A Computational Package for Design and Process Evaluation in Bandsawing - A User's Manual

by

James E. Borchelt

A. Galip Ulsoy

and

Panos Papalambros

UM-MEAM-83-14

Department of Mechanical Engineering
and Applied Mechanics

and

Integrated Design and Manufacturing
Unit of the Center for Robotics and
Integrated Manufacturing

The University of Michigan
Ann Arbor, MI 48109

Table of Contents

Description of Program

The stress analysis program developed was combined with the previously developed vibration analysis program ("Two Computer Codes for Band Saw Vibration and Stability Analysis", A.G. Ulsoy, University of California, Berkley, 1978.) to form a computationally efficient interactive design program. The report "Development of an Efficient Computational Procedure for Evaluating Band Saw Blade Stresses" (J.E. Borchelt, A.G. Ulsoy, P. Papalambros, University of Michigan, 1983) presents the theory used to develop the stress analysis program. The report "Vibration and Stability of Band Saw Blades: A Theoretical and Experimental Study" (Ph.D. Dissertation, A.G. Ulsoy, University of California, Berkley, 1979.) presents the theory used to develop the vibration analysis program. Figure 1 shows the total desired automatic design computation package. The process parameters, are $\underline{P}$, $\underline{\sigma}$ are the in-plane stresses, $\underline{\omega}$ are the natural frequencies, and J is the performance criterion. The existing program consists of the stress and vibration analysis segments and can stand alone as a design package. The user can iteratively use the existing program as a computer-aided design package. The stresses and/or natural frequencies can be computed for various design parameters and process conditions, allowing the designer to make direct comparisons.

Three possible paths exist that can be taken by the program. Figure 2 shows which path(s) the user can choose. The path taken by the program can be changed once a given problem has been solved without restarting the program. Figure 3 shows the subroutines called when only the in-plane stress analysis, Path 1, is chosen

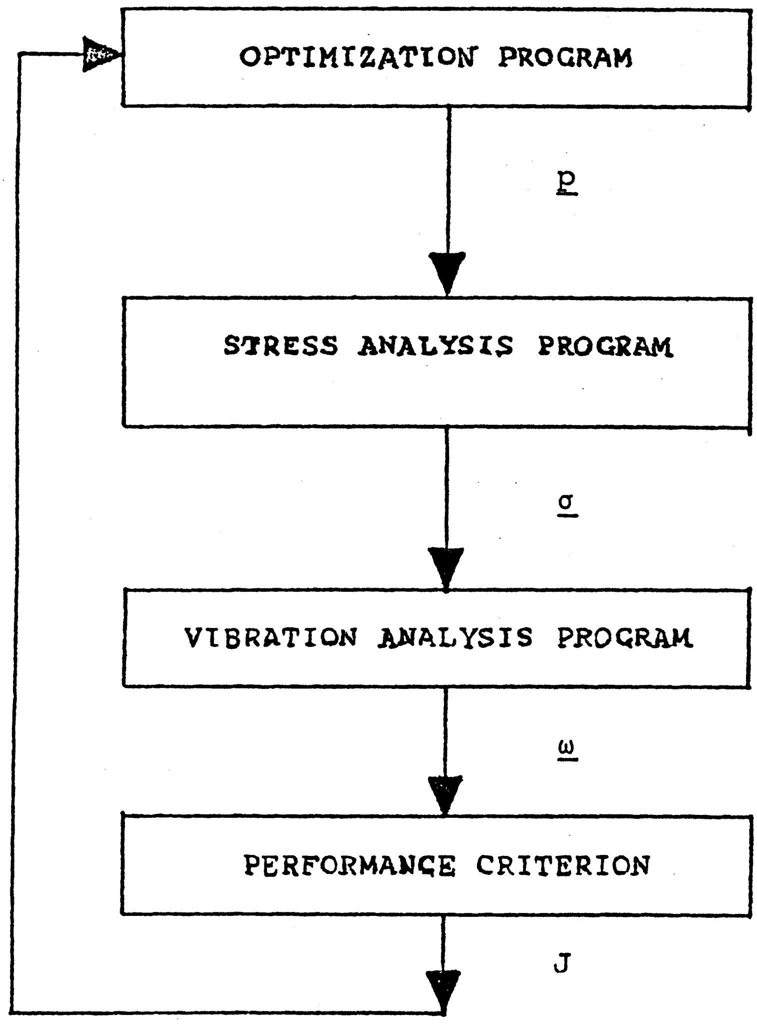
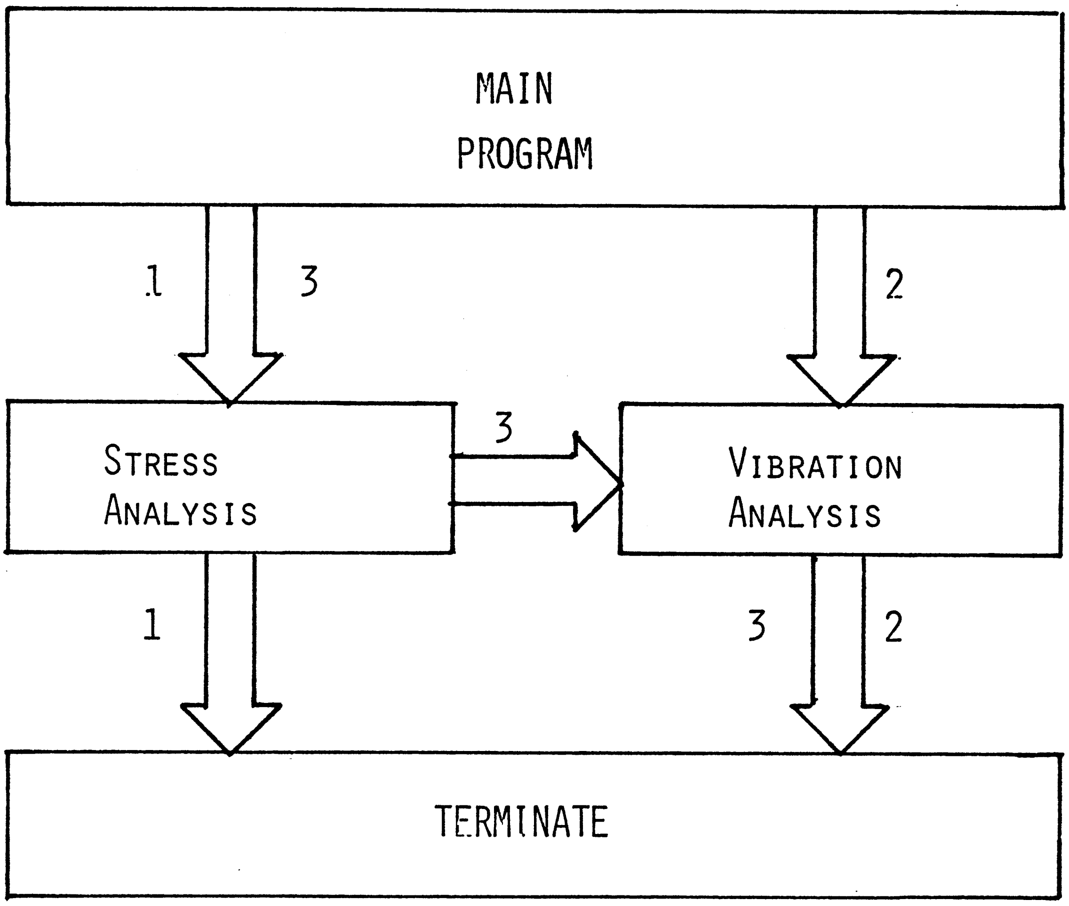


FIGURE 1 SCHEMATIC OF DESIGN APPROACH.



PATH 1: ANALYSIS OF IN-PLANE STRESSES ONLY
PATH 2: VIBRATION ANALYSIS ONLY
PATH 3: STRESS AND VIBRATION ANALYSIS TOGETHER

NOTE: THE ORDER OF THE APPROXIMATING FUNCTIONS FOR THE DISPLACEMENT FIELDS MUST BE 4 WHEN PATH 3 IS CHOSEN.

FIGURE 2 POSSIBLE PATHS THROUGH PROGRAM

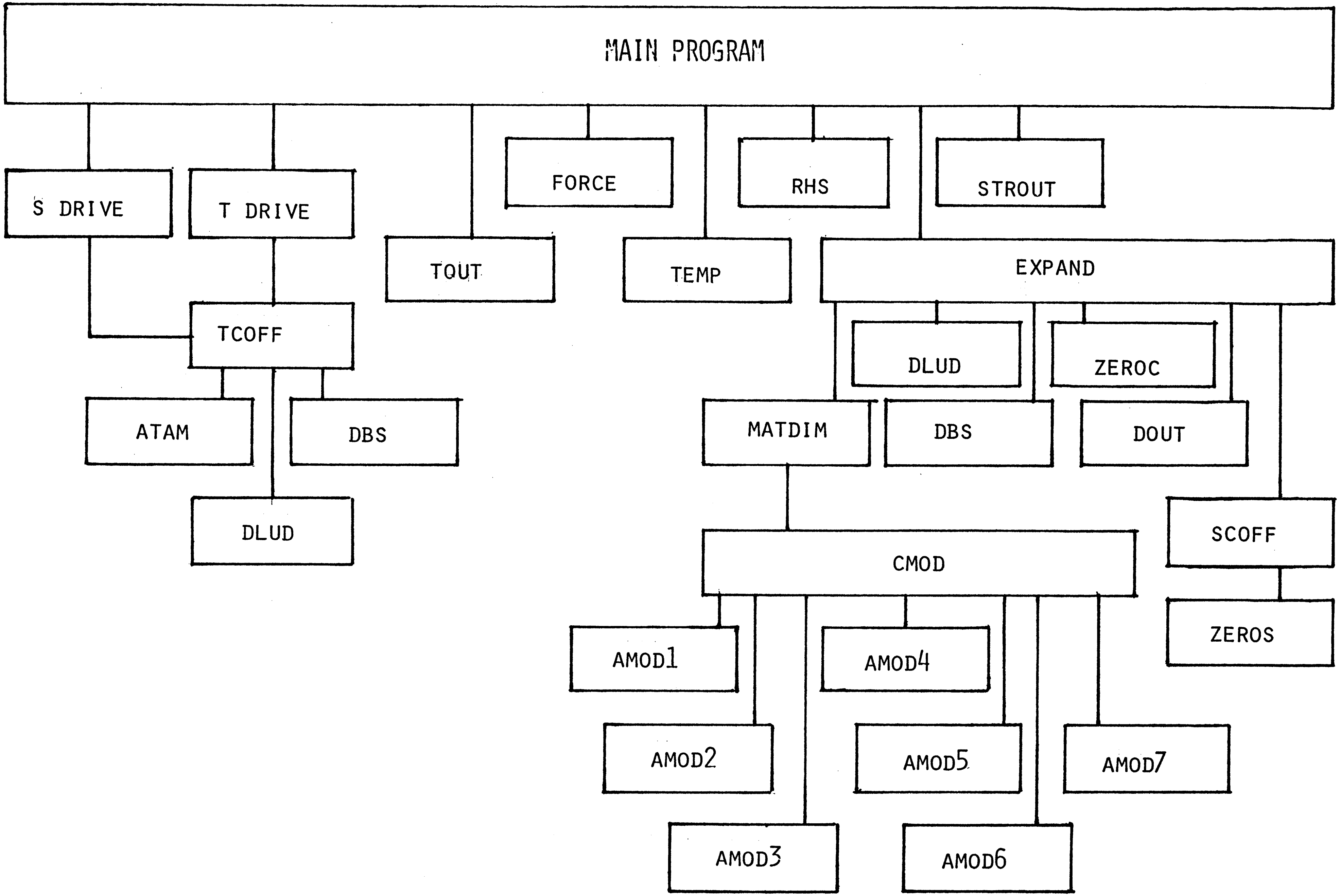


FIGURE 3 SUBROUTINES CALLED FOR PLANE STRESS ANALYSIS

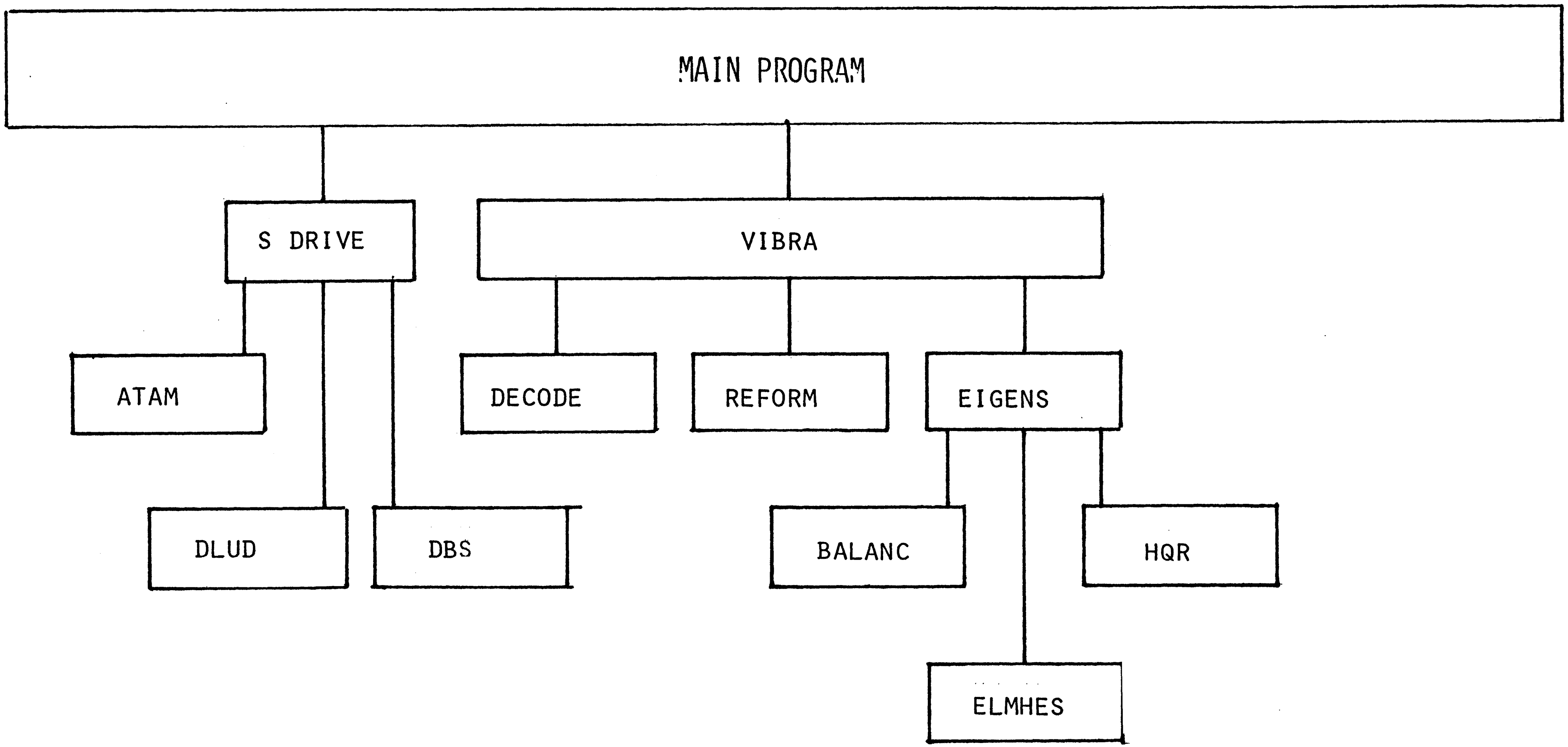


FIGURE 4 SUBROUTINES CALLED FOR VIBRATION ANALYSIS

by the user. The subroutines called when the vibration analysis, Path 2, alone is chosen by the user are shown in Figure 4. When the combined analysis, Path 3, is chosen by the user subroutines from both Figures 3 and 4 are used.

A brief description of what each subroutine does follows:

SUBROUTINE TDRIVE:	Reads temperature data for various points on the band saw blade. Calls subroutine TCOFF.
SUBROUTINE SDRIVE:	Reads (Pre) stress data for various points on the band saw blade. Calls subroutine TCOFF.
SUBROUTINE TCOFF:	Calculates temperature Field coefficients using the data read in subroutine TDRIVE by solving the linear regression problem. Temperature field coefficients are used in subroutine TEMP. Calls subroutines ATAM, DLUD, and DBS.
SUBROUTINE ATAM:	Calculates the matrix A-transposed times A for use in finding the temperature field coefficients.
SUBROUTINE TOUT:	Produces a table of the blade temperatures at various points using temperature field coefficients calculated in subroutine TCOFF.
SUBROUTINE FORCE:	Calculates the effect of the tensile, normal, and/or tangential loads for the righthand side vector. Also determines number of terms in approximating function for displacement field, (U).
SUBROUTINE TEMP:	Calculates effect of a temperature gradient if present for the righthand side vector.
SUBROUTINE RHS:	Adds the load and temperature effects together to form the righthand side vector.
SUBROUTINE EXPAND:	Calculates submatrices as given in report. Determines the displacement field coefficients for U and V. Prints out the displacement field coefficients A and B, if desired. Calls subroutines MATDIM, DLUD, DBS, DOUT, and SCOFF.
SUBROUTINE DLUD:	Decomposes a general matrix (A) into lower and upper triangular matrices.
SUBROUTINE DBS:	Solves a system of linear equations by forward elimination and back substitution.
SUBROUTINE MATDIM:	Calculates nondimensional integrals for use in subroutine expand to calculate submatrices K1,K2,K3, and K4. Calls subroutine CMOD.
SUBROUTINE CMOD:	Calulates the powers of x and y used to determine the elements of K1,K2,K3, and K4. Bridge subroutine that calls AMOD 1, AMOD2, AMOD3, AMOD4, AMOD5, AMOD6, AMOD7.

SUBROUTINE AMOD 1:	Calculates the nondimensional integrals that are multiplied by (EH)/(1-NU**2)*(B/L) in K1.
SUBROUTINE AMOD 2:	Calculates the nondimensional integrals that are multiplied by (EH)/(1+NU)*(L/B) in K1.
SUBROUTINE AMOD 3:	Calculates the nondimensional integrals that are multiplied by (EH)/(1-NU) in K2.
SUBROUTINE AMOD 4:	Calculates the nondimensional integrals that are multiplied by ((EH)NU)/(1-NU**2) in K2.
SUBROUTINE AMOD 5:	Calculates the nondimensional integrals that are multiplied by (EH)/(1+NU) in K2.
SUBROUTINE AMOD 6:	Calculates the nondimensional integrals that are multiplied by (EH/(1-NU**2)*(L/B) in K4.
SUBROUTINE AMOD7:	Calculates the nondimensional integrals that are multiplied by (EH)/(1+NU)*(B/L) in K4.
SUBROUTINE DOUT:	Produces displacement field tables for U and V at various points on the blade, using the displacement field coefficients determined in subroutine expand.
SUBROUTINE SCOFF:	Calculates the stress field coefficients for Sigma-X, Sigma-Y and Tau-XY, using the displacement field coefficients and Hooke's Law.
SUBROUTINE STROUT:	Produces tables of stress at various points on the blade. Tables made are Sigma-X, Sigma-Y, shear stress, and the maximum principal normal stress.
SUBROUTINE VIBRA:	Reads dimensionless integrals for vibration analysis from file. Calculates mass, gyro, and stiffness matrices.
SUBROUTINE DECODE:	Picks out the right dimensionless integral to be used in calculating the mass, gyro, and stiffness matrices.
SUBROUTINE REFORM:	Changes the system of second order differential equations to a system of first order differential equations.
SUBROUTINE EIGENS:	Calculates the natural frequencies of the band saw blade, and prints out the 5 lowest eigenvalues.

The temperature field and/or (pre) stress field coefficients are found by linear regression. The data points, blade coordinates x and y, and the temperature or (pre) stress values, are entered by the user or read from a data file. ATAM then calculates a NxN norm matrix, where N is the number of terms in the approximating function for the given temperature, or (pre) stress field. The norm matrix is formed by matrix multiplication of the transpose of a MxN matrix times the MxN matrix. M is the number of data points.

This method is inherently ill-conditioned so the method of singular-value decomposition and pseudo-inverse, (Numerical Methods, G. Dahlquist, and A. Bjorck, Prentice-Hall, Englewood Cliffs, N.J., 1974. Pp. 143-144), should be used to reduce the ill-conditioning problem.

Important remarks conerning the use of this program are:

1) When a combined stress and vibration analysis are chosen the order of the displacement field approximating functions must be 4. If a combined analysis is chosen initially the program automatically set the order of the displacement field approximating functions equal to 4. However, this is not the case if the combined analysis is not chosen initially. An error message is printed out, and the run is terminated if the above condition is not met.

2) The highest order of the displacement field approximating functions is 7. The program automatically sets the order to 6 if it is greater than 7, and sets the order to 2 if it is less than 2.

3) A subroutine is needed to combine the stress and prestress field coefficients. The subroutine for calculation of prestress coefficients is SDRIVE subroutine SDRIVE is not working properly.

4) Input/Output file numbers:

- 5 - Terminal input
- 6 - Terminal output
- 7 - Input file that contains the nondimensional integrals used for the vibration analysis
- 8 - Output file
- 10 - Input file that contains the blade coordinates, and stress values that are used with vibration analysis alone, or prestress values.
- 11 - Input file that contains the blade coordinates, and temperature data for use in calcualting the tempera-

ture field coefficients.

5) Maximum number of data points that can be entered in subroutines TDRIVE, or SDRIVE is 70.

Further information on subroutines DLUD, and DBS can be obtained from the University of Michigan Computing Center CCMEMO 426. Information concerning subroutines BALANC, ELMHES, and HQR can be obtained from the University of Michigan Computing Center by using the Archival utility program (NAASA: AUSP) to list documentation on the above subroutines. Also see the program listings of the above suroutines in a subsequent section for further references. The next section gives a sample of a terminal session using the present program. A listing of the program is provided in the last section.

SAMPLE TERMINAL SESSION

```
%LF10: LA36

  MTS Ann Arbor (LA36,LF10-GLAB-MP02-CD23,01142)

#   M.T.S. will be down from 11 30 pm Sat till 12 noon Sun (6-12-83)

#SIGNON K921
#Enter password.
?
#Terminal,Normal,Univ/Gov't
#Last signon was at 16:11:51, Fri Jun 10/83
#User K921 signed on at 16:48:57, Fri Jun 10/83

#Messages for mailbox K921: 2 new.  Use $MESSAGE to retrieve them.

#CON *SINK* UC=ON
#RUN *FTN SCARDS=BANDSAW SPUNCH=-B
#Execution begins
 No errors in MAIN
#Execution terminated
#RER SCARDS=STRESS.CODE SPUNCH=-S
#$Run *FTN SCARDS=STRESS.CODE SPUNCH=-S
#Execution begins
 No errors in EXPAND
 No errors in ZEROC
 No errors in MATDIM
 No errors in CMOD
 No errors in AMOD1
 No errors in AMOD2
 No errors in AMOD3
 No errors in AMOD4
 No errors in AMOD5
 No errors in AMOD6
 No errors in AMOD7
 No errors in SCOFF
 No errors in ZEROS
 No errors in DOUT
 No errors in STROUT
 No errors in TDRIVE
 No errors in TCOFF
 No errors in ATAM
 No errors in TOUT
 No errors in TEMP
 No errors in FORCE
 No errors in RHS
 No errors in SDRIVE
#Execution terminated
#RER SCARDS=VIBRATION.CO SPUNCH=-V
#$Run *FTN SCARDS=VIBRATION.CO SPUNCH=-V
#Execution begins
 No errors in VIBRA
 No errors in DECODE
 No errors in REFORM
 No errors in EIGENS
#Execution terminated
#
```

```
 RUN -B+-S+-V+NAAS:NAL+NAAS:EISPACK 7=TAPE 8=-OUT 11=IN1
#Execution begins



   BEGIN PROBLEM DESCRIPTION

      ENTER THE TITLE OF THIS RUN (50 CHARACTERS MAX)
 TEMPERATURE GRADIENT WITH A TENSILE LOAD

      ENTER NUMBER OF PROBLEM TO BE SOLVED
      1-PLATE STRESSES
      2-PLATE VIBRATION
      3-PLATE STRESSES AND VIBRATIONS
 1


   BEGIN DESCRIPTION OF APPROXIMATING FUNCTIONS FOR DISPLACEMENT FIELD COEFFICIENT CALCULATION

      ENTER DEGREE OF THE POLYNOMIAL FOR THE APPROXIMATING FUNCTIONS
      INTEGER 2 TO 7
 4


   BEGIN DESCRIPTION OF APPROXIMATING FUNCTION FOR TEMPERATURE FIELD COFFICIENT CALCULATION

      ENTER DEGREE OF THE POLYNOMIAL FOR THE APPROXIMATING FUNCTION
      IF ZERO NO TEMPERATURE FIELD IS PRESENT
      OTHERWISE ENTER INTEGER 1 TO 3
 3

         BEGIN ENTRY OF TEMPERATURE DATA

          IS THE DATA TO READ FROM A DATA FILE
          0-NO
          1-YES
 1

          ENTER THE NUMBER OF TEMPERATURE DATA POINTS (I2 FORMAT)
 66


   BEGIN DESCRIPTION OF GEOMETRY OF BANDSAW BLADE

      ENTER LENGTH OF BANDSAW BLADE
 39.37

      ENTER BANDSAW BLADE WIDTH
 10.827

      ENTER BANDSAW BLADE THICKNESS
 0.059055


   BEGIN DESCRIPTION OF MATERIAL PROPERTIES OF THE BANDSAW BLADE

      ENTER MODULUS OF ELASTICTY (YOUNG'S MODULUS)
 29010000.0

      ENTER POISSON'S RATIO
 0.3
```

```
    ENTER THE MASS DENSITY OF THE MATERIAL
0.283

    ENTER COEFFICIENT OF THERMAL EXPANSION
0.0000084

  BEGIN DESCRIPTION OF OPERATING CONDITIONS OF THE BANDSAW BLADE

    ENTER THE VELOCITY OF THE BANDSAW BLADE
0.0

    ENTER WHEEL SUPPORT COEFFICIENT
0.0

    ENTER THE INITIAL TENSION ON THE BANDSAW BLADE
11240.0

    ENTER NORMAL CUTTING FORCE ON THE BANDSAW BLADE
0.0

    ENTER TANGENTIAL CUTTING FORCE ON THE BANDSAW BLADE
0.0


  BEGIN OUTPUT CONTROL DESCRIPTION

    DO YOU WISH THE PRINTING OF THE DISPLACEMENT FIELD COEFFICIENTS (Y/N)?
Y

    DO YOU WISH THE PRINTING OF THE DISPLACEMENT FIELD TABLES (Y/N)?
Y

    DO YOU WISH THE PRINTING OF THE STRESS FIELD TABLES (Y/N)?
Y

    DO YOU WISH THE PRINTING OF THE TEMPERATURE FIELD COEFFICIENTS/TABLE (Y/N)?
Y

  ENTER NUMBER OF NEXT INSTRUCTION
    1- NEW PROBLEM TYPE
    2- NEW APPROXIMATING FUNCTION FOR DISPLACEMENTS
    3- NEW APPROXIMATING FUNCTION FOR TEMPERATURE
    4- NEW EIGENFUNCTIONS FOR VIBRATION ANALYSIS
    5- NEW BLADE GEOMETRY
    6- NEW MATERIAL PROPERTIES
    7- NEW OPERATING CONDITIONS
    8- SOLVE NEW PROBLEM
    9- STOP
1

    ENTER NUMBER OF PROBLEM TO BE SOLVED
    1-PLATE STRESSES
    2-PLATE VIBRATION
    3-PLATE STRESSES AND VIBRATIONS
2

  BEGIN DESCRIPTION OF EIGENFUNCTIONS FOR BANDSAW BLADE VIBRATIONS

    ENTER NUMBER OF SIMPLY SUPPORTED BEAM EIGENFUNCTIONS
    INTEGER 1 TO 6
4

    ENTER NUMBER OF FREE-FREE BEAM EIGENFUNCTIONS
    INTEGER 1 TO 6
2
```

```
  ENTER NUMBER OF NEXT INSTRUCTION
     1- NEW PROBLEM TYPE
     2- NEW APPROXIMATING FUNCTION FOR DISPLACEMENTS
     3- NEW APPROXIMATING FUNCTION FOR TEMPERATURE
     4- NEW EIGENFUNCTIONS FOR VIBRATION ANALYSIS
     5- NEW BLADE GEOMETRY
     6- NEW MATERIAL PROPERTIES
     7- NEW OPERATING CONDITIONS
     8- SOLVE NEW PROBLEM
     9- STOP
8

     ENTER THE TITLE OF THIS RUN (50 CHARACTERS MAX)
PLATE VIBRATION WITH TEMPERATURE GRADIENT


  DO YOU WISH TO USE THE STRESS FIELD COEFFICIENTS CALCULATED PREVIOUSLY (Y/N)?
Y

  ENTER NUMBER OF NEXT INSTRUCTION
     1- NEW PROBLEM TYPE
     2- NEW APPROXIMATING FUNCTION FOR DISPLACEMENTS
     3- NEW APPROXIMATING FUNCTION FOR TEMPERATURE
     4- NEW EIGENFUNCTIONS FOR VIBRATION ANALYSIS
     5- NEW BLADE GEOMETRY
     6- NEW MATERIAL PROPERTIES
     7- NEW OPERATING CONDITIONS
     8- SOLVE NEW PROBLEM
     9- STOP
1

     ENTER NUMBER OF PROBLEM TO BE SOLVED
     1-PLATE STRESSES
     2-PLATE VIBRATION
     3-PLATE STRESSES AND VIBRATIONS
1

  BEGIN DESCRIPTION OF APPROXIMATING FUNCTIONS FOR DISPLACEMENT FIELD COEFFICIENT CALCULATION

     ENTER DEGREE OF THE POLYNOMIAL FOR THE APPROXIMATING FUNCTIONS
     INTEGER 2 TO 7

6

  ENTER NUMBER OF NEXT INSTRUCTION
     1- NEW PROBLEM TYPE
     2- NEW APPROXIMATING FUNCTION FOR DISPLACEMENTS
     3- NEW APPROXIMATING FUNCTION FOR TEMPERATURE
     4- NEW EIGENFUNCTIONS FOR VIBRATION ANALYSIS
     5- NEW BLADE GEOMETRY
     6- NEW MATERIAL PROPERTIES
     7- NEW OPERATING CONDITIONS
     8- SOLVE NEW PROBLEM
     9- STOP
3

  BEGIN DESCRIPTION OF APPROXIMATING FUNCTION FOR TEMPERATURE FIELD COEFFICIENT CALCULATION

     ENTER DEGREE OF THE POLYNOMIAL FOR THE APPROXIMATING FUNCTION
     IF ZERO NO TEMPERATURE FIELD IS PRESENT
     OTHERWISE ENTER INTEGER 1 TO 5
0
```

```
   ENTER NUMBER OF NEXT INSTRUCTION
       1- NEW PROBLEM TYPE
       2- NEW APPROXIMATING FUNCTION FOR DISPLACEMENTS
       3- NEW APPROXIMATING FUNCTION FOR TEMPERATURE
       4- NEW EIGENFUNCTIONS FOR VIBRATION ANALYSIS
       5- NEW BLADE GEOMETRY
       6- NEW MATERIAL PROPERTIES
       7- NEW OPERATING CONDITIONS
       8- SOLVE NEW PROBLEM
       9- STOP
 5

   BEGIN DESCRIPTION OF GEOMETRY OF BANDSAW BLADE


     ENTER LENGTH OF BANDSAW BLADE
1.0

     ENTER BANDSAW BLADE WIDTH
0.25

     ENTER BANDSAW BLADE THICKNESS
0.00107


  ENTER NUMBER OF NEXT INSTRUCTION
     1- NEW PROBLEM TYPE
     2- NEW APPROXIMATING FUNCTION FOR DISPLACEMENTS
     3- NEW APPROXIMATING FUNCTION FOR TEMPERATURE
     4- NEW EIGENFUNCTIONS FOR VIBRATION ANALYSIS
     5- NEW BLADE GEOMETRY
     6- NEW MATERIAL PROPERTIES
     7- NEW OPERATING CONDITIONS
     8- SOLVE NEW PROBLEM
     9- STOP
6

  BEGIN DESCRIPTION OF MATERIAL PROPERTIES OF THE BANDSAW BLADE

     ENTER MODULUS OF ELASTICTY (YOUNG'S MODULUS)
207000000000.0

     ENTER POISSON'S RATIO
0.3

     ENTER THE MASS DENSITY OF THE MATERIAL
0.283


  ENTER NUMBER OF NEXT INSTRUCTION
     1- NEW PROBLEM TYPE
     2- NEW APPROXIMATING FUNCTION FOR DISPLACEMENTS
     3- NEW APPROXIMATING FUNCTION FOR TEMPERATURE
     4- NEW EIGENFUNCTIONS FOR VIBRATION ANALYSIS
     5- NEW BLADE GEOMETRY
     6- NEW MATERIAL PROPERTIES
     7- NEW OPERATING CONDITIONS
     8- SOLVE NEW PROBLEM
     9- STOP
```

```
7

  BEGIN DESCRIPTION OF OPERATING CONDITIONS OF THE BANDSAW BLADE

     ENTER THE VELOCITY OF THE BANDSAW BLADE
48.74

     ENTER WHEEL SUPPORT COEFFICIENT
0.2

     ENTER THE INITIAL TENSION ON THE BANDSAW BLADE
50000.0

     ENTER NORMAL CUTTING FORCE ON THE BANDSAW BLADE
250.0

        ENTER STARTING X-COORDINATE OF NORMAL CUTTING FORCE
0.487
        ENTER ENDING X-COORDINATE OF NORMAL CUTTING FORCE
0.513

     ENTER TANGENTIAL CUTTING FORCE ON THE BANDSAW BLADE
500.0

        ENTER STARTING X-COORDINATE OF TANGENTIAL CUTTING FORCE
0.487
        ENTER ENDING X-COORDINATE OF TANGENTIAL CUTTING FORCE

0.513

     ENTER TANGENTIAL CUTTING FORCE ON THE BANDSAW BLADE
500.0

        ENTER STARTING X-COORDINATE OF TANGENTIAL CUTTING FORCE
0.487
        ENTER ENDING X-COORDINATE OF TANGENTIAL CUTTING FORCE
0.513


  ENTER NUMBER OF NEXT INSTRUCTION
     1- NEW PROBLEM TYPE
     2- NEW APPROXIMATING FUNCTION FOR DISPLACEMENTS
     3- NEW APPROXIMATING FUNCTION FOR TEMPERATURE
     4- NEW EIGENFUNCTIONS FOR VIBRATION ANALYSIS
     5- NEW BLADE GEOMETRY
     6- NEW MATERIAL PROPERTIES
     7- NEW OPERATING CONDITIONS
     8- SOLVE NEW PROBLEM
     9- STOP
8

     ENTER THE TITLE OF THIS RUN (50 CHARACTERS MAX)
TENSILE, NORMAL, TANGENTIAL LOAD DEGREE EQUAL 6


  ENTER NUMBER OF NEXT INSTRUCTION
     1- NEW PROBLEM TYPE
     2- NEW APPROXIMATING FUNCTION FOR DISPLACEMENTS
     3- NEW APPROXIMATING FUNCTION FOR TEMPERATURE
     4- NEW EIGENFUNCTIONS FOR VIBRATION ANALYSIS
     5- NEW BLADE GEOMETRY
     6- NEW MATERIAL PROPERTIES
     7- NEW OPERATING CONDITIONS
     8- SOLVE NEW PROBLEM
     9- STOP
1

     ENTER NUMBER OF PROBLEM TO BE SOLVED
     1-PLATE STRESSES
     2-PLATE VIBRATION
     3-PLATE STRESSES AND VIBRATIONS
2

  BEGIN DESCRIPTION OF EIGENFUNCTIONS FOR BANDSAW BLADE VIBRATIONS
```

```
      ENTER NUMBER OF SIMPLY SUPPORTED BEAM EIGENFUNCTIONS
      INTEGER 1 TO 6
 5

      ENTER NUMBER OF FREE-FREE BEAM EIGENFUNCTIONS
      INTEGER 1 TO 6
 2


   **** ERROR THE STRESS FIELD POLYNOMIALS DO NOT MATCH THOSE REQUIRED FOR THE VIBRATION ANALYSIS! ***
#Execution terminated
#RUN *PAGEPR SCARDS=-OU PAR=ONESIDED
.Version Jun 02/83.  $Copy *PAGENEWS(1,30) for description of changes.
#Execution begins
 *PRINT* assigned receipt number 667585
 *PRINT* 667585 held

 This *PAGEPR run generated 634 lines, 28 pages, 28 images, and 28 sheets.

 *PRINT* 667585 released, 30 pages, route=CNTR, printer=PAGE.
#Execution terminated
#SIG
#Receipt Summary:
#*PRINT* 667499  61 pages, route=CNTR, printer=PAGE.
#*PRINT* 667585  30 pages, route=CNTR, printer=PAGE.
#
#K921 16:48:57 to 17:24:39, Fri Jun 10/83
#Terminal,Normal,Univ/Gov't
#Elapsed time          35.684 minutes       $.81
#CPU time used          5.553 seconds      $2.36
#CPU storage VMI        7.483 page-min.     $.95
#Wait storage VMI      67.475 page-hr.
#Page printer lines      2357
#Page printer pages        91
#Page printer images       91              $1.59
#Page printer sheets       91               $.55
#Page-ins                 577
#Disk I/O                1639
#     Approximate cost of this run:       $6.26

#Disk storage charge     101 page-hr.      $.01
#     Approximate remaining balance:    $43.43
```

OUTPUT FROM SAMPLE TERMINAL SESSION

TEMPERATURE GRADIENT WITH A TENSILE LOAD

THE DEGREE OF THE POLYNOMIAL FOR THE DISPLACEMENT FIELD CALCULATION IS 4

THE DEGREE OF THE POLYNOMIAL FOR THE TEMPERATURE FIELD CALCULATION IS 3

THE LENGTH OF THE BANDSAW BLADE IS   0.393700E+02

THE WIDTH OF THE BANDSAW BLADE IS   0.108270E+02

THE THICKNESS OF THE BANDSAW BLADE IS   0.590550E-01

THE MODULUS OF ELASTICITY IS   0.290100E+08

POISON"S RATIO IS   0.300000E+00

THE MASS DENSITY IS   0.283000E+00

THE COEFFICIENT OF THERMAL EXPANSION IS   0.840000E-05

THE VELOCITY OF THE BANDSAW BLADE IS   0.0

THE WHEEL SUPPORT COEFFICIENT IS   0.0

THE INITIAL TENSION IS   0.112400E+05

THE NORMAL CUTTING FORCE IS   0.0

THE TANGENTIAL CUTTING FORCE IS   0.0

TEMPERATURE GRADIENT WITH A TENSILE LOAD

TEMPERATURE FIELD COEFFICIENTS

```
   T(0,0)= 0.13925E+03
   T(1,0)=-0.31567E+01
   T(2,0)= 0.19117E+00
   T(3,0)=-0.30164E-02
   T(1,1)= 0.52621E+00
   T(2,1)=-0.14790E-01
   T(1,2)= 0.84362E-02
   T(0,1)=-0.51549E+01
   T(0,2)= 0.17958E+00
   T(0,3)=-0.33431E-03
```

TEMPERATURE GRADIENT WITH A TENSILE LOAD

TEMPERATURE DISTRIBUTION

Y COORDINATE											
0.108E+02	0.104E+03	0.118E+03	0.132E+03	0.145E+03	0.156E+03	0.163E+03	0.165E+03	0.162E+03	0.152E+03	0.134E+03	0.107E+03
0.866E+01	0.108E+03	0.117E+03	0.126E+03	0.136E+03	0.144E+03	0.149E+03	0.151E+03	0.149E+03	0.140E+03	0.125E+03	0.102E+03
0.650E+01	0.113E+03	0.117E+03	0.123E+03	0.129E+03	0.135E+03	0.139E+03	0.141E+03	0.139E+03	0.133E+03	0.120E+03	0.101E+03
0.433E+01	0.120E+03	0.119E+03	0.121E+03	0.125E+03	0.129E+03	0.132E+03	0.134E+03	0.134E+03	0.129E+03	0.120E+03	0.105E+03
0.217E+01	0.129E+03	0.123E+03	0.122E+03	0.123E+03	0.125E+03	0.129E+03	0.131E+03	0.132E+03	0.130E+03	0.124E+03	0.114E+03
0.0	0.139E+03	0.130E+03	0.125E+03	0.124E+03	0.125E+03	0.128E+03	0.132E+03	0.134E+03	0.135E+03	0.133E+03	0.127E+03
X-COORDINATE	0.0	0.394E+01	0.787E+01	0.118E+02	0.157E+02	0.197E+02	0.236E+02	0.276E+02	0.315E+02	0.354E+02	0.394E+02

TEMPERATURE GRADIENT WITH A TENSILE LOAD

COEFFICIENTS OF APPROXIMATING FUNCTION FOR DISPLACEMENT FIELD U

A(1, 0)=	0.54463E-03
A(2, 0)=	-0.16677E-04
A(3, 0)=	0.59267E-06
A(4, 0)=	-0.62723E-08
A(1, 1)=	-0.10565E-04
A(2, 1)=	0.36739E-05
A(3, 1)=	-0.67150E-07
A(1, 2)=	-0.11390E-05
A(2, 2)=	0.42909E-07
A(1, 3)=	-0.41193E-07
A(0, 1)=	-0.66577E-03
A(0, 2)=	0.31867E-05
A(0, 3)=	0.62317E-06
A(0, 4)=	0.76584E-08

TEMPERATURE GRADIENT WITH A TENSILE LOAD

COEFFICIENTS OF APPROXIMATING FUNCTION FOR DISPLACEMENT FIELD V

B(1, 0)=	0.67075E-03
B(2, 0)=	0.56762E-05
B(3, 0)=	-0.12378E-05
B(4, 0)=	0.16788E-07
B(1, 1)=	-0.59324E-05
B(2, 1)=	0.10317E-05
B(3, 1)=	-0.26166E-07
B(1, 2)=	-0.20579E-05
B(2, 2)=	0.66575E-07
B(1, 3)=	-0.30633E-07
B(0, 1)=	-0.31032E-04
B(0, 2)=	-0.71422E-04
B(0, 3)=	0.50149E-05
B(0, 4)=	-0.11227E-06

TEMPERATURE GRADIENT WITH A TENSILE LOAD

DISPLACEMENT FIELD U

Y COORDINATE											
0.108E+02	-.594E-02	-.455E-02	-.236E-02	0.529E-03	0.398E-02	0.781E-02	0.118E-01	0.157E-01	0.193E-01	0.222E-01	0.240E-01
0.866E+01	-.508E-02	-.345E-02	-.128E-02	0.140E-02	0.450E-02	0.789E-02	0.114E-01	0.149E-01	0.180E-01	0.207E-01	0.224E-01
0.650E+01	-.401E-02	-.222E-02	-.115E-03	0.231E-02	0.501E-02	0.793E-02	0.110E-01	0.140E-01	0.168E-01	0.191E-01	0.208E-01
0.433E+01	-.277E-02	-.886E-03	0.109E-02	0.321E-02	0.550E-02	0.794E-02	0.105E-01	0.130E-01	0.155E-01	0.176E-01	0.193E-01
0.217E+01	-.142E-02	0.505E-03	0.231E-02	0.410E-02	0.595E-02	0.791E-02	0.996E-02	0.121E-01	0.142E-01	0.162E-01	0.179E-01
0.0	0.0	0.192E-02	0.352E-02	0.496E-02	0.637E-02	0.784E-02	0.942E-02	0.111E-01	0.130E-01	0.148E-01	0.167E-01
X-COORDINATE	0.0	0.394E+01	0.787E+01	0.118E+02	0.157E+02	0.197E+02	0.236E+02	0.276E+02	0.315E+02	0.354E+02	0.394E+02

TEMPERATURE GRADIENT WITH A TENSILE LOAD

DISPLACEMENT FIELD V

Y COORDINATE											
0.108E+02	-.389E-02	-.231E-02	-.466E-03	0.123E-02	0.246E-02	0.301E-02	0.276E-02	0.168E-02	-.159E-03	-.258E-02	-.532E-02
0.866E+01	-.300E-02	-.103E-02	0.107E-02	0.291E-02	0.418E-02	0.471E-02	0.438E-02	0.319E-02	0.123E-02	-.130E-02	-.412E-02
0.650E+01	-.204E-02	0.227E-03	0.251E-02	0.443E-02	0.572E-02	0.620E-02	0.579E-02	0.451E-02	0.247E-02	-.112E-03	-.294E-02
0.433E+01	-.111E-02	0.137E-02	0.376E-02	0.572E-02	0.699E-02	0.741E-02	0.694E-02	0.559E-02	0.352E-02	0.952E-03	-.179E-02
0.217E+01	-.354E-03	0.225E-02	0.469E-02	0.664E-02	0.787E-02	0.823E-02	0.770E-02	0.634E-02	0.429E-02	0.180E-02	-.762E-03
0.0	0.0	0.266E-02	0.509E-02	0.700E-02	0.817E-02	0.848E-02	0.792E-02	0.657E-02	0.460E-02	0.229E-02	0.139E-16
X-COORDINATE	0.0	0.394E+01	0.787E+01	0.118E+02	0.157E+02	0.197E+02	0.236E+02	0.276E+02	0.315E+02	0.354E+02	0.394E+02

TEMPERATURE GRADIENT WITH A TENSILE LOAD

```
COEFFICIENTS OF NORMAL STRESS IN X-DIRECTION
  AX(0,0)= 0.17066E+05
  AX(1,0)=-0.11200E+04
  AX(2,0)= 0.66548E+02
  AX(3,0)=-0.10501E+01
  AX(1,1)= 0.19488E+03
  AX(2,1)=-0.51487E+01
  AX(1,2)= 0.18569E+01
  AX(0,1)=-0.17029E+04
  AX(0,2)= 0.10757E+03
  AX(0,3)=-0.56083E+01
```

TEMPERATURE GRADIENT WITH A TENSILE LOAD

```
COEFFICIENTS OF NORMAL STRESS IN Y-DIRECTION
  AX(0,0)= 0.42194E+04
  AX(1,0)=-0.50810E+03
  AX(2,0)= 0.49895E+02
  AX(3,0)=-0.10741E+01
  AX(1,1)=-0.60939E+02
  AX(2,1)= 0.23181E+01
  AX(1,2)=-0.21090E+01
  AX(0,1)=-0.46548E+04
  AX(0,2)= 0.46872E+03
  AX(0,3)=-0.14711E+02
```

TEMPERATURE GRADIENT WITH A TENSILE LOAD

```
COEFFICIENTS OF SHEAR STRESS
  AX(0,0)= 0.55516E+02
  AX(1,0)= 0.87845E+01
  AX(2,0)=-0.44214E+00
  AX(3,0)= 0.0
  AX(1,1)=-0.23936E+01
  AX(2,1)= 0.81674E-01
  AX(1,2)= 0.10680E+00
  AX(0,1)= 0.49201E+01
  AX(0,2)=-0.21024E+01
  AX(0,3)= 0.0
```

TEMPERATURE GRADIENT WITH A TENSILE LOAD

SIGMA X

Y COORDINATE											
0.108E+02	0.412E+04	0.898E+04	0.138E+05	0.182E+05	0.217E+05	0.241E+05	0.248E+05	0.236E+05	0.201E+05	0.138E+05	0.433E+04
0.866E+01	0.674E+04	0.980E+04	0.132E+05	0.164E+05	0.192E+05	0.212E+05	0.219E+05	0.209E+05	0.180E+05	0.127E+05	0.453E+04
0.650E+01	0.901E+04	0.103E+05	0.123E+05	0.145E+05	0.166E+05	0.182E+05	0.189E+05	0.183E+05	0.161E+05	0.118E+05	0.507E+04
0.433E+01	0.113E+05	0.109E+05	0.116E+05	0.128E+05	0.143E+05	0.156E+05	0.164E+05	0.162E+05	0.147E+05	0.115E+05	0.627E+04
0.217E+01	0.138E+05	0.119E+05	0.113E+05	0.117E+05	0.126E+05	0.137E+05	0.146E+05	0.149E+05	0.143E+05	0.122E+05	0.848E+04
0.0	0.171E+05	0.136E+05	0.119E+05	0.114E+05	0.118E+05	0.128E+05	0.139E+05	0.148E+05	0.150E+05	0.142E+05	0.120E+05
X-COORDINATE	0.0	0.394E+01	0.787E+01	0.118E+02	0.157E+02	0.197E+02	0.236E+02	0.276E+02	0.315E+02	0.354E+02	0.394E+02

TEMPERATURE GRADIENT WITH A TENSILE LOAD

SIGMA Y

Y COORDINATE											
0.108E+02	-.990E+04	-.144E+05	-.169E+05	-.179E+05	-.178E+05	-.169E+05	-.156E+05	-.144E+05	-.136E+05	-.137E+05	-.149E+05
0.866E+01	-.105E+05	-.142E+05	-.161E+05	-.166E+05	-.161E+05	-.151E+05	-.138E+05	-.127E+05	-.122E+05	-.127E+05	-.146E+05
0.650E+01	-.103E+05	-.132E+05	-.146E+05	-.147E+05	-.140E+05	-.128E+05	-.116E+05	-.108E+05	-.107E+05	-.117E+05	-.142E+05
0.433E+01	-.834E+04	-.107E+05	-.115E+05	-.113E+05	-.105E+05	-.929E+04	-.823E+04	-.767E+04	-.801E+04	-.963E+04	-.129E+05
0.217E+01	-.381E+04	-.558E+04	-.605E+04	-.560E+04	-.462E+04	-.352E+04	-.268E+04	-.250E+04	-.337E+04	-.568E+04	-.983E+04
0.0	0.422E+04	0.293E+04	0.279E+04	0.341E+04	0.440E+04	0.536E+04	0.590E+04	0.563E+04	0.415E+04	0.108E+04	-.399E+04
X-COORDINATE	0.0	0.394E+01	0.787E+01	0.118E+02	0.157E+02	0.197E+02	0.236E+02	0.276E+02	0.315E+02	0.354E+02	0.394E+02

TEMPERATURE GRADIENT WITH A TENSILE LOAD

SHEAR STRESS

Y COORDINATE											
0.108E+02	-.138E+03	-.149E+03	-.147E+03	-.130E+03	-.101E+03	-.571E+02	0.121E+00	0.711E+02	0.156E+03	0.254E+03	0.366E+03
0.866E+01	-.596E+02	-.710E+02	-.741E+02	-.691E+02	-.558E+02	-.343E+02	-.453E+01	0.334E+02	0.796E+02	0.134E+03	0.197E+03
0.650E+01	-.125E+01	-.876E+01	-.135E+02	-.156E+02	-.149E+02	-.114E+02	-.523E+01	0.370E+01	0.154E+02	0.298E+02	0.469E+02
0.433E+01	0.374E+02	0.377E+02	0.352E+02	0.300E+02	0.221E+02	0.114E+02	-.200E+01	-.182E+02	-.371E+02	-.587E+02	-.831E+02
0.217E+01	0.563E+02	0.684E+02	0.722E+02	0.678E+02	0.551E+02	0.343E+02	0.519E+01	-.321E+02	-.776E+02	-.131E+03	-.193E+03
0.0	0.555E+02	0.832E+02	0.973E+02	0.976E+02	0.842E+02	0.571E+02	0.163E+02	-.382E+02	-.106E+03	-.188E+03	-.284E+03
X-COORDINATE	0.0	0.394E+01	0.787E+01	0.118E+02	0.157E+02	0.197E+02	0.236E+02	0.276E+02	0.315E+02	0.354E+02	0.394E+02

TEMPERATURE GRADIENT WITH A TENSILE LOAD

MAXIMUM PRINCIPAL NORMAL STRESS

Y COORDINATE											
0.108E+02	0.412E+04	0.898E+04	0.138E+05	0.182E+05	0.217E+05	0.241E+05	0.248E+05	0.236E+05	0.201E+05	0.138E+05	0.434E+04
0.866E+01	0.674E+04	0.980E+04	0.132E+05	0.164E+05	0.192E+05	0.212E+05	0.219E+05	0.209E+05	0.180E+05	0.127E+05	0.454E+04
0.650E+01	0.901E+04	0.103E+05	0.123E+05	0.145E+05	0.166E+05	0.182E+05	0.189E+05	0.183E+05	0.161E+05	0.118E+05	0.507E+04
0.433E+01	0.113E+05	0.109E+05	0.116E+05	0.128E+05	0.143E+05	0.156E+05	0.164E+05	0.162E+05	0.147E+05	0.115E+05	0.627E+04
0.217E+01	0.138E+05	0.119E+05	0.113E+05	0.117E+05	0.126E+05	0.137E+05	0.146E+05	0.149E+05	0.143E+05	0.122E+05	0.848E+04
0.0	0.171E+05	0.136E+05	0.119E+05	0.114E+05	0.118E+05	0.128E+05	0.139E+05	0.148E+05	0.150E+05	0.142E+05	0.120E+05
X-COORDINATE	0.0	0.394E+01	0.787E+01	0.118E+02	0.157E+02	0.197E+02	0.236E+02	0.276E+02	0.315E+02	0.354E+02	0.394E+02

```
PLATE VIBRATION WITH TEMPERATURE GRADIENT

THE NUMBER OF SIMPLY SUPPORTED BEAM EIGENFUNCTIONS IS 4

THE NUMBER OF FREE-FREE BEAM EIGENFUNCTIONS IS 2

THE LENGTH OF THE BANDSAW BLADE IS   0.393700E+02

THE WIDTH OF THE BANDSAW BLADE IS   0.108270E+02

THE THICKNESS OF THE BANDSAW BLADE IS   0.590550E-01

THE MODULUS OF ELASTICITY IS   0.290100E+08

POISON"S RATIO IS   0.300000E+00

THE MASS DENSITY IS   0.283000E+00

THE COEFFICIENT OF THERMAL EXPANSION IS   0 840000E-05

THE VELOCITY OF THE BANDSAW BLADE IS   0.0

THE WHEEL SUPPORT COEFFICIENT IS   0.0
```

PLATE VIBRATION WITH TEMPERATURE GRADIENT

EIGEN VALUE	REAL PART	IMAGINARY PART	NON-DIMENSIONAL NATURAL FREQUENCY
1	-0.531418E+02	0.0	0.455240E+03
3	-0.299176E+02	0.844834E+02	0.767769E+03
5	-0.299176E+02	0.233046E+02	0.324870E+03
7	-0.299176E+02	0.318921E+02	0.374600E+03
9	-0.299176E+02	0.714623E+02	0.663667E+03
11	-0.299176E+02	0.468079E+02	0.475889E+03
13	-0.299176E+02	0.489795E+02	0.491666E+03
15	-0.299176E+02	0.648342E+02	0.611685E+03

LOWEST 5 EIGENVALUES IN INCREASING ORDER 5 7 1 11 13

```
                    TENSILE, NORMAL, TANGENTIAL LOAD DEGREE EQUAL 6

THE DEGREE OF THE POLYNOMIAL FOR THE DISPLACEMENT FIELD CALCULATION IS 6

THE DEGREE OF THE POLYNOMIAL FOR THE TEMPERATURE FIELD CALCULATION IS 0

THE LENGTH OF THE BANDSAW BLADE IS   0.100000E+01

THE WIDTH OF THE BANDSAW BLADE IS   0.250000E+00

THE THICKNESS OF THE BANDSAW BLADE IS   0.107000E-02

THE MODULUS OF ELASTICITY IS   0.207000E+12

POISON"S RATIO IS   0.300000E+00

THE MASS DENSITY IS   0.283000E+00

THE VELOCITY OF THE BANDSAW BLADE IS   0.487400E+02

THE WHEEL SUPPORT COEFFICIENT IS   0.200000E+00

THE INITIAL TENSION IS   0.500000E+05

THE NORMAL CUTTING FORCE IS   0.250000E+03

THE TANGENTIAL CUTTING FORCE IS   0.500000E+03

THE STARTING X-COORDINATE FOR THE NORMAL CUTTING FORCE IS   0.487000E+00

THE ENDING X-COORDINATE FOR THE NORMAL CUTTING FORCE IS   0.513000E+00

THE STARTING X-COORDINATE FOR THE TANGENTIAL CUTTING FORCE IS   0.487000E+00

THE ENDING X-COORDINATE FOR THE TANGENTIAL CUTTING FORCE IS   0.513000E+00
```

TENSILE, NORMAL, TANGENTIAL LOAD DEGREE EQUAL 6

COEFFICIENTS OF APPROXIMATING FUNCTION FOR DISPLACEMENT FIELD U

```
A( 1, 0)=  0.83999E-03
A( 2, 0)=  0.16685E-03
A( 3, 0)= -0.25155E-03
A( 4, 0)=  0.35664E-03
A( 5, 0)= -0.35950E-03
A( 6, 0)=  0.14366E-03
A( 1, 1)=  0.10311E-02
A( 2, 1)= -0.25467E-02
A( 3, 1)=  0.17339E-02
A( 4, 1)=  0.42432E-03
A( 5, 1)= -0.55051E-03
A( 1, 2)= -0.58239E-02
A( 2, 2)=  0.13222E-01
A( 3, 2)= -0.10183E-01
A( 4, 2)=  0.20595E-02
A( 1, 3)=  0.13402E-01
A( 2, 3)= -0.24878E-01
A( 3, 3)=  0.13451E-01
A( 1, 4)= -0.91731E-02
A( 2, 4)=  0.66890E-02
A( 1, 5)=  0.32471E-02
A( 0, 1)= -0.99129E-04
A( 0, 2)=  0.64076E-03
A( 0, 3)= -0.17807E-02
A( 0, 4)=  0.24074E-02
A( 0, 5)= -0.17646E-02
A( 0, 6)=  0.13378E-03
```

TENSILE, NORMAL, TANGENTIAL LOAD DEGREE EQUAL 6

COEFFICIENTS OF APPROXIMATING FUNCTION FOR DISPLACEMENT FIELD V

```
   B( 1, 0)=   0.97751E-05
   B( 2, 0)=  -0.14137E-03
   B( 3, 0)=   0.12220E-03
   B( 4, 0)=   0.37973E-03
   B( 5, 0)=  -0.63100E-03
   B( 6, 0)=   0.26066E-03
   B( 1, 1)=  -0.18996E-04
   B( 2, 1)=   0.21425E-03
   B( 3, 1)=  -0.84791E-03
   B( 4, 1)=   0.11563E-02
   B( 5, 1)=  -0.50844E-03
   B( 1, 2)=   0.10928E-04
   B( 2, 2)=   0.74170E-03
   B( 3, 2)=  -0.10066E-02
   B( 4, 2)=   0.28240E-03
   B( 1, 3)=  -0.11924E-02
   B( 2, 3)=  -0.29921E-03
   B( 3, 3)=   0.12896E-02
   B( 1, 4)=   0.38049E-02
   B( 2, 4)=  -0.30997E-02
   B( 1, 5)=  -0.80268E-03
   B( 0, 1)=  -0.26631E-03
   B( 0, 2)=  -0.71350E-04
   B( 0, 3)=   0.44606E-03
   B( 0, 4)=  -0.10174E-02
   B( 0, 5)=   0.55038E-03
   B( 0, 6)=  -0.16236E-03
```

TENSILE, NORMAL, TANGENTIAL LOAD DEGREE EQUAL 6

DISPLACEMENT FIELD U

Y COORDINATE											
0.250E+00	-.484E-05	0.860E-04	0.176E-03	0.265E-03	0.353E-03	0.442E-03	0.531E-03	0.621E-03	0.712E-03	0.802E-03	0.893E-03
0.200E+00	-.514E-05	0.854E-04	0.175E-03	0.264E-03	0.353E-03	0.443E-03	0.532E-03	0.622E-03	0.713E-03	0.803E-03	0.893E-03
0.150E+00	-.538E-05	0.850E-04	0.175E-03	0.264E-03	0.353E-03	0.443E-03	0.533E-03	0.623E-03	0.714E-03	0.804E-03	0.894E-03
0.100E+00	-.506E-05	0.847E-04	0.174E-03	0.264E-03	0.354E-03	0.443E-03	0.534E-03	0.624E-03	0.714E-03	0.804E-03	0.895E-03
0.500E-01	-.356E-05	0.848E-04	0.174E-03	0.263E-03	0.353E-03	0.444E-03	0.534E-03	0.625E-03	0.715E-03	0.805E-03	0.896E-03
0.0	0.0	0.854E-04	0.173E-03	0.262E-03	0.353E-03	0.444E-03	0.535E-03	0.626E-03	0.716E-03	0.806E-03	0.896E-03
X-COORDINATE	0.0	0.100E+00	0.200E+00	0.300E+00	0.400E+00	0.500E+00	0.600E+00	0.700E+00	0.800E+00	0.900E+00	0.100E+01

TENSILE, NORMAL, TANGENTIAL LOAD DEGREE EQUAL 6

DISPLACEMENT FIELD V

Y COORDINATE											
0.250E+00	-.675E-04	-.681E-04	-.698E-04	-.721E-04	-.740E-04	-.748E-04	-.743E-04	-.727E-04	-.707E-04	-.690E-04	-.682E-04
0.200E+00	-.540E-04	-.545E-04	-.564E-04	-.587E-04	-.606E-04	-.613E-04	-.607E-04	-.590E-04	-.571E-04	-.555E-04	-.546E-04
0.150E+00	-.405E-04	-.409E-04	-.428E-04	-.452E-04	-.471E-04	-.478E-04	-.471E-04	-.455E-04	-.436E-04	-.420E-04	-.410E-04
0.100E+00	-.270E-04	-.273E-04	-.293E-04	-.317E-04	-.336E-04	-.342E-04	-.335E-04	-.319E-04	-.300E-04	-.286E-04	-.273E-04
0.500E-01	-.134E-04	-.137E-04	-.157E-04	-.182E-04	-.200E-04	-.207E-04	-.200E-04	-.183E-04	-.165E-04	-.151E-04	-.136E-04
0.0	0.0	-.282E-06	-.230E-05	-.476E-05	-.656E-05	-.709E-05	-.632E-05	-.472E-05	-.299E-05	-.156E-05	-.542E-19
X-COORDINATE	0.0	0.100E+00	0.200E+00	0.300E+00	0.400E+00	0.500E+00	0.600E+00	0.700E+00	0.800E+00	0.900E+00	0.100E+01

TENSILE, NORMAL, TANGENTIAL LOAD DEGREE EQUAL 6

```
COEFFICIENTS OF NORMAL STRESS IN X-DIRECTION
  AX(0,0)= 0.17290E+09
  AX(1,0)= 0.74610E+08
  AX(2,0)=-0.15704E+09
  AX(3,0)= 0.26664E+09
  AX(4,0)=-0.32997E+09
  AX(5,0)= 0.16137E+09
  AX(1,1)=-0.11571E+10
  AX(2,1)= 0.12845E+10
  AX(3,1)= 0.24870E+09
  AX(4,1)=-0.58759E+09
  AX(1,2)= 0.57713E+10
  AX(2,2)=-0.70102E+10
  AX(3,2)= 0.21379E+10
  AX(1,3)=-0.10280E+11
  AX(2,3)= 0.83331E+10
  AX(1,4)= 0.27692E+10
  AX(0,1)= 0.22482E+09
  AX(0,2)=-0.12335E+10
  AX(0,3)= 0.27709E+10
  AX(0,4)=-0.18988E+10
  AX(0,5)= 0.67216E+09
```

TENSILE, NORMAL, TANGENTIAL LOAD DEGREE EQUAL 6

```
COEFFICIENTS OF NORMAL STRESS IN Y-DIRECTION
  AX(0,0)=-0.32551E+07
  AX(1,0)= 0.18451E+08
  AX(2,0)=-0.27620E+07
  AX(3,0)=-0.95527E+08
  AX(4,0)= 0.14036E+09
  AX(5,0)=-0.56837E+08
  AX(1,1)=-0.34261E+09
  AX(2,1)= 0.69241E+09
  AX(3,1)=-0.34212E+09
  AX(4,1)=-0.59365E+08
  AX(1,2)= 0.99088E+09
  AX(2,2)=-0.22889E+10
  AX(3,2)= 0.14422E+10
  AX(1,3)= 0.66624E+08
  AX(2,3)=-0.66624E+08
  AX(1,4)= 0.0
  AX(0,1)= 0.37907E+08
  AX(0,2)=-0.93037E+08
  AX(0,3)=-0.11104E+08
  AX(0,4)= 0.0
  AX(0,5)= 0.0
```

TENSILE, NORMAL, TANGENTIAL LOAD DEGREE EQUAL 6

```
COEFFICIENTS OF SHEAR STRESS
  AX(0,0)=-0.71140E+07
  AX(1,0)= 0.59585E+08
  AX(2,0)=-0.17357E+09
  AX(3,0)= 0.25898E+09
  AX(4,0)=-0.21740E+09
  AX(5,0)= 0.80685E+08
  AX(1,1)=-0.89323E+09
  AX(2,1)= 0.19029E+10
  AX(3,1)=-0.12532E+10
  AX(4,1)= 0.12554E+09
  AX(1,2)= 0.33192E+10
  AX(2,2)=-0.61824E+10
  AX(3,2)= 0.33027E+10
  AX(1,3)=-0.29689E+10
  AX(2,3)= 0.24382E+10
  AX(1,4)= 0.79904E+09
  AX(0,1)= 0.10052E+09
  AX(0,2)=-0.42443E+09
  AX(0,3)= 0.67173E+09
  AX(0,4)=-0.39952E+09
  AX(0,5)= 0.0
```

TENSILE, NORMAL, TANGENTIAL LOAD DEGREE EQUAL 6

SIGMA X

Y COORDINATE											
0.250E+00	0.189E+09	0.187E+09	0.185E+09	0.183E+09	0.183E+09	0.184E+09	0.185E+09	0.187E+09	0.188E+09	0.188E+09	0.188E+09
0.200E+00	0.188E+09	0.187E+09	0.185E+09	0.184E+09	0.184E+09	0.185E+09	0.186E+09	0.187E+09	0.187E+09	0.187E+09	0.186E+09
0.150E+00	0.187E+09	0.186E+09	0.185E+09	0.185E+09	0.185E+09	0.186E+09	0.186E+09	0.187E+09	0.187E+09	0.187E+09	0.186E+09
0.100E+00	0.186E+09	0.186E+09	0.185E+09	0.185E+09	0.186E+09	0.186E+09	0.187E+09	0.187E+09	0.187E+09	0.187E+09	0.187E+09
0.500E-01	0.181E+09	0.184E+09	0.185E+09	0.186E+09	0.187E+09	0.187E+09	0.187E+09	0.187E+09	0.187E+09	0.187E+09	0.188E+09
0.0	0.173E+09	0.179E+09	0.183E+09	0.186E+09	0.188E+09	0.189E+09	0.189E+09	0.188E+09	0.186E+09	0.186E+09	0.189E+09
X-COORDINATE	0.0	0.100E+00	0.200E+00	0.300E+00	0.400E+00	0.500E+00	0.600E+00	0.700E+00	0.800E+00	0.900E+00	0.100E+01

ILE, NORMAL, TANGENTIAL LOAD DEGREE EQUAL 6

SIGMA Y

Y COORDINATE											
0.250E+00	0.233E+06	-.620E+04	-.108E+06	-.250E+06	-.448E+06	-.617E+06	-.649E+06	-.470E+06	-.119E+06	0.192E+06	0.526E+04
0.200E+00	0.516E+06	-.132E+06	-.382E+06	-.495E+06	-.558E+06	-.556E+06	-.440E+06	-.189E+06	0.113E+06	0.215E+06	-.372E+06
0.150E+00	0.300E+06	-.350E+06	-.532E+06	-.544E+06	-.507E+06	-.427E+06	-.271E+06	-.299E+05	0.211E+06	0.204E+06	-.527E+06
0.100E+00	-.406E+06	-.657E+06	-.558E+06	-.402E+06	-.297E+06	-.232E+06	-.146E+06	0.401E+04	0.173E+06	0.162E+06	-.452E+06
0.500E-01	-.159E+07	-.105E+07	-.458E+06	-.695E+05	0.664E+05	0.233E+05	-.683E+05	-.892E+05	0.446E+03	0.926E+05	-.136E+06
0.0	-.326E+07	-.152E+07	-.233E+06	0.451E+06	0.581E+06	0.335E+06	-.422E+05	-.312E+06	-.306E+06	0.464E+03	0.426E+06
X-COORDINATE	0.0	0.100E+00	0.200E+00	0.300E+00	0.400E+00	0.500E+00	0.600E+00	0.700E+00	0.800E+00	0.900E+00	0.100E+01

TENSILE, NORMAL, TANGENTIAL LOAD DEGREE EQUAL 6

SHEAR STRESS

Y COORDINATE											
0.250E+00	0.423E+06	0.142E+06	-.387E+06	-.798E+06	-.981E+06	-.981E+06	-.902E+06	-.809E+06	-.635E+06	-.804E+05	0.148E+07
0.200E+00	0.747E+06	-.216E+06	-.997E+06	-.132E+07	-.119E+07	-.776E+06	-.309E+06	-.142E+05	0.992E+04	-.121E+06	0.241E+05
0.150E+00	0.479E+06	-.519E+06	-.128E+07	-.152E+07	-.126E+07	-.668E+06	-.421E+05	0.351E+06	0.342E+06	-.358E+05	-.452E+06
0.100E+00	-.675E+06	-.898E+06	-.124E+07	-.131E+07	-.103E+07	-.506E+06	0.321E+05	0.361E+06	0.343E+06	0.250E+05	-.261E+06
0.500E-01	-.307E+07	-.153E+07	-.915E+06	-.624E+06	-.384E+06	-.137E+06	0.573E+05	0.116E+06	0.294E+05	-.390E+05	0.341E+06
0.0	-.711E+07	-.265E+07	-.390E+06	0.568E+06	0.784E+06	0.592E+06	0.189E+06	-.263E+06	-.545E+06	-.281E+06	0.116E+07
X-COORDINATE	0.0	0.100E+00	0.200E+00	0.300E+00	0.400E+00	0.500E+00	0.600E+00	0.700E+00	0.800E+00	0.900E+00	0.100E+01

TENSILE, NORMAL, TANGENTIAL LOAD DEGREE EQUAL 6

MAXIMUM PRINCIPAL NORMAL STRESS

Y COORDINATE											
0.250E+00	0.189E+09	0.187E+09	0.185E+09	0.183E+09	0.183E+09	0.184E+09	0.185E+09	0.187E+09	0.188E+09	0.188E+09	0.188E+09
0.200E+00	0.188E+09	0.187E+09	0.185E+09	0.184E+09	0.184E+09	0.185E+09	0.186E+09	0.187E+09	0.187E+09	0.187E+09	0.186E+09
0.150E+00	0.187E+09	0.186E+09	0.185E+09	0.185E+09	0.185E+09	0.186E+09	0.186E+09	0.187E+09	0.187E+09	0.187E+09	0.186E+09
0.100E+00	0.186E+09	0.186E+09	0.185E+09	0.185E+09	0.186E+09	0.186E+09	0.187E+09	0.187E+09	0.187E+09	0.187E+09	0.187E+09
0.500E-01	0.181E+09	0.184E+09	0.185E+09	0.186E+09	0.187E+09	0.187E+09	0.187E+09	0.187E+09	0.187E+09	0.187E+09	0.188E+09
0.0	0.173E+09	0.179E+09	0.183E+09	0.186E+09	0.188E+09	0.189E+09	0.189E+09	0.188E+09	0.186E+09	0.186E+09	0.189E+09
X-COORDINATE	0.0	0.100E+00	0.200E+00	0.300E+00	0.400E+00	0.500E+00	0.600E+00	0.700E+00	0.800E+00	0.900E+00	0.100E+01

PROGRAM LISTING

MAIN PROGRAM

```
C       'BANDSAW' IS THE MAIN PROGRAM TO CALCULATE THE STRESS FIELD
C       COEFFICIENTS, DISPLACEMENT FIELD COEFFICIENTS, TEMPERATURE
C       FIELD COEFFICIENTS, AND NATURAL FREQUENCIES OF A GIVEN
C       BANDSAW BLADE.  THE INPUTS INTO THE PROGRAM ARE THE GEOMETRY
C       OF THE BANDSAW BLADE, THE OPERATING CONDITIONS, TEMPERATURES
C       AT VARIOUS POINTS ON THE BANDSAW BLADE, AND THE MATERIAL
C       PROPERTIES OF THE BANDSAW BLADE.  THE OPERATING CONDITIONS
C       INCLUDES THE MILL SUPPORT COEFFICIENT, THE BLADE VELOCITY,
C       THE INITIAL TENSILE LOAD, THE NORMAL AND TANGENTIAL CUTTING
C       FORCES.
C           A DISCUSSION OF THE THEORY LEADING TO THE EQUATIONS
C       USED BY THE VARIOUS SUBROUTINES CALLED BY 'BANDSAW' SEE
C       "DEVELOPMENT OF AN EFFICIENT COMPUTIONAL PROCEDURE FOR
C       EVALUATING BAND SAW BLADE STRESS" BY J. E. BORCHELT,
C       A. G. ULSOY, P. PAPALAMBROS (1983). EXAMPLE RUNS OF THIS
C       PROGRAM ARE GIVEN IN "A COMPUTIONAL PACKAGE FOR DESIGN AND
C       PROCESS EVALUATION IN BANDSAWING- A USER'S MANUAL".
C           'BANDSAW' CALLS SUBOUTINES TDRIVE, TOUT, FORCE, TEMP,
C       EXPAND, STROUT, AND VIBRA.
C
C       SUBROUTINE TDRIVE: READS TEMPERATURE DATA FOR VARIOUS POINTS
C                          ON THE BAND SAW BLADE. CALLS SUBROUTINE TCOFF
C       SUBROUTINE TCOFF: CALCULATES TEMPERAURE  FIELD COEFFICIENTS
C                         USING THE DATA READ IN SUBROUTINE TDRIVE BY
C                         SOLVING THE LINEAR REGRESSTION PROBLEM.
C                         TEMPERATURE FIELD COEFFICIENTS ARE USED IN
C                         SUBROUTINE TEMP. CALLS SUBROUTINES ATAM,
C                         DLUD, AND DBS.
C       SUBROUTINE ATAM: CALCULATES THE MATRIX A-TRANSPOSED TIMES A
C                        FOR USE IN FINDING THE TEMPERATURE FIELD
C                        COEFFICIENTS.
C       SUBROUTINE TOUT: PRODUCES A TABLE OF THE BLADE TEMPERATURES
C                        AT VARIOUS POINTS USING TEMPERATURE FIELD
C                        COEFFICIENTS CALCULATED IN SUBROUTINE TCOFF.
C       SUBROUTINE FORCE: CALCULATES THE EFFECT OF THE TENSILE,NORMAL,
C                         AND/OR TANGENTIAL LOADS FOR THE RIGHTHAND
C                         SIDE VECTOR. ALSO DETERMINES NUMBER OF TERMS
C                         IN APPROXIMATION FUNCTION FOR DISPLACEMENT
C                         FIELD,(U).
C       SUBROUTINE TEMP: CALCULATES EFFECT OF A TEMPERATURE GRADIENT
C                        IF PRESENT FOR THE RIGHTHAND SIDE VECTOR.
C       SUBROUTINE RHS: ADDS THE LOAD AND TEMPERATURE EFFECTS
C                       TOGETHER TO FORM THE RIGHTHAND SIDE VECTOR
C                       ,WHICH IS STORED IN VECTOR BX.
C       SUBROUTINE EXPAND: CALCULATES SUBMATRICES AS GIVEN IN REPORT.
C                          DETERMINES THE DISPLACEMENT FIELD COEFFICIENTS
C                          FOR U AND V. PRINTS OUT THE DISPLACEMENT
C                          FIELD COEFFICIENTS A AND B, IF DESIRED.
C                          CALLS SUBROUTINES MATDIM, DLUD, DBS, DOUT,
C                          AND SCOFF.
C       SUBROUTINE DLUD: DECOMPOSES A GENERAL MATRIX(A) INTO LOWER AND UPPER
C                        TRIANGULAR MATRICES. THE DECOMPOSED MATRICES ARE
C                        STORED IN MATRIX(T).
C       SUBROUTINE DBS: SOLVES A SYSTEM OF LINEAR EQUATIONS BY FORWARD
C                       ELIMINATION AND BACK SUBSTISUTION.
C       SUBRUOTINE MATDIM: CALCULATES NONDIMENSIONAL INTEGRALS FOR USE IN
C                          SUBROUTINE EXPAND TO CALCULATE SUBMATRICES
C                          K1,K2,K3, AND K4. CALLS SUBROUTINE CMOD.
C       SUBROUTINE CMOD: CALCULATES THE POWERS OF X AND Y USED TO
C                        DETERMINE THE ELEMENTS OF K1,K2,K3, AND K4.
```

```
C                       BRIDGE SUBROUTINE THAT CALLS AMOD1, AMOD2,
C                       AMOD3,AMOD4,AMOD5,AMOD6,AMOD7.
C       SUBROUTINE AMOD1: CALCULATES THE NONDIMENSIONAL INTEGRALS THAT
C                         ARE MULTIPLIED BY (EH)/(1-NU**2)*(B/L) IN K1.
C       SUBROUTINE AMOD2: CALCULATES THE NONDIMENSIONAL INTEGRALS THAT
C                         ARE MULTIPLIED BY (EH)/(1+NU)*(L/B) IN K1.
C       SUBROUTINE AMOD3: CALCULATES THE NONDIMENSIONAL INTEGRALS THAT
C                         ARE MULTIPLIED BY (EH)/(1-NU) IN K2.
C       SUBROUTINE AMOD4: CALCULATES THE NONDIMENSIONAL INTEGRALS THAT
C                         ARE MULTIPLIED BY ((EH)NU)/(1-NU**2) IN K2.
C       SUBROUTINE AMOD5: CALCULATES THE NONDIMENSIONAL INTEGRALS THAT
C                         ARE MULTIPLIED BY (EH)/(1+NU) IN K2.
C       SUBROUTINE AMOD6: CALCULATES THE NONDIMENSIONAL INTEGRALS THAT
C                         ARE MULTIPLIED BY (EH)/(1-NU**2)*(L/B) IN K4.
C       SUBROUTINE AMOD7: CALCULATES THE NONDIMENSIONAL INTEGRALS THAT
C                         ARE MULTIPLIED BY (EH)/(1+NU)*(B/L) IN K4.
C       SUBROUTINE DOUT: PRODUCES DISPLACEMENT FIELD TABLES FOR
C                        U AND V AT VARIOUS POINTS ON THE BLADE,
C                        USING THE DIPLACEMENT FIELD COEFFICIENTS
C                        DETERMINED IN SUBROUTINE EXPAND.
C       SUBROUTINE SCOFF: CALCULATES THE STRESS FIELD COEFFICIENTS
C                         ,FOR SIGMA-X, SIGMA-Y, AND TAU-XY, USING
C                         THE DISPLACEMENT FIELD COEFFICIENTS AND
C                         HOOKE'S LAW.
C       SUBROUTINE STROUT: PRODUCES TABLES OF STRESSES AT VARIOUS POINTS
C                          ON THE BLADE . TABLES MADE ARE SIGMA-X, SIGMA-Y,
C                          SHEAR STRESS, AND THE MAXIMUM PRINCIPAL
C                          NORMAL STRESS.
C       SUBROUTINE VIBRA: READS DIMENSIONLESS INTEGRALS FOR VIBRATION
C                          PROBLEM FROM FILE. CALCULATES MASS, GYRO,
C                          AND STIFFNESS MATRICES. CALLS SUBROUTINES
C                          DECODE, REFORM, AND EIGENS.
C       SUBROUTINE DECODE: PICKS OUT THE RIGHT DIMENSIONLESS INTEGRAL
C                          TO BE USED IN CALCULATING THE MASS, GYRO,
C                          AND STIFFNESS MATRICES.
C       SUBROUTINE REFORM: CHANGES THE SYSTEM OF SECOND ORDER DIFFERENTIAL
C                          EQUATIONS TO A SYSTEM OF FIRST ORDER DIFFERENTIAL
C                          EQUATIONS.
C       SUBROUTINE EIGENS: CALCULATES THE NATURAL FREQUENCIES OF THE
C                          BAND SAW BLADE, AND PRINTS OUT THE 5
C                          LOWEST EIGENVALUES.
C
C       **********************************************************
C       **********************************************************
C       **                                                      **
C       **          IMPORTANT CONTROL VARIBLES USED             **
C       **                   IN PROGRAM                         **
C       **                                                      **
C       **    IPROB: PROBLEM TYPE BEING SOLVED                  **
C       **           1) PLANE STRESSES ONLY                     **
C       **           2) VIBRATION PROBLEM ONLY                  **
C       **           3) BOTH STRESS AND VIBRATION PROBLEM       **
C       **                                                      **
C       **    KOUT: INITIALLY ZERO SET EQUAL TO ONE IF          **
C       **          SINGULAR MATRIX IS ECOUNTERED IN EITHER         **
C       **          SUBROUTINE TCOFF, OR EXPAND. PROGRAM WILL   **
C       **          STOP EXECUTION, IF KOUT=1 !                 **
C       **                                                      **
C       **    KON(1): IS SET EQUAL TO ZERO IF A NEW ORDER OF    **
```

```
C     **             SET EQUAL TO ONE UNLESS NEW ORDER IS           **
C     **             ENTERED. INITIALLY ZERO. USED IN               **
C     **             SUBROUTINE EXPAND.                             **
C     **                                                            **
C     **    KON(2):  INITIALLY ZERO. SET EQUAL TO ONE UNLESS        **
C     **             NEW BLADE GEOMETRY, OR MATERIAL PROPERTIES     **
C     **             ARE ENTERED. USED IN SUBROUTINE EXPAND.        **
C     **                                                            **
C     **    KON(3):  INITIALLY ZERO. SET EQUAL ONE AFTER            **
C     **             PRINTING OF TEMPERATURE TABLE, UNLESS          **
C     **             NEW ORDER OF TEMPERATURE FIELD POLYNOMIAL      **
C     **             IS ENTERED. PREVENTS REPEATED PRINTING         **
C     **             OF TEMPERATURE FIELD TABLE DURING RUN WHEN     **
C     **             THE TEMPERATURE DISTRIBUTION IS THE SAME.      **
C     **             USED IN MAIN PROGRAM.                          **
C     **                                                            **
C     **    KON(4):  INITIALLY ZERO. SET EQUAL TO ONE IF            **
C     **             PRINTING OF DISPLACEMENT FIELD                 **
C     **             COEFFICIENTS IS NOT TO BE DONE.                **
C     **             USED IN SUBROUTINE EXPAND.                     **
C     **                                                            **
C     **    KON(5):  INITIALLY ZERO. SET EQUAL TO ONE IF            **
C     **             PRINTING OF DISPLACEMENT TABLES IS NOT         **
C     **             TO BE DONE. USED IN SUBROUTINE EXPAND          **
C     **                                                            **
C     **    KON(6):  INITIALLY ZERO. SET EQUAL TO ONE IF            **
C     **             PRINTING OF STRESS TABLES IS NOT TO            **
C     **             BE DONE. USED IN MAIN PROGRAM.                 **
C     **                                                            **
C     **    KON(7):  INITIALLY ZERO. SET EQUAL TO ONE IF            **
C     **             PRINTING OF TEMPERATURE TABLE IS NOT           **
C     **             TO BE DONE. USED IN MAIN PROGRAM.              **
C     **                                                            **
C     **    KON(8):  INITIALLY ZERO. SET EQUAL TO ONE AFTER         **
C     **             READING OF DIMENSIONLESS INTEGRALS FOR         **
C     **             VIBRATION ANALYSIS, SO DATA IS NOT             **
C     **             REREAD. USED IN SUBROUTINE VIBRA.              **
C     **                                                            **
C     **    KON(9):  INITIALLY ZERO. SET EQUAL TO ONE AFTER         **
C     **             COMPLETION OF FIRST PROBLEM IN A RUN.          **
C     **             USED IN MAIN PROGRAM.                          **
C     **                                                            **
C     **    KON(10):  INITIALLY ONE. SET EQUAL TO ZERO IF           **
C     **              NEW TITLE WAS ENTERED FOR NEW PROBLEM.        **
C     **              USED IN MAIN PROGRAM.                         **
C     **                                                            **
C     **    KON(11):  USED TO CALL SUBROUTINE SDRIVE, WHICH         **
C     **              CALCULATES THE STRESS FIELD COEFFICIENTS      **
C     **              NEEDED WHEN ONLY THE VIBRATION ANALYSIS       **
C     **              DESIRED.                                      **
C     **                                                            **
C     **    NEW: TELLS PROGRAM WHERE TO GO WHEN A CHANGE IN         **
C     **         THE PROBLEM HAS BEEN MADE.                         **
C     **                                                            **
C     ****************************************************************
C     ****************************************************************
C
C
C
C
```

```
C
C
C     ***********************************************************
C     ***********************************************************
C     **                                                       **
C     **   NOTE: WHEN A STRESS AND VIBRATION PROBLEM ARE RUN   **
C     **         TOGETHER THE ORDER OF THE DISPLACEMENT FIELD  **
C     **         APPROXIMATING FUNCTIONS MUST BE 4!!           **
C     **         THIS IS DONE AUTOMATICALLY FOR THE INITIAL    **
C     **         COMBINED ANALYSIS.                            **
C     **                                                       **
C     ***********************************************************
C     ***********************************************************
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 L,NU,KAPPA
      DIMENSION PSAX(35,3),KON(11),ITITL(50)
      DIMENSION BX(70),TX(35),AX(35,3),X1(2),X2(2)
      DIMENSION TIX(35),TIY(35),FORCX(35),FORCY(35)
      DATA NO /'N'/
C
C     THIS PART OF THE PROGRAM SETS THE CONTROL PARAMETERS
C     .KON, EQUAL TO ZERO.  ESTABLISHES MAXN, MAXM, AND
C     MAXNM. THE CONTROL FOR A SINGULAR MATRIX .KOUT, IS
C     SET EQUAL TO ZERO, IF A SINGULAR MATRIX IS FOUND
C     KOUT IS SET EQUAL TO 1.
      MAXM=6
      MAXN=6
      MAXNM=36
      KAPPA=0.0D+00
      MAXT=0
      C=0.0D+00
      KOUT=0
      DO 60 I=1,2
      X1(I)=1.0D+00
   60 X2(I)=2.0D+00
      DO 65 I=1,9
   65 KON(I)=0
      KON(10)=1
C
C     THIS PART OF THE PROGRAM BEGINS THE INTERACTIVE
C     COMPUTER SESSION. FIRST THE TITLE IS READ INTO
C     THE LABEL ITITL.
      WRITE(6,610)
  610 FORMAT(///,3X,'BEGIN PROBLEM DESCRIPTION')
      WRITE(6,611)
  611 FORMAT(/,6X,'ENTER THE TITLE OF THIS RUN (50 CHARACTERS MAX)')
      READ(5,512) ITITL
  512 FORMAT(50A1)
C
C      THIS PART OF THE PROGRAM READS THE TYPE OF PROBLEM
C      TO BE SOLVED.
    2 WRITE(6,613)
  613 FORMAT(/,6X,'ENTER NUMBER OF PROBLEM TO BE SOLVED',/,
     $6X,'1-PLATE STRESSES',/,
     $6X,'2-PLATE VIBRATION ',/,
     $6X,'3-PLATE STRESSES AND VIBRATIONS')
       READ(5,513) IPROB
```

```
      IF(IPROB .EQ. 2 .AND. KON(9) .EQ. O) KON(11)=O
C
C     THIS IF STATEMENT CHECKS TO MAKE SURE THAT A PROPER
C     PROBLEM TYPE HAS BEEN CHOSEN.
      IF(IPROB .GT. 3) GO TO 2
      IF(IPROB .LE. O) GO TO 2
C
C     THIS IF STATEMENT CHECKS TO SEE IF ONLY THE VIBRATION
C     PROBLEM WAS CHOSEN.
      IF(IPROB .EQ. 2) GO TO 20
C
C     THIS PART OF THE PROGRAM ASKES THE USER THE DEGREE OF
C     THE POLYNOMIAL TO BE USED IN THE STRESS AND TEMPERATURE
C     FIELD COEFFICIENT CALCULATIONS.
    5 WRITE(6,624)
  624 FORMAT(//,3X,'BEGIN DESCRIPTION OF APPROXIMATING FUNCTIONS FOR
     $ ,'DISPLACEMENT FIELD COEFFICIENT CALCULATION')
      WRITE(6,614)
      WRITE(6,638)
  614 FORMAT(/,6X,'ENTER DEGREE OF THE POLYNOMIAL FOR THE APPROXIMATING
     $,' FUNCTIONS')
  638 FORMAT(6X,'INTEGER 2 TO 7')
      READ(5,513) MAXS
  513 FORMAT(I1)
      KON(1)=O
      KON(2)=O
      IF(KON(9) .EQ. 1) GO TO 52
C
C     THE IF STATEMENT CHECKS THAT MAXS IS SMALL ENOUGH SO
C     ILL CONDITIONING OF COEFFICIENT MATRIX DOES NOT OCCUR.
      IF(MAXS .GT. 7) MAXS=6
      IF(MAXS .LT. 2) MAXS=2
  615 FORMAT(/,6X,'ENTER DEGREE OF THE POLYNOMIAL FOR THE APPROXIMATING
     $,' FUNCTION')
    6 WRITE(6,636)
  636 FORMAT(//,3X,'BEGIN DESCRIPTION OF APPROXIMATING FUNCTION FOR
     $'TEMPERATURE FIELD COFFICIENT CALCULATION')
      WRITE(6,615)
      I2=MAXS-1
      WRITE(6,637) I2
  637 FORMAT(6X,'IF ZERO NO TEMPERATURE FIELD IS PRESENT',/,
     $6X,'OTHERWISE ENTER INTEGER 1 TO',I2)
      READ(5,513) MAXT
      IF(MAXT .EQ. O) KON(3)=1
      IF(KON(9) .EQ. 1 .AND. MAXT .EQ. O) GO TO 52
C
C     THE IF STATEMENT CHECKS TO SEE IF A TEMPERATURE FIELD
C     IS PRESENT.
      IF(MAXT .EQ. O) GO TO 30
      IF(MAXT .GE. MAXS) MAXT=MAXS-1
      CALL TDRIVE (MAXT,TX,KOUT)
      KON(3)=O
C
C     THIS IF STATEMENT AND OTHERS LIKE IT CHECK TO SEE IF
C     A SINGULAR MATRIX WAS GENERATED IN ANY SUBROUTINE THAT
C     USES THE MTS SYSTEM SUBROUTINES DLUD AND DBS.
      IF(KOUT .EQ. 1) GO TO 57
      IF(KON(9) .EQ. 1) GO TO 52
C
C     THIS IF STATEMENT CHECKS TO SEE IF ONLY THE STRESS FIELD
```

```
C       COEFFICIENTS ARE TO BE CALCULATED.
     30 IF(IPROB .EQ. 1) GO TO 10
C
C       THIS PART OF THE PROGRAM ASKES THE USER THE NUMBER OF
C       EIGENFUNCTIONS WHICH ARE TO BE USED IN THE NATURAL
C       FREQUENCY CALCULATION.
     20 WRITE(6,625)
    625 FORMAT(//,3X,'BEGIN DESCRIPTION OF EIGENFUNCTIONS FOR ',
       $'BANDSAW BLADE VIBRATIONS')
        WRITE(6,634)
    634 FORMAT(/,6X,'ENTER NUMBER OF SIMPLY SUPPORTED BEAM EIGENFUNCTIONS'
      ' $,/,6X,'INTEGER 1 TO 6')
        READ(5,513) N
        WRITE(6,635)
    635 FORMAT(/,6X,'ENTER NUMBER OF FREE-FREE BEAM EIGENFUNCTIONS',/,
       $6X,'INTEGER 1 TO 6')
        READ(5,513) M
C
C       THIS IF STATEMENT CHECKS TO MAKE SURE THAT THE DEGREE OF THE
C       CALCULATED STRESS FIELDS MATCHES THAT OF THE ONES REQUIRED IN
C       THE VIBRATION ANALYSIS.
        IF(KON(9) .EQ. O .AND. IPROB .EQ. 3) GO TO 825
        IF(KON(9) .EQ. O .AND. IPROB .EQ. 2) GO TO 825
        IF(KON(9) .EQ. 1 .AND. MAXS .EQ. 4) GO TO 825
        WRITE(6,650)
        GO TO 57
    650 FORMAT(//,3X,'**** ERROR THE STRESS FIELD POLYNOMIALS DO NOT ',
       1'MATCH THOSE REQUIRED FOR THE VIBRATION ANALYSIS! ***')
    825 MAXS=4
        IF(KON(9) .EQ. 1) GO TO 27
C
C       THIS PART OF THE PROGRAM ASKES THE USER FOR IFORMATION
C       CONCERNING THE GEOMETRY OF THE PLATE (LENGTH, WIDTH,
C       AND THICKNESS).
     10 WRITE(6,626)
    626 FORMAT(//,3X,'BEGIN DESCRIPTION OF GEOMETRY OF BANDSAW BLADE')
        WRITE(6,616)
    616 FORMAT(/,6X,'ENTER LENGTH OF BANDSAW BLADE')
        READ(5,514) L
    514 FORMAT(D15.6)
        WRITE(6,617)
    617 FORMAT(/,6X,'ENTER BANDSAW BLADE WIDTH')
        READ(5,514) B
        WRITE(6,618)
    618 FORMAT(/,6X,'ENTER BANDSAW BLADE THICKNESS')
        READ(5,514) H
        KON(2)=O
        IF(KON(9) .EQ. 1) GO TO 52
C
C       THIS PART OF THE PROGRAM ASKES THE USER TO ENTER THE
C       MATERIAL PROPERTIES OF THE BANDSAW BLADE (MODULUS OF
C       ELASTICITY, POISSON'S RATIO, MASS DENSITY, AND
C       COEFFICIENT OF THERMAL EXPANSION).
     12 WRITE(6,627)
    627 FORMAT(//,3X,'BEGIN DESCRIPTION OF MATERIAL PROPERTIES ',
       $'OF THE BANDSAW BLADE')
        WRITE(6,620)
    620 FORMAT(/,6X,'ENTER MODULUS OF ELASTICTY (YOUNG"S MODULUS)')
        READ(5,514) E
```

```
  619 FORMAT(/,6X,'ENTER POISSON'S RATIO')
      READ(5,514) NU
      WRITE(6,621)
  621 FORMAT(/,6X,'ENTER THE MASS DENSITY OF THE MATERIAL')
      READ(5,514) RHOH
      RHI=RHOH
      RHOH=RHOH*H
      KON(2)=0
      IF(KON(9) .EQ. 1 .AND. MAXT .EQ. 0) GO TO 52
      IF(MAXT .EQ. 0) GO TO 35
      WRITE(6,640)
  640 FORMAT(/,6X,'ENTER COEFFICIENT OF THERMAL EXPANSION')
      READ(5,514) ALPHA
      IF(KON(9) .EQ. 1) GO TO 52
C
C     THIS PART OF THE PROGRAM ASKES THE USER TO SUPPLY
C     INFORMATION CONCERNING THE OPERATING CONDIONS OF
C     THE BANDSAW.
   35 WRITE(6,628)
  628 FORMAT(//,3X,'BEGIN DESCRIPTION OF OPERATING CONDITIONS OF THE '
     $,'BANDSAW BLADE')
      WRITE(6,622)
  622 FORMAT(/,6X,'ENTER THE VELOCITY OF THE BANDSAW BLADE')
      READ(5,514) C
      WRITE(6,641)
  641 FORMAT(/,6X,'ENTER WHEEL SUPPORT COEFFICIENT')
      READ(5,514) KAPPA
      IF(IPROB .EQ. 2) GO TO 27
      WRITE(6,623)
  623 FORMAT(/,6X,'ENTER THE INITIAL TENSION ON',
     $' THE BANDSAW BLADE')
      READ(5,514) RO
      RIN=RO
C
C   THIS STATEMENT CHANGES RO FROM A FORCE TO A FORCE PER UNIT LENGTH.
      RO=RO/B
      WRITE(6,629)
  629 FORMAT(/,6X,'ENTER NORMAL CUTTING FORCE ON',
     $' THE BANDSAW BLADE')
      READ(5,514) FNOR
      FNRI=FNOR
      IF(FNOR .EQ. 0.0) GO TO 43
      WRITE(6,631)
  631 FORMAT(/,9X,'ENTER STARTING X-COORDINATE OF NORMAL CUTTING '
     $,'FORCE')
      READ(5,514) X1(1)
      WRITE(6,632)
  632 FORMAT(9X,'ENTER ENDING X-COORDINATE OF NORMAL CUTTING FORCE')
      READ(5,514) X2(1)
C
C     THIS STATEMENT CHANGES FNOR FROM A FORCE TO A FORCE PER
C     UNIT LENGTH. FNOR IS THE NORMAL CUTTING FORCE.
      FNOR=FNOR/(X2(1)-X1(1))
   43 WRITE(6,630)
  630 FORMAT(/,6X,'ENTER TANGENTIAL CUTTING FORCE ON',
     $' THE BANDSAW BLADE')
      READ(5,514) FTAN
      FTNI=FTAN
      IF(FTAN .EQ. 0.0) GO TO 27
      WRITE(6,633)
```

```
      633 FORMAT(/,9X,'ENTER STARTING X-COORDINATE OF TANGENTIAL CUTTING ',
         $'FORCE')
          READ(5,514) X1(2)
          WRITE(6,647)
      647 FORMAT(9X,'ENTER ENDING X-COORDINATE OF TANGENTIAL CUTTING ',
         $'FORCE')
          READ(5,514) X2(2)
C
C         THIS STATEMENT CHANGES FTAN FROM A FORCE TO A FORCE PER
C         UNIT LENGTH. FTAN IS THE TANGENTIAL CUTTING FORCE.
          FTAN=FTAN/(X2(2)-X1(2))
C  28 WRITE(6,827)
C 827 FORMAT(//,6X,'IS THERE A PRESTRESS ON THE BAND SAW BLADE (Y/N)?')
C         READ(5,801) ICPS
C
C         THIS IF STATEMENT CHECKS TO SEE IF A PRESTRESS FIELD IS PRESENT.
C         IF(ICPS .EQ. NO) GO TO 27
C         WRITE(6,828)
C 828 FORMAT(/,9X,'ENTER ORDER OF THE POLYNOMIAL FOR PRESTRESS FIELD '
C        1,'CALCULATION',/,9X,'INTERGER 1 TO 6')
C         READ(5,513) NPPS
C
C         HERE SDRIVE CALCULATES THE PRESTRESS FIELDS.
C         SDRIVE DOES NOT WORK PROPERLY AT THIS TIME.
C         CALL SDRIVE(PSAX,NPPS,KOUT)
C         IF(KOUT .EQ. 1) GO TO 57
       27 IF(KON(9) .EQ. 1) GO TO 52
C
C         THIS PART OF THE PROGRAM ASKS THE USER WHAT OUTPUT IS TO
C         BE WRITTEN TO THE OUTPUT FILE
          WRITE(6,802)
      802 FORMAT(///,3X,'BEGIN OUTPUT CONTROL DESCRIPTION')
          IPRD=4
          IF(MAXT .EQ. 0) IPRD=3
          DO 810 IPT=1,IPRD
          IF(IPT .EQ. 1) WRITE(6,800)
          IF(IPT .EQ. 2) WRITE(6,803)
          IF(IPT. EQ. 3) WRITE(6,804)
          IF(IPT .EQ. 4) WRITE(6,805)
          IKON=IPT+3
          READ(5,801) IDP
          IF(IDP .EQ. NO) KON(IKON)=1
      800 FORMAT(/,6X,'DO YOU WISH THE PRINTING OF THE DISPLACEMENT',
         1' FIELD COEFFICIENTS (Y/N)?')
      803 FORMAT(/,6X,'DO YOU WISH THE PRINTING OF THE DISPLACEMENT',
         1' FIELD TABLES (Y/N)?')
      804 FORMAT(/,6X,'DO YOU WISH THE PRINTING OF THE STRESS FIELD',
         1' TABLES (Y/N)?')
      805 FORMAT(/,6X,'DO YOU WISH THE PRINTING OF THE TEMPERATURE',
         1' FIELD COEFFICIENTS/TABLE (Y/N)?')
      801 FORMAT(1A1)
      810 CONTINUE
C
C         THIS PART OF THE PROGRAM WRITES TO THE OUTPUT FILE, DEVICE
C         #8, THE USER INPUT FROM THE INTERACTIVE SESSION.
       25 WRITE(8,710) ITITL
          IF(IPROB .EQ. 2) GO TO 80
          WRITE(8,720) MAXS,MAXT
          IF(IPROB .EQ. 1) GO TO 81
```

```
   81 WRITE(8,740) L,B,H
      WRITE(8,750) E,NU,RHI
      IF(MAXT .GT. O) WRITE(8,760) ALPHA
      WRITE(8,770) C,KAPPA
      IF(IPROB .EQ. 2) GO TO 55
      WRITE(8,780) RIN,FNRI,FTNI
      IF(FNRI .NE. O.O) WRITE(8,790) X1(1),X2(1)
      IF(FTNI .NE. O.O) WRITE(8,795) X1(2),X2(2)
  710 FORMAT(1H1,2(/,1H-),40X,50A1)
  720 FORMAT(1HO,5X,'THE DEGREE OF THE POLYNOMIAL FOR THE DISPLACEMENT'
     1,' FIELD CALCULATION IS',I2,/,1HO,5X,'THE DEGREE OF THE ',
     2'POLYNOMIAL FOR THE TEMPERATURE FIELD CALCULATION IS',I2)
  730 FORMAT(1HO,5X,'THE NUMBER OF SIMPLY SUPPORTED BEAM ',
     1'EIGENFUNCTIONS IS',I2,/,1HO,5X,'THE NUMBER OF FREE-FREE ',
     2'BEAM EIGENFUNCTIONS IS',I2)
  740 FORMAT(1HO,5X,'THE LENGTH OF THE BANDSAW BLADE IS',D15.6,/,
     11HO,5X,'THE WIDTH OF THE BANDSAW BLADE IS',D15.6,/,1HO,5X,
     2'THE THICKNESS OF THE BANDSAW BLADE IS',D15.6)
  750 FORMAT(1HO,5X,'THE MODULUS OF ELASTICITY IS',D15.6,/,1HO,5X,
     1'POISON"S RATIO IS',D15.6,/,1HO,5X,'THE MASS DENSITY IS',D15.6)
  760 FORMAT(1HO,5X,'THE COEFFICIENT OF THERMAL EXPANSION IS',D15.6)
  770 FORMAT(1HO,5X,'THE VELOCITY OF THE BANDSAW BLADE IS',D15.6,
     1/,1HO,5X,'THE WHEEL SUPPORT COEFFICIENT IS',D15.6)
  780 FORMAT(1HO,5X,'THE INITIAL TENSION IS',D15.6,/,1HO,5X,
     1'THE NORMAL CUTTING FORCE IS',D15.6,/,1HO,5X,'THE TANGENTIAL',
     2' CUTTING FORCE IS',D15.6)
  790 FORMAT(1HO,5X,'THE STARTING X-COORDINATE FOR THE NORMAL'
     1,' CUTTING FORCE IS',D15.6,/,1HO,5X,'THE ENDING X-COORDINATE',
     2' FOR THE NORMAL CUTTING FORCE IS',D15.6)
  795 FORMAT(1HO,5X,'THE STARTING X-COORDINATE FOR THE TANGENTIAL ',
     1'CUTTING FORCE IS',D15.6,/,1HO,5X,'THE ENDING X-COORDINATE FOR',
     2' THE TANGENTIAL CUTTING FORCE IS',D15.6)
C
C     THIS EQUATION CALCULATES THE RUNNING TENSION ON THE BLADE.
      R=RO-RHOH*KAPPA*C**2
      IF(MAXT .EQ. O) GO TO 40
      IF(KON(3) .EQ. 1) GO TO 40
      IF(KON(7) .EQ. 1) GO TO 40
      CALL TOUT(MAXT,L,B,TX,ITITL)
      KON(3)=1
C
C     SUBROUTINES FORCE,TEMP,RHS, AND EXPAND ARE USED
C     TO CALCULATE THE STRESS FIELD COEFFIECIENTS.
   40 CALL FORCE(MAXS,R,FTAN,FNOR,X1,X2,L,B,FORCX,FORCY,MSIZE)
      CALL TEMP(MAXS,MAXT,TX,MSIZE,L,B,H,E,NU,ALPHA,TIX,TIY)
      CALL RHS(FORCX,FORCY,TIX,TIY,MSIZE,MAXS,BX)
      CALL EXPAND(MAXS,MSIZE,L,B,H,E,NU,BX,AX,KOUT,KON,ITITL)
      IF(KOUT .EQ. 1) GO TO 57
      IF(KON(6) .EQ. 1) GO TO 50
      CALL STROUT(MAXS,L,B,AX,ITITL)
      IF(IPROB .EQ. 1) GO TO 50
   55 CONTINUE
      IF(N .GT. MAXN) N=MAXN
      IF(M .GT. MAXM) M=MAXM
      NM=N*M
C
C     D IS THE FLEXURE MODULUS OF THE PLATE, WHICH IS USED IN
C     THE VIBRATION ANALYSIS.
      D=(E*H**3)/((1.2D+O1)*((1.OD+OO)-NU**2))
      IF(IPROB .NE. 2) GO TO 77
```

```
      IF(KON(11) .EQ. 0) GO TO 76
      WRITE(6,850)
  850 FORMAT(///,3X,'DO YOU WISH TO USE THE STRESS FIELD COEFFICIENTS',
     1' CALCULATED PREVIOUSLY (Y/N)?')
      READ(5,801) ISDS
      IF(ISDS .EQ. NO) KON(11)=0
      IF(KON(11) .EQ. 1) GO TO 77
   76 NCSF=3
      IF(KON(11) .EQ. 0) CALL SDRIVE(AX,NCSF,KOUT)
      KON(11)=1
      IF(KOUT .EQ. 1) GO TO 57
   77 CONTINUE
      CALL VIBRA(N,M,NM,MAXNM,KON,L,B,H,D,RHOH,NU,C,AX,ITITL)
   50 CONTINUE
      KON(9)=1
C
C     THIS PART OF THE PROGRAM ASKS THE USER WHAT IS TO BE DONE
C     NEXT. ANOTHER PROBLEM, OR STOP.
   52 WRITE(6,700)
  700 FORMAT(//,3X,'ENTER NUMBER OF NEXT INSTRUCTION',/,
     $6X,'1- NEW PROBLEM TYPE',/,6X,'2- NEW APPROXIMATING',
     $' FUNCTION FOR DISPLACEMENTS',/,6X,'3- NEW APPROXIMATING',
     $' FUNCTION FOR TEMPERATURE',/,6X,'4- NEW EIGENFUNCTIONS FOR ',
     $'VIBRATION ANALYSIS',/,6X,'5- NEW BLADE GEOMETRY',
     $/,6X,'6- NEW MATERIAL PROPERTIES',/,6X,'7- NEW',
     $' OPERATING CONDITIONS',/,6X,'8- SOLVE NEW PROBLEM'
     $,/,6X,'9- STOP')
      READ(5,513) NEW
      IF(NEW .EQ. 9) GO TO 57
      IF(NEW .EQ. 8) KON(10)=0
      IF(KON(10) .NE. 0) GO TO 62
      WRITE(6,611)
      READ(5,512) ITITL
   62 KON(10)=1
      GO TO(2,5,6,20,10,12,35,25),NEW
   57 STOP
      END
End of file
```

STRESS ANALYSIS SUBROUTINES

```
      SUBROUTINE EXPAND(N,MSIZE,EL,B,H,E,UN,BX,AX,KOUT,KON,ITITL)
C
C     EXPAND CALCULATES THE DIMENSIONIZED MATRIX (A), WHICH IS
C     USED TO SOLVE FOR THE COEFFICIENTS OF THE DISPLACEMENT FIELD
C     APPROXIMATING FUNCTIONS FOR U AND V. THE SUBMATRICES K1, K2
C     , K3, AND K4 ARE CALCULATED USING THE NONDIMENSIONAL INTEGRALS
C     CALCULATED BY SUBROUTINE AMOD. THE MATRIX (A) IS THEN ASSEMBLED
C     WITH THE ROW CORRESPONDING TO THE VARION OF B(N,O), AND THE
C     COLUMN CORRESPONDING TO B(N,O) BEING ELIMINATED.
C
C     EXPAND CALLS SUBROUTINES MATDIM, DLUD, DBS, DOUT, AND SCOFF.
C     SUBROUTINE MATDIM: CALCULATES THE NONDIMENSIONAL INTEGRALS
C                        FOR USE IN CALCULATING THE SUBMATRICES
C                        K1, K2, K3, AND K4.
C     SUBROUTINE DLUD: DECOMPOSES THE MATRIX (A) INTO LOWER AND
C                      UPPER TRIANGULAR MATRICES.
C     SUBROUTINE DBS: CALCULATES THE UDETERMINED COEFFICIENTS A,
C                     AND B BY FORWARD ELIMATION AND BACK SUBSTITUTION.
C     SUBROUTINE DOUT: PRODUCES DISPLACEMENT FIELD TABLES FOR
C                      U AND V AT VARIOUS POINTS ON THE BLADE.
C     SUBROUTINE SCOFF: CALCULATES THE STRESS FIELD COEFFICIENTS
C                       USING THE DISPLACEMENT FIELD COEFFICIENTS
C                       AND HOOKE"S LAW.
C
C     *********************************************************
C     *********************************************************
C     **                                                     **
C     **      IMPORTANT VARIBLES IN SUBROUTINE EXPAND        **
C     **                                                     **
C     **   EL: BLADE LENGTH BETWEEN SUPPORTS.                **
C     **       VARIBLE (L) IN MAIN PROGRAM AND SUBROUTINES   **
C     **       VIBRA, AND EIGENS.                            **
C     **                                                     **
C     **   B: BLADE WIDTH.                                   **
C     **                                                     **
C     **   H: BLADE THICKNESS.                               **
C     **                                                     **
C     **   E: MODULUS OF ELASTICITY.                         **
C     **                                                     **
C     **   UN: POISSON"S RATIO. VARIBLE (NU) IN MAIN         **
C     **       PROGRAM AND SUBROUTINE VIBRA.                 **
C     **                                                     **
C     **   N: ORDER OF APPROXIMATING FUNCTIONS FOR           **
C     **      DISPLACEMENT FIELDS U AND V.                   **
C     **                                                     **
C     **   MSIZE: NUMBER OF TERMS IN APPROXIMATING FUNCTION  **
C     **          FOR DISPLACEMENT FIELD (U).                **
C     **                                                     **
C     **   NSIZE: NUMBER OF TERMS IN APPROXIMATING FUNCTION  **
C     **          FOR DISPLACEMENT FIELD (V).                **
C     **                                                     **
C     **   J2: SIZE OF THE MATRIX (A). MSIZE+NSIZE.          **
C     **                                                     **
C     **   NM: MAXIMUM SIZE OF THE MATRIX (A).               **
C     **                                                     **
C     **   KON(1): O- NEW ORDER OF APPROXIMATING FUNCTIONS.  **
C     **           1- ORDER OF APPROXIMATING FUNCTIONS DID   **
C     **              NOT CHANGE.                            **
C     **                                                     **
C     **   KON(2): O- NEW BLADE GEOMETRY, MATERIAL PROPERTIES **
```

```
C      **              , OR ORDER OF APPROXIMATING FUNCTIONS.  **
C      **            1- NO CHANGE IN BLADE GEOMETRY, MATERIAL    **
C      **               PROPERTIES, OR ORDER OF APPROXIMATING   **
C      **               FUNCTIONS.                              **
C      **                                                       **
C      **   KON(4): O- PRINT OUT DISPLACEMENT FIELD             **
C      **               COEFFICIENTS FOR U AND V TO FILE.       **
C      **            1- DO NOT PRINT OUT DISPLACEMENT FIELD     **
C      **               COEFFICIENTS TO FILE.                   **
C      **                                                       **
C      **   KON(5): O- PRODUCE DISPLACEMENT FIELD TABLES.       **
C      **            1- DO NOT PRODUCE DISPLACEMENT FIELD       **
C      **               TABLES                                  **
C      **                                                       **
C      **   COFF: DISPLACEMENT FIELD COEFFICIENTS.              **
C      **                                                       **
C      **   AX: STRESS FIELD COEFFICIENTS FOR SIGMA-X,          **
C      **       SIGMA, AND TAU-XY.                              **
C      **                                                       **
C      **********************************************************
C      **********************************************************
C
C
       IMPLICIT REAL*8(A-H,O-Z)
       REAL*8 K1,K2,K3,K4
       COMMON /SUB2/ XP,YP,XP1,YP1
       COMMON /SUB4/ AMOD
       DIMENSION ITITL(50),AMOD(7,1230),XP1(1230),YP1(1230),XP(730)
       DIMENSION YP(730),A(70,70),K1(35,35),K2(35,35),K3(35,35),K4(35,35)
       DIMENSION KON(11),AX(35,3),T(70,70),IV(70),BX(70),COFF(75)
C
C      INITIALIZE THE STRESS COEFFICIENTS TO ZERO
       DO 52 I=1,3
       DO 53 J=1,35
    53 AX(J,I)=0.0D+00
    52 CONTINUE
C
C      NM IS THE MAXIMUM MATRIX SIZE FOR SUBROUTINES DLUD AND DBS
       NM=70
C
C      MSIZE IS THE DIMENSION OF MATRICES K1,K2,K3,K4
C      ALSO, MSIZE IS THE NUMBER OF TERMS IN THE APPROXIMATING
C      FUNCTION FOR U.
       JJ=MSIZE
C
C      NSIZE IS THE NUMBER OF TERMS IN THE APPROXIAMATING
C      FUNCTION FOR V WHEN C(N,O) IS REMOVED
       NSIZE=MSIZE-1
C
C      J2 IS THE DIMENSION OF THE COFFECIENT MATRIX A.
       J2=JJ+NSIZE
C
C      SUBROUTINE MATDIM FORMS THE NONDIMENSIONAL
C      INTEGRATIONS OF THE VARIATIONAL EQUATION FOR U AND V.
C
C      THIS SET OF IF STATEMENTS CHECKS TO SEE IF THE NONDIMENSIONAL
C      INTEGRALS NEED TO BE RECALCULATED.
       IF(KON(1) .EQ. O ) GO TO 73
       IF(KON(1) .EQ. 1) GO TO 68
```

```
      KON(1)=1
C
C     THIS IF STATEMENT CHECKS TO SEE IF NEW BLADE GEOMETRY, OR NEW
C     MATERIAL PROPERTIES HAVE BEEN ENTERED.
   68 IF(KON(2) .EQ. 1 ) GO TO 65
      JK=1
      DO 15 I=1,MSIZE
      DO 15 J=I,MSIZE
C
C     K1 IS THE COEFFICIENT OF A(N,M) MULTIPLIED BY THE
C     VARIATION OF A(N,M)
      K1(I,J)=(E*H/((1.OD+OO)-UN**2)*AMOD(1,JK)*(B/EL)
     1+E*H/((1.OD+OO)+UN)*AMOD(2,JK)*(EL/B))*EL**XP(JK)*
     2B**YP(JK)
      K1(J,I)=K1(I,J)
C
C     K4 IS THE COEFFICIENT OF B(N,M) MULTIPLIED BY THE
C     VARIATION OF B(N,M)
      K4(I,J)=(E*H/((1.OD+OO)-UN**2)*AMOD(6,JK)*(EL/B)+E*H/((1.OD+OO)
     1+UN)*AMOD(7,JK)*(B/EL))*EL**XP(JK)*B**YP(JK)
      K4(J,I)=K4(I,J)
   15 JK=JK+1
      JK=1
      DO 16 I=1,MSIZE
      DO 16 J=1,MSIZE
C
C     K2 IS THE COEFFICIENT OF B(N,M) MULTIPLIED BY THE
C     VARIATION OF A(N,M)
      K2(I,J)=(E*H/((1.OD+OO)-UN)*AMOD(3,JK)-E*H*UN/((1.OD+OO)-UN**2)*
     1AMOD(4,JK)-E*H/((1.OD+OO)+UN)*AMOD(5,JK))*EL**XP1(JK)*B**YP1(JK)
C
C     K3 IS THE COFFICIENT OF A(N,M) MULTIPLIED BY THE
C     VARIATION OF B(N,M), K3 IS EQUAL TO K2 TRANSPOSED.
      K3(J,I)=K2(I,J)
   16 JK=JK+1
C
C     THIS NESTED DO LOOP ASSEMBLES THE COEFFICIENT MATRIX ,A,
C     ELIMINATING THE COLUMN CORRESPONDING TO B(N,O) AND ROW
C     CORRESPONDING TO THE VARIATION OF B(N,O)
      DO 25 I=1,MSIZE
      IA=I+JJ
      DO 26 J=1,MSIZE
      J1=J
      I1=I
      JA=J+JJ
      A(I,J)=K1(I1,J1)
      IF(J1 .EQ. MSIZE) GO TO 30
      IF(J .GE. N) J1=J1+1
      A(I,JA)=K2(I1,J1)
   30 J1=J
      IF( I1 .EQ. MSIZE) GO TO 26
      IF(I .GE. N) I1=I1+1
      A(IA,J)=K3(I1,J1)
      IF(J1 .EQ. MSIZE) GO TO 25
      IF(J .GE. N) J1=J1+1
      A(IA,JA)=K4(I1,J1)
   26 CONTINUE
   25 CONTINUE
C
C     SUBROUTINE DLUD IS A SYSTEM SUBROUTINE THAT DECOMPOSES THE
```

```
C     MATRIX(A) INTO LOWER AND UPPER TRIANGLULAR MATRICES.
C     THE DECOMPOSED MATRICES ARE STORED IN MATRIX(T).
      CALL DLUD(J2,NM,A,NM,T,IV)
      IF(IV(J2) .EQ. 0) GO TO 50
      KON(2)=1
C
C     SUBROUTINE DBS IS A SYSTEM SUBROUTINE THAT SOLVES
C     THE SYSTEM OF EQUATIONS [A]{X}={BX} BY FORWARD
C     ELIMINATION AND BACK SUBSTITUTION
   65 CONTINUE
      CALL DBS(J2,NM,T,IV,BX)
C
C     INITIALIZE DISPLACEMENT FIELD COEFFICIENTS FOR U AND V TO ZERO
      DO 62 J=1,45
   62 COFF(J)=0.0D+00
      JM=J2+1
      MINUS=0
      DO 75 J=1,JM
      J1=J
      IF(MINUS .EQ. 1) J1=J1-1
      IF(J .EQ. MSIZE+N) GO TO 60
C
C     COFF ARE THE COEFFICIENTS A(N,M) FOR DISPLACEMENT FIELD U,
C     AND B(N,M) FOR DISPLACEMENT FIELD V.
      COFF(J)=BX(J1)
      GO TO 75
C
C     THIS DO LOOP CALCULATES B(N,0) KNOWING
C     B(1,0),B(2,0),....,B(N-1,0)
   60 DO 85 I=1,N
      IF(N-I .EQ. 0) GO TO 75
      J3=J1-N+I
      F=-BX(J3)/EL**(N-I)
      COFF(J)=F+COFF(J)
      MINUS=1
   85 CONTINUE
   75 CONTINUE
C
C     SUBROUTINE ZEROC IS USED TO SET SMALL DISPLACEMENT FIELD
C     COEFFICIENTS TO ZERO.
      MD=J2+1
      CALL ZEROC(COFF,MD,N,EL,B)
      WRITE(8,500) ITITL
      WRITE(8,101)
C
C     THIS NESTED DO LOOP WRITES OUT THE COEFFICIENTS
C     FOR THE APPROXIMATING FUNCTIONS U AND V
C
C     THIS IF STATEMENT CHECKS TO SEE IF THE DISPLACEMENT COEFFICIENTS
C     ARE TO BE PRINTED TO THE OUTFILE.
      IF(KON(4) .EQ. 1) GO TO 79
      IJ=1
      DO 95 J=1,2
      IF(J .EQ. 2) WRITE(8,500) ITITL
      IF(J .EQ. 2) WRITE(8,102)
      DO 105 I1=1,N
      I=I1-1
      DO 115 J1=1,N
C
```

```
C      COEFFICIENT, AND I IS THE ORDER OF Y ASSOCIATED WITH EACH
C      DISPLACEMENT FIELD COEFFICIENT.
       IF(J1+I .GT. N) GO TO 105
       IF(J .EQ. 1) WRITE(8,100) J1,I,COFF(IJ)
       IF(J .EQ. 2) WRITE(8,103) J1,I,COFF(IJ)
   115 IJ=IJ+1
   105 CONTINUE
       I=0
       DO 120 J1=1,N
       IF(J .EQ. 1) WRITE(8,100) I,J1,COFF(IJ)
       IF(J .EQ. 2) WRITE(8,103) I,J1,COFF(IJ)
   120 IJ=IJ+1
    95 CONTINUE
   100 FORMAT(8X,'A(',I2,',',I2,')=',D14.5)
   101 FORMAT(1H-,4X,'COEFFICIENTS OF APPROXIMATING FUNCTION FOR'
      $,' DISPLACEMENT FIELD U')
   102 FORMAT(1H-,4X,'COEFFICIENTS OF APPROXIMATING FUNCTION FOR'
      $,' DISPLACEMENT FIELD V')
   103 FORMAT(8X,'B(',I2,',',I2,')=',D14.5)
   500 FORMAT(1H1,2(/,1H-),40X,50A1)
C
C      THIS IF STATEMENT CHECKS TO SEE IF THE DISPLACEMENT FIELD
C      TABLES ARE TO BE PRODUCED.
    79 IF(KON(5) .EQ. 1) GO TO 97
C
C      DOUT PRODUCES DISPLACEMENT FIELD TABLES OF U AND V.
       CALL DOUT(N,MSIZE,EL,B,COFF,ITITL)
C
C      SCOFF CALCULATES THE STRESS FIELD COEFFICIENTS, AX, FOR SIGMA-X,
C      SIGMA-Y, AND TAU-XY USING THE DISPLACEMENT FIELD COEFFICIENTS
C      AND HOOKE'S LAW.
    97 CALL SCOFF(EL,B,N,MSIZE,COFF,E,UN,AX,ITITL)
       GO TO 110
C
C      SINGULAR MATRIX ENCOUNTERED.
    50 WRITE(6,106)
       KOUT=1
   106 FORMAT(4X,'*** ERROR ** MATRIX IS SINGULAR ',
      $'IN SUBROUTINE EXPAND')
   110 RETURN
       END
       SUBROUTINE ZEROC(ZCOF,MD,N,EL,B)
C
C      ZEROC SETS SMALL DISPLACEMENT COEFFICIENTS TO ZERO. EACH COEFFICIENT
C      IS MULTIPLIED BY THE LENGHT(EL) RAISED TO THE APPROPIATE POWER
C      AND THE WIDTH(B) RAISED TO THE APPROPIATE POWER AND STORED IN ZCF.
C      THE MAXIMUM MAGNITUDE OF ZCF IS THEN FOUND. EACH VALUE OF ZCF IS
C      THEN DIVIDED BY THIS MAXIMUM VALUE, IF THE RESULT IS LESS THAN 1.0E-6.
C      THE COEFFICIENT ZCOF CORRESPONDING TO ZCF IS SET EQUAL ZERO. ZCOF IS
C      TRANSFERED BACK TO SUBROUTINE EXPAND AS COFF.
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION ZCF(75),ZCOF(75)
       JBL=1
C
C      THIS SET OF DO LOOPS MULTIPLIES THE DISPLACEMENT FIELD COEFFICIENTS
C      ZCOF BY EL AND B RAISED TO THE APPROPIATE POWER.
    30 DO 10 K=1,N
       ZCF(JBL)=ZCOF(JBL)*EL**K
    10 JBL=JBL+1
       DO 15 L=1,N
```

```
      DO 20 K=1,N
      IF(K+L .GT. N) GO TO 15
      ZCF(JBL)=ZCOF(JBL)*EL**K*B**L
   20 JBL=JBL+1
   15 CONTINUE
      DO 25 L=1,N
      ZCF(JBL)=ZCOF(JBL)*B**L
   25 JBL=JBL+1
      IF(JBL .LT. MD) GO TO 30
C
C     THIS DO LOOP FINDS THE MAXIMUM VALUE OF ZCF.
      ZMAX=DABS(ZCF(1))
      DO 35 I=2,MD
      ZC=DABS(ZCF(I))
      IF(ZC .GT. ZMAX) ZMAX=ZC
   35 CONTINUE
C
C     THIS DO LOOP SETS SMALL VALUES OF THE DISPLACEMENT FIELD
C     COEFFICIENTS ZCOF EQUAL TO ZERO.
      DO 40 I=1,MD
      ZC=DABS(ZCF(I))/ZMAX
      IF(ZC .LT. 1.OD-6) ZCOF(I)=O.OD+OO
   40 CONTINUE
      RETURN
      END
End of file
```

```
      SUBROUTINE MATDIM(N)
C
C     ***********************************************************
C     ***********************************************************
C     **                                                       **
C     **       IMPORTANT VARIALES IN SUBROUTINE MATDIM         **
C     **                                                       **
C     **    N: ORDER OF THE APPROXIMATING FUNCTIONS FOR        **
C     **       DISPLACEMENT FIELDS U AND V.                    **
C     **                                                       **
C     **    NN: ORDER OF X IN THE APPROXIMATING FUNCTIONS.     **
C     **                                                       **
C     **    MM: ORDER OF Y IN THE APPROXIMATING FUNCTIONS.     **
C     **                                                       **
C     **    KK: ORDER OF X IN THE VARIATION OF THE             **
C     **        APPROXIMATING FUNCTIONS.                       **
C     **                                                       **
C     **    LL: ORDER OF Y IN THE VARIATION OF THE             **
C     **        APPROXIMATING FUNCTIONS.                       **
C     **                                                       **
C     **    AMOD: NONDIMENSIONAL INTEGRALS.                    **
C     **                                                       **
C     ***********************************************************
C     ***********************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /SUB1/ NN,MM,KK,LL
      COMMON /SUB2/ XP,YP,XP1,YP1
      COMMON /SUB3/ IGO
      COMMON /SUB4/ AMOD
      DIMENSION AMOD(7,1230),XP1(1230),YP1(1230),XP(730),YP(730)
C
C     SUBROUTINE MATDIM CALCULATES THE NONDIMENSIONAL
C     INTEGRATIONS OF THE VARIATIONAL EQUATION FOR U AND V.
C     THE NONDIMENSIONAL INTEGRALS ARE FORMED IN THE FOLLOWING MANNER:
C     STARTING WITH A(1,O) MULTIPLIED BY THE VARIATION OF A(1,O). THEN
C     A(2,O) TIMES THE VARIATION OF A(1,O) TO A(N,O). THEN A(1,1) TO A(N-1,1)
C     ARE MULTIPLIED BY THE VARIATION OF A(1,O). THE LAST INTEGRALS TO BE
C     FORMED INVOLVE POWERS OF Y ONLYTIMES THE VARIATION OF A(1,O). THE
C     ABOVE PROCESS IS REPEATED FOR THE VARIATION OF A(2,O) TO THE
C     VARIATION OF A(O,N). THE ORDER OF THE VARIATIONS IS THE SAME AS
C     THAT USED WHEN THE COEFFICIENTS ARE MULTIPLIED BY THE VARIATION
C     OF A(1,O). THE NONDIMENSIONAL INTEGRALS OF B(NN,MM) TIMES
C     THE VARITION OF B(KK,LL) ARE FORMED IN THE SAME MANNER.
C
C     NOTE: THE FIRST IN ( ) CORRESPONDS TO THE POWER OF X ASSOCIATED
C           WITH EACH COEFFICIENT A AND B, AND THE SECOND NUMBER IN
C           ( ) CORRESPONDS TO THE POWER OF Y ASSOCIATED WITH EACH
C           COEFFICIENT A AND B.
C
C     INITIALIZE NONDIMENSIONAL INTEGRALS TO ZERO.
      DO 3 I=1,7
      DO 2 IJ=1,730
    2 AMOD(I,IJ)=0.0D+00
    3 CONTINUE
      J=0
      JL=0
      DO 20 L1=1,N
      LL=L1-1
      DO 30 KK=1,N
```

```
      IGO=0
      IF(KK+LL .GT. N) GO TO 20
      DO 40 M1=1,N
      MM=M1-1
      DO 50 NN=1,N
      IF(NN+MM .GT. N) GO TO 40
      CALL CMOD(N,J,JL)
   50 CONTINUE
   40 CONTINUE
      NN=0
      DO 60 MM=1,N
      CALL CMOD(N,J,JL)
   60 CONTINUE
   30 CONTINUE
   20 CONTINUE
      KK=0
      DO 70 LL=1,N
      IGO=0
      DO 80 M1=1,N
      MM=M1-1
      DO 90 NN=1,N
      IF(NN+MM .GT. N) GO TO 80
      CALL CMOD(N,J,JL)
   90 CONTINUE
   80 CONTINUE
      NN=0
      DO 95 MM=1,N
      CALL CMOD(N,J,JL)
   95 CONTINUE
   70 CONTINUE
      RETURN
      END
End of file
```

```
      SUBROUTINE CMOD(N,J,JL)
C
C     CMOD IS USED AS A BRIDGE SUBROUTINE BETWEEN SUBROUTINES MATDIM AND
C     AMOD1, AMOD2, AMOD3, AMOD4, AMOD5, AMOD6, AND AMOD7. CMOD ALSO
C     CALCULATES THE APPROPIATE POWERS OF X AND Y USED IN SUBROUTINE
C     EXPAND TO DIMENSIONALIZE THE NONDIMENSIONAL INTEGRALS.
C
C     **********************************************************
C     **********************************************************
C     **                                                      **
C     **      IMPORTANT VARIABLES IN SUBROUTINE CMOD          **
C     **                                                      **
C     **    IGO: O- DIAGONAL ELEMENT NOT ENCOUNTERED          **
C     **         1- DIAGONAL ELEMENT ENCOUNTERED              **
C     **                                                      **
C     **    AMOD: NONDIMENSIONAL INTEGRALS.                   **
C     **                                                      **
C     **********************************************************
C     **********************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /SUB1/ NN,MM,KK,LL
      COMMON /SUB2/ XP,YP,XP1,YP1
      COMMON /SUB3/ IGO
      COMMON /SUB4/ AMOD
      DIMENSION XP(730),YP(730),XP1(1230),YP1(1230),AMOD(7,1230)
C
C     AMOD(1,J),AMOD(2,J) IS USED TO CALCULATE K1
C     AMOD(3,J), AMOD(4,J), AND AMOD(5,J) IS USED TO
C     CALCULATE K2, AMOD(6,J) AND AMOD(7,J) IS USED
C     TO CALCULATE K4, IN SUBROUTINE EXPAND.
      IF(IGO .EQ. 1) GO TO 10
      IF(NN .EQ. KK .AND. MM .EQ. LL ) GO TO 10
      GO TO 15
   10 IGO=1
      JL=JL+1
      XP(JL)=NN+KK
      YP(JL)=MM+LL
      CALL AMOD1(JL)
      CALL AMOD2(JL)
      CALL AMOD6(JL)
      CALL AMOD7(N,JL)
   15 J=J+1
      XP1(J)=NN+KK
      YP1(J)=MM+LL
      CALL AMOD3(J)
      CALL AMOD4(J)
      CALL AMOD5(N,J)
      RETURN
      END
```
End of file

```
      SUBROUTINE AMOD1(J)
C
C     AMOD1 CALCULATES THE NONDIMENSIONAL INTEGRALS THAT ARE MULTIPLIED
C     BY (EH)/(1-NU**2)*(B/L) IN SUBMATRIX K1, WHICH IS CALCULATED IN
C     SUBROUTINE EXPAND.
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /SUB1/ NN,MM,KK,LL
      COMMON /SUB4/ AMOD
      DIMENSION AMOD(7,1230)
      IF(KK+NN .EQ. 1) GO TO 20
      AMOD(1,J)=-DFLOAT(NN*KK)/DFLOAT((NN+KK-1)*(MM+LL+1))
   20 RETURN
      END
End of file
```

```
      SUBROUTINE AMOD2(J)
C
C     AMOD2 CALCULATES THE NONDIMENSIONAL INTEGRALS THAT ARE MULTIPLIED
C     BY (EH)/(1+NU)*(L/B) IN SUBMATRIX K1, WHICH IS CALCULATED IN
C     SUBROUTINE EXPAND.
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /SUB1/ NN,MM,KK,LL
      COMMON /SUB4/ AMOD
      DIMENSION AMOD(7,1230)
      IF(MM+LL .EQ. 1) GO TO  20
      AMOD(2,J)=-DFLOAT(MM*LL)/DFLOAT((NN+KK+1)*(MM+LL-1))
      AMOD(2,J)=AMOD(2,J)/(2.0D+00)
   20 RETURN
      END
End of file
```

```
      SUBROUTINE AMOD3(J)
C
C     AMOD3 CALCULATES THE NONDIMENSIONAL INTEGRALS THAT ARE MULTIPLIED
C     BY (EH)/(1-NU) IN SUBMATRIX K2, WHICH IS CALCULATED IN SUBROUTINE
C     EXPAND
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /SUB1/ NN,MM,KK,LL
      COMMON /SUB4/ AMOD
      DIMENSION AMOD(7,1230)
      IF(NN .EQ. 0) GO TO 20
      IF(MM .EQ. 0) GO TO 20
      D1=DFLOAT(NN)/DFLOAT(NN+KK)
      D2=DFLOAT(MM)/DFLOAT(MM+LL)
      AMOD(3,J)=(D1*D2)/(2.0D+00)
   20 RETURN
      END
End of file
```

```
      SUBROUTINE AMOD4(J)
C
C     AMOD4 CALCULATES THE NONDIMENSIONAL INTEGRALS THAT ARE MULTIPLIED
C     BY ((EH)NU)/(1-NU**2) IN SUBMATRIX K2, WHICH IS CALCULATED IN
C     SUBROUTINE EXPAND.
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /SUB1/ NN,MM,KK,LL
      COMMON /SUB4/ AMOD
      DIMENSION AMOD(7,1230)
      IF(NN+KK .EQ. 0) GO TO 20
      IF(MM .EQ. 0) GO TO 20
      AMOD(4,J)=DFLOAT(MM)/DFLOAT(MM+LL)
   20 RETURN
      END
End of file
```

```
      SUBROUTINE AMOD5(N,J)
C
C     AMOD5 CALCULATES THE NONDIMENSIONAL INTEGRALS THAT ARE MULTIPLIED
C     BY (EH)/(1+NU) IN SUBMATRIX K2, WHICH IS CALCULATED IN SUBROUTINE
C     EXPAND
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /SUB1/ NN,MM,KK,LL
      COMMON /SUB4/ AMOD
      DIMENSION AMOD(7,1230)
      IF(MM+LL .EQ. 0) GO TO 20
      IF(MM .EQ. 0) GO TO 10
      IF(NN .EQ. 0) GO TO 20
      AMOD(5,J)=(DFLOAT(NN)/DFLOAT(2*(NN+KK)))
      GO TO 20
   10 D1=DFLOAT(NN)/DFLOAT(NN+KK)
      D2=DFLOAT(N)/DFLOAT(KK+N)
      AMOD(5,J)=(D1-D2)/(2.0D+00)
   20 RETURN
      END
End of file
```

```
      SUBROUTINE AMOD6(J)
C
C     AMOD6 CALCULATES THE NONDIMENSIONAL INTEGRALS THAT ARE MULTIPLIED
C     BY (EH)/(1-NU**2)*(L/B) IN SUBMATRIX K4, WHICH IS CALCULATED IN
C     SUBROUTINE EXPAND.
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /SUB1/ NN,MM,KK,LL
      COMMON /SUB4/ AMOD
      DIMENSION AMOD(7,1230)
      IF(MM+LL .EQ. 1) GO TO 20
      AMOD(6,J)=-DFLOAT(MM*LL)/DFLOAT((NN+KK+1)*(MM+LL-1))
   20 RETURN
      END
End of file
```

```
      SUBROUTINE AMOD7(N,J)
C
C     AMOD7 CALCULATES THE NONDIMENSIONAL INTEGRALS THAT ARE MULTIPLIED
C     BY (EH)/(1+NU)*(B/L) IN SUBMATRIX K4, WHICH IS CALCULATED IN
C     SUBROUTINE EXPAND.
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /SUB1/ NN,MM,KK,LL
      COMMON /SUB4/ AMOD
      DIMENSION AMOD(7,1230)
      IF(LL .EQ. 0) GO TO 10
      IF(MM .EQ. 0) GO TO 15
      IF(NN+KK .EQ. 1) GO TO 20
      AMOD(7,J)=-DFLOAT(NN*KK)/DFLOAT((NN+KK-1)*(MM+LL+1))
      AMOD(7,J)=AMOD(7,J)/(2.0D+00)
      GO TO 20
   10 IF(MM .EQ. 0) GO TO 30
      IF(NN+KK .EQ. 1) GO TO 20
      D1=DFLOAT(NN*(NN-1))/DFLOAT((NN+KK-1)*(MM+1))
      D2=DFLOAT(NN*(NN-1))/DFLOAT((NN+N-1)*(MM+1))
      AMOD(7,J)=(D1-D2)/(2.0D+00)
      GO TO 20
   15 D1=0.0D+00
      IF(NN+KK .EQ. 1) GO TO 25
      D1=DFLOAT(NN*(NN-1))/DFLOAT((NN+KK-1)*(LL+1))
   25 D2=DFLOAT(N*(N-1))/DFLOAT((KK+N-1)*(LL+1))
      D3=DFLOAT(NN-N)/DFLOAT(LL+1)
      AMOD(7,J)=(D1-D2-D3)/(2.0D+00)
      GO TO 20
   30 D1=0.0D+00
      D2=0.0D+00
      IF(NN+KK .EQ. 1) GO TO 40
      D1=DFLOAT(NN*(NN-1))/DFLOAT(NN+KK-1)
      D2=DFLOAT(NN*(NN-1))/DFLOAT(NN+N-1)
   40 D3=DFLOAT(N*(N-1))/DFLOAT(KK+N-1)
      D4=DFLOAT(N*(N-1))/DFLOAT(2*N-1)
      AMOD(7,J)=(D1-D2-D3+D4)/(2.0D+00)
   20 RETURN
      END
End of file
```

```
C NAASA  2.1.001 DLUD      FTN-A 10-29-75      THE UNIV OF MICH COMP CTR
      SUBROUTINE DLUD (N,ADIM,A,TDIM,T,IV)
C
C COMPUTES THE LU-DECOMPOSITION OF THE N X N MATRIX A USING
C GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING. THIS FACTOR-
C IZATION MAY BE EXPRESSED IN THE FORM
C           L(N-1)*P(N-1)*...*L(1)*P(1)*A = U,
C WHERE EACH L(J) IS THE IDENTITY MATRIX EXCEPT FOR THE SUB-
C DIAGONAL ELEMENTS IN COLUMN J, EACH P(J) IS A PERMUTATION
C MATRIX, AND U IS AN UPPER TRIANGULAR MATRIX.  THIS IS THE
C PREPARATORY STEP IN SOLVING A SYSTEM OF LINEAR EQUATIONS,
C INVERTING A MATRIX, OR CALCULATING A DETERMINANT. A DISCUSSION
C OF GAUSSIAN ELIMINATION AND THE LU-DECOMPOSITION AND THEIR
C RELATIONSHIP TO THE NUMERICAL SOLUTION OF SYSTEMS OF LINEAR
C EQUATIONS MAY BE FOUND IN EITHER WILKINSON (1965,CHAPTER 4)
C OR FORSYTHE AND MOLER (1967).
C
      INTEGER N,ADIM,TDIM,IV(1)
      DOUBLE PRECISION A(ADIM,N),T(TDIM,N)
C
C   N  -> ORDER OF THE MATRIX A.
C ADIM -> ROW DIMENSION OF THE ARRAY A. BECAUSE A IS AN N X N
C         MATRIX, ADIM SHOULD NOT BE LESS THAN N.  IF ADIM IS LESS
C         THAN N, THE CONTENTS OF A ARE IGNORED, AND THE MATRIX
C         TO BE FACTORED IS ASSUMED TO BE STORED IN THE ARRAY T.
C         SINCE ADIM MUST BE A POSITIVE INTEGER, IT IS RECOMMENDED
C         THAT THE ACTUAL ARGUMENTS A AND T COINCIDE WHEN ADIM IS
C         LESS THAN N TO AVOID THE INCONSISTENCY WHICH ARISES WHEN
C         N EQUALS 1.
C   A  -> TWO-DIMENSIONAL ARRAY CONTAINING THE N X N MATRIX TO
C         BE FACTORED, I.E., THE COEFFICIENT MATRIX OF THE SYSTEM
C         OF LINEAR EQUATIONS OR THE MATRIX TO BE INVERTED. THE
C         CONTENTS OF A ARE NOT ALTERED.
C TDIM -> ROW DIMENSION OF THE ARRAY T.
C   T  <- TWO-DIMENSIONAL ARRAY FOR RETURNING THE LU-DECOMPOSITION
C         OF A. THE SUBDIAGONAL ELEMENTS OF THE J-TH COLUMN OF THE
C         L(J) AND THE UPPER TRIANGULAR MATRIX U ARE RETURNED IN
C         THE CORRESPONDING ELEMENTS OF T.  IF ADIM IS LESS THAN N,
C         T MUST CONTAIN THE MATRIX TO BE FACTORED WHEN THIS SUB-
C         ROUTINE IS CALLED.
C  IV  <- VECTOR OF LENGTH N DEFINING THE PERMUTATION MATRICES
C         P(J): MULTIPLICATION ON THE LEFT BY P(J) INTERCHANGES
C         ROWS J AND IV(J).  IF IV(J) IS NOT EQUAL TO J, THEN
C         DET(A) = - DET(P(J)*A).  AND TO AID IN THE COMPUTATION
C         OF DET(A), IV(N) WILL CONTAIN +1 IF AN EVEN NUMBER OF
C         INTERCHANGES ARE PERFORMED AND -1 IF AN ODD NUMBER. THUS
C              DET(A) = IV(N)*T(1,1)* ... *T(N,N).
C         IV(N) WILL CONTAIN 0 IF A IS COMPUTATIONALLY SINGULAR.
C
      INTEGER I,J,K,KP1,L
      DOUBLE PRECISION TMP
      DOUBLE PRECISION PIV
C
C IF ADIM IS GREATER THAN OR EQUAL TO N, THE CONTENTS OF A ARE MOVED
C TO T; WHILE IF ADIM IS LESS THAN N, THIS INITIAL DATA MOVEMENT IS
C SKIPPED.  SINCE THE DATA MOVEMENT TIME IS PROPORTIONAL TO N**2 AND
C THE COMPUTATION TIME PROPORTIONAL TO N**3/3, SIGNIFICANT SAVINGS
C SHOULD NOT BE EXPECTED.
C
      IF ( ADIM .LT. N ) GO TO 8110
```

```
      DO 8100 J = 1, N
          DO 8100 I = 1, N
 8100         T(I,J) = A(I,J)
 8110 CONTINUE
C
C GAUSSIAN ELIMINATION CONSISTS OF N-1 STAGES. DURING THE K-TH
C STAGE, THE PERMUTATION MATRIX P(K), THE LOWER TRIANGULAR MATRIX
C L(K), AND THE K-TH ROW OF U ARE COMPUTED BASED ON THE CURRENT
C ELEMENTS OF THE K-TH RESIDUAL MATRIX, I.E., THE ELEMENTS T(I,J),
C I,J=K...N. ONLY THE ELEMENTS OF THE K-TH RESIDUAL MATRIX ARE
C REFERENCED DURING THE K-TH STAGE OF GAUSSIAN ELIMINATION.
C
      IV(N) = 1
      DO 8260 K = 1, N
          PIV = DABS(T(K,K))
          IF ( K .GE. N ) GO TO 8260
C
C SELECT THE PIVOT ROW FOR THE K-TH STAGE BY PARTIAL PIVOTING,
C I.E., THE MAXIMUM ELEMENT IN THE 1-ST COLUMN OF THE K-TH RESIDUAL
C MATRIX, AND SET IV(K) ACCORDINGLY. THE VARIABLE L HOLDS THE
C SUBSCRIPT OF THE PIVOT ROW, AND PIV THE ABSOLUTE VALUE OF THE
C PIVOT ELEMENT.
C
          L = K
          KP1 = K + 1
          DO 8200 I = KP1, N
              IF ( PIV .GE. DABS(T(I,K)) ) GO TO 8200
                PIV = DABS(T(I,K))
                L = I
 8200     CONTINUE
          IV(K) = L
C
C SAVE THE PIVOT ELEMENT IN TMP.  IF P(K) IS NONTRIVIAL, I.E., IV(K)
C IS NOT EQUAL TO K, THE PIVOT ELEMENT IS ALWAYS NONZERO; OTHERWISE,
C THE PIVOT ELEMENT MUST BE CHECKED.  IF THE PIVOT IS ZERO, I.E.,
C THE MATRIX IS COMPUTATIONALLY SINGULAR, THEN T(I,K) IS ZERO FOR
C I=K...N, AND THE COMPUTATION MAY PROCEED TO THE NEXT STAGE.
C
          TMP = T(L,K)
          IF ( K .NE. L ) GO TO 8210
            IF ( PIV .GT. O.DO ) GO TO 8220
              IV(N) = O
              GO TO 8260
 8210     CONTINUE
            IV(N) = -IV(N)
            T(L,K) = T(K,K)
            T(K,K) = TMP
 8220     CONTINUE
C
C COMPUTE THE NONTRIVIAL ELEMENTS OF L(K).  BECAUSE OF THE PARTIAL
C PIVOTAL STRATEGY, THE ABSOLUTE VALUE OF L(I,K)=-T(I,K)/T(K,K) IS
C LESS THAN OR EQUAL TO 1. IF T(K) DENOTES THE K-TH RESIDUAL MATRIX,
C THEN THE SUBDIAGONAL ELEMENTS OF THE 1-ST COLUMN OF THE MATRIX
C L(K) P(K) T(K) ARE ZERO. THESE ELEMENTS ARE NOT ACTUALLY CALCU-
C LATED, THEY ARE REPLACED BY THE ELEMENTS OF L(K).
C
          TMP = -TMP
          DO 8230 I = KP1, N
 8230         T(I,K) = T(I,K) / TMP
```

```
C APPLY P(K) AND L(K) TO THE K-TH RESIDUAL MATRIX COLUMNWISE, I.E.,
C FOR J=K+1...N, INTERCHANGE T(K,J) AND T(L,J), THE (K,J)-ELEMENT
C OF U, AND THEN FOR I=K+1...N, REPLACE T(I,J) BY
C                T(I,J) + L(K)(I,K) * T(K,J).
C
          DO 8250 J = KP1, N
             TMP = T(L,J)
             T(L,J) = T(K,J)
             T(K,J) = TMP
             DO 8240 I = KP1, N
 8240           T(I,J) = T(I,J) + T(I,K) * TMP
 8250     CONTINUE
 8260 CONTINUE
      IF (PIV .EQ. O.DO) IV(N) = O
      RETURN
C
C FORSYTHE,G.E. AND MOLER,C.B. 1967. COMPUTER SOLUTION OF LINEAR
C   ALGEBRAIC SYSTEMS. ENGLEWOOD CLIFFS,N.J.: PRENTICE-HALL.
C WILKINSON,J.H. 1965. THE ALGEBRAIC EIGENVALUE PROBLEM. OXFORD:
C   CLARENDON PRESS.
C                       THE UNIVERSITY OF MICHIGAN COMPUTING CENTER
C                       NUMERICAL ANALYSIS LIBRARY - JULY 1975
      END
End of file
```

```
C NAASA  2.1.003 DBS      FIN-A 10-29-75      THE UNIV OF MICH COMP CTR
      SUBROUTINE DBS (N,TDIM,T,IV,B)
C
C SOLVES THE SYSTEM OF LINEAR EQUATIONS AX=B, WHERE A DENOTES
C THE N X N COEFFICIENT MATRIX AND X AND B ARE N-VECTORS, BY
C BACK-SUBSTITUTION IN THE LU-DECOMPOSITION OF A. THIS
C DECOMPOSITION MUST BE PROVIDED IN THE ARRAY T AND VECTOR IV
C VIS-A-VIS THE SUBROUTINE DLUD. THE LU-DECOMPOSITION MAY BE
C EXPRESSED IN THE FORM
C               L(N-1)*P(N-1)*...*L(1)*P(1)*A = U,
C WHERE EACH L(J) IS THE IDENTITY MATRIX EXCEPT FOR THE SUB-
C DIAGONAL ELEMENTS IN COLUMN J, EACH P(J) IS A PERMUTATION
C MATRIX, AND U IS AN UPPER TRIANGULAR MATRIX. USING THIS
C NOTATION, THE BACK-SUBSTITUTION CONSISTS OF FORMING
C               Y = L(N-1)*P(N-1)*...*L(1)*P(1)*B
C AND SOLVING THE UPPER TRIANGULAR SYSTEM OF LINEAR EQUATIONS
C UX = Y, I.E., FOR I=N...1
C   X(I)=(Y(I)-U(I,N)*X(N)-...-U(I,I+1)*X(I+1))/U(I,I).
C THIS BACK-SUBSTITUTION YIELDS A VECTOR X WHICH IS THE EXACT
C SOLUTION OF A SYSTEM OF LINEAR EQUATIONS (A+E)X=B, WHERE
C //E// IS GENERALLY ON THE ORDER OF N*//A//*MACHEPS. THIS METHOD
C OF SOLVING SYSTEMS OF LINEAR EQUATIONS IS DESCRIBED IN BOTH
C WILKINSON (1965,CHAPTER 4) AND FORSYTHE AND MOLER (1967).
C
      INTEGER N,TDIM,IV(1)
      DOUBLE PRECISION T(TDIM,N),B(1)
C
C   N  -> ORDER OF THE SYSTEM OF LINEAR EQUATIONS.
C TDIM -> ROW DIMENSION OF THE ARRAY T.
C   T  -> TWO-DIMENSIONAL ARRAY CONTAINING THE LU-DECOMPOSITION
C         OF THE COEFFICIENT MATRIX VIS-A-VIS THE SUBROUTINE DLUD.
C  IV  -> VECTOR CONTAINING THE INTERCHANGE INFORMATION GENERATED
C         BY THE SUBROUTINE DLUD DURING THE COMPUTATION OF THE
C         LU-DECOMPOSITION OF THE COEFFICIENT MATRIX.
C   B  -- VECTOR CONTAINING THE RIGHT-HAND SIDE OF THE SYSTEM OF
C         LINEAR EQUATIONS. THE CONTENTS OF B ARE REPLACED BY THE
C         ELEMENTS OF THE SOLUTION.
C
      INTEGER I,K,KP1,L
      DOUBLE PRECISION TMP
C
C REPLACE THE CONTENTS OF THE VECTOR B BY THE VECTOR
C        ( L(N-1) (P(N-1) ... ( L(1) (P(1) B)...).
C THIS COMPUTATION IS PERFORMED IN N-1 STAGES, WHERE DURING THE
C K-TH STAGE, THE CONTENTS OF B ARE REPLACED BY L(K)( P(K) B ).
C THE PERMUTATION MATRICES P(K) ARE DEFINED BY THE VECTOR IV, I.E.,
C MULTIPLICATION BY P(K) SIMPLY INTERCHANGES THE K-TH AND IV(K)-TH
C ELEMENTS OF THE VECTOR.  THE LOWER TRIANGULAR MATRIX L(K) IS THE
C IDENTITY MATRIX EXCEPT FOR THE SUBDIAGONAL ELEMENTS OF THE K-TH
C COLUMN, WHICH ARE STORED IN THE CORRESPONDING ELEMENTS OF THE
C ARRAY T.
C
      DO 8110 K = 1, N
         IF ( K .GE. N ) GO TO 8110
         L = IV(K)
         TMP = B(L)
         B(L) = B(K)
         B(K) = TMP
         KP1 = K + 1
         DO 8100 I = KP1, N
```

```
 8100         B(I) = B(I) + T(I,K) * TMP
 8110 CONTINUE
C
C REPLACE THE CONTENTS OF THE VECTOR B BY THE SOLUTION TO THE SYSTEM
C OF LINEAR EQUATIONS WITH UPPER TRIANGULAR COEFFICIENT MATRIX U
C AND RIGHT-HAND SIDE VECTOR B.  THE USUAL FORMULAS FOR THE BACK-
C SUBSTITUTION, WHICH ARE BASED ON THE SUCCESSIVE ROWS OF THE MATRIX
C AND ARE SUITABLE WHEN INNER-PRODUCTS ARE ACCUMULATED, ARE NOT
C EMPLOYED.  THE COMPUTATION HAS INSTEAD BEEN ARRANGED TO REFERENCE
C THE SUCCESSIVE COLUMNS OF U.  THUS AFTER B(I) HAS BEEN COMPUTED,
C IT IS REMOVED FROM THE SYSTEM BY SUBTRACTING B(I) TIMES THE I-TH
C COLUMN OF U FROM THE RESIDUAL VECTOR B(1)...B(I-1).
C
      K = N
 8200     B(K) = B(K) / T(K,K)
          IF ( K .LE. 1 ) RETURN
          TMP = -B(K)
          KP1 = K
          K = K - 1
          DO 8210 I = 1, K
 8210         B(I) = B(I) + T(I,KP1) * TMP
          GO TO 8200
C
C FORSYTHE,G.E. AND MOLER,C.B. 1967. COMPUTER SOLUTION OF LINEAR
C   ALGEBRAIC SYSTEMS. ENGLEWOOD CLIFFS,N.J.: PRENTICE-HALL.
C WILKINSON,J.H. 1965. THE ALGEBRAIC EIGENVALUE PROBLEM. OXFORD:
C   CLARENDON PRESS.
C                         THE UNIVERSITY OF MICHIGAN COMPUTING CENTER
C                         NUMERICAL ANALYSIS LIBRARY - JULY 1975
      END
End of file
```

```
      SUBROUTINE SCOFF(EL,B,N,MSIZE,COFF,E,UN,AX,ITITL)
C
C     SCOFF CALCULATES THE STRESS FIELD COEFFICIENTS FOR SIGMA-X, AX(J,1),
C     SIGMA-Y, AX(J,2), AND TAU-XY,AX(J,3) USING THE DISPLACEMENT FIELD
C     COEFFICIENTS FOUND IN SUBROUTINE EXPAND AND HOOKE'S LAW.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION COFF(75),AX(35,3),DUX(35),DUY(35),DVX(35),DVY(35)
      DIMENSION ITITL(50)
      J=1
C
C     THIS SET OF DO LOOPS CALCULATES THE DERIVATIVES OF THE THE APPROXIMATING
C     FUNCTIONS FOR THE DISPLACEMENT FIELDS U AND V WITH RESPECT TO X AND
C     WITH RESPECT TO Y.
      DO 10 M1=1,N
      MM=M1-1
      DO 20 NN=1,N
      IF(NN+MM .GT. N) GO TO 10
      J1=MSIZE+J
C
C     DUX IS THE DERIVATIVE OF THE APPROXIMATING FUNCTION FOR DISPLACEMENT
C     FIELD, U, WITH RESPECT TO X.
C     DUY IS THE DERIVATIVE OF THE APPROXIMATING FUNCTION FOR DISPLACEMENT
C     FIELD, U, WITH RESPECT TO Y.
C     DVX IS THE DERIVATIVE OF THE APPROXIMATING FUNCTION FOR DISPLACEMENT
C     FIELD, V, WITH RESPECT TO X.
C     DVY IS THE DERIVATIVE OF THE APPROXIMATING FUNCTION FOR DISPLACEMENT
C     FIELD, V, WITH RESPECT TO Y.
      DUX(J)=DFLOAT(NN)*COFF(J)
      DUY(J)=DFLOAT(MM)*COFF(J)
      DVX(J)=DFLOAT(NN)*COFF(J1)
      DVY(J)=DFLOAT(MM)*COFF(J1)
   20 J=J+1
   10 CONTINUE
      NN=0
      DO 30 MM=1,N
      J1=MSIZE+J
      DUX(J)=DFLOAT(NN)*COFF(J)
      DUY(J)=DFLOAT(MM)*COFF(J)
      DVX(J)=DFLOAT(NN)*COFF(J1)
      DVY(J)=DFLOAT(MM)*COFF(J1)
   30 J=J+1
      J1=MSIZE-N+1
      E1=E/((1.0D+00)-UN**2)
      G=E/((2.0D+00)*((1.0D+00)+UN))
C
C     THESE STATEMENTS CALCULATE THE CONSTANT TERMS FOR THE STRESS
C     FIELD COEFFICIENTS.
      AX(1,1)=E1*(DUX(1)+UN*DVY(J1))
      AX(1,2)=E1*(UN*DUX(1)+DVY(J1))
      AX(1,3)=G*(DUY(J1)+DVX(1))
      NN=2
      N1=N
      IF=0
      J=2
C
C     THIS PART OF SCOFF CALCULATES THE VARIOUS TERMS OF THE STRESS
C     FIELDS THAT ARE ASSOCIATED WITH VARIOUS POWERS OF X AND Y.
   40 IF(NN .GT. N1) GO TO 50
      J1=J
      J2=N+J-1
```

```
      IF(IF .GE. 1) J1=J1+IF
      AX(J,1)=E1*(DUX(J1)+UN*DVY(J2))
      AX(J,2)=E1*(UN*DUX(J1)+DVY(J2))
      AX(J,3)=G*(DUY(J2)+DVX(J1))
      J=J+1
      NN=NN+1
      GO TO 40
   50 NN=2
      IF=IF+1
      N1=N1-1
      IF(NN .GT. N1) GO TO 60
      GO TO 40
   60 J1=N
      IF=1
      IF1=1
   65 J1=J1+IF
      IF(J1 .GE. MSIZE-N+1) GO TO 70
      J2=N+J
      AX(J,1)=E1*(DUX(J1)+UN*DVY(J2))
      AX(J,2)=E1*(UN*DUX(J1)+DVY(J2))
      AX(J,3)=G*(DUY(J2)+DVX(J1))
      J=J+1
      IF=N-IF1
      IF1=IF1+1
      GO TO 65
   70 J=J-1
C
C     ZEROS SETS STRESS FIELD COEFFICIENTS THAT ARE SMALL IN RELATIVE
C     MAGNITUDE EQUAL TO ZERO.
      CALL ZEROS(EL,B,N,AX)
C
C     THIS PART OF SCOFF PRINTS THE STRESS COEFFICIENTS TO THE OUTFILE.
C     JJ IS THE POWER OF X ASSOCIATED WITH EACH STRESS FIELD COEFFICIENT
C     AND JM IS THE POWER OF Y ASSOCIATED WITH EACH STRESS FIELD
C     COEFFICIENT.
      WRITE(8,500) ITITL
      WRITE(8,100)
      DO 80 I=1,3
      IF(I .NE. 1) WRITE(8,500) ITITL
      IF(I .EQ. 2) WRITE(8,101)
      IF(I .EQ. 3) WRITE(8,102)
  100 FORMAT(1H-,3X,'COEFFICIENTS OF NORMAL STRESS IN X-DIRECTION'
     $)
  101 FORMAT(1H-,3X,'COEFFICIENTS OF NORMAL STRESS IN Y-DIRECTION'
     $)
  102 FORMAT(1H-,3X,'COEFFICIENTS OF SHEAR STRESS')
  500 FORMAT(1H1,2(/,1H-),40X,50A1)
      J=1
      JM=0
      N1=N-1
      DO 90 J1=1,N
      JJ=J1-1
      WRITE(8,103) JJ,JM,AX(J,I)
   90 J=J+1
      DO 91 JM=1,N1
      DO 92 JJ=1,N1
      IF(JJ+JM .GT. N1) GO TO 91
      WRITE(8,103) JJ,JM,AX(J,I)
   92 J=J+1
```

```
      JJ=O
      DO 93 JM=1,N1
      WRITE(8,103) JJ,JM,AX(J,I)
   93 J=J+1
   80 CONTINUE
  103 FORMAT(6X,'AX(',I1,',',I1,')=',D12.5)
      RETURN
      END
End of file
```

```
      SUBROUTINE ZEROS(EL,B,N,AX)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AXC(35,3),AX(35,3)
      N1=N-1
      DO 10 K=1,3
      J=1
      DO 15 II=1,N
      I=II-1
      AXC(J,K)=AX(J,K)*EL**I
   15 J=J+1
      DO 20 L=1,N1
      DO 25 I=1,N1
      IF(I+L .GT. N1) GO TO 20
      AXC(J,K)=AX(J,K)*EL**I*B**L
   25 J=J+1
   20 CONTINUE
      DO 30 L=1,N1
      AXC(J,K)=AX(J,K)*B**L
   30 J=J+1
      J=J-1
      AXMAX=DABS(AXC(1,K))
      DO 35 I=2,J
      AC=DABS(AXC(I,K))
      IF(AC .GT. AXMAX) AXMAX=AC
   35 CONTINUE
      DO 40 I=1,J
      AC=DABS(AXC(I,K))
      AB=AC/AXMAX
      IF(AB .LT. 1.0D-06) AX(I,K)=0.0D+00
   40 CONTINUE
   10 CONTINUE
      RETURN
      END
End of file
```

```
      SUBROUTINE DOUT(N,MSIZE,EL,B,COFF,ITITL)
C
C     DOUT PRODUCES TABLES OF THE DISPLACEMENT FIELDS U AND V AT VARIOUS
C     POINTS ON THE BAND SAW BLADE USING THE DISPLACEMENT FIELD COEFFICIENTS
C     FOUND IN SUBROUTINE EXPAND.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION DISV(6,11),COFF(75),DISU(6,11),X1(11),Y1(6)
      DIMENSION ITITL(50)
      Y2=0.0D+00
      DELX=EL/(1.0D+01)
      DELY=B/(5.0D+00)
      DO 20 I=1,6
      X2=0.0D+00
      DO 30 J=1,11
      J1=1
      J2=1+MSIZE
      DISU(I,J)=0.0D+00
      DISV(I,J)=0.0D+00
      DO 40 K=1,N
      DISU(I,J)=DISU(I,J)+COFF(J1)*X2**K
      DISV(I,J)=DISV(I,J)+COFF(J2)*X2**K
      J2=J2+1
   40 J1=J1+1
      DO 50 L=1,N
      DO 60 K=1,N
      IF(K+L .GT. N) GO TO 50
      DISU(I,J)=DISU(I,J)+COFF(J1)*X2**K*Y2**L
      DISV(I,J)=DISV(I,J)+COFF(J2)*X2**K*Y2**L
      J2=J2+1
   60 J1=J1+1
   50 CONTINUE
      DO 70 L=1,N
      DISU(I,J)=DISU(I,J)+COFF(J1)*Y2**L
      DISV(I,J)=DISV(I,J)+COFF(J2)*Y2**L
      J2=J2+1
   70 J1=J1+1
      X1(J)=X2
   30 X2=X2+DELX
      Y1(I)=Y2
   20 Y2=Y2+DELY
      WRITE(8,500) ITITL
      WRITE(8,100)
  100 FORMAT(1H-,56X,'DISPLACEMENT FIELD U')
      DO 90 IJ=1,2
      IF(IJ .EQ. 2) WRITE(8,500) ITITL
      IF(IJ .EQ. 2) WRITE(8,120)
  120 FORMAT(1H-,56X,'DISPLACEMENT FIELD V')
      WRITE(8,107)
  107 FORMAT(1H ,124('_'))
  101 FORMAT(12X,'I',50X,'X-COORDINATE',T125,'I')
  102 FORMAT(12X,'I',T125,'I',/,12X,'I',11(1X,D9.3),T125,'I')
  108 FORMAT(1X,124('-'))
  109 FORMAT(1H ,124('_'))
  500 FORMAT(1H1,2(/,1H-),40X,50A1)
      WRITE(8,105)
  105 FORMAT(5X,'Y',6X,'I',T125,'I')
      WRITE(8,106)
  106 FORMAT(1X,'COORDINATE',1X,'I',T125,'I')
      DO 80 J=1,6
      J1=7-J
```

```
      IF(IJ .EQ. 1) WRITE(8,104) Y1(J1),(DISU(J1,I),I=1,11)
   80 IF(IJ .EQ. 2) WRITE(8,104) Y1(J1),(DISV(J1,I),I=1,11)
      WRITE(8,108)
      WRITE(8,102) (X1(I),I=1,11)
      WRITE(8,101)
      WRITE(8,109)
  104 FORMAT(12X,'I',T125,'I',/,1X,D9.3,2X,'I',11(1X,D9.3),T125,'I')
   90 CONTINUE
      RETURN
      END
End of file
```

```
      SUBROUTINE STROUT(N,EL,B,AX,ITITL)
C
C     STROUT PRODUCES TABLES OF THE STRESS FIELDS SIGMA-X, SIGMA-Y,
C     TAU-XY, AND THE MAXIMUM NPRINCIPAL NORMAL STRESS.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AX(35,3),SIGX(6,11),SIGY(6,11),TAUXY(6,11)
     $,SIGMAX(6,11),X1(11),Y1(6),ITITL(50)
      N1=N-1
      DELX=EL/(1.OD+01)
      DELY=B/(5.OD+00)
      Y2=0.OD+00
      DO 20 I=1,6
      X2=0.OD+00
      DO 30 J=1,11
      J1=2
      SIGX(I,J)=AX(1,1)
      SIGY(I,J)=AX(1,2)
      TAUXY(I,J)=AX(1,3)
      DO 40 K=1,N1
      SIGX(I,J)=SIGX(I,J)+AX(J1,1)*X2**K
      SIGY(I,J)=SIGY(I,J)+AX(J1,2)*X2**K
      TAUXY(I,J)=TAUXY(I,J)+AX(J1,3)*X2**K
   40 J1=J1+1
      DO 50 L=1,N1
      DO 60 K=1,N1
      IF(L+K .GT. N1) GO TO 50
      SIGX(I,J)=SIGX(I,J)+AX(J1,1)*X2**K*Y2**L
      SIGY(I,J)=SIGY(I,J)+AX(J1,2)*X2**K*Y2**L
      TAUXY(I,J)=TAUXY(I,J)+AX(J1,3)*X2**K*Y2**L
   60 J1=J1+1
   50 CONTINUE
      DO 70 L=1,N1
      SIGX(I,J)=SIGX(I,J)+AX(J1,1)*Y2**L
      SIGY(I,J)=SIGY(I,J)+AX(J1,2)*Y2**L
      TAUXY(I,J)=TAUXY(I,J)+AX(J1,3)*Y2**L
   70 J1=J1+1
      X1(J)=X2
      CEN=(SIGX(I,J)+SIGY(I,J))/(2.OD+00)
      SRAD=((SIGX(I,J)-SIGY(I,J))/(2.OD+00))**2+TAUXY(I,J)**2
      RAD=DSQRT(SRAD)
      SIGMAX(I,J)=CEN+RAD
   30 X2=X2+DELX
      Y1(I)=Y2
   20 Y2=Y2+DELY
      DO 90 IJ=1,4
      WRITE(8,500) ITITL
  500 FORMAT(1H1,2(/,1H-),40X,50A1)
      IF(IJ .EQ. 1) WRITE(8,100)
      IF(IJ .EQ. 2) WRITE(8,108)
      IF(IJ .EQ. 3) WRITE(8,109)
      IF(IJ .EQ. 4) WRITE(8,110)
  100 FORMAT(1H-,64X,'SIGMA X')
  108 FORMAT(1H-,64X,'SIGMA Y')
  109 FORMAT(1H-,61X,'SHEAR STRESS')
  110 FORMAT(1H-,48X,'MAXIMUM PRINCIPAL NORMAL STRESS')
      WRITE(8,107)
  107 FORMAT(1H ,124('_'))
  101 FORMAT(12X,'I',50X,'X-COORDINATE',T125,'I')
  102 FORMAT(12X,'I',T125,'I',/,12X,'I',11(1X,D9.3),T125,'I')
  103 FORMAT(1X,124('-'))
```

```
      WRITE(8,104)
  104 FORMAT(5X,'Y',6X,'I',T125,'I')
      WRITE(8,105)
  105 FORMAT(1X,'COORDINATE',1X,'I',T125,'I')
      DO 80 J=1,6
      J1=7-J
      IF(IJ .EQ. 1) WRITE(8,106) Y1(J1),(SIGX(J1,I),I=1,11)
      IF(IJ .EQ. 2) WRITE(8,106) Y1(J1),(SIGY(J1,I),I=1,11)
      IF(IJ .EQ. 3) WRITE(8,106) Y1(J1),(TAUXY(J1,I),I=1,11)
   80 IF(IJ .EQ. 4) WRITE(8,106) Y1(J1),(SIGMAX(J1,I),I=1,11)
      WRITE(8,103)
      WRITE(8,102) (X1(I),I=1,11)
      WRITE(8,101)
      WRITE(8,107)
  106 FORMAT(12X,'I',T125,'I',/,1X,D9.3,2X,'I',11(1X,D9.3),T125,'I')
   90 CONTINUE
      RETURN
      END
End of file
```

```
      SUBROUTINE TDRIVE(NT,TX,KOUT)
C
C     SUBROUTINE TDRIVE IS USED TO LET THE USER ENTER
C     X, Y COORDINATES AND TEMPERATURE DATA FOR VARIOUS
C     POINTS ON THE BANDSAW BLADE.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION TX(35),XT(70),YT(70),TXI(70)
      IO=5
      WRITE(6,102)
  102 FORMAT(/,9X,'BEGIN ENTRY OF TEMPERATURE DATA')
      WRITE(6,106)
  106 FORMAT(/,11X,'IS THE DATA TO READ FROM A DATA FILE',/,
     $11X,'O-NO',/,11X,'1-YES')
      READ(5,107) IR
  107 FORMAT(I1)
      IF(IR .EQ. 1) IO=11
      IF(IO .EQ. 11) REWIND 11
      WRITE(6,104)
  104 FORMAT(/,11X,'ENTER THE NUMBER OF TEMPERATURE DATA POINTS',
     $' (I2 FORMAT)')
      READ(5,105) NPTS
  105 FORMAT(I2)
  103 FORMAT(/,12X,'ENTER X-COORDINATE,Y-COORDINATE,AND TEMPERATURE',
     $' FOR POINT',I3, ' ON THE BANDSAW BLADE')
      DO 60 I1=1,NPTS
      IF(IO .EQ. 5) WRITE(6,103) I1
   60 READ(IO,201) XT(I1),YT(I1),TXI(I1)
C
C     SUBROUTINE TCOFF CALCULATES THE TEMPERATURE FIELD
C     COEFFICIENTS USING LINER REGRESSION.
      CALL TCOFF(NT,NPTS,XT,YT,TXI,TX,KOUT)
  201 FORMAT(3D15.6)
      RETURN
      END
End of file
```

```
      SUBROUTINE TCOFF(NT,NPTS,XT,YT,TXI,TX,KOUT)
C
C     SUBROUTINE TCOFF CALCULATES THE TEMPERATURE FIELD
C     COEFFICIENTS GIVEN TEMPERATURES AT VARIOUS POINTS
C     ON THE BANDSAW BLADE. THE TEMPERATURE DATA IS READ
C     IN SUBROUTINE TDRIVE. THE TEMPERATURE COEFFICIENTS
C     ARE USED IN SUBROUTINE TEMP TO CALCULATE THE EFFECTS
C     OF A TEMPERATURE FIELD FOR THE RIGHT HAND SIDE VECTOR
C     OF THE SYSTEM OF EQUATIONS.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION IV(35),TX(35),A(70,35),T(35,35)
      DIMENSION ATA(35,35),TXI(70),XT(70),YT(70)
      NM=35
      MTZ=2*NT+1
      DO 10 I=1,NT
   10 MTZ=MTZ+(NT-I)
C
C     THIS PART OF TCOFF CALCULATES THE COEFFICIENT
C     MATRIX USED BY SUBROUTINES DLUD AND DBS. SUBROUTINE
C     DLUD DECOMPOSES THE MATRIX(A) INTO LOWER AND UPPER
C     TRIANGLUR MATRICES. SUBROUTINE DBS CALCULATES THE
C     TEMPERATURE FIELD COEFFICIENTS USING THE DECOMPOSITION
C     OF THE MATRIX(A) AND THE TEMPERATURE DATA TX.
      DO 45 I=1,NPTS
      J=2
      A(I,1)=1.0D+00
      DO 25 K=1,NT
      A(I,J)=XT(I)**K
   25 J=J+1
      DO 30 L=1,NT
      DO 35 K=1,NT
      IF(K+L .GT. NT) GO TO 30
      A(I,J)=XT(I)**K*YT(I)**L
   35 J=J+1
   30 CONTINUE
      DO 40 L=1,NT
      A(I,J)=YT(I)**L
   40 J=J+1
   45 CONTINUE
C
C     SUBROUTINE ATA FORMS THE NORM MATRIX
C     (ATRANSPOSE) TIMES A AND (ATRANSPOSE) TIMES TXI
      CALL ATAM(MTZ,NPTS,A,TXI,TX,ATA)
      CALL DLUD(MTZ,NM,ATA,NM,T,IV)
      IF(IV(MTZ) .EQ. 0) GO TO 75
      CALL DBS(MTZ,NM,T,IV,TX)
      GO TO 80
   75 WRITE(6,102)
  102 FORMAT(4X,'***ERROR** MATRIX IS SINGULAR IN SUBROUTINE TCOFF')
      KOUT=1
   80 RETURN
      END
```

End of file

```
      SUBROUTINE ATAM(MTZ,NPTS,A,TXI,TX,ATA)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(70,35),TXI(70),ATA(35,35),TX(35)
C
C     CALCULATE (ATRANSPOSE) TIMES A AND (ATRANSPOSE) TIMES TXI
      DO 25 I=1,MTZ
      DO 25 J=1,MTZ
      SUMA=0.0D+00
      SUMB=0.0D+00
      DO 30 K=1,NPTS
      SUMA=SUMA+A(K,J)*A(K,I)
   30 SUMB=SUMB+TXI(K)*A(K,I)
      ATA(I,J)=SUMA
   25 TX(I)=SUMB
      RETURN
      END
End of file
```

```
      SUBROUTINE TOUT(N,EL,B,TFL,ITITL)
C
C     TOUT PRODUCES A TABLE OF TEMPERATURES AT VARIOUS POINTS ON THE
C     BAND SAW BLADE USING THE TEMPERATURE FIELD COEFFICIENTS FOUND IN
C     SUBROUTINE TCOFF.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ITITL(50),TFL(35),TOU(6,11),X1(11),Y1(6)
      WRITE(8,500) ITITL
      WRITE(8,510)
  510 FORMAT(1H-,4X,'TEMPERATURE FIELD COEFFICIENTS')
      I1=1
      N2=0
      M2=0
      WRITE(8,520) N2,M2,TFL(I1)
      I1=2
      DO 55 I=1,N
      WRITE(8,520) I,M2,TFL(I1)
   55 I1=I1+1
      DO 65 J=1,N
      DO 75 I=1,N
      IF(I+J .GT. N) GO TO 65
      WRITE(8,520) I,J,TFL(I1)
   75 I1=I1+1
   65 CONTINUE
      DO 85 J=1,N
      WRITE(8,520) N2,J,TFL(I1)
   85 I1=I1+1
  520 FORMAT(8X,'T(',I1,',',I1,')=',D12.5)
      Y2=0.0D+00
      DELX=EL/(1.0D+01)
      DELY=B/(5.0D+00)
      DO 20 I=1,6
      X2=0.0D+00
      DO 30 J=1,11
      J1=2
      TOU(I,J)=TFL(1)
      DO 40 K=1,N
      TOU(I,J)=TOU(I,J)+TFL(J1)*X2**K
   40 J1=J1+1
      DO 50 L=1,N
      DO 60 K=1,N
      IF(K+L .GT. N) GO TO 50
      TOU(I,J)=TOU(I,J)+TFL(J1)*X2**K*Y2**L
   60 J1=J1+1
   50 CONTINUE
      DO 70 L=1,N
      TOU(I,J)=TOU(I,J)+TFL(J1)*Y2**L
   70 J1=J1+1
      X1(J)=X2
   30 X2=X2+DELX
      Y1(I)=Y2
   20 Y2=Y2+DELY
      WRITE(8,500) ITITL
  500 FORMAT(1H1,2(/,1H-),56X,50A1)
      WRITE(8,100)
  100 FORMAT(1H-,56X,'TEMPERATURE DISTRIBUTION')
      WRITE(8,107)
  107 FORMAT(1H ,124('_'))
  101 FORMAT(12X,'I',50X,'X-COORDINATE',T125,'I')
  102 FORMAT(12X,'I',T125,'I',/,12X,'I',11(1X,D9.3),T125,'I')
```

```
  108 FORMAT(1X,124('-'))
      WRITE(8,105)
  105 FORMAT(5X,'Y',6X,'I',T125,'I')
      WRITE(8,106)
  106 FORMAT(1X,'COORDINATE',1X,'I',T125,'I')
      DO 80 J=1,6
      J1=7-J
   80 WRITE(8,104) Y1(J1),(TOU(J1,I),I=1,11)
      WRITE(8,108)
      WRITE(8,102) (X1(I),I=1,11)
      WRITE(8,101)
      WRITE(8,107)
  104 FORMAT(12X,'I',T125,'I',/,1X,D9.3,2X,'I',11(1X,D9.3),T125,'I')
      RETURN
      END
End of file
```

```
      SUBROUTINE TEMP(N,NT,TX,MSIZE,EL,B,H,E,UN,ALPHA,TIX,TIY)
C
C     TEMP CALCULATES THE EFFECTS OF A TEMPERATURE GRADIENT, IF PRESENT,
C     ON BLADE STRESS THROUGH ITS CONTRIBUTION TO THE RIGHT HAND SIDE
C     VECTOR, WHICH IS FORMED IN SUBROUTINE RHS.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION TX(35),TIX(35),TIY(35)
      DO 85 JJ=1,MSIZE
      TIX(JJ)=0.0D+00
   85 TIY(JJ)=0.0D+00
      IF(NT .EQ. 0) GO TO 25
      KK=1
      LL=0
      DO 70 JJ=1,MSIZE
      DTX=0.0D+00
      DTY=0.0D+00
      J1=2
      DO 60 I=1,NT
      IF(KK .EQ. 0) GO TO 35
      D1=-DFLOAT(KK)/DFLOAT((I+KK)*(LL+1))
      DTX=TX(J1)*EL**(I+KK)*B**(LL+1)*D1+DTX
   35 IF(LL .EQ. 0) GO TO 60
      D2=-1.0/DFLOAT(I+KK+1)
      DTY=DTY+D2*TX(J1)*EL**(I+KK+1)*B**LL
   60 J1=J1+1
      DO 50 J=1,NT
      DO 55 I=1,NT
      IF(I+J .GT. NT) GO TO 50
      IF(KK .EQ. 0) GO TO 75
      D1=-DFLOAT(KK)/DFLOAT((I+KK)*(J+LL+1))
      DTX=TX(J1)*EL**(I+KK)*B**(J+LL+1)*D1+DTX
   75 IF(LL .EQ. 0) GO TO 55
      D2=-DFLOAT(LL)/DFLOAT((I+KK+1)*(J+LL))
      DTY=TX(J1)*EL**(I+KK+1)*B*(J+LL)*D2+DTY
   55 J1=J1+1
   50 CONTINUE
      DO 40 J=1,NT
      IF(KK .EQ. 0) GO TO 65
      D1=-1.0/DFLOAT(J+LL+1)
      DTX=DTX+D1*TX(J1)*EL**KK*B**(J+LL+1)
   65 IF(LL .EQ. 0) GO TO 40
      D2=-DFLOAT(LL)/DFLOAT((KK+1)*(J+LL))
      DTY=TX(J1)*EL**(KK+1)*B**(J+LL)*D2+DTY
   40 J1=J1+1
      TIX(JJ)=E*ALPHA*H*DTX/((1.0D+00)-UN)
      TIY(JJ)=E*ALPHA*H*DTY/((1.0D+00)-UN)
      IF(KK .EQ. 0) GO TO 20
      KK=KK+1
      IF(KK+LL .GT. N) GO TO 15
      GO TO 70
   15 LL=LL+1
      KK=1
      IF(KK+LL .GT. N) GO TO 10
      GO TO 70
   10 KK=0
      LL=1
      GO TO 70
   20 LL=LL+1
      IF(LL .GT. N) GO TO 2
      KK=0
```

```
   70 CONTINUE
   25 RETURN
      END
End of file
```

```
      SUBROUTINE FORCE(N,R,FTAN,FNOR,X1,X2,EL,B,FORCX,FORCY,MSIZE)
C
C     FORCE CALCULATES THE BLADE STRESSES CAUSED BY A NORMAL, TANGENTIAL
C     , AND/OR TENSILE LOADS THROUGH ITS CONTRIBUTION TO THE RIGHT HAND
C     SIDE VECTOR, WHICH IS FORMED IN SUBROUTINE RHS.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X1(2),X2(2),FORCX(35),FORCY(35)
C
C     MSIZE IS THE NUMBER OF TERMS IN THE APPROXIMATING FUNCTION FOR
C     DISPLACEMENT FIELD, U.
      MSIZE=2*N
      DO 10 I=1,N
   10 MSIZE=N-I+MSIZE
      J=1
      DO 20 L1=1,N
      LL=L1-1
      DO 30 KK=1,N
      IF(KK+LL .GT. N) GO TO 20
      FORCX(J)=-R*EL**KK*B**(LL+1)/DFLOAT(LL+1)
     $+FTAN*(X2(2)**(KK+1)-X1(2)**(KK+1))*B**LL/DFLOAT(KK+1)
      FNI=FNOR*(X2(1)**(KK+1)-X1(1)**(KK+1))*B**LL/DFLOAT(KK+1)
      IF(LL .NE. 0) GO TO 40
      D1=DFLOAT(KK+1)
      D2=DFLOAT(N+1)*EL**(N-KK)
      D3=X2(1)**(KK+1)-X1(1)**(KK+1)
      D4=X2(1)**(N+1)-X1(1)**(N+1)
      FNI=FNOR*((D3/D1)-(D4/D2))
      IF(KK .EQ. N) FNI=0.0D+00
   40 FORCY(J)=FNI
   30 J=J+1
   20 CONTINUE
      DO 50 LL=1,N
      FORCX(J)=FTAN*(X2(2)-X1(2))*B**LL
      FORCY(J)=FNOR*(X2(1)-X1(1))*B**LL
   50 J=J+1
      RETURN
      END
End of file
```

```
      SUBROUTINE RHS(FORCX,FORCY,TIX,TIY,MSIZE,N,BX)
C
C     RHS COMBINES THE EFFECTS OF THE EXTERNAL LOADS, STORED IN FORCX AND
C     FORCY, WITH THE EFFECTS OF A TEMPERATURE GRADIENT, STORED IN TIX
C     AND TIY, TO FORM THE RIGHT HAND SIDE VECTOR, WHICH IS STORED IN BX.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION FORCX(35),FORCY(35),TIX(35),TIY(35),BX(70)
      DO 10 J=1,MSIZE
      J1=J+MSIZE
      J2=J
      IF(J .GE. N) J2=J2+1
      BX(J)=FORCX(J)+TIX(J)
      IF(J2 .GT. MSIZE) GO TO 10
      BX(J1)=FORCY(J2)+TIY(J2)
   10 CONTINUE
      RETURN
      END
End of file
```

```
      SUBROUTINE SDRIVE(AX,NTS,KOUT)
C     SUBROUTINE SDRIVE DOES NOT WORK PROPERLY AT THIS TIME.
C     SDRIVE IS USED TO LET THE USER ENTER X, AND Y COORDINATES AND
C     PRESTRESS OR STRESS DATA FOR VARIOUS POINTS ON THE BAND SAW BLADE.
C     THUS, STRESS FIELD COEFFICIENTS CAN BE CALCULATED IF ONLY THE
C     VIBRATION PROBLEM IS BEING SOLVED. PRESTRESS FIELDS COEFFICIENTS
C     CAN BE CALCULATED TO BE COMBINED WITH THE STRESS FIELD COEFFICIENTS
C     LATER IN THE PROGRAM.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AX(35,3),XT(70),YT(70),SXI(70),SX(35)
      IPI=0
      IO=5
      DO 5 I=1,3
      DO 7 J=1,35
    7 AX(I,J)=0.0D+00
    5 CONTINUE
      WRITE(6,102)
  102 FORMAT(///,3X,'BEGIN ENTRY OF STRESS DATA')
   15 WRITE(6,103)
  103 FORMAT(//,6X,'ENTER THE NUMBER OF THE DESIRED STRESS FIELD',/,9X,
     1'1- SIGMA X',/,9X,'2- SIGMA Y',/,9X,'3- TAU XY',/,9X,
     2'4- ALL THREE STRESS FIELDS',/,9X,'5- STOP')
      READ(5,100) IS
  100 FORMAT(I1)
      IF(IS .NE. 4) GO TO 30
      IPI=1
      IS=1
   30 IF(IS .EQ. 5) GO TO 20
      WRITE(6,104)
  104 FORMAT(/,12X,'IS DATA TO BE READ FROM A DATA FILE',/,12X,'0- NO,
     1,'1- YES')
      READ(5,100) IR
      IF(IR .EQ. 1) IO=10
      IF(IO .EQ. 10) REWIND 10
      WRITE(5,105)
  105 FORMAT(/,9X,'ENTER THE NUMBER OF DATA POINTS (I2 FORMAT)')
      READ(5,101) NPTS
  101 FORMAT(I2)
   40 DO 45 I1=1,NPTS
      IF(IO .EQ. 5) WRITE(6,106) I1
  106 FORMAT(12X,'ENTER X-COORDINATE, Y-COORDINATE, AND STRESS FOR POINT
     1 ',I2,'  ON THE BAND SAW BLADE')
   45 READ(IO,201) XT(I1),YT(I1),SXI(I1)
  201 FORMAT(3D15.6)
      CALL TCOFF(NTS,NPTS,XT,YT,SXI,SX,KOUT)
      MT=(NTS+1)*(NTS+2)/2
      DO 50 IJ=1,MT
   50 AX(IJ,IS)=SX(IJ)
      IS=IS+1
      IF(IS .EQ. 4) GO TO 20
      IF(IPI .EQ. 1) GO TO 40
      GO TO 15
   20 RETURN
      END
```

End of file

VIBRATION ANALYSIS SUBROUTINES

```
      SUBROUTINE VIBRA(N,M,NM,MAXNM,KON,L,B,H,D,RHOH,NU,C,AX,ITITL)
C
C     SUBROUTINE VIBRA SOLVES FOR THE FIVE LOWEST EIGENVALUES OF
C     THE BAND SAW BLADE,BASED ON THE CLASSICAL RITZ METHOD CODE
C     PRESENTED IN THE REPORT "TWO COMPUTER CODES FOR BAND SAW
C     BLADE VIBRATION AND STABILITY ANALYSIS" BY A.G. ULSOY, 1978.
C     THE THEORY IS PRESENTED IN THE DOCTORAL DISSERTATION BY
C     A. G. ULSOY "VIBRATION AND STABILTY OF BAND SAW BLADES: A
C     THEORETICAL AND EXPERIMENTAL STUDY".
C
C     VIBRA USES NON DIMENSIONAL INTEGRALS STORED ON TAPE(FILE)
C     TO COMPUTE THE MASS, GYROSCOPIC AND STIFFNESS MATRICES.
C     VIBRA CALLS SUBROUTINES DECODE, REFORM, EIGENS.
C
C     SUBROUTINE DECODE: PROVIDES A CONVERSION BETWEEN THE INDEXING
C                        NOTATION USED IN VIBRA AND THAT USED FOR
C                        THE NONDIMENSIONAL INTEGRALS ON TAPE(FILE).
C     SUBROUTINE REFORM: CALCULATES A MATRIX (A) FROM THE MASS,
C                        GYRO, AND STIFF MATRICES TO CONVERT FROM
C                        A SYSTEM OF SECOND ORDER DIFFERENTIAL
C                        EQUATIONS TO A SYSTEM OF FIRST ORDER
C                        DIFFERENTIAL EQUATIONS.
C     SUBROUTINE EIGENS: CALCULATES THE FIVE LOWEST EIGENVALUES OF
C                        THE BLADE, USING THE MATRIX (A) FORMED
C                        IN SUBROUTINE REFORM.
C
C     **********************************************************
C     **********************************************************
C     **                                                      **
C     **      IMPORTANT VARIBLES IN SUBROUTINE VIBRA          **
C     **                                                      **
C     **    KON(8): PREVENTS REREADING OF NONDIMENSIONAL      **
C     **            INTEGRALS AFTER INITIAL READING.          **
C     **                                                      **
C     **    N: NUMBER OF SIMPLE SUPPORTED BEAM EIGENFUNCTIONS. **
C     **                                                      **
C     **    M: NUMBER OF FREE-FREE BEAM EIGENFUNCTIONS.       **
C     **                                                      **
C     **    NM: N TIMES M.                                    **
C     **                                                      **
C     **    AX: STRESS FIELD COEFFICIENTS FOR SIGMA-X,        **
C     **        SIGMA-Y, AND TAU-XY.                          **
C     **                                                      **
C     **********************************************************
C     **********************************************************
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 L,NU,MASS(36)
      DIMENSION ITITL(50),KON(11),GYRO(36,36),A(72,72),STIF(36,36)
      DIMENSION AX(35,3),SXO1(6,6),SXO2(6,6),SXO3(6,6),SXO4(6,6),
     $SXO5(6,6),SXO6(6,6),SXO7(6,6),SXO8(6,6),SXO9(6,6),SX10(6,6),
     $SX11(6,6),SX12(6,6),SX13(6,6),SX14(6,6),SYO1(6,6),SYO2(6,6),
     $SYO3(6,6),SYO4(6,6),SYO5(6,6),SYO6(6,6),SYO7(6,6),SYO8(6,6),
     $SYO9(6,6),SY10(6,6),SY11(6,6),SY12(6,6),SY13(6,6),SY14(6,6),
     $SXT1(6),SXT2(6),SXT3(6),SXT4(6),SYT1(6),SYT2(6),SYT3(6),SYT4(6)
      IF(KON(8) .EQ. 1) GO TO 60
      REWIND 7
C
C     READ NONDIMENSIONAL INTEGRALS FROM TAPE (FILE).
```

```
      READ(7) SXO1,SXO2,SXO3,SXO4,SXO5,SXO6,SXO7
      READ(7) SXO8,SXO9,SX10,SX11,SX12,SX13,SX14
      READ(7) SYO1,SYO2,SYO3,SYO4,SYO5,SYO6,SYO7
      READ(7) SYO8,SYO9,SY10,SY11,SY12,SY13,SY14
      READ(7) SXT1,SXT2,SXT3,SXT4,SYT1,SYT2,SYT3,SYT4
C     FORMULATE THE MASS,GYRO,AND STIFFNESS MATRICES
   60 CONTINUE
      DO 50 I=1,NM
      CALL DECODE(I,N,M,IX,IY)
      MASS(I)=RHOH*L*B*SXO1(IX,IX)*SYO1(IY,IY)
      DO 50 J=1,NM
      CALL DECODE(J,N,M,JX,JY)
      GYRO(I,J)=(2.OD+OO)*RHOH*B*C*SXO2(IX,JX)*SYO1(IY,JY)
     $+L*B*SXO1(IX,JX)*SYO1(IY,JY)
      STIF1=(B/L)*(AX(1,1)*SXO4(IX,JX)*SYO1(IY,JY)-AX(2,1)*
     1      SXO8(IX,JX)*SYO1(IY,JY)-AX(8,1)*SXO3(IX,JX)*SYO6(IY,JY)-
     2      AX(3,1)*SX11(IX,JX)*SYO1(IY,JY)-AX(5,1)*SXO8(IX,JX)*
     3      SYO6(IY,JY)-AX(9,1)*SXO3(IX,JX)*SYO9(IY,JY)-AX(4,1)*
     4      SX14(IX,JX)*SYO1(IY,JY)-AX(6,1)*SX11(IX,JX)*SYO6(IY,JY)-
     5      AX(7,1)*SXO8(IX,JX)*SYO9(IY,JY)-AX(10,1)*SXO3(IX,JX)*
     6      SY12(IY,JY))
      STIF2=(L/B)*(AX(1,2)*SXO1(IX,JX)*SYO4(IY,JY)-AX(2,2)*SXO6(IX,JX)
     1      *SYO3(IY,JY)-AX(8,2)*SXO1(IX,JX)*SYO8(IY,JY)-AX(3,2)*
     2      SXO9(IX,JX)*SYO3(IY,JY)-AX(5,2)*SXO6(IX,JX)*SYO8(IY,JY)-
     3      AX(9,2)*SXO1(IX,JX)*SY11(IY,JY)-AX(4,2)*SX12(IX,JX)*
     4      SYO3(IY,JY)-AX(6,2)*SXO9(IX,JX)*SYO8(IY,JY)-AX(7,2)*
     5      SXO6(IX,JX)*SY11(IY,JY)-AX(10,2)*SXO1(IX,JX)*SY14(IY,JY))
      STIF3=(2.OD+OO)*(AX(1,3)*SXO2(JX,IX)*SYO2(IY,JY)*(5.OD-O1)+
     1      AX(1,3)*SXO2(IX,JX)*SYO2(JY,IY)*(5.OD-O1)-AX(2,3)*
     2      SXO7(IX,JX)*SYO2(IY,JY)-AX(8,3)*SXO2(IX,JX)*SYO7(IY,JY)-
     $      AX(3,3)*SX10(IX,JX)*SYO2(IY,JY)
     3      -AX(5,3)*SXO7(IX,JX)*SYO7(IY,JY)-AX(9,3)*SXO2(IX,JX)*
     4      SY10(IY,JY)-AX(4,3)*SX13(IX,JX)*SYO2(IY,JY)-AX(6,3)*
     5      SX10(IX,JX)*SYO7(IY,JY)-AX(7,3)*SXO7(IX,JX)*SY10(IY,JY)-
     6      AX(10,3)*SXO2(IX,JX)*SY13(IY,JY))
      STIF(I,J)=D*((B/(L**3))*SXO5(IX,JX)*SYO1(IY,JY)+(L/(B**3))*
     1    SXO1(IX,JX)*SYO5(IY,JY)+(((2.OD+OO)*((1.OD+OO)-NU))/(L*B))*
     2    SXO4(IX,JX)*SYO4(IY,JY)+(NU/(L*B))*(SXO3(JX,IX)*SYO3(IY,JY)+
     3          SXO3(IX,JX)*SYO3(JY,IY)))+(STIF1+STIF2+STIF3)*H
   50 CONTINUE
      KON(8)=1
      NMT=2*NM
      MAXNMT=2*MAXNM
C
C     SUBROUTINE REFORM REFORMULATES SET OF SECOND ORDER DIFFERENTIAL
C     EQUATIONS AS A SET OF FIRST ORDER DIFFERENTIAL EQUATIONS.
      CALL REFORM(NM,NMT,MASS,GYRO,STIF,A)
C
C     SOLVES THE EIGENVALUE PROBLEM FOR THE FIVE LOWEST
C     EIGENVALUES OF THE BLADE.
      CALL EIGENS(NMT,MAXNMT,KON,L,RHOH,D,A,ITITL)
      RETURN
      END
End of file
```

```
      SUBROUTINE DECODE(I,N,M,IX,IY)
C
C     USED TO PROVIDE A CONVERSION BETWEEN THE INDEXING NOTATION
C     USED IN VIBRA AND THAT USED FOR THE NONDIMENSIONAL INTEGRALS
C     STORED ON TAPE (FILE).
C
      DO 10 K=1,N
      KMI=K-1
      IF(I .LE. (KMI*M) .OR. I .GT. (K*M)) GO TO 10
      IX=K
      IY=I-KMI*M
      RETURN
   10 CONTINUE
      RETURN
      END
End of file
```

```
      SUBROUTINE REFORM(NM,NMT,MASS,GYRO,STIF,A)
C
C     CALCULATES A MATRIX (A) FROM THE MASS, GYRO, AND STIFF MATRICES
C     TO CONVERT FROM A SYSTEM OF SECOND ORDER DIFFERENTIAL EQUATIONS
C     TO A SYSTEM OF FIRST ORDER DIFFERENTIAL EQUATIONS.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION GYRO(36,36),STIF(36,36),A(72,72)
      REAL*8 MASS(36)
      DO 10 I=1,NM
      II=I+NM
      DO 10 J=1,NM
      JJ=J+NM
      A(I,J)=-GYRO(I,J)/MASS(I)
      A(I,JJ)=-STIF(I,J)/MASS(I)
      A(II,J)=0.0D+00
      IF(II .EQ. JJ) A(II,J)=1.0D+00
      A(II,JJ)=0.0D+00
   10 CONTINUE
      RETURN
      END
End of file
```

```
      SUBROUTINE EIGENS(N,NM,KON,L,RHOH,D,A,ITITL)
C
C     EIGENVALUES ARE OBTAINED USING EISPACK ROUTINES BALANC, ELMHES,
C     AND HQR. THE FIVE LOWEST EIGENVALUES AND NONDIMENSIONAL NATURAL
C     FREQUENCIES ARE PRINTED TO THE OUTPUT FILE. THE EIGEN SOLUTION
C     ROUTINES USED ARE FOR THE CASE OF A REAL MATRIX (A)
C
C     ************************************************************
C     ************************************************************
C     **                                                        **
C     **   NOTE: SUBROUTINE HQR DESTROYS THE ORIGINAL           **
C     **           MATRIX (A).                                  **
C     **                                                        **
C     ************************************************************
C     ************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 L
      DIMENSION A(72,72),SCALE(72),WR(72),WI(72),OMEGA(36),INT(72),
     $ITITL(50),KON(11),LEV(5)
      CALL BALANC(NM,N,A,LOW,IGH,SCALE)
      CALL ELMHES(NM,N,LOW,IGH,A,INT)
      CALL HQR(NM,N,LOW,IGH,A,WR,WI,IERR)
      IF(IERR .NE. 0) WRITE(6,680) IERR
      WRITE(8,610) ITITL
      WRITE(8,611)
      NM1=N-1
      DO 20 I=1,NM1,2
      II=(I/2)+1
      FREQ=DSQRT(RHOH/D)*DSQRT((WR(I)**2)+(WI(I)**2))
      OMEGA(II)=L*L*FREQ
   20 WRITE(8,620) I,WR(I),WI(I),OMEGA(II)
      DO 30 I=1,5
      LEV(I)=1
      OMEGAO=OMEGA(1)
      DO 40 J=1,NM1,2
      JJ=(J/2)+1
      IF(OMEGA(JJ) .LT. OMEGAO) LEV(I)=J
      IF(OMEGA(JJ) .LT. OMEGAO) OMEGAO=OMEGA(JJ)
   40 CONTINUE
      JJ=(LEV(I)/2)+1
      OMEGA(JJ)=1.0D+20
   30 CONTINUE
      WRITE(8,625) (LEV(I),I=1,5)
  610 FORMAT(1H1,/,1H-,50A1,/,1H-,4X,'EIGEN',6X,'REAL',9X,'IMAGINARY'
     $,9X,'NON-DIMENSIONAL')
  611 FORMAT(5X,'VALUE',6X,'PART',12X,'PART',9X,'NATURAL FREQUENCY')
  620 FORMAT(5X,I3,2X,2(1X,D14.6),5X,D14.6)
  625 FORMAT(3X,'LOWEST 5 EIGENVALUES IN INCREASING ORDER ',5(2X,I2))
  680 FORMAT(//,'ERROR**** SUBROUTINE HQR HAS FAILED TO CONVERGE',
     $' TO EIGENVALUE NUMBER',I5)
      RETURN
      END
End of file
```

```
C NAASA  3.2.001 BALANC   FTN   06-24-75     THE UNIV OF MICH COMP CTR
C
C     ------------------------------------------------------------------
C
      SUBROUTINE BALANC(NM,N,A,LOW,IGH,SCALE)
C
      INTEGER I,J,K,L,M,N,JJ,NM,IGH,LOW,IEXC
      REAL*8 A(NM,N),SCALE(N)
      REAL*8 C,F,G,R,S,B2,RADIX
      REAL*8 DABS
      LOGICAL NOCONV
C
C     THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE BALANCE,
C     NUM. MATH. 13, 293-304(1969) BY PARLETT AND REINSCH.
C     HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 315-326(1971).
C
C     THIS SUBROUTINE BALANCES A REAL MATRIX AND ISOLATES
C     EIGENVALUES WHENEVER POSSIBLE.
C
C     ON INPUT:
C
C        NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
C          ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
C          DIMENSION STATEMENT;
C
C        N IS THE ORDER OF THE MATRIX;
C
C        A CONTAINS THE INPUT MATRIX TO BE BALANCED.
C
C     ON OUTPUT:
C
C        A CONTAINS THE BALANCED MATRIX;
C
C        LOW AND IGH ARE TWO INTEGERS SUCH THAT A(I,J)
C          IS EQUAL TO ZERO IF
C           (1) I IS GREATER THAN J AND
C           (2) J=1,...,LOW-1 OR I=IGH+1,...,N;
C
C        SCALE CONTAINS INFORMATION DETERMINING THE
C           PERMUTATIONS AND SCALING FACTORS USED.
C
C     SUPPOSE THAT THE PRINCIPAL SUBMATRIX IN ROWS LOW THROUGH IGH
C     HAS BEEN BALANCED, THAT P(J) DENOTES THE INDEX INTERCHANGED
C     WITH J DURING THE PERMUTATION STEP, AND THAT THE ELEMENTS
C     OF THE DIAGONAL MATRIX USED ARE DENOTED BY D(I,J).  THEN
C        SCALE(J) = P(J),    FOR J = 1,...,LOW-1
C                 = D(J,J),      J = LOW,...,IGH
C                 = P(J)         J = IGH+1,...,N.
C     THE ORDER IN WHICH THE INTERCHANGES ARE MADE IS N TO IGH+1,
C     THEN 1 TO LOW-1.
C
C     NOTE THAT 1 IS RETURNED FOR IGH IF IGH IS ZERO FORMALLY.
C
C     THE ALGOL PROCEDURE EXC CONTAINED IN BALANCE APPEARS IN
C     BALANC  IN LINE.  (NOTE THAT THE ALGOL ROLES OF IDENTIFIERS
C     K,L HAVE BEEN REVERSED.)
C
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C
```

```
C     ----------------------------------------------------------------
C
C     :::::::::: RADIX IS A MACHINE DEPENDENT PARAMETER SPECIFYING
C                THE BASE OF THE MACHINE FLOATING POINT REPRESENTATION.
C                RADIX = 16.0D0 FOR LONG FORM ARITHMETIC
C                ON S360 ::::::::::
      DATA RADIX/Z4210000000000000/
C
      B2 = RADIX * RADIX
      K = 1
      L = N
      GO TO 100
C     :::::::::: IN-LINE PROCEDURE FOR ROW AND
C                COLUMN EXCHANGE ::::::::::
   20 SCALE(M) = J
      IF (J .EQ. M) GO TO 50
C
      DO 30 I = 1, L
         F = A(I,J)
         A(I,J) = A(I,M)
         A(I,M) = F
   30 CONTINUE
C
      DO 40 I = K, N
         F = A(J,I)
         A(J,I) = A(M,I)
         A(M,I) = F
   40 CONTINUE
C
   50 GO TO (80,130), IEXC
C     :::::::::: SEARCH FOR ROWS ISOLATING AN EIGENVALUE
C                AND PUSH THEM DOWN ::::::::::
   80 IF (L .EQ. 1) GO TO 280
      L = L - 1
C     :::::::::: FOR J=L STEP -1 UNTIL 1 DO -- ::::::::::
  100 DO 120 JJ = 1, L
         J = L + 1 - JJ
C
         DO 110 I = 1, L
            IF (I .EQ. J) GO TO 110
            IF (A(J,I) .NE. 0.0D0) GO TO 120
  110    CONTINUE
C
         M = L
         IEXC = 1
         GO TO 20
  120 CONTINUE
C
      GO TO 140
C     :::::::::: SEARCH FOR COLUMNS ISOLATING AN EIGENVALUE
C                AND PUSH THEM LEFT ::::::::::
  130 K = K + 1
C
  140 DO 170 J = K, L
C
         DO 150 I = K, L
            IF (I .EQ. J) GO TO 150
            IF (A(I,J) .NE. 0.0D0) GO TO 170
  150    CONTINUE
```

```
         M = K
         IEXC = 2
         GO TO 20
  170 CONTINUE
C     :::::::::: NOW BALANCE THE SUBMATRIX IN ROWS K TO L ::::::::::
      DO 180 I = K, L
  180 SCALE(I) = 1.0D0
C     :::::::::: ITERATIVE LOOP FOR NORM REDUCTION ::::::::::
  190 NOCONV = .FALSE.
C
      DO 270 I = K, L
         C = 0.0D0
         R = 0.0D0
C
         DO 200 J = K, L
            IF (J .EQ. I) GO TO 200
            C = C + DABS(A(J,I))
            R = R + DABS(A(I,J))
  200    CONTINUE
C
         G = R / RADIX
         F = 1.0D0
         S = C + R
  210    IF (C .GE. G) GO TO 220
         F = F * RADIX
         C = C * B2
         GO TO 210
  220    G = R * RADIX
  230    IF (C .LT. G) GO TO 240
         F = F / RADIX
         C = C / B2
         GO TO 230
C     :::::::::: NOW BALANCE ::::::::::
  240    IF ((C + R) / F .GE. 0.95D0 * S) GO TO 270
         G = 1.0D0 / F
         SCALE(I) = SCALE(I) * F
         NOCONV = .TRUE.
C
         DO 250 J = K, N
  250    A(I,J) = A(I,J) * G
C
         DO 260 J = 1, L
  260    A(J,I) = A(J,I) * F
C
  270 CONTINUE
C
      IF (NOCONV) GO TO 190
C
  280 LOW = K
      IGH = L
      RETURN
C     :::::::::: LAST CARD OF BALANC ::::::::::
      END
End of file
```

```
C NAASA   3.2.006 ELMHES    FTN    06-24-75      THE UNIV OF MICH COMP CTR
C
C     ------------------------------------------------------------------
C
      SUBROUTINE ELMHES(NM,N,LOW,IGH,A,INT)
C
      INTEGER I,J,M,N,LA,NM,IGH,KP1,LOW,MM1,MP1
      REAL*8 A(NM,N)
      REAL*8 X,Y
      REAL*8 DABS
      INTEGER INT(IGH)
C
C     THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE ELMHES,
C     NUM. MATH. 12, 349-368(1968) BY MARTIN AND WILKINSON.
C     HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 339-358(1971).
C
C     GIVEN A REAL GENERAL MATRIX, THIS SUBROUTINE
C     REDUCES A SUBMATRIX SITUATED IN ROWS AND COLUMNS
C     LOW THROUGH IGH TO UPPER HESSENBERG FORM BY
C     STABILIZED ELEMENTARY SIMILARITY TRANSFORMATIONS.
C
C     ON INPUT:
C
C        NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
C          ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
C          DIMENSION STATEMENT;
C
C        N IS THE ORDER OF THE MATRIX;
C
C        LOW AND IGH ARE INTEGERS DETERMINED BY THE BALANCING
C          SUBROUTINE  BALANC.  IF  BALANC  HAS NOT BEEN USED,
C          SET LOW=1, IGH=N;
C
C        A CONTAINS THE INPUT MATRIX.
C
C     ON OUTPUT:
C
C        A CONTAINS THE HESSENBERG MATRIX.  THE MULTIPLIERS
C          WHICH WERE USED IN THE REDUCTION ARE STORED IN THE
C          REMAINING TRIANGLE UNDER THE HESSENBERG MATRIX;
C
C        INT CONTAINS INFORMATION ON THE ROWS AND COLUMNS
C          INTERCHANGED IN THE REDUCTION.
C          ONLY ELEMENTS LOW THROUGH IGH ARE USED.
C
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C
C     ------------------------------------------------------------------
C
      LA = IGH - 1
      KP1 = LOW + 1
      IF (LA .LT. KP1) GO TO 200
C
      DO 180 M = KP1, LA
         MM1 = M - 1
         X = 0.0D0
         I = M
C
         DO 100 J = M, IGH
```

```
                IF (DABS(A(J,MM1)) .LE. DABS(X)) GO TO 100
                X = A(J,MM1)
                I = J
  100       CONTINUE
C
            INT(M) = I
            IF (I .EQ. M) GO TO 130
C     :::::::::: INTERCHANGE ROWS AND COLUMNS OF A ::::::::::
            DO 110 J = MM1, N
                Y = A(I,J)
                A(I,J) = A(M,J)
                A(M,J) = Y
  110       CONTINUE
C
            DO 120 J = 1, IGH
                Y = A(J,I)
                A(J,I) = A(J,M)
                A(J,M) = Y
  120       CONTINUE
C     :::::::::: END INTERCHANGE ::::::::::
  130       IF (X .EQ. 0.0D0) GO TO 180
            MP1 = M + 1
C
            DO 160 I = MP1, IGH
                Y = A(I,MM1)
                IF (Y .EQ. 0.0D0) GO TO 160
                Y = Y / X
                A(I,MM1) = Y
C
                DO 140 J = M, N
  140           A(I,J) = A(I,J) - Y * A(M,J)
C
                DO 150 J = 1, IGH
  150           A(J,M) = A(J,M) + Y * A(J,I)
C
  160       CONTINUE
C
  180 CONTINUE
C
  200 RETURN
C     :::::::::: LAST CARD OF ELMHES ::::::::::
      END
```

End of file

```
      END
C NAASA  3.2.010 HQR      FTN   06-24-75     THE UNIV OF MICH COMP CTR
C
C     ------------------------------------------------------------------
C
      SUBROUTINE HQR(NM,N,LOW,IGH,H,WR,WI,IERR)
C
      INTEGER I,J,K,L,M,N,EN,LL,MM,NA,NM,IGH,ITS,LOW,MP2,ENM2,IERR
      REAL*8 H(NM,N),WR(N),WI(N)
      REAL*8 P,Q,R,S,T,W,X,Y,ZZ,NORM,MACHEP
      REAL*8 DSQRT,DABS,DSIGN
      INTEGER MINO
      LOGICAL NOTLAS
C
C     THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE HQR,
C     NUM. MATH. 14, 219-231(1970) BY MARTIN, PETERS, AND WILKINSON.
C     HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 359-371(1971).
C
C     THIS SUBROUTINE FINDS THE EIGENVALUES OF A REAL
C     UPPER HESSENBERG MATRIX BY THE QR METHOD.
C
C     ON INPUT:
C
C        NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
C          ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
C          DIMENSION STATEMENT;
C
C        N IS THE ORDER OF THE MATRIX;
C
C        LOW AND IGH ARE INTEGERS DETERMINED BY THE BALANCING
C          SUBROUTINE  BALANC.  IF  BALANC  HAS NOT BEEN USED,
C          SET LOW=1, IGH=N;
C
C        H CONTAINS THE UPPER HESSENBERG MATRIX.  INFORMATION ABOUT
C          THE TRANSFORMATIONS USED IN THE REDUCTION TO HESSENBERG
C          FORM BY  ELMHES  OR  ORTHES, IF PERFORMED, IS STORED
C          IN THE REMAINING TRIANGLE UNDER THE HESSENBERG MATRIX.
C
C     ON OUTPUT:
C
C        H HAS BEEN DESTROYED.  THEREFORE, IT MUST BE SAVED
C          BEFORE CALLING  HQR  IF SUBSEQUENT CALCULATION AND
C          BACK TRANSFORMATION OF EIGENVECTORS IS TO BE PERFORMED;
C
C        WR AND WI CONTAIN THE REAL AND IMAGINARY PARTS,
C          RESPECTIVELY, OF THE EIGENVALUES.  THE EIGENVALUES
C          ARE UNORDERED EXCEPT THAT COMPLEX CONJUGATE PAIRS
C          OF VALUES APPEAR CONSECUTIVELY WITH THE EIGENVALUE
C          HAVING THE POSITIVE IMAGINARY PART FIRST.  IF AN
C          ERROR EXIT IS MADE, THE EIGENVALUES SHOULD BE CORRECT
C          FOR INDICES IERR+1,...,N;
C
C        IERR IS SET TO
C          ZERO       FOR NORMAL RETURN,
C          J          IF THE J-TH EIGENVALUE HAS NOT BEEN
C                     DETERMINED AFTER 30 ITERATIONS.
C
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C
```

```
C     ------------------------------------------------------------------
C
C     :::::::::: MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING
C                THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.
C                MACHEP = 16.0D0**(-13) FOR LONG FORM ARITHMETIC
C                ON S360 ::::::::::
      DATA MACHEP/Z3410000000000000/
C
      IERR = 0
      NORM = 0.0D0
      K = 1
C     :::::::::: STORE ROOTS ISOLATED BY BALANC
C                AND COMPUTE MATRIX NORM ::::::::::
      DO 50 I = 1, N
C
         DO 40 J = K, N
   40    NORM = NORM + DABS(H(I,J))
C
         K = I
         IF (I .GE. LOW .AND. I .LE. IGH) GO TO 50
         WR(I) = H(I,I)
         WI(I) = 0.0D0
   50 CONTINUE
C
      EN = IGH
      T = 0.0D0
C     :::::::::: SEARCH FOR NEXT EIGENVALUES ::::::::::
   60 IF (EN .LT. LOW) GO TO 1001
      ITS = 0
      NA = EN - 1
      ENM2 = NA - 1
C     :::::::::: LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT
C                FOR L=EN STEP -1 UNTIL LOW DO -- ::::::::::
   70 DO 80 LL = LOW, EN
         L = EN + LOW - LL
         IF (L .EQ. LOW) GO TO 100
         S = DABS(H(L-1,L-1)) + DABS(H(L,L))
         IF (S .EQ. 0.0D0) S = NORM
         IF (DABS(H(L,L-1)) .LE. MACHEP * S) GO TO 100
   80 CONTINUE
C     :::::::::: FORM SHIFT ::::::::::
  100 X = H(EN,EN)
      IF (L .EQ. EN) GO TO 270
      Y = H(NA,NA)
      W = H(EN,NA) * H(NA,EN)
      IF (L .EQ. NA) GO TO 280
      IF (ITS .EQ. 30) GO TO 1000
      IF (ITS .NE. 10 .AND. ITS .NE. 20) GO TO 130
C     :::::::::: FORM EXCEPTIONAL SHIFT ::::::::::
      T = T + X
C
      DO 120 I = LOW, EN
  120 H(I,I) = H(I,I) - X
C
      S = DABS(H(EN,NA)) + DABS(H(NA,ENM2))
      X = 0.75D0 * S
      Y = X
      W = -0.4375D0 * S * S
  130 ITS = ITS + 1
```

```
C                    SUB-DIAGONAL ELEMENTS.
C                    FOR M=EN-2 STEP -1 UNTIL L DO -- ::::::::::
         DO 140 MM = L, ENM2
            M = ENM2 + L - MM
            ZZ = H(M,M)
            R = X - ZZ
            S = Y - ZZ
            P = (R * S - W) / H(M+1,M) + H(M,M+1)
            Q = H(M+1,M+1) - ZZ - R - S
            R = H(M+2,M+1)
            S = DABS(P) + DABS(Q) + DABS(R)
            P = P / S
            Q = Q / S
            R = R / S
            IF (M .EQ. L) GO TO 150
            IF (DABS(H(M,M-1)) * (DABS(Q) + DABS(R)) .LE. MACHEP * DABS(P)
     X       * (DABS(H(M-1,M-1)) + DABS(ZZ) + DABS(H(M+1,M+1)))) GO TO 150
  140    CONTINUE
C
  150    MP2 = M + 2
C
         DO 160 I = MP2, EN
            H(I,I-2) = 0.0D0
            IF (I .EQ. MP2) GO TO 160
            H(I,I-3) = 0.0D0
  160    CONTINUE
C        :::::::::: DOUBLE QR STEP INVOLVING ROWS L TO EN AND
C                    COLUMNS M TO EN ::::::::::
         DO 260 K = M, NA
            NOTLAS = K .NE. NA
            IF (K .EQ. M) GO TO 170
            P = H(K,K-1)
            Q = H(K+1,K-1)
            R = 0.0D0
            IF (NOTLAS) R = H(K+2,K-1)
            X = DABS(P) + DABS(Q) + DABS(R)
            IF (X .EQ. 0.0D0) GO TO 260
            P = P / X
            Q = Q / X
            R = R / X
  170       S = DSIGN(DSQRT(P*P+Q*Q+R*R),P)
            IF (K .EQ. M) GO TO 180
            H(K,K-1) = -S * X
            GO TO 190
  180       IF (L .NE. M) H(K,K-1) = -H(K,K-1)
  190       P = P + S
            X = P / S
            Y = Q / S
            ZZ = R / S
            Q = Q / P
            R = R / P
C        :::::::::: ROW MODIFICATION ::::::::::
            DO 210 J = K, EN
               P = H(K,J) + Q * H(K+1,J)
               IF (.NOT. NOTLAS) GO TO 200
               P = P + R * H(K+2,J)
               H(K+2,J) = H(K+2,J) - P * ZZ
  200          H(K+1,J) = H(K+1,J) - P * Y
               H(K,J) = H(K,J) - P * X
  210       CONTINUE
```

```
C
         J = MINO(EN,K+3)
C     :::::::::: COLUMN MODIFICATION ::::::::::
         DO 230 I = L, J
            P = X * H(I,K) + Y * H(I,K+1)
            IF (.NOT. NOTLAS) GO TO 220
            P = P + ZZ * H(I,K+2)
            H(I,K+2) = H(I,K+2) - P * R
  220       H(I,K+1) = H(I,K+1) - P * Q
            H(I,K) = H(I,K) - P
  230    CONTINUE
C
  260 CONTINUE
C
      GO TO 70
C     :::::::::: ONE ROOT FOUND ::::::::::
  270 WR(EN) = X + T
      WI(EN) = 0.0D0
      EN = NA
      GO TO 60
C     :::::::::: TWO ROOTS FOUND ::::::::::
  280 P = (Y - X) / 2.0D0
      Q = P * P + W
      ZZ = DSQRT(DABS(Q))
      X = X + T
      IF (Q .LT. 0.0D0) GO TO 320
C     :::::::::: REAL PAIR ::::::::::
      ZZ = P + DSIGN(ZZ,P)
      WR(NA) = X + ZZ
      WR(EN) = WR(NA)
      IF (ZZ .NE. 0.0D0) WR(EN) = X - W / ZZ
      WI(NA) = 0.0D0
      WI(EN) = 0.0D0
      GO TO 330
C     :::::::::: COMPLEX PAIR ::::::::::
  320 WR(NA) = X + P
      WR(EN) = X + P
      WI(NA) = ZZ
      WI(EN) = -ZZ
  330 EN = ENM2
      GO TO 60
C     :::::::::: SET ERROR -- NO CONVERGENCE TO AN
C                EIGENVALUE AFTER 30 ITERATIONS ::::::::::
 1000 IERR = EN
 1001 RETURN
C     :::::::::: LAST CARD OF HQR ::::::::::
      END
End of file
```