MLBeam

A Finite Element Analysis Program For
Multi-Layered Wood Beams

Developed By

David R. Bohnhoff, Ph.D., P.E.
University of Wisconsin-Madison
Department of Biological Systems Engineering
460 Henry Mall
Madison, Wisconsin 53706

DEVELOPING A STRUCTURAL ANALOG FOR MLBeam

The first step in modeling a multi-layered, horizontally nail-laminated beam is to develop a structural analog. This is accomplished by vertically sectioning the beam at locations where there is (1) a support, (2) a fastener, (3) an applied load, and (4) a member end. Additional sectionings should be made at any point where displacements are desired. Number the section lines consecutively from left to right.

The next step is to place a node where each section line intersects the centroidal axis of a wood layer. If there are "n" layers and "s" section lines this results in n x s structural nodes. At each section line there are n+2 degrees of freedom. The first degree of freedom is the rotation of all nodes on the section line. The second degree of freedom is the vertical displacement of all nodes on the section line. The third degree of freedom is the horizontal displacement of the node in layer 1. The fourth degree of freedom is the horizontal displacement of the node in layer 2. The ith + 2 degree of freedom is the horizontal displacement of the node in layer i.

In MLBeam, the wood layers between two adjacent section lines make up one wood element. This results in a total number of wood elements that is one less than the total number of section lines.

Each wood element is assigned a configuration number. The configuration number is used to define the width, depth, and MOE of each layer in the element. Two different wood elements are assigned the same configuration number if the layers in each element have the same properties, that is, the width, depth, and MOE of layer 1 are the same in both elements, the width, depth, and MOE of layer 2 are the same in both elements, the width, depth, and MOE of layer 3 are the same in each element, etc.

Nail elements are used to model nails at a section line. A single nail element can be comprised of a number of nails just as long as all the nails are located somewhere on the same section line. The number, type, and location of the nails in a particular nail element is defined by the element's configuration number. A configuration number specifies what load-slip properties will be assigned to all nails in the element, and also specifies how many nails are at each of the wood layer interfaces. If nails with different load-slip properties are located on the same section line, they will have to be modeled with different nail elements. Note that the load-slip behavior of all nails is described by the following Foschi equation:

$$Load = (p_o + p_1*Slip)*[1 - exp(-k*Slip/p_o)]$$

where:

Load = shear force transferred by a single nail
Slip = slip between the layers connected by the nail
p_o, p_1, k = load-slip parameters

When running a linear analysis, the program defaults to the equation: Load = k*slip, that is, load-slip parameter k in the Foschi equation defines the linear stiffness of the nail connection(s).

MLBeam INPUT FORMAT

Following are instructions for creating an input data file for MLBeam. Data files can be given any name that is acceptable to the Disk Operating System (DOS). Once the file is created, the program can be run. This is accomplished by simply typing "ML". The operator will then be prompted for the name of the input file. After a filename is input, an output file is created by the program. The output file will have the same name as the input file with the exception of the filename extension (MLBeam strips off the input filename extension (if used) and replaces it with the extension ".MLB"). A path of directory names can precede the input filename.

In the following discussion, the term "line" refers to one line of type in the input data file. All data are entered free format. Two data entries on the same line must be separated by either a comma or one or more blank spaces. A zero data value must be entered as a "0", a blank space between commas is not acceptable.

All measurements must be input in like units. For example, if the newton is chosen as the unit of force and the meter as the unit of length, all loads must be expressed in newtons, all lengths in meters and all stresses in N/m^2.

STEP 1 Title

First 2 lines - any heading desired, maximum of 80 characters per line.

STEP 2 Program Control Data

Next line - NLAYER,NUMNOD,NWOOD,NNAIL,NNZD,NZD,NCL,NDWE,NDWP,NDNE,NNLSP

NLAYER is the number of wood layers.

NUMNOD is the number of "vertical" section lines used to divide the beam.

NWOOD is the total number of wood elements.

NNAIL is the total number of nail elements.

NNZD is the number of degrees of freedom with prescribed nonzero displacements.

NZD is the number of degrees of freedom with zero displacement.

NCL is the number of degrees of freedom with concentrated loads.

NDWE is the number of different wood element configurations.

NDWP is the number of different wood member properties. Wood members (i.e., layers) have different properties if one or more of the following properties differ: width, depth, or MOE.

NDNE is the number of different nail element configurations. Two nail elements differ in configuration if they contain a different number of connectors (nails), if the connectors are not identically located with respect to individual layers, or if the load-slip parameters of the connectors differ. NNLSP is the number of sets of load-slip parameters that will be input for nail elements.

STEP 3 X Coordinates of Section Lines

Next NUMNOD lines - N,XCORD

N is the section line number (input for user convenience). Section line coordinates must be entered in order beginning with section line 1.

XCORD is the X coordinate of the section line. The X coordinate axis always runs parallel to the beam.

STEP 4 Wood Element Data

Next NWOOD lines - N,MAT(N),LINK(1),LINK(2)

N is the element number (input for user convenience). Element data must be entered in order beginning with element number 1.

MAT(N) is wood element configuration number.

LINK(1) and LINK(2) are the section line numbers to the left and right of wood element, respectively.

STEP 5 Nail Element Data

Next NNAIL lines - N,MAT(N),LINK(1)

N is the element number (input for user convenience). Element data must be entered in order beginning with element number 1.

MAT(N) is nail element configuration number.

LINK(1) is the number of the section line upon which the nail element lies.

STEP 6 Prescribed Nonzero Displacments

If NNZD=0 go to STEP 7

Next NNZD lines - NSEC,NX,VALUE

NSEC is the number of the section line with the prescribed nonzero displacement.

NX identifies the degree of freedom with the prescribed nonzero displacement.
- NX = -1 for a prescribed rotation of all nodes on section line NSEC
- NX = 0 for a prescribed vertical translation of all nodes on section line NSEC
- NX = 1 for a prescribed horizontal translation of layer 1 at section line NSEC
- NX = 2 for a prescribed horizontal translation of layer 2 at section line NSEC
- NX = i for a prescribed horizontal translation of layer i at section line NSEC

VALUE is the value of the prescribed displacement. Be consistent with sign convention when entering these values. Generally, forces and translations are positive upward and to the right, moments and rotations are positive counterclockwise.

STEP 7 Fixed Displacement Degree of Freedoms

If NZD=0 go to STEP 8

Next NZD lines - NSEC,NX

NSEC is the number of the section line with the fixed degree of freedom (i.e., prescribed zero displacement).

NX identifies the degree of freedom that is fixed.
- NX = -1 for no rotation of the nodes on section line NSEC.
- NX = 0 for no vertical translation of the nodes on section line NSEC.
- NX = 1 for no horizontal translation of layer 1 at section line NSEC
- NX = 2 for no horizontal translation of layer 2 at section line NSEC
- NX = i for no horizontal translation of layer i at section line NSEC

STEP 8 Concentrated Loads

If NCL=0 go to STEP 9

Next NCL lines - NSEC,NX,VALUE

NSEC is the number of the section line with the concentrated load.

NX identifies the degree of freedom with the concentrated load.
- NX = -1 for a prescribed moment at section line NSEC
- NX = 0 for a prescribed vertical force at section line NSEC
- NX = 1 for a horizontal force applied to the layer 1 node
- NX = 2 for a horizontal force applied to the layer 1 node
- NX = i for a horizontal force applied to the layer i node

VALUE is the value of the concentrated force. Be consistent with sign convention when entering these values. Generally, forces and translations are positive upward and to the right, moments and rotations are positive counterclockwise.

STEP 9 Wood Element Configurations

Next NDWE lines - N,(MPID(J,N),J=1,NX) where NX is the number of layers

N is the wood element configuration number (input for user convenience). Wood element configurations must be entered in order beginning with configuration 1.

MPID(J,N) is the wood member property number for layer J. This number assigns a set of wood properties (input in STEP 10) to the wood member comprising layer J.

STEP 10 Wood Member Properties

Next NDWP lines - N,(WOOD(J,N),J=1,3)

N is the wood member property number (input for user convenience). Wood member properties must be entered in order beginning with wood member type 1.

WOOD(1,N) is the width of wood member N.

WOOD(2,N) is the depth (vertical dimension) of the wood member N.

WOOD(3,N) is the MOE of wood member N.

STEP 11 Nail Element Configurations
Next NDNE lines - N,(NEC(J,N),J=1,NX)

N is the nail element configuration number (input for user convenience). FFC element configurations must be entered in order beginning with configuration 1.

NEC(1,N) is the load-slip parameter (LSP) number. This number assigns a set of load-slip parameters (input in STEP 12) to all connectors within the element.

NFFC(2,N) is the number of wood element. This element is used to obtain the depth of each layer (an important variable in nail element stiffness calculations).

NFFC(3,N) is the number of connectors (nails) located between the first and second layers.

NFFC(4,N) is the number of connectors (nails) located between the second and third layers.

NFFC(i,N) is the number of connectors (nails) located between the i-2 and i-3 layers.

STEP 12 Load-Slip Parameters For Nail Connections
Next NNLSP lines - I,(FLSP(J,I),J=1,3)

I is the load-slip parameter (LSP) number. Each set of load-slip parameters must be entered in the following order beginning with LSP number 1.

FLSP(1,I) is the p_0 for load-slip parallel to the wood grain.

FLSP(2,I) is the p_1 for load-slip parallel to the wood grain.

FLSP(3,I) is the k for load-slip parallel to the wood grain.

STEP 13 Displacement Output Information
Next line - NNOUT

NNOUT is the number of section lines for which displacements will be output.

STEP 14 Displacement Output Information
If NNOUT=NUMNOD or if NNOUT=0 go to STEP 15
Next NNOUT lines - NOUT(I)

NOUT(I) is one of NNOUT section line numbers for which displacements will be output. Each of the NNOUT section line numbers is entered on a separate line. If all section line displacements are to be output (NNOUT=NUMNOD), this step is skipped.

STEP 15 Wood Element Output Information
Next line - NWEOUT

NWEOUT is the number of wood elements to be included in the program output.

STEP 16 Element Output Information

If NWEOUT=NWOOD or if NWEOUT=0 go to STEP 17

Next NWEOUT lines - LOUT(I)

LOUT(I) is the wood element number of one of the elements which will be included in the program output. Each of the NWEOUT element numbers is entered on a separate line. If all elements are to be included in the program output (NWEOUT=NWOOD) this step is skipped.

STEP 17 Nail Element Output Information

Next line - NNEOUT

NNEOUT is the number of nail elements to be included in the program output.

STEP 18 Element Output Information

If NNEOUT=NNAIL or if NNEOUT=0 go to STEP 19

Next NNEOUT lines - LOUT(I+NWOOD)

LOUT(I+NWOOD) is the nail element number of one of the elements which will be included in the program output. Each of the NNEOUT element numbers is entered on a separate line. If all elements are to be included in the program output (NNEOUT=NNAIL) this step is skipped.

STEP 19 Program Control Data

Next line - ILOAD,NSTEP,NELAS

ILOAD is the number of "load steps" making up the initial load. The values input in STEP 6 and STEP 8 are treated as a single "load step". If ILOAD is equal to 1, the values input in STEPS 6 and 8 will constitute the initial load applied to the structure.

NSTEP is the number of load steps (including the initial load) for which solutions are desired. If, for example, NSTEP=3 and ILOAD=10, solutions will be found for loads equal to 10 x the "load step", 11 x the "load step" and 12 x the "load step".

NELAS is a parameter used to identify the use of a linear or non-linear solution. If NELAS=1, the solution obtained is only linear and corresponds to connectors considered with their initial stiffness. If NELAS=0, the connectors are considered with their complete non-linear response law.

STEP 20 Program Control Data For Non-linear Analysis

If NELAS=1 skip this STEP

Next line - NITER,TOL,CONFAC

NITER is the maximum number of iterations desired per load step. A value of 50 is recommended.

TOL is the tolerance for convergence of the iterative process. A value of 0.0001 is generally adequate.

CONFAC is a factor used to change the initial stiffness of the structure. When nail elements are highly loaded, it is possible that the final stiffness of the structure will be much less than the initial stiffness of the structure. This may slow convergence. A value of CONFAC greater than

1.0 will increase the initial stiffness of the structure, a value less than 1.0 will decrease the initial stiffness of the structure.

Limitations On Input

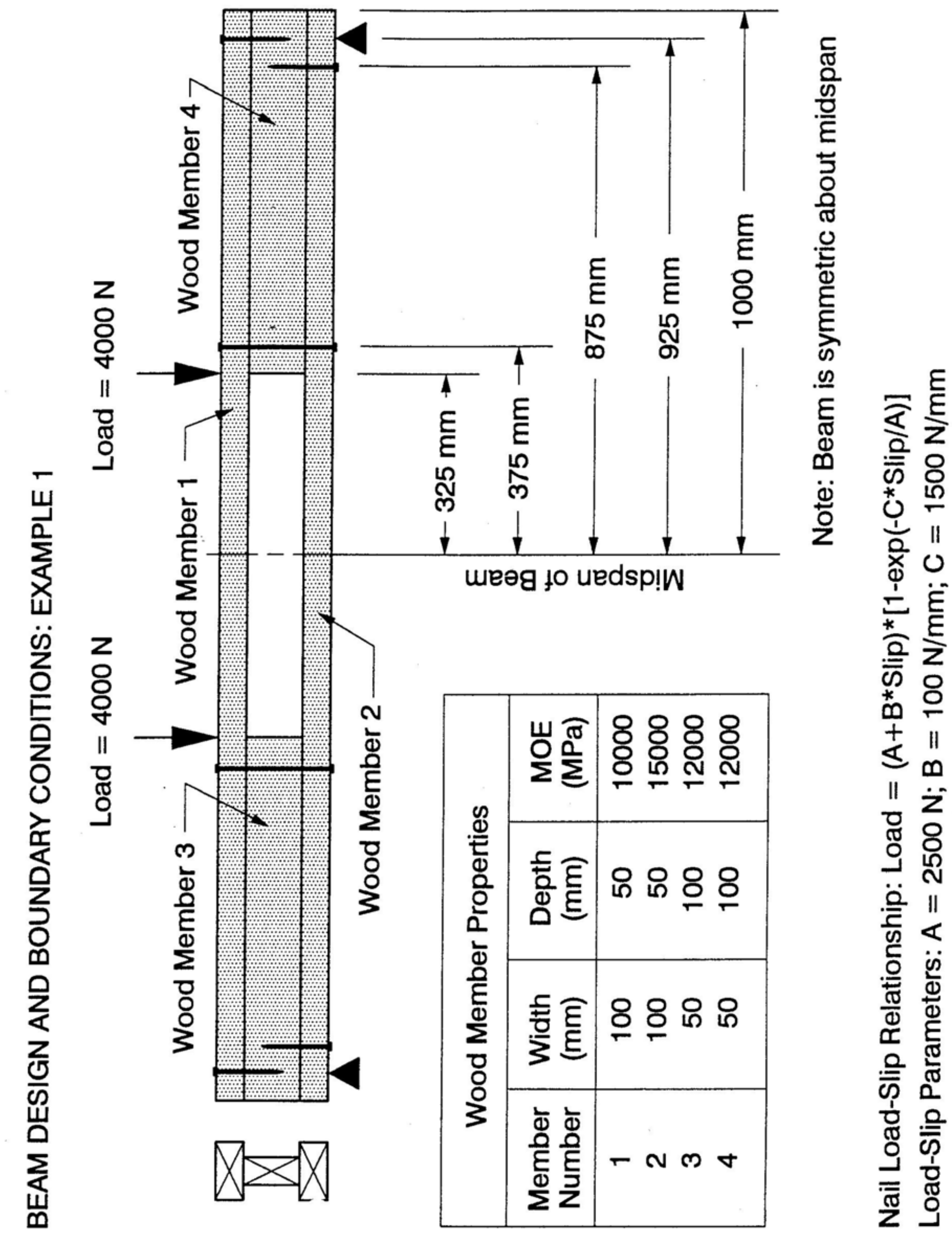
There are two, one-dimensional, primary storage arrays in MLBeam, one is used to store integer values, the other is used to store real numbers. The real array {A} contains nodal coordinates, loads, displacements, and the global stiffness matrix. The integer array {IA} stores information on nodal fixities, element types, element nodes, and element fixities. If either of these two arrays is not large enough to store the required information the following message will appear, "MEMORY ERROR: INSUFFICIENT MEMORY". The integer storage array {IA} is currently dimensioned at 2000 and the real array {A} at 9000. The actual portion of each array that is used during a run is listed in the program output.

Most of the remaining input, such as element properties and element configuration data, is stored in two-dimensional arrays. The pre-dimensioned size of these arrays places limitations on data input. If the size of these arranges is to small, redimension them or rewrite the program placing this data in dynamic storage arrays. Note the following current limitations:

1. Number of wood layers = 8
2. Number of different wood element configurations (MPID) = 8
3. Number of different wood member properties (WOOD) = 8
4. Number of different nail element configurations (NEC) = 8
5. Number of sets of load-slip parameters for nails (FLSP) = 3

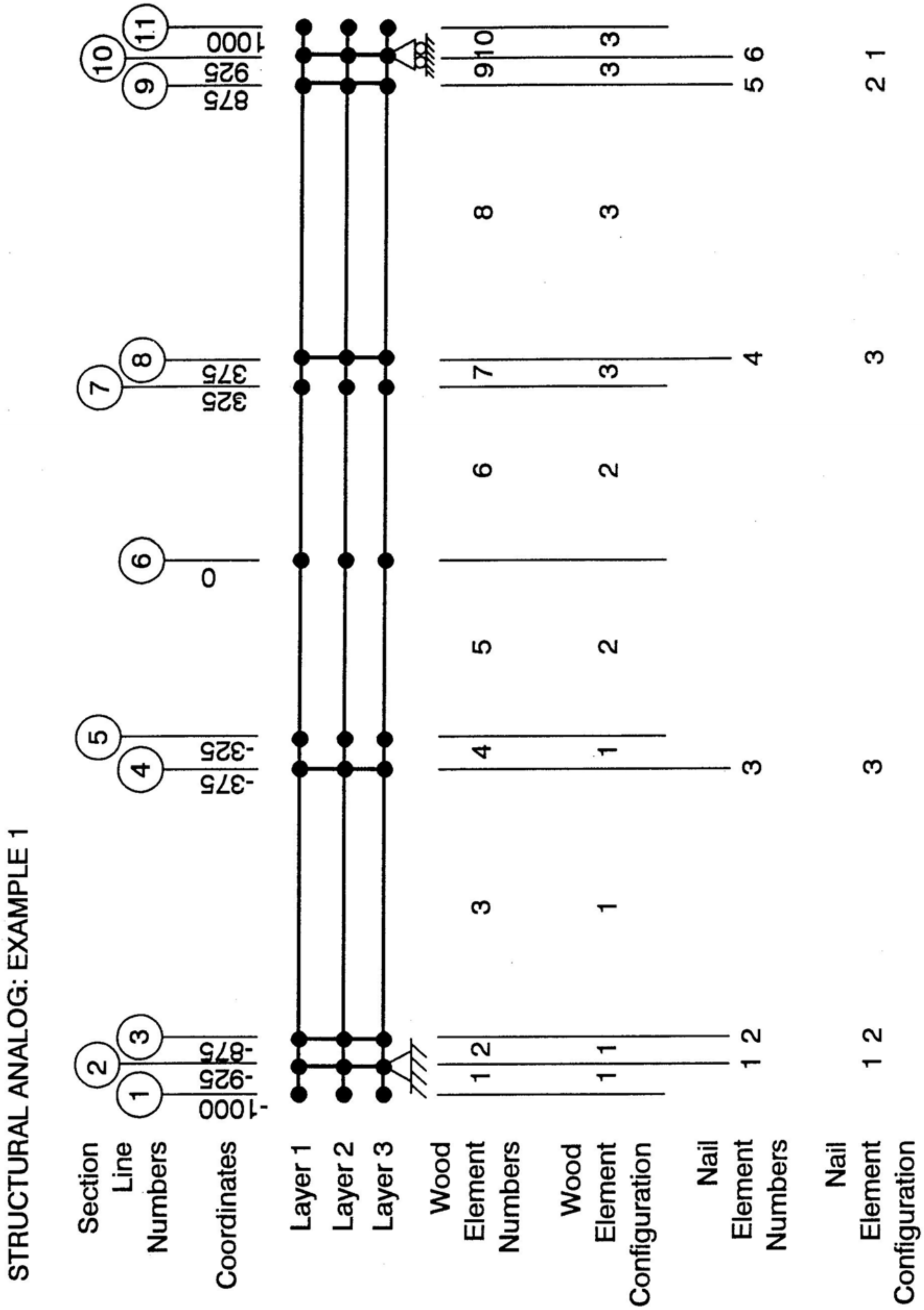
EXAMPLE 1

SCHEMATIC: EXAMPLE 1



Wood Member Properties

Member Number	Width (mm)	Depth (mm)	MOE (MPa)
1	100	50	10000
2	100	50	15000
3	50	100	12000
4	50	100	12000


STRUCTURAL ANALOG: EXAMPLE 1
Section Line Numbers
Coordinates
Layer 1
Layer 2
Layer 3
Wood Element Numbers
Wood Element Configuration
Nail Element Numbers
Nail Element Configuration
-1000
-925
-875
-375
-325
0
325
375
875
925
1000

INPUT FILE: EXAMPLE 1

MLBeam Example 1 - 3-Layer Beam By Dave Bohnhoff for P.A.Favre on 3-15-1996	*Step 1*
3 11 10 6 0 3 2 3 5 3 1	*Step 2*
1 -1000 2 -925 3 -875 4 -375 5 -325 6 0 7 325 8 375 9 875 10 925 11 1000	*Step 3*
1 1 1 2 2 1 2 3 3 1 3 4 4 1 4 5 5 2 5 6 6 2 6 7 7 3 7 8 8 3 8 9 9 3 9 10 10 3 10 11	*Step 4*
1 1 2 2 2 3 3 3 4 4 3 8 5 2 9 6 1 10	*Step 5*
2 0 2 3 10 0	*Step 7*
5 0 -4000 7 0 -4000	*Step 8*
1 1 3 2 2 1 5 2 3 1 4 2	*Step 9*
1 100 50 10000 2 100 50 15000 3 50 100 12000 4 50 100 12000 5 1 1 0	*Step 10*
1 1 1 1 0 2 1 1 0 1 3 1 1 1 1	*Step 11*
1 2500 100 1500	*Step 12*
11	*Step 13*
10	*Step 15*
6	*Step 17*
1 1 0	*Step 19*
100 .0001 .5	*Step 20*

OUTPUT FILE: EXAMPLE 1

```
MLBeam Example 1 – 3-Layer Beam
By Dave Bohnhoff for P.A.Favre on 3-15-1996


PROGRAM CONTROL DATA
---------------------------------------------------------------
Number of wood layers                                         =  3
Number of section lines                                       = 11
Number of wood elements                                       = 10
Number of nail elements                                       =  6
Number of DOF's with a prescribed nonzero displacement        =  0
Number of DOF's that are fixed from displacing                =  3
Number of DOF's with a concentrated load                      =  2
Number of different wood element configurations               =  3
Number of different wood pieces                               =  5
Number of nail connector element configurations               =  3
Number of sets of nail load-slip parameters                   =  1


WOOD ELEMENT DATA
-----------------
Element    Configuration      Section Line Number
Number         Number       Left Side    Right Side
---------------------------------------------------
    1            1               1            2
    2            1               2            3
    3            1               3            4
    4            1               4            5
    5            2               5            6
    6            2               6            7
    7            3               7            8
    8            3               8            9
    9            3               9           10
   10            3              10           11


NAIL ELEMENT DATA
-----------------
Element #  Configuration #    Section Line Number
-------------------------------------------------
    1            1                  2
    2            2                  3
    3            3                  4
    4            3                  8
    5            2                  9
    6            1                 10

SECTION LINE DATA
-----------------
Section #  X Coordinate         Prescribed Displacements
--------------------------------------------------------------
   1      -1000.0000
   2       -925.0000      Y Translation of all layers =    .00000
                             X Translation of layer 3 =    .00000
   3       -875.0000
```

4	-375.0000	
5	-325.0000	
6	.0000	
7	325.0000	
8	375.0000	
9	875.0000	
10	925.0000	Y Translation of all layers = .00000
11	1000.0000	

APPLIED LOADS

Section #	Degree of Freedom	Force
5	Vertical Force =	-4000.000
7	Vertical Force =	-4000.000

WOOD ELEMENT CONFIGURATIONS

Configuration Number	Wood Material Specification For Layer.... 1	2	3	4	5	6	7	8	9
1	1	3	2						
2	1	5	2						
3	1	4	2						

WOOD MATERIAL SPECIFICATIONS

Matl#	Width	Depth	MOE
1	100.0000	50.0000	10000.0000
2	100.0000	50.0000	15000.0000
3	50.0000	100.0000	12000.0000
4	50.0000	100.0000	12000.0000
5	1.0000	1.0000	.0000

NAIL ELEMENT CONFIGURATIONS

Conf#	LSP#	Wood Element Conf#	Number of Identical Connectors On The Same Section Line Between Layers... 1&2	2&3	3&4	4&5	5&6	6&7	7&8	8&9
1	1	1	1	0						
2	1	1	0	1						
3	1	1	1	1						

NAIL LOAD-SLIP PARAMETERS

LSP#	Parallel to Grain M0	M1	K
1	2500.0	100.0	1500.0

PROGRAM CONTROL DATA

Number of load steps comprising the initial load	=	1
Number of load steps	=	1
Non-linear analysis	=	Yes
Maximum number of iterations/load step	=	100
Tolerance for convergence of iterative process	=	1.00E-04
Factor for increasing the initial stiffness of the structure	=	5.00E-01
Length of integer storage array {IA}	=	131
Length of real storage array {A}	=	947

Diagonal value associated with equation 29 set equal to one.

LOAD STEP 1

NUMBER OF ITERATIONS TO CONVERGE = 4

DISPLACEMENT VALUES

Section	Z Rot.	Y Trans.	X Translations (Number in bracket is layer number)		
1	-.0327933	2.45950	.05779(1)	.02647(2)	.00000(3)
2	-.0327933	.00000	.05779(1)	.02647(2)	.00000(3)
3	-.0327975	-1.64032	.05637(1)	.02765(2)	.00000(3)
4	-.0263075	-16.96453	.04218(1)	.02764(2)	.00947(3)
5	-.0250533	-18.24910	.03957(1)	.02764(2)	.01122(3)
6	.0000000	-22.32026	.02256(1)	.00000(2)	.02256(3)
7	.0250533	-18.24910	.00555(1)	.01747(2)	.03390(3)
8	.0263075	-16.96453	.00293(1)	.01747(2)	.03564(3)
9	.0327975	-1.64032	-.01126(1)	.01746(2)	.04511(3)
10	.0327933	.00000	-.01268(1)	.01864(2)	.04511(3)
11	.0327933	2.45950	-.01268(1)	.01864(2)	.04511(3)

WOOD ELEMENT FORCES

Element	Layer	Axial Force (tension +)	Shear Force (/\+\/)	Bending Moment (ccw +) Left End	Right End
1	1	.0000	.0000	.0000	.0000
	2	.0000	.0000	.0000	.0000
	3	.0000	.0000	.0000	.0000
2	1	-1418.9976	547.9452	14578.7424	12818.5178
	2	1418.9976	2630.1370	69977.9637	61528.8856
	3	.0000	821.9178	21868.1137	19227.7767
3	1	-1418.9976	547.9452	1778.5995	272194.0032
	2	-1.7885	2630.1370	8537.2777	1306531.2155
	3	1420.7861	821.9178	2667.8993	408291.0048
4	1	-2616.8187	547.9452	-247599.6156	274996.8759
	2	.0000	2630.1370	-1188478.1548	1319985.0041
	3	2616.8187	821.9178	-371399.4234	412495.3138
5	1	-2616.8187	.0000	-802990.8775	802990.8775
	2	.0000	.0000	.0000	.0000
	3	2616.8187	.0000	-1204486.3162	1204486.3162

6	1	-2616.8187	.0000	-802990.8775	802990.8775
	2	.0000	.0000	.0000	.0000
	3	2616.8187	.0000	-1204486.3162	1204486.3162
7	1	-2616.8187	-547.9452	-274996.8759	247599.6156
	2	.0000	-2630.1370	-1319985.0041	1188478.1548
	3	2616.8187	-821.9178	-412495.3138	371399.4234
8	1	-1418.9976	-547.9452	-272194.0032	-1778.5995
	2	-1.7885	-2630.1370	-1306531.2155	-8537.2777
	3	1420.7861	-821.9178	-408291.0048	-2667.8993
9	1	-1418.9976	-547.9452	-12818.5178	-14578.7424
	2	1418.9976	-2630.1370	-61528.8856	-69977.9637
	3	.0000	-821.9178	-19227.7767	-21868.1137
10	1	.0000	.0000	.0000	.0000
	2	.0000	.0000	.0000	.0000
	3	.0000	.0000	.0000	.0000

MAXIMUM WOOD STRESSES

Elmt.	Layer	Axial Comp.(-) Tension(+)	Shear	Bending Left End (Absolute)	Bending Right End (Absolute)	Axial & Bending
1	1	.0	.0	.0	.0	.0
	2	.0	.0	.0	.0	.0
	3	.0	.0	.0	.0	.0
2	1	-.3	.2	.3	.3	-.6
	2	.3	.8	.8	.7	1.1
	3	.0	.2	.5	.5	-.5
3	1	-.3	.2	.0	6.5	-6.8
	2	.0	.8	.1	15.7	-15.7
	3	.3	.2	.1	9.8	10.1
4	1	-.5	.2	5.9	6.6	-7.1
	2	.0	.8	14.3	15.8	-15.8
	3	.5	.2	8.9	9.9	10.4
5	1	-.5	.0	19.3	19.3	-19.8
	2	.0	.0	.0	.0	.0
	3	.5	.0	28.9	28.9	29.4
6	1	-.5	.0	19.3	19.3	-19.8
	2	.0	.0	.0	.0	.0
	3	.5	.0	28.9	28.9	29.4
7	1	-.5	-.2	6.6	5.9	-7.1
	2	.0	-.8	15.8	14.3	15.8
	3	.5	-.2	9.9	8.9	10.4
8	1	-.3	-.2	6.5	.0	-6.8
	2	.0	-.8	15.7	.1	-15.7
	3	.3	-.2	9.8	.1	10.1
9	1	-.3	-.2	.3	.3	-.6
	2	.3	-.8	.7	.8	1.1
	3	.0	-.2	.5	.5	.5
10	1	.0	.0	.0	.0	.0
	2	.0	.0	.0	.0	.0
	3	.0	.0	.0	.0	.0

NAIL ELEMENT FORCES

Element	Nails Between Layers-	Number of Nails per Interface	Interlayer Slip	Shear Force per Nail
1	1 & 2	1	-2.42818	2103.862
	2 & 3	0	-2.43303	
2	1 & 2	0	-2.43109	
	2 & 3	1	-2.43216	2105.694
3	1 & 2	1	-1.95852	1863.416
	2 & 3	1	-1.95490	1861.355
4	1 & 2	1	1.95852	1863.416
	2 & 3	1	1.95490	1861.355
5	1 & 2	0	2.43109	
	2 & 3	1	2.43216	2105.694
6	1 & 2	1	2.42818	2103.862
	2 & 3	0	2.43303	

COMMENTS: EXAMPLE 1

1. Layer 2 of wood elements 5 and 6 does not exist. To handle this, a dummy member with a depth of 1 mm, width of 1 mm, and <u>zero</u> modulus of elasticity (MOE) was used (see wood property member 5). Note that this method can be used to model butt joints between wood members.

2. Because layer 2 does not exist in the middle of the beam, there really is no layer 2 node on section line 6. Another way to look at this is that there is a node but it is not attached to anything. Because this node is not attached, the diagonal value in the global stiffness matrix that is associated with this degree of freedom will remain null (i.e., stay equal to zero). Unless this zero value is set equal to a nonzero values, the structural equations cannot be solved for the unknown displacement. MLBeam recognizes this and automatically sets any zero diagonal terms equal to 1. A note is placed in the output file to alert the user that this was done (i.e., "Diagonal value associated with equation 29 set equal to one.").

3. Because this beam was symmetric about the midspan with respect to both design and boundary conditions, it was not really necessary to model the whole beam. Instead, the beam could have been split at midspan, and half the beam modeled. To enable this to work, all degrees of freedom on the midspan section line, with the exception of the vertical translation DOF, must be fixed.

MLBeam FORTRAN Source Code

```
C *******************************************************************
C *                                                                 *
C *                            MLBeam                               *
C *                                                                 *
C *       Finite Element Analysis of a Multi-Layered Wood Beam      *
C *            (Layers joined with nails which exhibit              *
C *                    nonlinear load-slip behavior)                *
C *       Written by Dave Bohnhoff                 Spring 1988      *
C *                                                                 *
C *******************************************************************
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z),INTEGER(I-N)
      COMMON IA(2000),A(9000)
      COMMON /MPOINT/ MPNOD,MPFEXT,MPDISP,MPSTIF,MPEND,MAXREL,
     *                IPEQN,IPLINK,IPMAT,IPEND,MAXINT,
     *                MPSTEP,MPLOAD,MPMDIS,MPLDIS,MPESM
      COMMON /KONTRL/ NUMNOD,NDOF,NELE,NWOOD,NNAIL,NNPE,NSD,NEQ,
     *                IBAND,NDIM
      COMMON /LIMITS/ ILOAD,NSTEP,NITER,NELAS,TOL,CONFAC
      COMMON /WOODPROP/ NDWE,NDWP,MPID(9,8),WOOD(3,8)
      COMMON /NAILPROP/ NDNE,NNLSP,FLSP(3,3),NEC(10,8)
      COMMON /DEVICE/ IIN,IOUT,IBUG,NNOUT,NWEOUT,NNEOUT
      CHARACTER CARD1*80,CARD2*80,DATAIN*30,OUTPUT*30
C
C *** Read name of data file, create output file, open files.
C
      IIN=5
      IOUT=6
      WRITE(*,'(5X,18HINPUT FILE NAME ? ,\)')
      READ(*,'(A30)')DATAIN
      OPEN(IIN,FILE=DATAIN)
      DO 2 I=1,30
      IF(DATAIN(I:I).EQ.'.'.OR.DATAIN(I:I).EQ.' ') GOTO 3
    2 CONTINUE
    3 OUTPUT=DATAIN
      OUTPUT(I:I+3) = '.MLB'
      WRITE(*,4)OUTPUT
    4 FORMAT(5X,'OUTPUT FILE NAME = ',A30)
      OPEN (IOUT,FILE=OUTPUT)
C
C *** Read and write title cards and control data.
C
      READ(IIN,1000)CARD1
      READ(IIN,1000)CARD2
      WRITE(IOUT,2000)CARD1,CARD2
      READ(IIN,*)NLAYER,NUMNOD,NWOOD,NNAIL,NNZD,NZD,NCL,
     *                NDWE,NDWP,NDNE,NNLSP
      WRITE(IOUT,2001)
      WRITE(IOUT,2002)NLAYER,NUMNOD,NWOOD,NNAIL,NNZD,NZD,NCL,
     *                NDWE,NDWP,NDNE,NNLSP
C
C *** Initialize variables.
C
      MAXREL=9000
      MAXINT=2000
```

```
      IBUG=1
      NDOF=NLAYER+2
      NNPE=2
      NSD=1
      NELE=NWOOD+NNAIL
      NEQ=NUMNOD*NDOF
      NDIM=NDOF*NNPE
C
C *** Initialize memory pointers for arrays IA and A.
C
      MPNOD=1
      MPFEXT=MPNOD+NUMNOD*NSD
      MPSTEP=MPFEXT+NUMNOD*NDOF
      MPLOAD=MPSTEP+NUMNOD*NDOF
      MPDISP=MPLOAD+NUMNOD*NDOF
      MPLDIS=MPDISP+NUMNOD*NDOF
      MPMDIS=MPLDIS+NUMNOD*NDOF
      MPESM=MPMDIS+NUMNOD*NDOF
      MPSTIF=MPESM+NDIM*NDIM
      MPEND=MPSTIF
C
      IPEQN=1
      IPLINK=IPEQN+NUMNOD*NDOF
      IPMAT=IPLINK+NELE*NNPE
      IPNOUT=IPMAT+NELE
      IPEOUT=IPNOUT+NUMNOD
      IPEND=IPEOUT+NELE
C
C *** Check available memory.
C
      CALL MCHECK(MPEND,IPEND,MAXREL,MAXINT)
C
C *** Complete data input.
C
      CALL INPUT(A(MPNOD),A(MPSTEP),IA(IPEQN),IA(IPLINK),IA(IPMAT),
     *    IA(IPNOUT),IA(IPEOUT),NNZD,NZD,NCL)
C
C *** Determine equation bandwidth.
C
      CALL EQBAND(IA(IPEQN),IA(IPLINK))
C
C *** Allocate storage for stiffness matrix.
C
      LENGTH=NEQ+(IBAND-1)*(2*NEQ-IBAND)/2
      MPEND=MPSTIF+LENGTH
      WRITE(IOUT,2003)IPEND
      WRITE(IOUT,2004)MPEND
C
C *** Check available memory.
C
      CALL MCHECK(MPEND,IPEND,MAXREL,MAXINT)
C
C *** Clear stiffness.
C
      DO 10 I=1,LENGTH
   10 A(MPSTIF+I-1)=0.
```

```
C
C *** Form stiffness matrix.
C
      IF(NELAS.EQ.1) CONFAC=1.0D0
      CALL WOODSTF(A(MPNOD),IA(IPEQN),IA(IPLINK),IA(IPMAT),A(MPSTIF),
     *             A(MPESM))
      CALL NAILSTF(A(MPNOD),IA(IPEQN),IA(IPLINK),IA(IPMAT),A(MPSTIF),
     *             A(MPESM))
C
C *** Modify stiffness and load vector to account for prescribed displ.
C
      CALL MODIFY(IA(IPEQN),A(MPSTEP),A(MPSTIF))
C
C *** Factorize stiffness.
C
      IOP=0
      EPS=1.E-06
      MUD=IBAND-1
      ZERO=0.
      CALL MCHB(ZERO,A(MPSTIF),NEQ,0,MUD,IOP,EPS,IER)
      IF(IER.EQ.0) GO TO 20
C *** Error in factorization.
      CALL WI('IERFACT',IER,1)
   20 CONTINUE
C
C *** Transfer external force to displacement array.
C
      DO 30 I=1,NEQ
   30 A(MPLDIS+I-1)=A(MPSTEP+I-1)
C
C *** Solve simultaneous equations for linear-elastic "step"
C *** displacements.
C
      IOP=-1
      CALL MCHB(A(MPLDIS),A(MPSTIF),NEQ,1,MUD,IOP,EPS,IER)
      IF(IER.EQ.0) GO TO 40
C *** Error in back substitution.
      CALL WI('IER-BACK',IER,1)
   40 CONTINUE
C
C *** Loop over the number of load steps. Calculate current linear
C *** displacements and loads from step disp. and step load vectors.
C
      DO 50 LS=1,NSTEP
      WRITE(IOUT,2005)LS
      DO 60 I=1,NEQ
      A(MPDISP+I-1)=A(MPLDIS+I-1)*(ILOAD+LS-1)
   60 A(MPFEXT+I-1)=A(MPSTEP+I-1)*(ILOAD+LS-1)
C
C *** Output displacement if only a linear-elastic analysis.
C
      IF(NELAS.EQ.1) GOTO 130
      IBETA = 0
C
C *** Begin non-linear analysis.
C
      DO 80 ITER=1,NITER
      WRITE(*,'(3X,I2,2X,1H-,I4)')LS,ITER
```

```
C
C *** Zero and then assemble load modifying vector.
C
      DO 90 I=1,NEQ
   90 A(MPLOAD+I-1)=0.D0
      CALL NAILNLB(A(MPNOD),IA(IPLINK),IA(IPMAT),A(MPDISP),A(MPLOAD))
C
C *** Zero loads associated with prescribed DOF's.  Add modifying load
C *** vector to external load vector and transfer sum to disp. vector.
C
      DO 100 I=1,NEQ
      IF(IA(IPEQN+I-1).LE.0) A(MPLOAD+I-1)=0.D0
  100 A(MPMDIS+I-1)=A(MPLOAD+I-1)+A(MPFEXT+I-1)
C
C *** Solve simultaneous equations.
C
      IOP=-1
      CALL MCHB(A(MPMDIS),A(MPSTIF),NEQ,1,MUD,IOP,EPS,IER)
      IF(IER.EQ.0) GOTO 110
C *** Error in back substitution
      CALL WI('IER-BACK',IER,1)
  110 CONTINUE
C
C *** Check convergence.
C
      CALL CONVRG(A(MPMDIS),A(MPDISP),CHECK)
      IF(CHECK.LE.0.D0) THEN
           DO 125 I=1,NEQ
  125      A(MPDISP+I-1)=A(MPMDIS+I-1)
           WRITE(IOUT,2030)ITER
           GOTO 130
      END IF
      IF(ITER.EQ.NITER) THEN
           WRITE(IOUT,2035)ITER
           GOTO 200
      END IF
C
C *** Update displacements.
C
      DO 129 I=1,NEQ
  129 A(MPDISP+I-1) = A(MPMDIS+I-1)
C
   80 CONTINUE
C
C *** Output nodal displacements.
C
  130 IF(NNOUT.EQ.0) GOTO 145
      WRITE(IOUT,2040)
      NNN=NDOF-1
      DO 140 I=1,NUMNOD
      IF(IA(IPNOUT+I-1).NE.1) GOTO 140
      LL=MPDISP+I*NDOF
      IF(NDOF.LE.5)WRITE(IOUT,2050)I,A(LL-NDOF),(A(LL-NDOF+J),J=1,NNN)
      IF(NDOF.LE.8.AND.NDOF.GT.5)
     *             WRITE(IOUT,2051)I,A(LL-NDOF),(A(LL-NDOF+J),J=1,NNN)
      IF(NDOF.LE.11.AND.NDOF.GT.8)
     *             WRITE(IOUT,2052)I,A(LL-NDOF),(A(LL-NDOF+J),J=1,NNN)
  140 CONTINUE
```

```
C
C *** Post-process data.
C
  145 IF(NWEOUT.EQ.0) GOTO 150
      CALL WOODFOR(A(MPNOD),IA(IPLINK),IA(IPMAT),IA(IPEOUT),A(MPDISP))
  150 IF(NNEOUT.EQ.0) GOTO 50
      CALL NAILFOR(A(MPNOD),IA(IPLINK),IA(IPMAT),IA(IPEOUT),A(MPDISP))
C
   50 CONTINUE
  200 STOP
 1000 FORMAT(A80)
 1001 FORMAT(6I5)
 2000 FORMAT(/,5X,1H ,A80,/,5X,1H ,A80)
 2001 FORMAT(//,5X,21H PROGRAM CONTROL DATA,/,6X,64(1H-))
 2002 FORMAT(6X,21HNumber of wood layers,39X,1H=,I3,/,6X,
     *   23HNumber of section lines,37X,1H=,I3,/,6X,
     *   23HNumber of wood elements,37X,1H=,I3,/,6X,
     *   23HNumber of nail elements,37X,1H=,I3,/,6X,
     *   54HNumber of DOF's with a prescribed nonzero displacement,
     *     6X,1H=,I3,/,6X,
     *   46HNumber of DOF's that are fixed from displacing,
     *     14X,1H=,I3,/,6X,
     *   40HNumber of DOF's with a concentrated load,
     *     20X,1H=,I3,/,6X,
     *   47HNumber of different wood element configurations,
     *     13X,1H=,I3,/,6X,
     *   31HNumber of different wood pieces,
     *     29X,1H=,I3,/,6X,
     *   47HNumber of nail connector element configurations,
     *     13X,1H=,I3,/,6X,
     *   43HNumber of sets of nail load-slip parameters,
     *     17X,1H=,I3)
 2003 FORMAT(6X,36HLength of integer storage array {IA},14X,1H=,I6)
 2004 FORMAT(6X,32HLength of real storage array {A},18X,1H=,I6)
 2005 FORMAT(///,6X,10HLOAD STEP ,I2,/,6X,12(1H-))
 2030 FORMAT(/,6X,35HNUMBER OF ITERATIONS TO CONVERGE = ,I3)
 2035 FORMAT(/,6X,28HCONVERGENCE NOT ACHIEVED IN ,I3,11H ITERATIONS)
 2040 FORMAT(/,6X,19HDISPLACEMENT VALUES,/,6X,19(1H-),/,
     *       6X,7HSection,3X,6HZ Rot.,2X,8HY Trans.,14X,
     *       14HX Translations,/,37X,27H(Number in bracket is layer,
     *       8H number),/,6X,66(1H-))
 2050 FORMAT(7X,I3,F11.7,F11.5,F10.5,'(1)',F10.5,'(2)',F10.5,'(3)')
 2051 FORMAT(7X,I3,F11.7,F11.5,F10.5,'(1)',F10.5,'(2)',F10.5,'(3)',
     *                   /,33X,F10.5,'(4)',F10.5,'(5)',F10.5,'(6)')
 2052 FORMAT(7X,I3,F11.7,F11.5,F10.5,'(1)',F10.5,'(2)',F10.5,'(3)',
     *                   /,33X,F10.5,'(4)',F10.5,'(5)',F10.5,'(6)',
     *                   /,33X,F10.5,'(7)',F10.5,'(8)',F10.5,'(9)')
      END

      SUBROUTINE MCHECK(MPEND,IPEND,MAXREL,MAXINT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z),INTEGER(I-N)
      COMMON /DEVICE/ IIN,IOUT,IBUG,NNOUT,NWEOUT,NNEOUT
      IF(MPEND.GT.MAXREL .OR. IPEND.GT.MAXINT) WRITE(IOUT,100)
      RETURN
  100 FORMAT(//,5X,34H MEMORY ERROR: INSUFFICIENT MEMORY /)
      END
```

```
      SUBROUTINE EQBAND(KFIX,LINK)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z),INTEGER(I-N)
      COMMON /KONTRL/ NUMNOD,NDOF,NELE,NWOOD,NNAIL,NNPE,NSD,NEQ,
     *                IBAND,NDIM
      DIMENSION KFIX(NDOF,1),LINK(NNPE,1)
C
C *** Scan elements to determine the maximum eqn difference of
C *** unconstrained nodes - then compute half bandwidth.
C
      IBAND=0
      DO 20 I=1,NELE
      MEQMIN=10000
      MEQMAX=0
      DO 10 J=1,NNPE
      NODE=LINK(J,I)
      DO 10 K=1,NDOF
      IEQNUM=(NODE-1)*NDOF+K
      IF(IEQNUM.LT.MEQMIN .AND. KFIX(K,NODE).NE.0) MEQMIN=IEQNUM
      IF(IEQNUM.GT.MEQMAX .AND. KFIX(K,NODE).NE.0) MEQMAX=IEQNUM
   10 CONTINUE
      IF(MEQMIN.EQ.10000) GO TO 20
      IF(MEQMAX-MEQMIN+1.GT.IBAND) IBAND=MEQMAX-MEQMIN+1
   20 CONTINUE
      RETURN
      END



      SUBROUTINE INPUT(X,F,KFIX,LINK,MAT,NOUT,LOUT,NNZD,NZD,NCL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z),INTEGER(I-N)
      CHARACTER CHK(2)*3,FIXNOTE(3)*30,TYPEIT*30,FORNOTE(3)*24,
     *          NUMBER(9)*1
      COMMON /KONTRL/ NUMNOD,NDOF,NELE,NWOOD,NNAIL,NNPE,NSD,NEQ,
     *                IBAND,NDIM
      COMMON /LIMITS/ ILOAD,NSTEP,NITER,NELAS,TOL,CONFAC
      COMMON /WOODPROP/ NDWE,NDWP,MPID(9,8),WOOD(3,8)
      COMMON /NAILPROP/ NDNE,NNLSP,FLSP(3,3),NEC(10,8)
      COMMON /DEVICE/ IIN,IOUT,IBUG,NNOUT,NWEOUT,NNEOUT
      DIMENSION X(NSD,1),F(NDOF,1),KFIX(NDOF,1),LINK(NNPE,1),MAT(1),
     *     NOUT(1),LOUT(1)
      DATA CHK/'No ','Yes'/,
     *     FIXNOTE/'     Rotation of all layers = ',
     *             'Y Translation of all layers = ',
     *             '   X Translation of layer   = '/,
     *     FORNOTE/'       Bending Moment = ',
     *             '       Vertical Force = ',
     *             'Horizontal Force (Layer '/
     *     NUMBER/'1','2','3','4','5','6','7','8','9'/
C
C *** Initialize Arrays
C
      DO 1 I=1,NUMNOD
      DO 1 J=1,NDOF
      KFIX(J,I)=1
    1 F(J,I)=0.D0
```

```
C
C *** Read X-coordinates of section lines.
C
      DO 10 I=1,NUMNOD
   10 READ(IIN,*)N,X(1,I)
C
C *** Read and write wood element data.
C
      WRITE(IOUT,2000)
      DO 15 I=1,NWOOD
      READ(IIN,*)N,MAT(I),(LINK(J,I),J=1,NNPE)
      WRITE(IOUT,2010)I,MAT(I),(LINK(J,I),J=1,NNPE)
   15 CONTINUE
C
C *** Read and write nail element data.
C
      WRITE(IOUT,2020)
      DO 20 I=1,NNAIL
      J=I+NWOOD
      READ(IIN,*)N,MAT(J),LINK(1,J)
      LINK(2,J)=LINK(1,J)
      WRITE(IOUT,2030)I,MAT(J),LINK(1,J)
   20 CONTINUE
CC *** Read prescribed nonzero displacements.
C
      IF(NNZD.EQ.0)GOTO 30
      DO 25 I=1,NNZD
      READ(IIN,*)NSEC,NX,VALUE
      NNDOF=NX+2
      KFIX(NNDOF,NSEC)=-1
   25 F(NNDOF,NSEC)=VALUE
C
C *** Read fixed DOF's.
C
   30 DO 35 I=1,NZD
      READ(IIN,*)NSEC,NX
      NNDOF=NX+2
   35 KFIX(NNDOF,NSEC)=0
C
C *** Write section line data (X-coordinate, fixities, etc).
C
      WRITE(IOUT,2040)
      DO 40 I=1,NUMNOD
      NNOTES=0
      DO 45 J=1,NDOF
      IF(KFIX(J,I).EQ.1)GOTO 45
      NNOTES=NNOTES+1
      KK=J
      IF(J.GE.3)KK=3
      TYPEIT=FIXNOTE(KK)
      IF(J.GE.3)TYPEIT(27:27)=NUMBER(J-2)
      IF(NNOTES.EQ.1)WRITE(IOUT,2050)I,X(1,I),TYPEIT,F(J,I)
      IF(NNOTES.GT.1)WRITE(IOUT,2060)TYPEIT,F(J,I)
   45 CONTINUE
      IF(NNOTES.EQ.0)WRITE(IOUT,2070)I,X(1,I)
   40 CONTINUE
```

```
C
C *** Read and write concentrated load values.
C
      IF(NCL.EQ.0)GOTO 48
      WRITE(IOUT,2080)
      DO 47 I=1,NCL
      READ(IIN,*)NSEC,NX,VALUE
      NNDOF=NX+2
      F(NNDOF,NSEC)=VALUE
      IF(NNDOF.EQ.1.OR.NNDOF.EQ.2)WRITE(IOUT,2090)NSEC,FORNOTE(NNDOF),
     *                                                   VALUE
   47 IF(NNDOF.GE.3)WRITE(IOUT,2100)NSEC,FORNOTE(3),NX,VALUE
C
C *** Read and write data on wood element configurations.
C
   48 NX=NDOF-2
      WRITE(IOUT,2110)
      DO 49 I=1,NDWE
      READ(IIN,*)N,(MPID(J,I),J=1,NX)
   49 WRITE(IOUT,2120)I,(MPID(J,I),J=1,NX)
C
C *** Read and write data on wood pieces.
C
      WRITE(IOUT,2130)
      DO 50 I=1,NDWP
      READ(IIN,*)N,(WOOD(J,I),J=1,3)
   50 WRITE(IOUT,2140)I,(WOOD(J,I),J=1,3)
C
C *** Read/write data on nail element configurations.
C
      NX=NDOF-1
      WRITE(IOUT,2150)
      DO 60 I=1,NDNE
           READ(IIN,*)N,(NEC(J,I),J=1,NX)
   60 WRITE(IOUT,2160)N,(NEC(J,I),J=1,NX)
C
C *** Read/write nail load-slip parameters.
C
      WRITE(IOUT,2170)
      DO 70 I=1,NNLSP
      READ(IIN,*)N,(FLSP(J,I),J=1,3)
   70 WRITE(IOUT,2180)I,(FLSP(J,I),J=1,3)
C
C *** Read information on desired output.
C
   75 READ(IIN,*)NNOUT
      IF(NNOUT.EQ.NUMNOD) THEN
           DO 160 I=1,NUMNOD
  160      NOUT(I) = 1
      ELSE
           DO 170 I=1,NUMNOD
  170      NOUT(I) = 0
           DO 180 I=1,NNOUT
           READ(IIN,*) NN
  180      NOUT(NN)=1
      END IF
C
```

```
      READ(IIN,*)NWEOUT
      IF(NWEOUT.EQ.NWOOD) THEN
           DO 190 I=1,NWOOD
  190      LOUT(I) = 1
      ELSE
           DO 200 I=1,NWOOD
  200      LOUT(I) = 0
           DO 210 I=1,NWEOUT
           READ(IIN,*) NN
  210      LOUT(NN) = 1
      END IF
C
      READ(IIN,*)NNEOUT
      IF(NNEOUT.EQ.NNAIL) THEN
           DO 220 I=1,NNAIL
  220      LOUT(I+NWOOD) = 1
      ELSE
           DO 230 I=1,NNAIL
  230      LOUT(I+NWOOD) = 0
           DO 240 I=1,NNEOUT
           READ(IIN,*) NN
  240      LOUT(NN+NWOOD) = 1
      END IF
C
C *** Read and write additional control data.
C
      WRITE(IOUT,2190)
      READ(IIN,*)ILOAD,NSTEP,NELAS
      NEL=1
      IF(NELAS.NE.1) NEL=2
      WRITE(IOUT,2200)ILOAD,NSTEP,CHK(NEL)
      IF(NELAS.EQ.1) GOTO 250
      READ(IIN,*)NITER,TOL,CONFAC
      WRITE(IOUT,2210)NITER,TOL,CONFAC
C
  250 CONTINUE
      RETURN
C
 2000 FORMAT(//,5X,' WOOD ELEMENT DATA',/,6X,17(1H-),/,5X,' Element    ',
     *      'Configuration      Section Line Number',/,5X,' Number     ',
     *      '    Number       Left Side    Right Side',/,6X,51(1H-))
 2010 FORMAT(2X,I9,I13,6X,I10,3X,I10)
 2020 FORMAT(//,5X,' NAIL ELEMENT DATA',/,6X,17(1H-),/,5X,' Element #  ',
     *      'Configuration #    Section Line Number',/,6X,49(1H-))
 2030 FORMAT(2X,I9,I14,10X,I10)
 2040 FORMAT(//,5X,' SECTION LINE DATA',/,6X,17(1H-),/,5X,' Section #  ',
     *     'X Coordinate         Prescribed Displacements ',
     *     /,6X,62(1H-))
 2050 FORMAT(6X,I4,6X,F10.4,6X,A30,F9.5)
 2060 FORMAT(32X,A30,F9.5)
 2070 FORMAT(6X,I4,6X,F10.4)
 2080 FORMAT(//,5X,' APPLIED LOADS',/,6X,13(1H-),/,5X,' Section #  ',
     *     '   Degree of Freedom          Force',/,6X,48(1H-))
 2090 FORMAT(6X,I4,9X,A24,F11.3)
 2100 FORMAT(6X,I4,4X,A24,I1,') = ',F11.3)
 2110 FORMAT(//,5X,' WOOD ELEMENT CONFIGURATIONS',/6X,27(1H-),/,5X,
     *  ' Configuration    Wood Material Specification For Layer....',
     */,8X,'  Number        1    2    3    4    5    6    7    8    9',
```

```
     *   /,6X,59(1H-))
 2120 FORMAT(12X,I2,6X,9I5)
 2130 FORMAT(//,6X,28HWOOD MATERIAL SPECIFICATIONS,/,6X,
     *      28(1H-),/,6X,5HMatl#,5X,5HWidth,6X,5HDepth,
     *      9X,3HMOE,/,6X,43(1H-))
 2140 FORMAT(5X,I4,4X,F9.4,1X,F9.4,2X,F15.4)
 2150 FORMAT(//,6X,27HNAIL ELEMENT CONFIGURATIONS,/,6X,27(1H-),/,23X,
     *    'Wood     Number of Identical Connectors On The',/,5X,
     *    ' Conf#   LSP#    Element     Same Section Line Between',
     *    ' Layers...',/,21X,'  Conf#    1&2  2&3  3&4  4&5  5&6',
     *    '  6&7  7&8  8&9',/,6X,64(1H-))
 2160 FORMAT(8X,I1,6X,I2,7X,I2,3X,8(I5))
 2170 FORMAT(//,5X,' NAIL LOAD-SLIP PARAMETERS',/,6X,25(1H-),/,5X,
     *      ' LSP#                 Parallel to Grain ',/,
     *      16X,'    M0             M1              K',/,6X,48(1H-))
 2180 FORMAT(5X,I4,3(3X,F12.1))
 2190 FORMAT(//,5X,21H PROGRAM CONTROL DATA,/,6X,57(1H-))
 2200 FORMAT(6X,48HNumber of load steps comprising the initial load,2X,
     *      1H=,I6,/,6X,20HNumber of load steps,30X,1H=,I6,/,6X,
     *      19Hnon-linear analysis,31X,1H=,3X,A3)
 2210 FORMAT(6X,38HMaximum number of iterations/load step,12X,1H=,I6,/,
     *      6X,46HTolerance for convergence of iterative process,4X,1H=,
     *      1X,1PE8.2,/,6X,25HFactor for increasing the,/,20X,
     *      34Hinitial stiffness of the structure,2X,1H=,1X,1PE8.2)
      END


      SUBROUTINE ADSTIF(KFIX,S,LINK,T,ITROW,KELE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
      COMMON /KONTRL/ NUMNOD,NDOF,NELE,NWOOD,NNAIL,NNPE,NSD,NEQ,
     *                IBAND,NDIM
      DIMENSION KFIX(1),S(1),T(ITROW,1),LINK(NNPE,1)
C
C *** Assemble element stiffness into upper triangle of global stiff.
C
      DO 100 IGEN=1,NNPE
      DO 100 JGEN=1,NNPE
      IGLB=NDOF*(LINK(IGEN,KELE)-1)
      JGLB=NDOF*(LINK(JGEN,KELE)-1)
C
      DO 100 IDOF=1,NDOF
      DO 100 JDOF=1,NDOF
C
C *** If lower triangular element, skip assembly
C
      IF(IGLB+IDOF.GT.JGLB+JDOF) GO TO 100
C
C *** Enforce boundary condition:
C *** If constrained DOF and nondiagonal term, skip assembly.
C
      IF((KFIX(IGLB+IDOF).EQ.0 .OR. KFIX(JGLB+JDOF).EQ.0)  .AND.
     *   IGLB+IDOF.NE.JGLB+JDOF) GO TO 100
C
      JADD=LLOC(IGLB+IDOF,JGLB+JDOF)
      S(JADD)=S(JADD)+T(NDOF*(IGEN-1)+IDOF,NDOF*(JGEN-1)+JDOF)
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MODIFY(KFIX,F,S)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z),INTEGER(I-N)
      COMMON /KONTRL/ NUMNOD,NDOF,NELE,NWOOD,NNAIL,NNPE,NSD,NEQ,
     *                IBAND,NDIM
      COMMON /DEVICE/ IIN,IOUT,IBUG,NNOUT,NWEOUT,NNEOUT
      DIMENSION KFIX(1),F(1),S(1)
C
C *** Subroutine to modify stiffness matrix and load vector to
C *** account for prescribed displacements.
C
C *** Make sure force is zero for each prescribed zero D.O.F.
C *** Setting zero diagonal values to one.

      DO 5 MEQ=1,NEQ
      IF (KFIX(MEQ) .EQ. 0) F(MEQ) = 0.D0
      IF(S(LLOC(MEQ,MEQ)).GT.0.00001) GOTO 5
      S(LLOC(MEQ,MEQ))=1.0
      WRITE(IOUT,1000)MEQ
 1000 FORMAT(/,5X,' Diagonal value associated with equation ',I3,
     *' set equal to one.')
    5 CONTINUE
C
      DO 100 MEQ=1,NEQ
      IF(KFIX(MEQ).GE.0) GO TO 100
C
C *** First: Modify load vector for equation  MEQ.
C
      MSTART=MEQ-IBAND+1
      IF(MSTART.LT.1) MSTART=1
      MSTOP=MEQ+IBAND-1
      IF(MSTOP.GT.NEQ) MSTOP=NEQ
C
      DO 10 K=MSTART,MSTOP
      IF(K.EQ.MEQ) GO TO 10
      IF(K.LT.MEQ) KLOC=LLOC(K,MEQ)
      IF(K.GT.MEQ) KLOC=LLOC(MEQ,K)
C
C *** Skip modification if equation K is also a prescribed displ.
C
      IF(KFIX(K).GE.0) F(K)=F(K)-S(KLOC)*F(MEQ)
   10 CONTINUE
C
C *** Second: Modify stiffness - zero row and col and leave diag.
C
      DO 30 K=MSTART,MSTOP
      IF(K.NE.MEQ) GO TO 20
C
C *** Diagonal term- modify prescribed displ and leave diag stiff.
C
      F(MEQ)=F(MEQ)*S(LLOC(MEQ,MEQ))
      GO TO 30
C
C *** Off diagonal term - zero stiffness.
C
   20 IF(K.GT.MEQ) S(LLOC(MEQ,K))=0.
      IF(K.LT.MEQ) S(LLOC(K,MEQ))=0.
   30 CONTINUE
C
```

```
  100 CONTINUE
      RETURN
      END



      FUNCTION LLOC(I,J)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
      COMMON /KONTRL/ NUMNOD,NDOF,NELE,NWOOD,NNAIL,NNPE,NSD,NEQ,
     *                IBAND,NDIM
C
C *** Find location in 1-D array of I,J address in 2-D array
C *** (location is based on constant bandwidth upper triangular
C *** column-wise storage).
C
      LLOC=IBAND*(I-1)+J-I+1
C
C *** Adjust address if in last IBAND-2 rows.
C
      IADJST=NEQ-IBAND+2
      IF(I.LE.IADJST) RETURN
      NHT=I-IADJST
      LLOC=LLOC-NHT*(NHT+1)/2
      RETURN
      END



      SUBROUTINE WI(ID,I,IOP)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
      COMMON /DEVICE/ IIN,IOUT,IBUG,NNOUT,NWEOUT,NNEOUT
      CHARACTER*8 ID
      IF(IOP.GT.IBUG) RETURN
      WRITE(IOUT,100)ID,I
  100 FORMAT(6X,A8,1H=,I10)
      RETURN
      END



      SUBROUTINE WR(ID,R,IOP)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
      COMMON /DEVICE/ IIN,IOUT,IBUG,NNOUT,NWEOUT,NNEOUT
      CHARACTER*8 ID
      IF(IOP.GT.IBUG) RETURN
      WRITE(IOUT,100)ID,R
  100 FORMAT(6X,A8,1H=,G15.5)
      RETURN
      END



      SUBROUTINE CONVRG(DISM,DISP,CHECK)
C
C *** Convergence can be considered achieved when CHECK is a
C *** negative number.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER(I-N)
```

```
      COMMON /KONTRL/ NUMNOD,NDOF,NELE,NWOOD,NNAIL,NNPE,NSD,NEQ,
     *                IBAND,NDIM
      COMMON /LIMITS/ ILOAD,NSTEP,NITER,NELAS,TOL,CONFAC
      DIMENSION DISM(1),DISP(1)
C
      EI=0.D0
      SD=0.D0
      DO 10 I=1,NEQ
      EI=EI+(DISM(I)-DISP(I))**2
   10 SD=SD+DISP(I)**2
      EI=SQRT(EI)
      SD=SQRT(SD)*TOL
      CHECK=EI-SD
      RETURN
      END



      SUBROUTINE   MCHB (R,A,M,N,MUD,IOP,EPS,IER)
C
C *** For a given positive definite M by M matrix A with symmetric band
C *** structure and (if necessary) a given general M by N matrix R, the
C *** following calculations (dependent on the value of the decision
C *** parameter IOP) are performed.
C *** (1) Matrix A is factorized (If IOP is not negative), that means
C ***     band matrix TU with upper codiagonals only is generated on the
C ***     locations of A such that TRANSPOSE(TU)*TU = A.
C *** (2) Matrix R is multiplied on the left by INVERSE(TU) and/or
C ***     INVERSE(TRANSPOSE(TU)) and the result is stored in the
C ***     location of R.
C *** This subroutine especially can be used to solve the system of
C *** simultaneous linear equations A*X=R with positive-definite
C *** coefficient matrix A of symmetric band structure.
C
C *** Description of Parameters.
C ***   R    Input  IOP = -3,-2,-1,1,2,3  M by N right hand matrix,
C ***               IOP = 0  irrelevant.
C ***        Output IOP = 1,-1  INVERSE(A)*R,
C ***               IOP = 2,-2  INVERSE(TU)*R,
C ***               IOP = 3,-3  INVERSE(TRANSPOSE(TU))*R,
C ***               IOP = 0  unchanged.
C ***   A    Input  IOP = 0,1,2,3  M x M positive-definite coefficient
C ***                     matrix of symmetric band structure stored in
C ***                     compressed form (see remarks),
C ***               IOP = -1,-2,-3  M x M band matrix TU with upper
C ***                     codiagonals only, stored in compressed form.
C ***        Output in all cases band matrix TU with upper codiagonals
C ***                     only, stored in compressed form (that means
C ***                     unchanged if IOP = -1,-2,-3).
C ***   M    Input value specifying the number of rows and columns of A
C ***        and number of rows of R.
C ***   N    Input value specifying the number of columns of R
C ***        (irrelevant in case IOP = 0).
C ***   MUD  Input value specifying the no. of upper codiagonals of A.
C ***   IOP  Decision parameter (either -3,-2,-1,0,1,2,3).
C ***   EPS  Input value used as relative tolerance for test on loss of
C ***        significant digits.
C ***   IER  Resulting error parameter coded as follows,
```

```
C ***          IER = 0  - No error,
C ***          IER = -1 - No result because of wrong input parameters M,
C ***                     MUD, IOP, or because of nonpositive radicand at
C ***                     some factorization step, or because of zero
C ***                     diagonal element at some division step.
C ***          IER = K  - Warning due to possible loss of significance
C ***                     indicated at factorization step K+1 where radi-
C ***                     cand was no longer greater than EPS*A(K+1,K+1).
C
C *** Remarks.
C *** Upper part of symmetric band matrix A consisting of main diagonal
C *** and MUD upper codiagonals (resp. band matrix TU consisting of main
C *** diagonal and MUD upper codiagonals) is assumed to be stored in
C *** compressed form, i.e. rowwise in totally needed M+MUD*(2M-MUD-1)/2
C *** successive storage locations.  On return upper band factor TU (on
C *** the locations of A) is stored in the same way.  Right hand side
C *** matrix R is assumed to be stored columnwise in N*M successive
C *** storage locations.  On return, INVERSE(A)*R or INVERSE(TU)*R or
C *** INVERSE(TRANSPOSE(TU))*R will be stored in the locations of R.
C *** Input parameters M, MUD, IOP should satisfy the following
C *** restrictions, 1) MUD not less than zero, 2) 1+MUD not greater than
C *** M, 3) ABS(IOP) not greater than 3.  No action besides error mes-
C *** sage IER=-1 takes place if these restrictions are not satisfied.
C *** The procedure gives results if the restrictions on input para-
C *** meters are satisfied, if radicands at all factorization steps are
C *** positive and/or if all diagonal elements of upper band factor TU
C *** are nonzero.  Factorization is done using Cholesky-S square-root
C *** method, which generates the upper band matrix TU such that
C *** TRANSPOSE(TU)*TU=A.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
      DIMENSION R(1),A(1)
C
C *** Test on wrong input parameters.
C
      IF(IABS(IOP)-3)100,100,520
  100 IF(MUD)520,110,110
  110 MC=MUD+1
      IF(M-MC)520,120,120
  120 MR=M-MUD
      IER=0
C
C *** MC is the maximum number of elements in the rows of array A.
C *** MR is the index of the last row in array A with MC elements.
C
C
C *** Start factorization of matrix A.
C
      IF(IOP)330,130,130
  130 IEND=0
      LLDST=MUD
      DO 320 K=1,M
      IST=IEND+1
      IEND=IST+MUD
      J=K-MR
      IF(J)150,150,140
  140 IEND=IEND-J
  150 IF(J-1)170,170,160
```

```
  160 LLDST=LLDST-1
  170 LMAX=MUD
      J=MC-K
      IF(J)190,190,180
  180 LMAX=LMAX-J
  190 ID=0
      TOL=A(IST)*EPS
C
C *** Start factorization-loop over K-th row.
C
      DO 320 I=IST,IEND
      SUM=0.000
      IF(LMAX)230,230,200
CC *** Prepare inner loop.
C
  200 LL=IST
      LLD=LLDST
C
C *** Start inner loop.
C
      DO 220 L=1,LMAX
      LL=LL-LLD
      LLL=LL+ID
      SUM=SUM+A(LL)*A(LLL)
      IF(LLD-MUD)210,220,220
  210 LLD=LLD+1
  220 CONTINUE
C
C *** End of inner loop.
C
C *** Transform element A(I).
C
  230 SUM=A(I)-SUM
      IF(I-IST)240,240,290
C
C *** A(I) is diagonal element. Error test.
C
  240 IF(SUM)540,540,250
C
C *** Test on loss of significant digits and warnings.
C
  250 IF(SUM-TOL)260,260,280
  260 IF(IER)270,270,280
  270 IER=K-1
C
C *** Computation of pivot element.
C
  280 PIV= SQRT(SUM)
      A(I)=PIV
      PIV= 1.0/PIV
      GO TO 300
C
C *** A(I) is not diagonal element.
C
  290 A(I)=SUM*PIV
C
C *** Update ID and LMAX.
C
```

```
  300 ID=ID+1
      IF(ID-J)320,320,310
  310 LMAX=LMAX-1
  320 CONTINUE
C
C *** End of factorization-loop over K-th row.
C *** End of factorization of matrix A.
C
C *** Prepare matrix divisions.
C      IF(IOP)330,530,330
  330 ID=N*M
      IEND=IABS(IOP)-2
      IF(IEND)340,440,340
C
C *** Start division by transpose of matrix TU (TU is stored in
C *** locations of A).
C
  340 IST=1
      LMAX =0
      J =-MR
      LLDST =MUD
      DO 430 K=1,M
      PIV = A(IST)
      IF(PIV)350,560,350
  350 PIV= 1.0/PIV
C
C *** Start backsubstitution-loop for K-th row of matrix R.
C
      DO 390 I=K,ID,M
      SUM=0.000
      IF(LMAX)390,390,360
C
C *** Prepare inner loop.
C
  360 LL=IST
      LLL=I
      LLD=LLDST
C
C *** Start inner loop.
C
      DO 380 L=1,LMAX
      LL=LL-LLD
      LLL=LLL-1
      SUM=SUM+A(LL)*R(LLL)
      IF(LLD-MUD)370,380,380
  370 LLD=LLD+1
  380 CONTINUE
C
C *** End of inner loop.
C
C *** Transform element R(I).
C
  390 R(I)=PIV*(R(I)-SUM)
C
C *** End of back substitution-loop for K-th row of matrix R.
```

```
C
C *** Update parameters LMAX, IST AND LLDST.
C
      IF(MC-K)410,410,400
  400 LMAX=K
  410 IST=IST+MC
      J=J+1
      IF(J)430,430,420
  420 IST=IST-J
      LLDST=LLDST-1
  430 CONTINUEC
C *** End of division by transpose of matrix TU.
C *** Start division by matrix TU (TU is stored on locations of A).
C
      IF(IEND)440,440,530
  440 IST=M+(MUD*(M+M-MC))/2+1
      LMAX=0
      K=M
  450 IEND=IST-1
      IST=IEND-LMAX
      PIV=A(IST)
      IF(PIV)460,520,460
  460 PIV= 1.0/PIV
      L=IST+1
C
C *** Start back substitution-loop for K-th row of matrix R.
C
      DO 490 I=K,ID,M
      SUM=0.000
      IF(LMAX)490,490,470
  470 LLL=I
C
C *** Start inner loop.
C
      DO 480 LL=L,IEND
      LLL=LLL+1
  480 SUM=SUM+A(LL)*R(LLL)
C
C *** End of inner loop.
C
C *** Transform element R(I).
C
  490 R(I)=PIV*(R(I)-SUM)
C
C *** End of back substitution-loop for K-th row of matrix R.
C
C *** Update parameters LMAX and K.
C
      IF(K-MR)510,510,500
  500 LMAX=LMAX+1
  510 K=K-1
      IF(K)530,530,450
C
C *** End of division by matrix TU.
C
C *** Error exit in case of wrong input parameters or pivot element
C *** less than or equal to zero.
C
```

```
  520 IER=-1
  530 RETURN
  540 IER=-2
      GO TO 530
  560 IER=-3
      GO TO 530
      END



      SUBROUTINE WOODSTF(XX,KFIX,LINK,MATNUM,S,ESM)
C
C *** Subroutine locates each wood element and its corresponding
C *** wood members and wood member properties. The element
C *** stiffness matrix is assembled and sent to subroutine ADSTIF.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
      COMMON /KONTRL/ NUMNOD,NDOF,NELE,NWOOD,NNAIL,NNPE,NSD,NEQ,
     *                IBAND,NDIM
      COMMON /WOODPROP/ NDWE,NDWP,MPID(9,8),WOOD(3,8)
      COMMON /DEVICE/ IIN,IOUT,IBUG,NNOUT,NWEOUT,NNEOUT
      DIMENSION XX(NSD,1),KFIX(1),LINK(NNPE,1),MATNUM(1),S(1),X(2),
     *     ESM(NDIM,NDIM),N(2),B(9),D(9),E(9),XI(9),A(9)
C
      NL=NDOF-2
C
C *** Loop over elements - generate [k] and assemble.
C
      DO 100 NN=1,NWOOD
C
C *** Get wood element configuration number (NCONF).  Get wood
C *** properties for all layers in wood element.
C
      NCONF = MATNUM(NN)
      EI=0.0D0
C
      DO 5 NW=1,NL
      NOOD = MPID(NW,NCONF)
      B(NW)=WOOD(1,NOOD)
      D(NW)=WOOD(2,NOOD)
      E(NW)=WOOD(3,NOOD)
      A(NW)=B(NW)*D(NW)
      XI(NW)=(B(NW)*D(NW)**3)/12.0D0
    5 EI=EI+E(NW)*XI(NW)
C
C *** Set initial X coordinate of the section lines and determine
C *** element length.
C
      DO 20 L=1,2
      N(L) = LINK(L,NN)
   20 X(L) = XX(1,N(L))
      H=X(2)-X(1)
      EL=SQRT(H*H)
C
C *** Zero and then assemble wood element stiffness matrix.
C
      DO 30 K = 1,NDIM
      DO 30 L = 1,K
```

```
   30 ESM(K,L)= 0.0D0
C
      ESM(1,1)=4.0D0*EI/EL
      ESM(2,1)=6.0D0*EI/EL/EL
      ESM(2,2)=12.0D0*EI/(EL**3)
      ESM((NDOF+1),1)=ESM(1,1)/2.0d0
      ESM((NDOF+1),2)=ESM(2,1)
      ESM((NDOF+1),(NDOF+1))=ESM(1,1)
      ESM((NDOF+2),1)=-ESM(2,1)
      ESM((NDOF+2),2)=-ESM(2,2)
      ESM((NDOF+2),(NDOF+1))=-ESM(2,1)
      ESM((NDOF+2),(NDOF+2))=ESM(2,2)
C
      DO 33 NW=1,NL
      AEOL=A(NW)*E(NW)/EL
      ESM((NW+2),(NW+2))=AEOL
      ESM((NDOF+NW+2),(NW+2))=-AEOL
  33  ESM((NDOF+NW+2),(NDOF+NW+2))=AEOL
C
      NDI=NDIM-1
      DO 35 I=1,NDI
      K=I+1
      DO 35 J=K,NDIM
   35 ESM(I,J)=ESM(J,I)
      CALL ADSTIF(KFIX,S,LINK,ESM,NDIM,NN)
C
  100 CONTINUE
C
      RETURN
      END



      SUBROUTINE WOODFOR(XX,LINK,MATNUM,LOUT,DISP)
C
C *** Subroutine calculates and outputs nodal forces and stresses
C *** for the frame elements.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
      COMMON /KONTRL/ NUMNOD,NDOF,NELE,NWOOD,NNAIL,NNPE,NSD,NEQ,
     *                IBAND,NDIM
      COMMON /WOODPROP/ NDWE,NDWP,MPID(9,8),WOOD(3,8)
      COMMON /DEVICE/ IIN,IOUT,IBUG,NNOUT,NWEOUT,NNEOUT
      DIMENSION XX(NSD,1),LINK(NNPE,1),MATNUM(1),DISP(1),X(2),N(2),
     *          DI(6),F(6),NFRM(100),S(1200),LOUT(1),ESM(6,6)
C
      WRITE(IOUT,2000)
      NFRME=0
      NL=NDOF-2
C
C *** Loop over elements.
C
      DO 100 NN=1,NWOOD
      IF(LOUT(NN).NE.1) GOTO 100
C
C *** Set X coordinate of nodes, set Y displacement & rotation of
C *** element nodes (DI(1),DI(2),DI(4),DI(5)), and calculate element
C *** length.
```

```
C
      DO 5 L=1,2
      N(L) = LINK(L,NN)
    5 X(L) = XX(1,N(L))
C
      DI(1) = DISP(N(1)*NDOF-NDOF+1)
      DI(2) = DISP(N(1)*NDOF-NDOF+2)
      DI(4) = DISP(N(2)*NDOF-NDOF+1)
      DI(5) = DISP(N(2)*NDOF-NDOF+2)
C
      H = X(2)-X(1)
      EL = SQRT(H*H)
C
C *** Set wood element configuration number (NCONF).
C *** Loop over each layer in the element.
C *** Get wood properties for layer NW.
C
      NCONF = MATNUM(NN)
      DO 20 NW=1,NL
C
      NOOD = MPID(NW,NCONF)
      WIDTH = WOOD(1,NOOD)
      DEPTH = WOOD(2,NOOD)
      XMOE = WOOD(3,NOOD)
      AREA = WIDTH*DEPTH
      XI = (WIDTH*DEPTH**3)/12.0D0
      EI = XMOE*XI
C
C *** For layer NW, get X displacements of ends and then calculate
C *** the 6-by-6 stiffness matrix ESM.  For this element stiffness
C *** maxtix: DOF 1&4 = Rot; DOF 2&5 = Y Trans; DOF 3&6 = X Trans.
C
      DI(3)=DISP(N(1)*NDOF-NDOF+2+NW)
      DI(6)=DISP(N(2)*NDOF-NDOF+2+NW)
      ESM(1,1) = 4.0D0*EI/EL
      ESM(2,1) = 6.0D0*EI/EL/EL
      ESM(2,2) = 12.0D0*EI/EL/EL/EL
      ESM(3,1) = 0.0D0
      ESM(3,2) = 0.0D0
      ESM(3,3) = XMOE*AREA/EL
      ESM(4,1) = ESM(1,1)/2.0D0
      ESM(4,2) = ESM(2,1)
      ESM(4,3) = 0.0D0
      ESM(4,4) = ESM(1,1)
      ESM(5,1) =-ESM(2,1)
      ESM(5,2) =-ESM(2,2)
      ESM(5,3) = 0.0D0
      ESM(5,4) =-ESM(2,1)
      ESM(5,5) = ESM(2,2)
      ESM(6,1) = 0.0D0
      ESM(6,2) = 0.0D0
      ESM(6,3) =-ESM(3,3)
      ESM(6,4) = 0.0D0
      ESM(6,5) = 0.0D0
      ESM(6,6) = ESM(3,3)
      DO 40 I=1,5
      K=I+1
      DO 40 J=K,6
```

```
   40 ESM(I,J)=ESM(J,I)
C
C *** Calculation of element forces in the global coordinate system.
C
      DO 50 I=1,6
      F(I) = 0.0D0
      DO 50 J=1,6
   50 F(I) = F(I) + ESM(I,J)*DI(J)
C
C *** Output of frame element nodal forces.
C
      IF(NW.EQ.1)WRITE(IOUT,2010)NN,NW,F(6),F(2),F(1),F(4)
      IF(NW.NE.1)WRITE(IOUT,2015)NW,F(6),F(2),F(1),F(4)
C
C *** Stress calculations.
C
      IF(NW.EQ.1)THEN
         NFRME = NFRME + 1
         NFRM(NFRME) = NN
      END IF
      AXL = F(6)/AREA
      BSL = ABS(F(1)*DEPTH/XI/2.0)
      BSR = ABS(F(4)*DEPTH/XI/2.0)
      IF(AXL.GT.0.0D0)THEN
        BSLM=AXL+BSL
        BSRM=AXL+BSR
      ELSE
        BSLM=AXL-BSL
        BSRM=AXL-BSR
      END IF
      BMAX=BSLM
      IF(ABS(BSRM).GT.ABS(BSLM))BMAX=BSRM
      LL=(NFRME-1)*NL*5+NW*5
      S(LL-4) = AXL
      S(LL-3) = 1.5*F(2)/AREA
      S(LL-2) = BSL
      S(LL-1) = BSR
      S(LL) = BMAX
  199 FORMAT(5(F12.0))
C
   20 CONTINUE
  100 CONTINUE
C
C *** Output of stresses.
C
      WRITE(IOUT,2020)
      DO 110 I=1,NFRME
      NN = NFRM(I)
      DO 110 NW=1,NL
      LL=(I-1)*NL*5+NW*5-5
      IF(NW.EQ.1)WRITE(IOUT,2030)NN,NW,(S(LL+J),J=1,5)
  110 IF(NW.NE.1)WRITE(IOUT,2040)NW,(S(LL+J),J=1,5)
C
 2000 FORMAT(//,6X,19HWOOD ELEMENT FORCES,/,6X,19(1H-),/,6X,
     *     7HElement,2X,5HLayer,2X,11HAxial Force,2X,11HShear Force,
     *     3X,22HBending Moment (ccw +),/,22X,11H(tension +),5X,
     *     7H(/\+\/),4X,22HLeft End     Right End,/,6X,65(1H-))
 2010 FORMAT(8X,I2,5X,I3,1X,4F13.4)
```

```
 2015 FORMAT(15X,I3,1X,4F13.4)
 2020 FORMAT(//,6X,21HMAXIMUM WOOD STRESSES,/,6X,21(1H-),/,21X,
     *     5HAxial,18X,7HBending,7X,7HBending,5X,7H Axial ,
     *     /,6X,46HElmt.  Layer  Comp.(-)     Shear      Left End,5X,
     *     9HRight End,
     *     7X,1H&,/,19X,37HTension(+)              (Absolute)   ,
     *     21H(Absolute)    Bending,/,6X,71(1H-))
 2030 FORMAT(6X,I3,4X,I3,5(2X,F10.1))
 2040 FORMAT(13X,I3,5(2X,F10.1))
C
      RETURN
      END



      SUBROUTINE NAILSTF(XX,KFIX,LINK,MATNUM,S,ESM)
C
C *** Subroutine to assembly the element stiffness matrix for the
C *** nail elements.
C
C *** Description of some subroutine variables:
C ***  NL      Number of wood layers.
C ***  NI      Number of interlayers or slip surfaces (NL-1).
C ***  NC      Configuration number associated with the element.
C ***  NP      Nail load-slip parameter set for element.
C ***  NW      Number of a wood element that's adjacent to element.
C ***  UK      Initial stiffness of connector parallel to grain.
C ***  SK(I)   Shear stiffness at interlayer I.  Equal to the
C ***            product of UK and the number of connectors at I.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
      COMMON /KONTRL/ NUMNOD,NDOF,NELE,NWOOD,NNAIL,NNPE,NSD,NEQ,
     *                IBAND,NDIM
      COMMON /LIMITS/ ILOAD,NSTEP,NITER,NELAS,TOL,CONFAC
      COMMON /WOODPROP/ NDWE,NDWP,MPID(9,8),WOOD(3,8)
      COMMON /NAILPROP/ NDNE,NNLSP,FLSP(3,3),NEC(10,8)
      COMMON /DEVICE/ IIN,IOUT,IBUG,NNOUT,NWEOUT,NNEOUT
      DIMENSION XX(NSD,1),KFIX(1),LINK(NNPE,1),MATNUM(1),
     *          S(1),ESM(NDIM,NDIM),SK(9),D(8),H(9)
C
      NI=NDOF-3
      NL=NDOF-2
C
C *** Loop over elements. Check to see if element is connector type.
C
      DO 100 NNN=1,NNAIL
      NN=NWOOD+NNN
C
C *** Set connector element properties.
C
      NC = MATNUM(NN)
      NP = NEC(1,NC)
      UK = FLSP(3,NP)*CONFAC
      NW = NEC(2,NC)
C
C *** Get heights of layers (H(I) = height of layer I).  Set vertical
C *** distance between centroid of adjacent layers (Note: D(I) is the
C *** distance between layer I and layer I+1).
```

```
C
      NWC = MATNUM(NW)
      DO 20 I=1,NL
   20 H(I) = WOOD(2,MPID(I,NWC))
      DO 30 I=1,NI
      SK(I)=UK*NEC((I+2),NC)
   30 D(I)=(H(I)+H(I+1))/2.0D0
C
C *** Zero and then assemble connector element stiffness matrix.
C
      DO 40 I=1,NDIM
      DO 40 J=1,NDIM
   40 ESM(I,J)=0.D0
C
      DO 50 K=1,NI
      I=K+2
      J=K+3
      ESM(1,1) = ESM(1,1) + SK(K)*D(K)**2
      ESM(I,I) = ESM(I,I) + SK(K)
      ESM(J,J) = ESM(J,J) + SK(K)
      ESM(J,I) = ESM(J,I) - SK(K)
      ESM(I,1) = ESM(I,1) + SK(K)*D(K)
   50 ESM(J,1) = ESM(J,1) - SK(K)*D(K)
C
      NDO=NDOF-1
      DO 60 I=1,NDO
      K=I+1
      DO 60 J=K,NDOF
   60 ESM(I,J)=ESM(J,I)
C
C *** Assemble the stiffness matrix into the global stiffness matrix.
C
      CALL ADSTIF(KFIX,S,LINK,ESM,NDIM,NN)
C
  100 CONTINUE
      RETURN
      END



      SUBROUTINE NAILNLB(XX,LINK,MATNUM,DISP,W)
C
C *** Subroutine to assemble load-modifying vector so that nonlinear
C *** behavior of frame-to-frame connectors is accounted for.
C
C *** Description of some subroutine variables:
C ***  Load-slip parameters - disp. parallel to grain.
C ***      UM0=m0           UM1=m1           UK=k
C ***  See also Subroutine NAILSTF.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
      COMMON /KONTRL/ NUMNOD,NDOF,NELE,NWOOD,NNAIL,NNPE,NSD,NEQ,
     *                IBAND,NDIM
      COMMON /LIMITS/ ILOAD,NSTEP,NITER,NELAS,TOL,CONFAC
      COMMON /WOODPROP/ NDWE,NDWP,MPID(9,8),WOOD(3,8)
      COMMON /NAILPROP/ NDNE,NNLSP,FLSP(3,3),NEC(10,8)
      DIMENSION XX(NSD,1),LINK(NNPE,1),MATNUM(1),
     *          DISP(NDOF,1),W(NDOF,1),N(8),D(8),H(9)
```

```
C
      NI=NDOF-3
      NL=NDOF-2
C
C *** Loop over elements. Check to see if element is connector type.
C
      DO 100 NNN=1,NNAIL
      NN=NWOOD+NNN
C
C *** Set connector element properties.
C
      N1 = LINK(1,NN)
      NC = MATNUM(NN)
      NP = NEC(1,NC)
      UK = FLSP(3,NP)*CONFAC
      UM0 = FLSP(1,NP)
      UM1 = FLSP(2,NP)
      NW = NEC(2,NC)
C
C *** Get heights of layers (H(I) = height of layer I).  Set vertical
C *** distance between centroid of adjacent layers (Note: D(I) is the
C *** distance between layer I and layer I+1).
C
      NWC = MATNUM(NW)
      DO 20 I=1,NL
   20 H(I) = WOOD(2,MPID(I,NWC))
      DO 30 I=1,NI
      N(I)= NEC((I+2),NC)
   30 D(I)=(H(I)+H(I+1))/2.0D0
C
C *** Loop over each interface. Calculate slip.  Calculate shear force
C *** at the interface. Load forces into global load modifying vector.
C
      DO 50 K=1,NI
      I = K+2
      J = K+3
      SLIP = DISP(I,N1)-DISP(J,N1)+DISP(1,N1)*D(K)
      AS = ABS(SLIP)
      FOR = (UM0+UM1*AS)*(1.0D0-EXP(-UK*AS/UM0))
      WF = N(K)*SLIP*(UK - FOR/AS)
      W(1,N1) = W(1,N1) + WF*D(K)
      W(I,N1) = W(I,N1) + WF
   50 W(J,N1) = W(J,N1) - WF
C
  100 CONTINUE
      RETURN
      END



      SUBROUTINE NAILFOR(XX,LINK,MATNUM,LOUT,DISP)
C
C *** Subroutine calculates and outputs shear forces the total slip
C *** of each connector in a frame-to-frame connector element.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER (I-N)
      COMMON /KONTRL/ NUMNOD,NDOF,NELE,NWOOD,NNAIL,NNPE,NSD,NEQ,
     *                IBAND,NDIM
```

```
      COMMON /LIMITS/ ILOAD,NSTEP,NITER,NELAS,TOL,CONFAC
      COMMON /WOODPROP/ NDWE,NDWP,MPID(9,8),WOOD(3,8)
      COMMON /NAILPROP/ NDNE,NNLSP,FLSP(3,3),NEC(10,8)
      COMMON /DEVICE/ IIN,IOUT,IBUG,NNOUT,NWEOUT,NNEOUT
      DIMENSION XX(NSD,1),LINK(NNPE,1),MATNUM(1),DISP(NDOF,1),LOUT(1),
     *          H(9),D(8),N(8)
C
      WRITE(IOUT,2000)
      NI=NDOF-3
      NL=NDOF-2
C
C *** Loop over elements. Check to see if element is connector type.
C
      DO 100 NNN=1,NNAIL
      NN=NWOOD+NNN
      IF(LOUT(NN).NE.1) GOTO 100
C
C *** Set connector element properties.
C
      N1 = LINK(1,NN)
      NC = MATNUM(NN)
      NP = NEC(1,NC)
      UK = FLSP(3,NP)
      UM0 = FLSP(1,NP)
      UM1 = FLSP(2,NP)
      NW = NEC(2,NC)
C
C *** Get heights of layers (H(I) = height of layer I).  Set vertical
C *** distance between centroid of adjacent layers (Note: D(I) is the
C *** distance between layer I and layer I+1).
C
      NWC = MATNUM(NW)
      DO 20 I=1,NL
   20 H(I) = WOOD(2,MPID(I,NWC))
      DO 30 I=1,NI
      N(I)= NEC((I+2),NC)
   30 D(I)=(H(I)+H(I+1))/2.0D0
C
C *** Loop over each interface. Calculate slip.
C
      DO 50 K=1,NI
      I = K+2
      J = K+3
      SLIP = DISP(I,N1)-DISP(J,N1)+DISP(1,N1)*D(K)
      AS = ABS(SLIP)
C
C *** Calculate shear force at the interface (FOR).
C
      IF(NELAS.EQ.0.AND.UM0.GT.0.0001)THEN
           FOR = (UM0+UM1*AS)*(1.0D0-EXP(-UK*AS/UM0))
      ELSE
           FOR = UK*AS
      END IF
      KK=K+1
      IF(N(K).EQ.0)THEN
          IF(K.EQ.1)WRITE(IOUT,2030)NNN,K,KK,N(K),SLIP
          IF(K.GT.1)WRITE(IOUT,2040)K,KK,N(K),SLIP
      ELSE
```

```
          IF(K.EQ.1)WRITE(IOUT,2010)NNN,K,KK,N(K),SLIP,FOR
          IF(K.GT.1)WRITE(IOUT,2020)K,KK,N(K),SLIP,FOR
   50 ENDIF
C
  100 CONTINUE
C
 2000 FORMAT(//,6X,19HNAIL ELEMENT FORCES,/,6X,19(1H-),/,14X,
     *   '   Nails     Number of   Interlayer    Shear                ',
     *   /,6X,'Element   Between    Nails per      Slip       Force',
     *    /,14X,'  Layers-    Interface               per Nail',
     *   /6X,53(1H-))
 2010 FORMAT(7X,I3,7X,I1,' & ',I1,8X,I2,5X,F10.5,1X,F11.3)
 2020 FORMAT(17X,I1,' & ',I1,8X,I2,5X,F10.5,1X,F11.3)
 2030 FORMAT(7X,I3,7X,I1,' & ',I1,8X,I2,5X,F10.5)
 2040 FORMAT(17X,I1,' & ',I1,8X,I2,5X,F10.5)
      RETURN
      END
```