6/1X,'NO. OF MODS=
     >',I2/1X,'IMPACT POS. XP,YP',F12.6,1X,F12.6/1X,'COEFF. OF REST.=',
     >F12.6/1X,'JOINT DAMPING C=',F12.6/1X,'MODAL DAMP. ZETA=',F12.6/
     >1X,'INITL. CONDS:'/1X,'TETA0=',F12.6/1X,'TETADOT0=',F12.6)

      N2=2+NU+NU
```

```
      INDEX=1
      IMPAC=0
      IMPA=0
      AJJ=RO*L*(A*A+A*L+L*L/3.D0)*(IR+1.D0)

      G=9.81
      IMPAN=0
      IMPAS=0
C
C     ***** MAIN LOOP STARTS *****
C
      NROW1=0
      KJ=0
      TI=TINITL
      J=0
      STEP1=STEP

201   J=J+1
      ABSCIS(J)=T0
      ORD(J)=0.0
      ORRDT(J)=0.0
C
C
      ORDT(J)=V(2+NU)
      ORDTT(J)=V(1)
      ORRT(J)=V(NU+2)*(XP+A)
      AKIN(J)=AJJ*V(1)*V(1)
      POT(J)=RO*L*L*G*DSIN(V(NU+2))

      DO 1500 I=1,NU
      LAM=ALAMDA(I)
      CALL MODSHAPE(LAM,X,F)
      ORRT(J)=ORRT(J)+V(I+NU+2)*EP(I)
 1500 ORD(J)=ORD(J)+F*V(2+NU+I)
C
      DO 150 I=1,NU
      LAM=ALAMDA(I)
      CALL STRAIN(LAM,X,F)
      POT(J)=POT(J)+(V(I+NU+2)**2*FLMOD*(LAM**4.D0)/(L**3.D0))
      AKIN(J)=AKIN(J)+RO*L*V(I+1)**2+RO*L*V(1)*V(1)*V(I+NU+2)**2
     >+2.D0*SR(I)*V(I+1)*V(1)
  150 ORRDT(J)=ORRDT(J)+F*V(2+NU+I)*R1

      ENERGY(J)=(POT(J)+AKIN(J))/2.D0

      IF(T0.GT.TFINAL) GO TO 202
C
      WP=(XP+A)*V(NU+2)
      DO 1151 I=1,NU
1151  WP=WP+V(2+NU+I)*EP(I)

      WPT=WP
       TTEMP=T0
      TTEMP1=T0
      DO 23 I=1,2*(NU+1)
23    VTEMP(I)=V(I)
2111   CONTINUE

C
C     INTEGRATE
C
2113  CALL DDRIVE(N2,T0,H0,V,TOUT,EPS,IERROR,MF,INDEX)

      WPP=V(1)*(XP+A)
      WP=(A+XP)*V(2+NU)
      DO 151 I=1,NU
```

```
      WPP=WPP+V(I+1)*EP(I)
151   WP=WP+V(2+NU+I)*EP(I)

C
C     FIRST CHECK FOR IMPACT
C
      IF(WP.GT.EPSI) GO TO 2002
      IF(WPP.GT.0.D0) GO TO 2002

211   T0=TTEMP
      DO 24 I=1,2*(NU+1)
24    V(I)=VTEMP(I)

      ISTEP=1

2112  IF(IMPA.EQ.0) STEP=STEP/10.D0
      IMPA=1
C
C     *** INNER LOOP STARTS ****
C
556   TOUT=T0+STEP
      WRITE(6,*) T0,STEP,STEP1
      DO 222 JJ=1,10

      DO 564 I=1,2*(NU+1)
564   VTEMP1(I)=V(I)

C
C     INTEGRATE THROUGH A SMALLER TIME STEP
C
      CALL DDRIVE(N2,T0,H0,V,TOUT,EPS,IERROR,MF,INDEX)

559   WP=(XP+A)*V(NU+2)
      DO 1152 I=1,NU
1152  WP=WP+V(2+NU+I)*EP(I)
      WPP=V(1)*(XP+A)
      DO 152 I=1,NU
152   WPP=WPP+V(I+1)*EP(I)
C
C     WRITE INFO. FOR MONITORING THE INTEGRATION
C
      WRITE(6,2234) T0, V(NU+2),WP,WPP
2234  FORMAT(/,2X, 'TIME=',F12.6,'TETA=',F14.8,2X,'WP=',F14.8,
     >'WPP=',F14.8)
C
C     ** SECOND CHECK FOR IMPACT **
C
C
C     CHECK DISPLACEMENT FOR PENETRATION
C
      IF(WP.GT.EPSI) GO TO 223
C
C     CHECK VELOCITY FOR RETURNING FROM AN IMPACT
C
212   IF(WPP.GE.-EPSV) GO TO 223

C
C     *** CONTROL STATEMENTS FOR COUNTING IMPACTS ***
C

333   IF(IMPAS.EQ.1) THEN
      IMPAN=IMPAN+1
      ENDIF

      WRITE(6,2233) IMPAN
2233  FORMAT(/1X,'NO. OF IMPACT=',I3,'TH*** IMPACT ***')
```

```
C
C     $$$$ FORM THE MOMENTUM BALANCE EQUATIONS $$$$
C

      DO 73 IJ=1,12
      B(IJ)=0.D0
      DO 73 IK=1,12
73    AA(IJ,IK)=0.D0
      AA(1,1)=RO*L*(A*A+L*L/3.D0)*(IR+1.D0)
C
C     CALCULATE THE COEFFICIENT OF RESTITUTION
C
      VV=DABS(WPP)
      IF(VV.LT..2) THEN
      RES=2.578*VV*VV-1.7428*VV+1.D0
      ELSE
      RES=TRES
      ENDIF
      IF(VV.GE..2) RES=.5014*VV**(-.25)
C
C
      DO 301 IJ=1,NU
      AA(IJ+1,1)=SR(IJ)
      AA(1,IJ+1)=SR(IJ)
      AA(IJ+1,IJ+1)=RO*L
      AA(NU+2,IJ+1)=-1.D0*EP(IJ)
301   AA(IJ+1,NU+2)=-1.D0*EP(IJ)
      AA(NU+2,1)=-1.D0*(A+XP)
      AA(1,NU+2)=-1.D0*(A+XP)
      B(1)=AA(1,1)*V(1)
      DO 303 IJ=1,NU
303   B(1)=B(1)+SR(IJ)*V(IJ+1)
      DO 304 IJ=1,NU
304   B(IJ+1)=SR(IJ)*V(1)+RO*L*V(IJ+1)
      B(NU+2)=(A+XP)*V(1)*RES
      DO 302 IJ=1,NU
302   B(NU+2)=B(NU+2)+EP(IJ)*V(IJ+1)*RES
      NUMM=NU+2
C
C     SOLVE FOR  JUMP DISCONTINUITIES IN VELOCITIES
C
      CALL DDEC(NUMM,12,AA,IP,IER)
      CALL DSOL(NUMM,12,AA,B,IP)
C
C     UPDATE THE VELOCITY VECTOR
C
      DO 399 I=1,NU+1
399   V(I)=B(I)

      WPPO=WPP
      WPP=V(1)*(XP+A)
      DO 1552 I=1,NU
1552   WPP=WPP+V(I+1)*EP(I)
C
C     DISPLAY VELOCITIES BEFORE AND AFTER IMPACT
C
      WRITE(6,1553) WPPO,WPP,RES
1553  FORMAT(2X,'VELOCITIES : ***',3F14.6)
      INX=1
      INDEX=1
      GO TO 2122
C
C     *** INNER LOOP ENDS ***
C
223   CONTINUE
      TOUT=TOUT+STEP
```

```
222   CONTINUE

2122  IF(WP.LE.EPSI) THEN
      IMPAS=0
      ELSE
      STEP=STEP1
      IMPAS=1
      ENDIF
2002  CONTINUE
      IF(WP.GT.EPSI) THEN
      STEP=STEP1
      IMPA=0
      IMPAS=1
      ENDIF
C
C     CONTROL STATEMENTS FOR OUTPUT
C
      IF((T0-TI).GE.STEP1.OR.ABSCIS(J).EQ.TINITL) THEN
      TI=T0
      KJ=KJ+1
C
C     OUTPUT
C
2001  WRITE(3,5000) ABSCIS(J),ORD(J),ORDTT(J),ORDT(J),ORRT(J),
     >ORRDT(J),ENERGY(J)
      ELSE
      ENDIF
C
C     **** FIRST LOOP ENDS ****
C
2000  TOUT=TOUT+STEP


  200 GO TO 201
202   CONTINUE

      NROW1=KJ

      WRITE(6,5050)(V(I),I=1,NU*2+2)
 5050 FORMAT(8(2X,F12.6/))
 5000 FORMAT(7F12.6)

C
C     WRITE INPUT DATA
C
C


      WRITE(8,4501)T0,NROW1,IMPAN
4501  FORMAT(1X,'TEND=:',F12.6/1X,'NO. OF POINTS=',
     >I4/1X,'NO. OF IMPACTS=',I3)

      CLOSE(UNIT=3)
      CLOSE(UNIT=8)

208   RETURN
      END
C
C
C     SUBROUTINE FOR INTEGRATION CONSTANTS
C
      SUBROUTINE NTIMES(TINITL,TFINAL,STEP)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
```

```
      INTEGER LU1,LU2,LU3,IPLT,IERR
      WRITE(LU1,1)
    1 FORMAT(/1X,'ENTER INITIAL TIME,FINAL TIME AND INCREMENT :',$)
      READ(LU2,*)TINITL,TFINAL,STEP
      RETURN
      END
C
C
C     INITIAL CONDITIONS
C

      SUBROUTINE AIC(A1,A2,A3)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
      INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM
      WRITE(LU1,1)
    1 FORMAT('0ENTER QDOT(0),TETA(0),TETDOT(0):',$)
      READ(LU2,*) A1,A2,A3
      RETURN
      END

C
C    SUBROUTINE FOR FORMING THE EQUATIONS OF MOTION
C
C
      SUBROUTINE EQNS(SM,SK,SF,V,TIME,UDT2)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      DOUBLE PRECISION L
      DOUBLE PRECISION MA,IR

      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ,T1,A,IR,R1
      COMMON/DAMPIN/DAMP,SDAMP
      COMMON/EQN/ALAMDA,SR,NU
      COMMON/EQNN/NUN
      EXTERNAL SRMAT,AMASS
      INTEGER IPS,NU,I,J,NDIM,NUM,NUN
      INTEGER IP,IER
      DIMENSION IP(100)
C
      DIMENSION V(100),DV(100),SM(11,11),SK(11,11),SF(100),IPS(100)
      DIMENSION ALAMDA(100),SR(100),UDT2(100),CEN(7,7)
      DIMENSION ATEMP(12),TEMP(20),TEMP1(20),SR1(10)
      NDIM=100
      EPS=1.E-7
      NUM=NU+1
      SUM=0.D0
      SUMD=0.D0
      SUME=0.D0
      SUMA=0.D0
      NUN=NU

      PI=3.14156
C
C     ENTER THE VALUES OF MODAL INTEGRALS
C
      CEN(1,1)= 1.193336*RO*L+A*RO*1.570878
      CEN(2,2)=6.478225*RO*L+A*RO*8.647143
      CEN(3,3)=17.859520*RO*L+A*RO*24.952113
      CEN(4,4)=36.055388*RO*L+A*RO*51.459105
      CEN(5,5)=60.801076*RO*L+A*RO*87.792327
      CEN(6,6)=92.129142*RO*L+A*RO*133.999024
      CEN(7,7)=130.036752*RO*L+A*RO*190.075040

      SR1(1)=RO*L*.782992
```

```
      SR1(2)=RO*L*.433936
      SR1(3)=RO*L*.254425
      SR1(4)=RO*L*.181898
      SR1(5)=RO*L*.141471
      SR1(6)=RO*L*.115749
      SR1(7)=RO*L*.097942

      DO 100 I=1,NU
      SR(1)=.568826*L*L*RO+A*RO*L*.782992
      SR(2)=.090767*L*L*RO+A*RO*L*.433936
      SR(3)=0.032416*L*L*RO+A*RO*L*.254425
      SR(4)=0.016542*L*L*RO+A*RO*L*.181898
      SR(5)=0.010007*L*L*RO+A*RO*L*.141471
      SR(6)=0.006699*L*L*RO+A*RO*L*.115749
      SR(7)=0.004796*L*L*RO+A*RO*L*.004796
C
      G=9.81
C
      SUM=SUM+V(I+NU+2)*V(I+NU+2)*RO*L
      SUME=SUME+CEN(I,I)*V(2+NU+I)*V(2+NU+I)
  145 SUMA=SUMA+CEN(I,I)*V(I+1)*V(NU+I+2)
  100 SUMD=SUMD+RO*L*V(2+NU+I)*V(I+1)
C
      DO 105 I=1,11
      DO 105 J=1,11
      SK(I,J)=0.D0
  105 SM(I,J)=0.D0
      SM(1,1)=RO*L*(A*A+A*L+L*L/3.D0)*(IR+1.D0)
C
C     CHECK IF THE MASS MATRIX IS POSITIVE DEFINITE
C
      IF(SM(1,1).LT.0.D0) THEN
      WRITE(6,222) SM(1,1),SUM,SUME
  222 FORMAT(2X,3(F12.6,2X))
      STOP
      ENDIF
C
C     DEFINE THE APPLIED TORQUE PROFILE
C
      TT=(1+T1)*TTORQ
      TTT=(2.+T1)*TTORQ
      TOR=TORQ
      IF(TIME.GT.TTORQ.AND.TIME.LE.TT) TOR=0.D0
      IF(TIME.GT.TT.AND.TIME.LE.TTT) TOR=-TORQ
      IF(TIME.GT.TTT) TOR=0.D0
C*
      SUMD=0.D0
      SF(1)=-(2.D0*SUMD)*V(1)+TOR-DAMP*V(1)
     >-RO*L*G*L*DCOS(V(NU+2))/2.D0
      DO 101 I=1,NU
      SM(1,I+1)=SR(I)
  101 SM(I+1,1)=SR(I)
      DO 200 J=1,NU
C
      SM(J+1,J+1)=RO*L
      SK(J+1,J+1)=FLMOD*(ALAMDA(J)**4.D0)/(L**3.D0)
      ADAMP=SDAMP*SQRT(2.D0*SK(J+1,J+1)*SM(J+1,J+1))
      SF(J+1)=V(2+NU+J)*(RO*L-CEN(J,J))*V(1)*V(1)-ADAMP*V(J+1)
     >-G*SR1(J)*DCOS(V(NU+2))

  200 CONTINUE

      DO 15 I=1,NUM
      TEMP(I)=0.D0
      TEMP1(I)=0.D0
      TEMP(I)=SK(I,I)*V(1+NU+I)
```

```
   15 TEMP1(I)=SF(I)-TEMP(I)
      DO 20 I=1,NUM
   20 UDT2(I)=TEMP1(I)
C

      CALL DDEC(NUM,11,SM,IP,IER)
      CALL DSOL(NUM,11,SM,UDT2,IP)
C
      RETURN
      END
C
C     SUBROUTINE FOR PASSING THE EQUATIONS OF
C     MOTION TO THE INTEGRATOR
C
      SUBROUTINE DDIFUN(N,T,V,VDOT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      INTEGER I,NU,NUM,N
C
      DIMENSION V(100),VDOT(100),UDT2(100)
      DIMENSION SM(11,11),SK(11,11),SC(4,4),SF(100)
      COMMON/EQNN/NU
      COMMON/DAMPIN/DAMP,SDAMP
      NUM=NU+1
      CALL EQNS(SM,SK,SF,V,T,UDT2)
      DO 10 I=1,NUM
      VDOT(I+NU+1)=V(I)
   10 VDOT(I)=UDT2(I)
      RETURN
      END
C
      SUBROUTINE DPDERV(N,T,Y,PD,MO)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      RETURN
      END
```

```
N     N   A       A   U     U
NN    N  A A     A A  U     U
N N   N A   A   A   A U     U
N  N  N A     A A     A U     U
N   N N AAAAAAA AAAAAAA U     U
N    NN A     A A     A U     U
N     N A     A A     A  UUUUU
```

```
i   m     m  ppppp     aa      cccc    ttttt  bbbbb          ffffff  ttttt  n    n
i   mm   mm  p    p   a  a    c    c     t    b    b         f         t    nn   n
i   m mm m   p    p  a    a   c          t    bbbbb          fffff     t    n n  n
i   m     m  ppppp   aaaaaa   c          t    b    b   ...   f         t    n  n n
i   m     m  p       a    a   c    c     t    b    b   ...   f         t    n   nn
i   m     m  p       a    a    cccc      t    bbbbb    ...   f         t    n    n
```

//meam/users/naau/soft.dir/impactb.ftn

LAST MODIFIED ON: 88/04/28 3:38 PM
FILE PRINTED: 88/05/03 10:27 AM

```
C
C      IMPACT BEAM MODEL WITH HERTZIAN LAW
C      USES GEAR ALGORITHM TO INTEGRATE , APPROPRIATE FOR DIFFERENT LENGTHS
C      MAIN PROGRAM, INTEGRALS ARE FROM OUTSIDE, INPUT: STEP ZERO STEP
C      INCLUDES ROOT LENGTH A
C      ALLOWS MULTIPLE IMPACTS
C      linearized with respect to elastic motion
c      strains are calculated instead of deflections.
c      INCLUDES GRAVITY
C      DAMPING INCLUDED JOINT DAMPING (C) AND MODAL DAMP. ZETA
C      INPUT PARAMETERS ARE RECORDED ALONG WITH THE RESULTS
C      total energy is computed
c      STEP IS REDUCED IN THE BAND
C      writing is adjusted for step size
c      coeff. of rest. is calculated
c

       IMPLICIT DOUBLE PRECISION (A-H,O-Z)


       DOUBLE PRECISION L
       DOUBLE PRECISION MA,IR
       INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM,NU
       INTEGER CHOICE
       COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
       COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ,T1,A,IR,R1
       DATA LU1,LU2,LU3,IPLT,IERR,ISUM/6,5,7,8,0,0/
       CALL INPUT
       CALL TRESP(CHOICE)
       STOP
       END
C
C
C
C      SUBROUTINE FOR INPUT PARAMATERS
C
       SUBROUTINE INPUT
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)


       DOUBLE PRECISION L
       DOUBLE PRECISION MA,IR
       COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
       INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM
       COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ,T1,A,IR,R1
       WRITE(LU1,1)
     1 FORMAT(/1X,'PARAMATER INPUTS ;')
       WRITE(LU1,2)
     2 FORMAT(/3X,'ENTER MASS PER UNIT LENGTH OF BEAM (KG/M)',
      >        ' : ',$)
       READ(LU2,*) RO
       WRITE(LU1,3)
     3 FORMAT(/3X,'ENTER FLEXURAL RIGIDITY OF BEAM (NM**2) :',$)
       READ(LU2,*) FLMOD
       WRITE(LU1,4)
     4 FORMAT(3X,'ENTER LENGTH OF THE BEAM (M) :',$)
       READ(LU2,*) L
       WRITE(LU1,8)
     8 FORMAT(3X,'ENTER THE LENGTH OF THE ROOT (M) :',$)
       READ(LU2,*) A
       WRITE(LU1,9)
     9 FORMAT(3X,'ENTER RATIO OF INERTIAS (IR/JBEAM):',$)
       READ(LU2,*) IR
       WRITE(LU1,5)
     5 FORMAT(3X,'ENTER APPLIED JOINT TORQUE (N-M) ;',$)
       READ(LU2,*)TORQ
```

```
      WRITE(LU1,6)
    6 FORMAT(3X,'ENTER DURATION OF PULSE (SECS) :',$)
      READ(LU2,*)TTORQ
      WRITE(LU1,7)
    7 FORMAT(3X,'ENTER FRACTION OF ZERO PULSE :',$)
      READ(LU2,*)T1
      WRITE(LU1,10)
   10 FORMAT(3X,'ENTER RADIUS OF BEAM FOR STRAIN CALC.(M):',$)
      READ(LU2,*)R1
C
      RETURN
      END
C
C
C     SUBROUTINE FOR DETERMINING THE NUMBER OF MODES TO BE INCLUDED
C
      SUBROUTINE NUMRT(NU)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)


      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
      INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM,NU
      WRITE(LU1,1)
    1 FORMAT(3X,'ENTER THE NUMBER OF ROOTS YOU WANT FOR EIGENVLS :',$)
      READ(LU2,2) NU
    2 FORMAT(I1)
      RETURN
      END
C
C
C     SUBROUTINE FOR IMPACT PARAMETERS
C
      SUBROUTINE IMPACT(XP,YP,CRES,CDAMP)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)


      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
      INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM
      DOUBLE PRECISION XP,YP,CRES,CDAMP
      WRITE(LU1,1)
    1 FORMAT(3X,'ENTER X AND Y COORDINATES OF THE STOP :',$)
      READ(LU2,*) XP,YP
      WRITE(LU1,2)
    2 FORMAT(3X,'ENTER SPRING COEFFICIENT : ',$)
      READ(LU2,*) CRES
      WRITE(LU1,3)
    3 FORMAT(3X,'ENTER DAMPING COEFFICIENT : ',$)
      READ(LU2,*) CDAMP
      RETURN
      END
C
C
C
      SUBROUTINE NMODE(NU)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)


      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
      INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM,NU
      WRITE(LU1,1)
    1 FORMAT(3X,'ENTER THE NUMBER OF MODES YOU WANT :',$)
      READ(LU2,2)NU
    2 FORMAT(I1)
      RETURN
      END
```

```
      SUBROUTINE MODSHAPE(LAM,X,F)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)


      DOUBLE PRECISION LAM,L
      DOUBLE PRECISION MA,IR
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ,T1,A,IR,R1
      A1=0.5D0*(1.D0+DEXP(-2.D0*LAM))+DCOS(LAM)*DEXP(-LAM)
      A2=0.5D0*(1.D0-DEXP(-2.D0*LAM))+DSIN(LAM)*DEXP(-LAM)
      A3=-1.D0*A1/A2
      ARGU=LAM*X/L
      DENOM=DEXP(-ARGU)
      F=(0.5D0*(1.D0+DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DCOS(ARGU)+
     >A3*(0.5D0*(1.D0-DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DSIN(ARGU)))
     >/(DENOM)
      RETURN
      END
C
C     SUBROUTINE FOR THIRD DERIVATIVE OF MODE SHAPE
C
C
      SUBROUTINE MODSHAPE2(LAM,X,F)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)


      DOUBLE PRECISION LAM,L
      DOUBLE PRECISION MA,IR
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ,T1,A,IR,R1
      A1=0.5D0*(1.D0+DEXP(-2.D0*LAM))+DCOS(LAM)*DEXP(-LAM)
      A2=0.5D0*(1.D0-DEXP(-2.D0*LAM))+DSIN(LAM)*DEXP(-LAM)
      A3=-1.D0*A1/A2
      ARGU=LAM*X/L
      DENOM=DEXP(-ARGU)
C
      F=DSINH(ARGU)-DSIN(ARGU)+A3*(DCOSH(ARGU)+DCOS(ARGU))
      F=F*LAM*LAM*LAM/(L*L*L)
C       F=(0.5D0*(1.D0+DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DCOS(ARGU)+
C     >A3*(0.5D0*(1.D0-DEXP(-2.D0*ARGU))-DEXP(-ARGU)*DSIN(ARGU)))
C     >/(DENOM)
      RETURN
      END
C
C
C     SUBROUTINE FOR COMPUTING DERIVATIVES OF EIGENFUNCTIONS
C

      SUBROUTINE STRAIN(LAM,X,F)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)


      DOUBLE PRECISION LAM,L
      DOUBLE PRECISION MA,IR
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ,T1,A,IR,R1
      A1=0.5D0*(1.D0+DEXP(-2.D0*LAM))+DCOS(LAM)*DEXP(-LAM)
      A2=0.5D0*(1.D0-DEXP(-2.D0*LAM))+DSIN(LAM)*DEXP(-LAM)
      A3=-1.D0*A1/A2
      ARGU=LAM*X/L
      DENOM=DEXP(-ARGU)
      A1=0.5D0*(1.D0+DEXP(-2.D0*ARGU))+DCOS(ARGU)*DEXP(-ARGU)
      A2=0.5D0*(1.D0-DEXP(-2.D0*ARGU))+DSIN(ARGU)*DEXP(-ARGU)
      F=(A1+A3*A2)/DENOM
C       F=DCOSH(ARGU)+DCOS(ARGU)+A3*(DSINH(ARGU)+DSIN(ARGU))
      F=F*LAM*LAM/(L*L)
```

```
C        F=(0.5D0*(1.D0+DEXP(-2.D0*ARGU))+DEXP(-ARGU)*DCOS(ARGU)+
C       >A3*(0.5D0*(1.D0-DEXP(-2.D0*ARGU))+DEXP(-ARGU)*DSIN(ARGU)))
C       >/(DENOM)
        RETURN
        END

C
C       SUBROUTINE FOR POSITION ON THE BEAM
C

        SUBROUTINE POSIT(X)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)


        COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
        INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM
        WRITE(LU1,1)
      1 FORMAT(/,'0ENTER BEAM POSITION IN METERS FOR RESPONSE :',$)
        READ(LU2,*) X
        RETURN
        END
C
C       ************
C
C       SUBROUTINE FOR ROOTS OF EIGENVL.EQN
C
        SUBROUTINE AROOT(ALAMDA,NU)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)


        COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
        INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM
        DIMENSION ALAMDA(10)
        ALAMDA(1)=1.87510407
        ALAMDA(2)= 4.69409113
        ALAMDA(3)= 7.85475744
        ALAMDA(4)= 10.99554073
        ALAMDA(5)= 14.13716839
        ALAMDA(6)= 17.27875953
        ALAMDA(7)= 20.42035225
        ALAMDA(8)= 23.56194490
        ALAMDA(9)= 26.70353756
        ALAMDA(10)= 29.84513021
        RETURN
        END
C

C
C       SUBROUTINE FOR TRANSIENT RESPONSE AND IMPACT
C
C
        SUBROUTINE TRESP(CHOICE)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)


        DOUBLE PRECISION LAM
        DOUBLE PRECISION MA,IR,L

        COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
        COMMON/DAMPIN/DAMP,SDAMP,FORCE
        INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM
        COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ,T1,A,IR,R1
        COMMON/DIFLST/TIME,STEP
        COMMON/IC/ A1,A2,A3
        DIMENSION Q(100),ALAMDA(100),QDOT(100),V(100),E(100),UDT2(100)
        DIMENSION SM(11,11),SK(11,11),SF(100),SR(100),VDOT(100)
```

```
      DIMENSION EP(100),AA(12,12),B(12),C(12),VTEMP(100)
      INTEGER NEQ,NU,I,J,NCUTS,NCUTSP,NSTEPS,CHOICE,NUM
      INTEGER JK,INDEX,IERROR,MF,N2,NUN,NSTEPS1
      INTEGER IJ,IK,JJ,NUM2,NUMM
      INTEGER IMPAC
      COMMON/EQN/ALAMDA,SR,NU
      COMMON/EQ/SM,SK,SF,V
      COMMON/TCON/TC,EE,IMPY,IMPZ
      COMMON/VCON/V1,V2,TSTR,TSTP
      COMMON/EQNN/NUN

      DIMENSION EP1(100)
      DIMENSION IP(100)
      COMMON/CON/RES,CDAMP,XP,EP,EP1


      LOGICAL STPSZ

C
C
      OPEN(UNIT=3,FILE='*FILE NAME FOR RESULTS: ')
      OPEN(UNIT=8,FILE='*FILE NAME FOR INPUT DATA AND OTHER RESULTS: ')


C     INTEGRATION
C
      CALL NMODE(NU)
      CALL NTIMES(TINITL,TFINAL,STEP)
      CALL IMPACT(XP,YP,RES,CDAMP)

C
C     PARAMETERS FOR INTEGRATION AND IMPACT
C

      NUM=NU+1
      NEQ=2*NUM
      INX=0
      T0=TINITL
       TOUT=STEP
      H0=1.E-8
      EPS=1.E-5
      EPSI=1.E-3
      IERROR=1
      MF=22
      INDEX=1

      NSTEPS=IDINT((TFINAL-TINITL)/STEP+0.1)+1



C
C     INITIAL COND'S
C
      IF(NU.GT.0) CALL AROOT(ALAMDA,NU)
 1003 CALL AIC(A1,A2,A3,A4)
      NUN=NU
      V(1)=A3
      V(2+NU)=A2
      DO 100 I=1,NU
      V(I+1)=0.D0
  100 V(2+NU+I)=0.D0
      V(2)=A1/2.D0
      V(NU+3)=A4/2.D0
      DO 1200 I=1,2*NU+2
 1200 UDT2(I)=0.0
C
```

```
C
C        V(2)=.01
      WRITE(LU1,4000) RO,FLMOD,L,TORQ,TTORQ,NU
 4000 FORMAT(2X,5(F12.4,2X),I2)
C
      DO 74 I=1,NU
      LAM=ALAMDA(I)
      CALL MODSHAPE2(LAM,XP,F)
      EP1(I)=F
      CALL MODSHAPE(LAM,XP,F)
74    EP(I)=F
      CALL POSIT(X)
      WRITE(LU1,111)
  111 FORMAT('0ENTER DAMPING AND MODAL DAMPING :',$)
      READ(LU2,*)DAMP,SDAMP



       N2=2+NU+NU
      INDEX=1
      IMPAC=0
      IMPA=0
      IMPAR=0
      IMPY=0
      IMPZ=0

C
C     WRITE INPUT PARAMETERS
C

      WRITE(8,4500)RO,FLMOD,L,A,IR,TORQ,TTORQ,T1,R1,NU,XP,YP,RES,
     >CDAMP,DAMP,SDAMP,A2,A3,X
4500  FORMAT(1X,'RO=',F12.6/1X,'EI=',F12.6/1X,'L=',F12.6/1X,'A=',
     >F12.6/1X,'IR/IB =',F12.6/1X,'TORQUE=',F12.6/1X,'TTORQ=',
     >F12.6/1X,'ZERO PULSE =',F12.6/1X,'RADIUS=',F12.6/1X,'NO. OF MODS=
     >',I2/1X,'IMPACT POS. XP,YP',F12.6,1X,F12.6/1X,'SPRING COEFF.=',
     >G16.6/1X,'IMPACT DAMPING CDAMP=',F12.6/1X,'JOINT DAMPING C='
     >,F12.6,/1X,'MODAL DAMP. ZETA=',F12.6/
     >,1X,'INITL. CONDS:'/1X,'TETA0=',F12.6/1X,'TETADOT0=',F12.6
     >,/1X,'OUTPUT POS. (M).:',F12.6)


      AJJ=RO*L*(A*A+A*L+L*L/3.D0)*(IR+1.D0)
      G=9.81

      NROW1=0
      KJ=0
      TI=TINITL
      J=0
      STEP1=STEP
      STEP2=STEP/100.D0
201   J=J+1

      ABSCIS=T0
      ORD=0.0
      ORRDT=0.0
C      UDT2=0.0
C
C
      ORDT=V(2+NU)
      ORDTT=V(1)
      ORRT=(XP+A)*V(1)
      AKIN=AJJ*V(1)*V(1)
      POT=RO*L*L*G*DSIN(V(NU+2))

      ORRVT=A3+S1*OMEG*A1*SIN(OMEG*T0)/AJ
```

```
      DO 1500 I=1,NU
      LAM=ALAMDA(I)
      CALL MODSHAPE(LAM,X,F)
      ORRT=ORRT+V(I+1)*EP(I)
 1500 ORD=ORD+F*V(2+NU+I)
C
      DO 150 I=1,NU
      LAM=ALAMDA(I)
      CALL STRAIN(LAM,X,F)
      POT=POT+(V(I+NU+2)**2*FLMOD*(LAM**4.D0)/(L**3.D0))
      AKIN=AKIN+RO*L*V(I+1)**2+RO*L*V(1)*V(1)*V(I+NU+2)**2
     >+2.D0*SR(I)*V(I+1)*V(1)

  150 ORRDT=ORRDT+F*V(2+NU+I)*R1
      ENERGY=(POT+AKIN)/2.D0

      IF(T0.GT.TFINAL) GO TO 202
C

      WP=(XP+A)*V(NU+2)
      DO 1151 I=1,NU
1151  WP=WP+V(2+NU+I)*EP(I)
      WPT=WP
       TTEMP=T0
      DO 23 I=1,2*(NU+1)
23    VTEMP(I)=V(I)

      WP=(A+XP)*V(2+NU)
      DO 151 I=1,NU
151   WP=WP+V(2+NU+I)*EP(I)

C
C     FIRST IMPACT CHECK TO REDUCE THE TIME STEP
C
      IF(WP.LE.EPSI) THEN
      STEP=STEP2
      TOUT=T0+STEP
      ELSE
      STEP=STEP1
      TOUT=T0+STEP
      ENDIF


      WP=(XP+A)*V(NU+2)
      DO 1152 I=1,NU
1152  WP=WP+V(2+NU+I)*EP(I)
      WPP=V(1)*(XP+A)
      DO 152 I=1,NU
152   WPP=WPP+V(I+1)*EP(I)

      WPN=0.D0
      IF(WP.LE.0.D0) THEN
      WPN=-WP
      ELSE
      WPN=0.D0
      ENDIF
C
C     INTEGRATE
C
      CALL DDRIVE(N2,T0,H0,V,TOUT,EPS,IERROR,MF,INDEX)
      IF(FORCE.GT.0.D0) THEN
      WRITE(6,2233) T0,FORCE,WP,WPP
      CONTINUE
2233  FORMAT(1X,'TIME=',F12.6,'FORCE=',F12.6,
     >'WP=',F12.6,'WPP=',F12.6,'* IMPACT *')
      ENDIF
```

```
C
C     *** INNER LOOP ENDS ***
C
223   CONTINUE

222   CONTINUE

2002  CONTINUE

      IF((T0-TI).GE.STEP1.OR.ABSCIS.EQ.TINITL) THEN
      TI=T0


      KJ=KJ+1
2001  WRITE(3,5000) ABSCIS,ORD,ORDTT,ORDT,ORRT,
     >ORRDT,ENERGY
      ELSE
      ENDIF
C
C     **** FIRST LOOP ENDS ****
C
2000  TOUT=TOUT+STEP


  200 GO TO 201
202   CONTINUE

      NROW1=KJ


      WRITE(6,5050)(V(I),I=1,NU*2+2)
 5050 FORMAT(8(2X,F12.6/))
 5000 FORMAT(7F12.6)

C
C     WRITE INPUT DATA
C
C


      WRITE(8,4501)T0,NROW1,TC,EE,V1,TSTR
4501  FORMAT(1X,'TEND=:',F12.6/1X,'NO. OF POINTS=',
     >I6/1X,'DURATION OF CONT.=',F14.8,
     >/1X,'COEFF. OF REST.=',F12.6,
     >/1X,'IMPACT VEL. (M/S)=',F12.6,
     >/1X,'FIRST IMPACT OCCURS AT T=',F12.6)

C


      CLOSE(UNIT=3)
      CLOSE(UNIT=8)

      RETURN
      END
C
C
C     SUBROUTINE FOR INTEGRATION CONSTANTS
C
      SUBROUTINE NTIMES(TINITL,TFINAL,STEP)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)


      DOUBLE PRECISION TINITL,TFINAL,STEP
      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
```

```
      INTEGER LU1,LU2,LU3,IPLT,IERR
      WRITE(LU1,1)
    1 FORMAT(/1X,'ENTER INITIAL TIME,FINAL TIME AND INCREMENT :',$)
      READ(LU2,*)TINITL,TFINAL,STEP
      RETURN
      END
C
C
C     INITIAL CONDITIONS
C

      SUBROUTINE AIC(A1,A2,A3,A4)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)


      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
      INTEGER LU1,LU2,LU3,IPLT,IERR,ISUM
      WRITE(LU1,1)
    1 FORMAT('0ENTER WDOT(L,0),TETA(0),TETDOT(0),W(L,0):',$)
      READ(LU2,*) A1,A2,A3,A4
      RETURN
      END
C
C
C     SUBROUTINE FOR OUTPUT
C
      SUBROUTINE RESOUT(X,Y,PT,T1,T2,STEP,CHOICE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)


      COMMON/EQN/ALAMDA,SR,NU
      COMMON/LUN/LU1,LU2,LU3,IPLT,IERR,ISUM
      INTEGER LU1,LU2,LU3,IPLT,IERR,NSTEPS,ISUM
      DIMENSION X(1),Y(1),ALAMDA(100),SR(100)
      INTEGER I,J,CHOICE,NU
      WRITE(LU1,1) PT,NU
    1 FORMAT(///1X,65('.'),//1X,'ASSUMED MODE TIME RESPONSE',
     >          'OF POINT X= ',F6.2, 'ON BEAM USING NU=',I2,/1X,65('.')/)
      WRITE(LU1,2)
    2 FORMAT(/1X,6X,'TIME',19X,'W(X,T)',//)
      J=0
      NSTEPS=IDINT((T2-T1)/STEP+.1)+1
      DO 100 I=1,NSTEPS
      J=J+1
  100 WRITE(LU1,3) X(J),Y(J)
    3 FORMAT(5X,F7.3,10X,E20.12)
      RETURN
      END

C
C    SUBROUTINE FOR FORMING THE EQUATIONS OF MOTION
C
      SUBROUTINE EQNS(SM,SK,SF,V,TIME,UDT2)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)


      DOUBLE PRECISION L
      DOUBLE PRECISION MA,IR
      COMMON/PLIST/FLMOD,SIGMA,L,RO,MA,TORQ,TTORQ,T1,A,IR,R1
      COMMON/DAMPIN/DAMP,SDAMP,FORCE
      COMMON/CON/RES,CDAMP,XP,EP,EP1
      COMMON/EQN/ALAMDA,SR,NU
      COMMON/EQNN/NUN
      COMMON/TCON/TC,EE,IMPY,IMPZ
      COMMON/VCON/V1,V2,TSTR,TSTP
```

```
      EXTERNAL SRMAT,AMASS
      INTEGER IPS,NU,I,J,NDIM,NUM,NUN
      INTEGER IP,IER
      DIMENSION IP(100)
C
      DIMENSION V(100),DV(100),SM(11,11),SK(11,11),SF(100),IPS(100)
      DIMENSION ALAMDA(100),SR(100),UDT2(100),CEN(7,7)
      DIMENSION ATEMP(12),TEMP(20),TEMP1(20),SR1(10)
      DIMENSION EP(100), EP1(100)
      NDIM=100
      EPS=1.E-7
      EPSA=1.E-4
      NUM=NU+1
      SUM=0.D0
      SUMD=0.D0
      SUME=0.D0
      SUMA=0.D0
      NUN=NU
      PI=3.14156
C
C     CONTACT FORCE CALCULATION
C
      FORCE=0.D0
      WP=(XP+A)*V(NU+2)
      DO 1152 I=1,NU
1152  WP=WP+V(2+NU+I)*EP(I)
      WPP=V(1)*(XP+A)
      DO 152 I=1,NU
152   WPP=WPP+V(I+1)*EP(I)
      WPN=-WP

C
C     CALCULATE THE DAMPING COEFFICIENT
C
      DDAMP=.75*CDAMP*RES
      IF(WPN.LE.0.D0) WPN=0.D0

      FORCE=RES*WPN*DSQRT(WPN)+DDAMP*(-WPP)*WPN
C
C     CHECK IF SEPARATION OCCURS
C
      IF(FORCE.LE.0.D0) FORCE=0.D0

C
C     DETERMINE THE BEGINNING OF IMPACT
C
      IF(IMPY.EQ.1) THEN
      IMPR=0
      GO TO 201
      ELSE
      ENDIF
      IF(WP.LE.0.D0.AND.WP.GT.-EPSA.AND.
     >WPP.LT.0.D0) THEN
      IMPR=1
      IMPY=1
      ELSE
      IMPR=0
      ENDIF
201   CONTINUE

      IF(IMPR.EQ.1.AND.IMPY.EQ.1) THEN
      V1=-WPP
      TSTR=TIME
      WRITE(6,221) TSTR,V1
      ELSE
      ENDIF
```

```
      IF(IMPZ.EQ.1) THEN
      IMPP=0
      GO TO 210
      ELSE
      ENDIF
c       IF(WP.GE.0.D0.AND.WP.LT.EPSA.AND.
c      >WPP.GT.0.D0) THEN
      IF((WP.GE.0.D0.OR.FORCE.LE.0.D0).AND.WP.LT.EPSA.
     >AND.WPP.GT.0.D0) THEN

C
C     DETERMINE THE END OF IMPACT AND THE COEFFICIENT OF
C     RESTITUTION
C
      IMPP=1
      IMPZ=1
      ELSE
      IMPP=0
      ENDIF
210   CONTINUE
C
C     CALCULATE THE COEFFICIENT OF RESTITUTION
C     AND DETERMINE THE CONTACT DURATION
C
      IF(IMPP.EQ.1.AND.IMPZ.EQ.1)  THEN
      V2=WPP
      TSTP=TIME
      WRITE(6,222) TSTP,V2
      IF(V1.NE.0) EE=V2/V1
      TC=TSTP-TSTR

      ELSE
      ENDIF
221   FORMAT(/2X,'TSTR=',F14.8,'V1=',F12.6)
222   FORMAT(/2X,'TSTP=',F14.8,'V2=',F12.6)

C
C     VALUES OF THE MODAL INTEGRALS
C
      CEN(1,1)= 1.193336*RO*L+A*RO*1.570878
      CEN(2,2)=6.478225*RO*L+A*RO*8.647143
      CEN(3,3)=17.859520*RO*L+A*RO*24.952113
      CEN(4,4)=36.055388*RO*L+A*RO*51.459105
      CEN(5,5)=60.801076*RO*L+A*RO*87.792327
      CEN(6,6)=92.129142*RO*L+A*RO*133.999024
      CEN(7,7)=130.036752*RO*L+A*RO*190.075040

      G=9.81
      SR1(1)=RO*L*.782992
      SR1(2)=RO*L*.433936
      SR1(3)=RO*L*.254425
      SR1(4)=RO*L*.181898
      SR1(5)=RO*L*.141471
      SR1(6)=RO*L*.115749
      SR1(7)=RO*L*.097942

      DO 100 I=1,NU
      SR(1)=.568826*L*L*RO+A*RO*L*.782992
      SR(2)=.090767*L*L*RO+A*RO*L*.433936
      SR(3)=0.032416*L*L*RO+A*RO*L*.254425
      SR(4)=0.016542*L*L*RO+A*RO*L*.181898
      SR(5)=0.010007*L*L*RO+A*RO*L*.141471
      SR(6)=0.006699*L*L*RO+A*RO*L*.115749
      SR(7)=0.004796*L*L*RO+A*RO*L*.004796
C
```

```
      SUM=SUM+V(I+NU+2)*V(I+NU+2)*RO*L
      SUME=SUME+CEN(I,I)*V(2+NU+I)*V(2+NU+I)
  145 SUMA=SUMA+CEN(I,I)*V(I+1)*V(NU+I+2)
  100 SUMD=SUMD+RO*L*V(2+NU+I)*V(I+1)
C
      DO 105 I=1,11
      DO 105 J=1,11
      SK(I,J)=0.D0
  105 SM(I,J)=0.D0
      SM(1,1)=RO*L*(A*A+A*L+L*L/3.D0)*(IR+1.D0)
C
C     STOP IF THE MASS MATRIX IS NOT POS. DEFINITE
C
      IF(SM(1,1).LT.0.D0) THEN
      STOP
      ENDIF
C
C     COMPUTE THE APPLIED TORQUE PROFILE
C
      TT=(1+T1)*TTORQ
      TTT=(2.+T1)*TTORQ
      TOR=TORQ
      IF(TIME.GT.TTORQ.AND.TIME.LE.TT) TOR=0.D0
      IF(TIME.GT.TT.AND.TIME.LE.TTT) TOR=-TORQ
      IF(TIME.GT.TTT) TOR=0.D0
C*
      SUMD=0.D0
      SF(1)=-(2.D0*SUMD)*V(1)+TOR-DAMP*V(1)
     >-RO*L*G*L*DCOS(V(NU+2))/2.D0+FORCE*XP
      DO 101 I=1,NU
      SM(1,I+1)=SR(I)
  101 SM(I+1,1)=SR(I)
      DO 200 J=1,NU
C
      AI=1.0
      SM(J+1,J+1)=RO*L
      IF(FORCE.EQ.0.D0) AI=0.0
      SK(J+1,J+1)=FLMOD*(ALAMDA(J)**4.D0)/(L**3.D0)
      DO 102 I=1,NU
      IF(I.EQ.J) GO TO 102
      AI=1.0
      IF(FORCE.EQ.0.D0) AI=0.0
  102 CONTINUE
      ADAMP=SDAMP*SQRT(2.D0*SK(J+1,J+1)*SM(J+1,J+1))
      SF(J+1)=V(2+NU+J)*(RO*L-CEN(J,J))*V(1)*V(1)-ADAMP*V(J+1)
     >-G*SR1(J)*DCOS(V(NU+2))+FORCE*EP(J)

  200 CONTINUE

      DO 15 I=1,NUM
      TEMP(I)=0.D0
      TEMP1(I)=0.D0
      TEMP(I)=SK(I,I)*V(1+NU+I)
   15 TEMP1(I)=SF(I)-TEMP(I)
      DO 20 I=1,NUM
   20 UDT2(I)=TEMP1(I)
C

      CALL DDEC(NUM,11,SM,IP,IER)
      CALL DSOL(NUM,11,SM,UDT2,IP)
C
      RETURN
      END
c
c
      SUBROUTINE DDIFUN(N,T,V,VDOT)
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      INTEGER I,NU,NUM,N
C
      DIMENSION V(100),VDOT(100),UDT2(100)
      DIMENSION SM(11,11),SK(11,11),SC(4,4),SF(100)
      COMMON/EQNN/NU
      COMMON/TCON/TC,EE,IMPY,IMPZ
      COMMON/VCON/V1,V2,TSTR,TSTP


      COMMON/DAMPIN/DAMP,SDAMP,FORCE
      NUM=NU+1
      CALL EQNS(SM,SK,SF,V,T,UDT2)
      DO 10 I=1,NUM
      VDOT(I+NU+1)=V(I)
   10 VDOT(I)=UDT2(I)
      RETURN
      END
C
      SUBROUTINE DPDERV(N,T,Y,PD,MO)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      RETURN
      END
```

References

[1] Yigit, A., *Dynamics of A RAdially Rotating Beam with Impact: Implications for Robotics*, Ph.D Thesis, The University of Michigan, May 1988.

[2] Harding, L.J., "Numerical Analysis and Applications Software Abstracts," University of Michigan Computing Center Memo 407, 1979.

[3] Byrne, G.D., Hindmarsh, A.C., "A Polyalgorithm for the Numerical Solution of Ordinary Differential Equations," *ACM Transactions on Mathematical Software*, Vol.1, 1975, pp.71-76.