AD-A269 001

ARMY RESEARCH LABORATORY

User-Defined Subroutine for Implentation of Higher-Order Shell Element in ABAQUS

Erik Saether and Alexander Tessler

ARL-TR-145

August 1993

DTIC
ELECTE
SEP 09 1993
S E D

93-20842

Approved for public release; distribution unlimited.

93 9 03 023

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1993	Final Report

4. TITLE AND SUBTITLE
User-Defined Subroutine for Implementation of Higner-Order Shell Element in ABAQUS

5. FUNDING NUMBERS

6. AUTHOR(S)
Erik Saether and Alexander Tessler*

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Watertown, MA 02172-0001
ATTN: AMRL-MA-DB

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-145

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
*Presently at NASA Langley Research Center, Structural Mechanics Division, Hampton, VA

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A user-defined subroutine has been developed to implement a higher-order three-node facet shell finite element in the commercial finite element program ABAQUS. The element, designated HOT3, is based on a {1,2}-order plate theory specifically formulated to model thin and thick composite laminates and exhibits improved accuracy compared to conventional shear-deformable shell finite elements. This report describes the use of HOT3 in ABAQUS for static and vibration analyses by presenting the required element input data together with numerical examples. Sample input and output datasets and the complete FORTRAN subroutine which performs all computations for the HOT3 element are included.

14. SUBJECT TERMS
Finite element analysis, Plate theory, Composites, Computer applications

15. NUMBER OF PAGES
108

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Contents

	Page
Introduction	1
HOT3 Shell Element	1
User-Defined Elements in ABAQUS	3
Use of HOT3 in ABAQUS	4
Demonstration Problem I - Static Analysis	6
Demonstration Problem II - Dynamic Analysis	9
Concluding Remarks	13
Acknowledgment	13
References	14
Appendix A Source Code Listing of Subroutine UEL Supporting HOT3 Element in ABAQUS	15
Appendix B Demonstration Problem I: Static Analysis	83
Appendix C Demonstration Problem II: Dynamic Analysis	97

Accesion For
NTIS CRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution /
Availability Codes
Dist | Avail and / or Special
A-1

DTIC QUALITY INSPECTED 1

Introduction

Research at Army Research Laboratory-Watertown has led to the development of a higher-order plate theory [1,2] which represents the state of the art in modeling thick composites using an equivalent single-layer type formulation. The theory has been incorporated into a 3-node anisoparametric facet shell finite element designated HOT3. The HOT3 element is a versatile displacement-based finite element incorporating higher-order kinematic expansions for use within the conventional 'h-version' finite element framework where nodal degrees of freedom are restricted to displacements and rotations. The higher-order field representation in HOT3 provides greater accuracy in the thick regime when compared to Reissner-Mindlin shear deformable elements with virtually the same computational cost.

This report outlines the use of the HOT3 element in ABAQUS via a user-defined element subroutine. A discussion of the HOT3 element and ABAQUS support of user-defined elements is presented in the following sections followed by two numerical examples illustrating the use of HOT3 in ABAQUS for static and dynamic analysis. Sample input and output datasets and the complete FORTRAN source code supporting the HOT3 element in ABAQUS are contained in separate appendices.

HOT3 Shell Element

HOT3 has been formulated as a triangular shell element incorporating transverse shear and transverse normal deformation modes. As shown in Figure 1, HOT3 has three vertex nodes with all 6 degrees of freedom defined together with an internal node which represents two constant element degrees of freedom, w1 and w2, which represent the higher-order transverse displacement variables. In elastostatic analysis these two element degrees of freedom are statically condensed out at the element level in the computation of the element stiffness matrix and the internal node is not defined by the user. In dynamic analysis, however, the inertial contributions of these higher-order functions are not condensed out and an internal node must be defined and assigned two degrees of freedom. The location of this node is arbitrary and is used to represent the stiffness and mass associated with the higher-order element variables which are internally stored as the first two degrees of freedom. The remaining four degrees of freedom at the interior node - which physically do not exist - must therefore be restrained through single-point boundary constraints when using HOT3 in a dynamic analysis. Complete details of the element formulation may be found in [2,3,4].

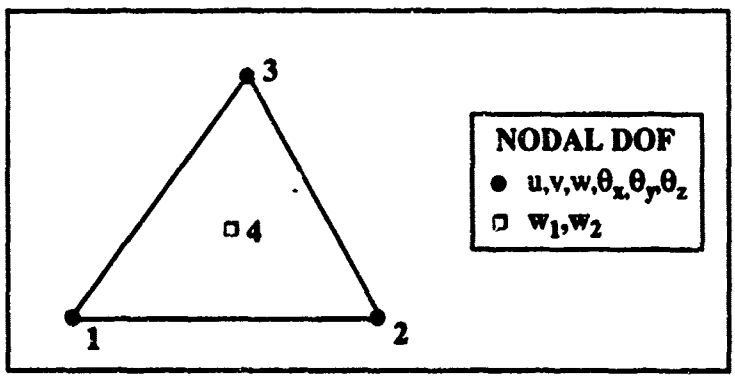

Figure 1. HOT3 : A facet shell element incorporating transverse shear and transverse normal deformation modes.

HOT3 may be defined arbitrarily in space using a counterclockwise node numbering convention. The element coordinate system (x',y',z') is shown in Figure 2. The x' axis is defined along the element side from node 1 to node 2 with the y' axis perpendicular to x' and lying in the element plane. The transverse axis z' is defined as normal to the element plane. A local coordinate system (x'',y'',z'') is also defined with the origin at node 1 and coordinate axes alligned with the global coordinate system (x,y,z). In static analysis, element output can be selected in either the local or element coordinate system.

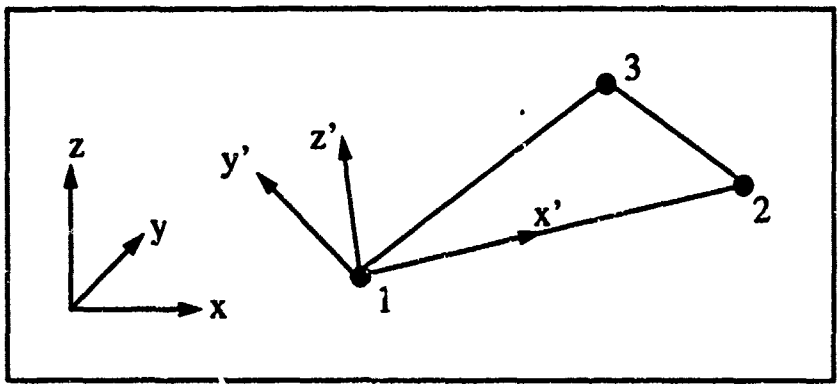

Figure 2. HOT3 element coordinate system.

For complicated geometries where the material axes do not correspond to the global coordinate system, a material coordinate system may be defined for each HOT3 element by specifying the longitudinal (M1) and transverse (M2) material axes. The normal material axis (M3) is assumed perpendicular to the plane formed by M1 and M2 .The material coordinate system is centered at the origin of the element local coordinate system and the vector components of the material axes are computed in the local system. This is depicted in figure 3.

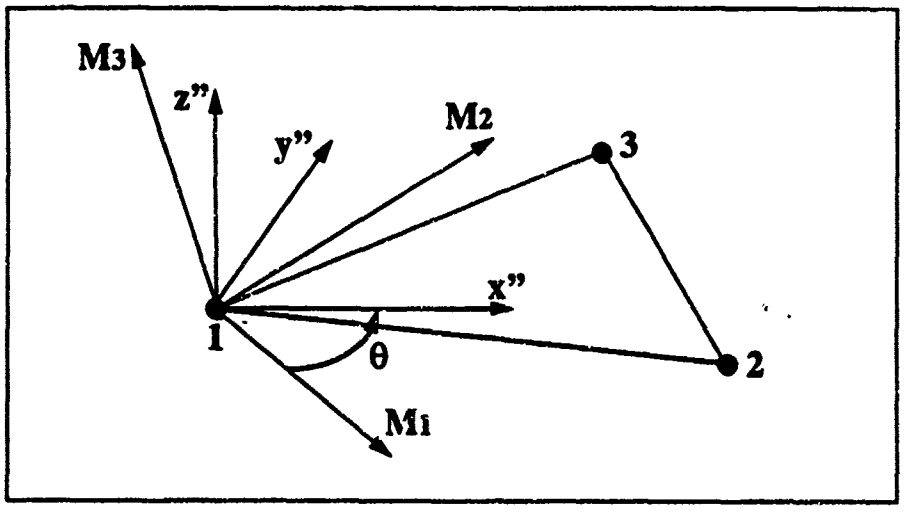

Figure 3. Specification of material axes in HOT3.

User-Defined Elements in ABAQUS

New finite elements may be implemented in ABAQUS by developing a subroutine denoted UEL (for User ELement) which perfoms the neccessary element computations and interfaces with the main ABAQUS program through a standardized parameter list in the subroutine call statement.

The *USER SUBROUTINE statement in the input deck alerts ABAQUS to the presence of user-defined subroutines which either immediatelly follow this data entry or are contained in a separate file. These subroutines are then compiled and linked to the main ABAQUS executable prior to job execution. A complete description of this and other user-defined capabilities in ABAQUS may be found in reference [5].

Shown in Figure 4 is the basic format of the UEL subroutine with the argument list used by ABAQUS to pass into the user-defined subroutine all the neccessary information needed to compute element stiffness and mass matrices. Once computed, these matrices are then passed back to ABAQUS for global assembly and problem solution. In static analysis, data recovery is performed during a second pass through the user-defined subroutine after the global solution has been obtained. During this phase, ABAQUS passes in the nodal displacements for the current element from which all element stresses, strains and forces can be computed. In dynamic analysis, frequencies and mode shapes are computed within ABAQUS; however, individual eigenvectors are not passed back into the user-defined subroutine and, hence, modal stresses and strains are not available. It should be noted that, in addition to linear static and dynamic analysis, the information passed into the UEL subroutine is sufficient to support material and geometric nonlinear analysis.

The complete source code for the HOT3 element supporting linear static and dynamic analysis in ABAQUS is listed in appendix A.

```
      SUBROUTINE UEL ( RHS,  AMATRX, SVARS, ENERGY, NDOFEL, NRHS
     1                 NSVARS, PROPS, NPROPS, COORDS, NCRD, NNODE,
     2                 U, DU, V, A, JTYPE, TIME, DTIME, KSTEP, KINC,
     3                 JELEM, PARAMS, NDLOAD, JDLTYP, ADLMAG
     4                 PREDEF, NPREDF, LFLAGS, MLVARX, DDLMAG,
     5                 MDLOAD, PNEWDT )
C
      REAL DOUBLE PRECISION ( A-H, O-Z )
C
      DIMENSION RHS(MLVARX,*),AMATRX(NDOFEL,NDOFEL), SVARS(1),
     1          ENERGY(8),PROPS(*), COORDS(MCRD,NNODE),
     2          U(NDOFEL), DU(NDOFEL), V(NDOFEL, A(NDOFEL),
     3          TIME(2), PARAMS(*), JDLTYPE(MDLOAD,*), LFLAGS(5),
     4          ADLMAG(MDLOAD,*), DDLMAG(MDLOAD,*),
     5          PREDEF(2,NPREDF,NNODE)

                 * Source code for the HOT3 element *

      RETURN
      END
```

Figure 4. Format of the user-defined subroutine UEL supporting the HOT3 element in ABAQUS.

Use of HOT3 in ABAQUS

Three sets of statements are used to describe HOT3 in the ABAQUS input deck. Each of these statement sets may be identified in the sample input files presented in Appendices B and C. The first set, *USER ELEMENT, defines the basic parameters of the HOT3 element. All parameters are manditory and the user must set the N and M values in the NODES and PROPERTIES parameters as described below.

Statement set I:

(i) *USER ELEMENT, NODES=N, TYPE=U1, PROPERTIES=M, COORDINATES=6
(ii) 1,2,3,4,5,6

where the various input parameters are:

Card (i): NODES=3 specifies that only 3 corner nodes are being defined for static analysis. NODES=4 specifies that an additional internal node is being defined for dynamic analysis.
TYPE=U1 specifies the internal designation of the HOT3 element in the user-defined subroutine as U1.
PROPERTIES=M specifies that a user-defined property list of length M is to be established for each HOT3 element as expained below.
COORDINATES=6 indicates that cartesian coordinates are assumed for the element together with possible specification of direction cosines for a nomal vector to the element plane at each node.

Card (ii): Specifies that all 6 degrees of freedom are defined as active at each HOT3 node. If an internal node is defined, degrees of freedom 3 through 6 must be constrained at this node.

The second entry, *UEL PROPERTY, is the primary list of input data used to compute element quantities for each HOT3 element. The size of this list is determined by the user as a function of the total number of layers in the element. Only a single layer would be specified for a homogeneous plate whereas, for a composite, the number of layers would correspond to the number of plies in the laminate. ABAQUS requires for each line in the property list that all quantities be written as real numbers in free format with eight entries per line - missing entries are simply treated as zeros. In the format of the property list shown below, the total length of the property list is calculated as 8*(2*NLAY+3). This length is entered as a parameter on the *USER ELEMENT entry.

Statement set II:

(i) *UEL PROPERTY, ELSET=NM
(ii) NLAY, RHO, L1, L2, L3, OUTPUT, NPTS
(iii) E1, E2, E3, G12, G23, G31
(iv) NU12, NU23, NU31, THICK, THETA
(v) S1, S2, S3, D1, D2, D3
(vi) NSOL, NSYS, M1x, M1y, M1z, M2x, M2y, M2z

where the input parameters are defined as:

Card (i): NM is the set Id of HOT3 elements for which the following properties are to be used.

Card (ii): NLAY is the number of layers in the element.
RHO is the material density.
L1-L3 are the nodal magnitudes of linearily distributed loads over the element in the order in which the element nodes are specified on the *ELEMENT input data entry.
OUTPUT is an output control flag for HOT3 element data;
OUTPUT=0 for surpression of all HOT3 element output.
OUTPUT=1 for stresses , strains and displacements through the element thickness, grid point forces and element strain energy.
OUTPUT=2 for element stiffness matrix and consistent load vector only.
OUTPUT=3 for all of the above element data
NPTS is the number of equally spaced recovery points through the thickness.

Card (iii): E1-G31 are the layer normal and shear moduli.

Card (iv): NU12-THETA are the layer Poisson ratios,thickness and orientation with respect to the global or material coordinate system. Note that the two material property cards, (iii) and (iv), are repeated NLAY times in order to specify material properties for each layer in the element.

Card (v): S1-S3 and D1-D3 are the locations for stress and displacement recovery given in local triangular coordinates where the vertices correspond to the node order specified on the *ELEMENT bulk data entry.

Card (vi): NSOL = 1 or 2 for static and dynamic analysis respectively.
NSYS = 1 for element output in the global coordinate system
= 2 for element output in the element coordinate system.
M1x,M1y,M1z = vector components defining the longitudinal material axis
M2x,M2y,M2z = vector components defining the transverse material axis.

The last entry is the *USER SUBROUTINE statement. As stated above, this alerts ABAQUS to the presence of source code which is to be included together with the main executable code prior to running the requested job. This data statement is depicted below.

Statement set III:

(i) *USER SUBROUTINE, INPUT=uel_hot3.f

where the optional parameter INPUT specifies the name of the external file containing the source code for the HOT3 element. If this parameter is omitted, ABAQUS assumes that the source code immediately follows this statement.

Demonstration Problem I - Static Analysis

In Reference [2] the problem of cylindrical bending of a thick composite laminate has been solved using the HOT3 element. As shown in figure 5, the finite element model consists of a single strip of HOT3 elements spanning one quarter of the loading wavelength (L/2) in which appropriate multipoint constraints are applied to simulate the cylindrical bending features of this infinite plate. The composite selected is a carbon/epoxy symmetric angle-ply ($[30/-30]_s$) laminate subjected to a sinusoidal transverse pressure. The ply material properties are taken as

$$E_l = 25\text{x}10^6 \text{ psi}, \quad E_t = 10^6 \text{ psi}$$

$$G_{lt} = .5\text{x}10^6 \text{ psi}, \quad G_{tt} = .2\text{x}10^6 \text{ psi}$$

$$\nu_{lt} = \nu_{tt} = .25$$

where l and t denote the longitudinal and transverse ply material directions, respectively. A sample input dataset showing the use of the various data statements defining the HOT3 element geometry, material properties, and distributed loading, together with the complete ABAQUS output are summarized in Appendix B.

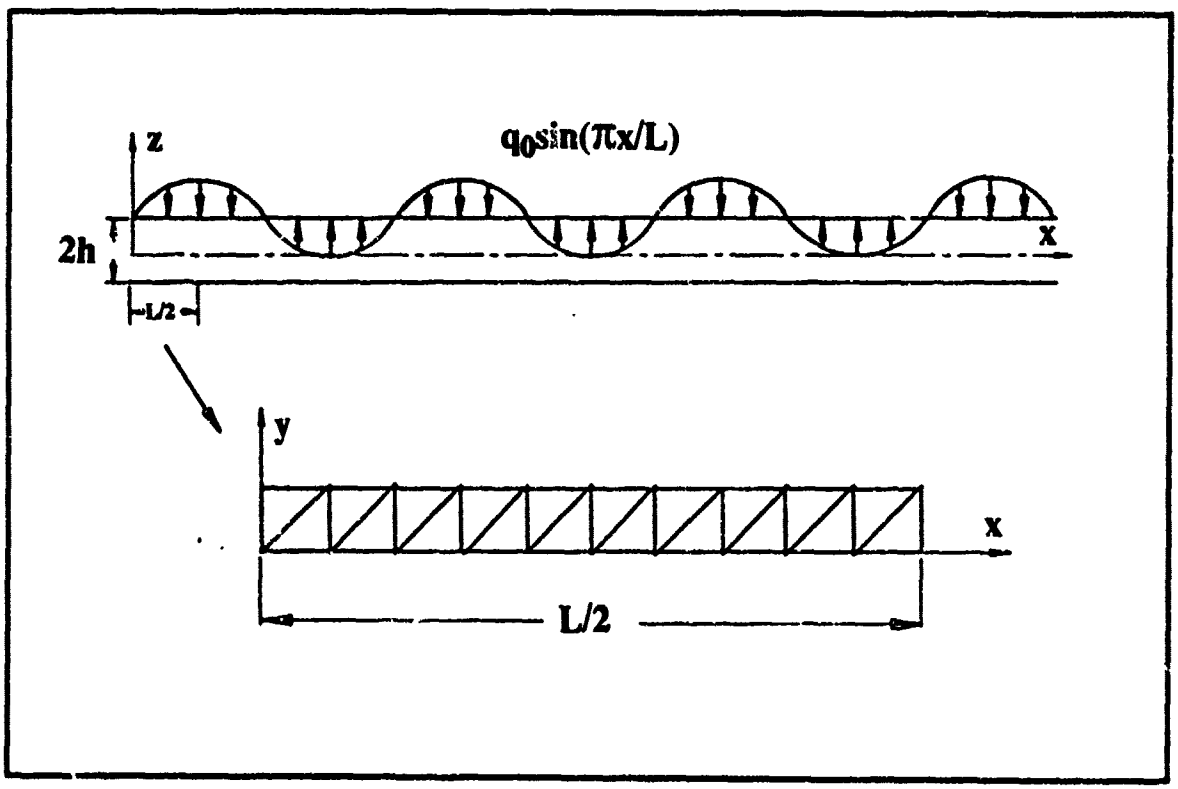

Figure 5. Finite element discretization of a carbon/epoxy $[30/-30]_s$ laminate subject to sinusoidal pressure in cylindrical bending.

Figure 6 shows the percent error of the maximum midplane transverse displacement as a function of the loading-to-thickness ratio (L/2h). In this and subsequent figures, results are shown variously as HOT_ANALYTIC when obtained analytically by way of the higher-order theory, as EXACT using an exact elasticity approach [6], as CPT using classical laminate plate theory, and as HOT_FEM results from the present finite element analysis using HOT3 in ABAQUS. Also, in Figure 6, results corresponding to standard shear-deformable plate theory are presented and labeled as SHEAR_DEF. All displacement and stress quantities have been normalized in accordance with [2].

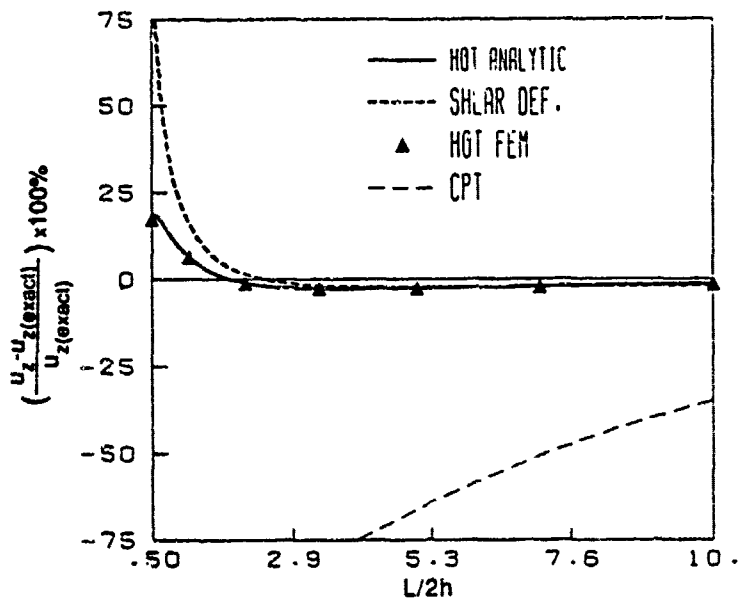

Figure 6. Percent error of maximum midplane deflection versus L/2h ratio

Figure 7 depicts the through-thickness distributions of the maximum normal (ε_{xx}) and inplane shear (γ_{xy}) strains. As noted in reference [2], close agreement with the exact solution is obtained for L/2h = 10. In the thick regime with L/2h = 4 the departure from the exact solution is due to the linear approximations made for the inplane u_x and u_y displacements which, as evidenced by the elasticity solution, become increasingly nonlinear particularily in the outer plies.

Figure 8 shows the through-thickness distributions of the transverse shear (γ_{xz}) and transverse normal (ε_{zz}) strains. It may be noted that the transverse shear strain distribution compares favorably with the exact solution in a resultant sense; the assumed parabolic distribution in the theory cannot match the exact distribution for general laminates. The assumed cubic variation in the transverse normal strain is seen to compare satisfactorily with the exact solution even in the thick regime for the particular laminate selected.

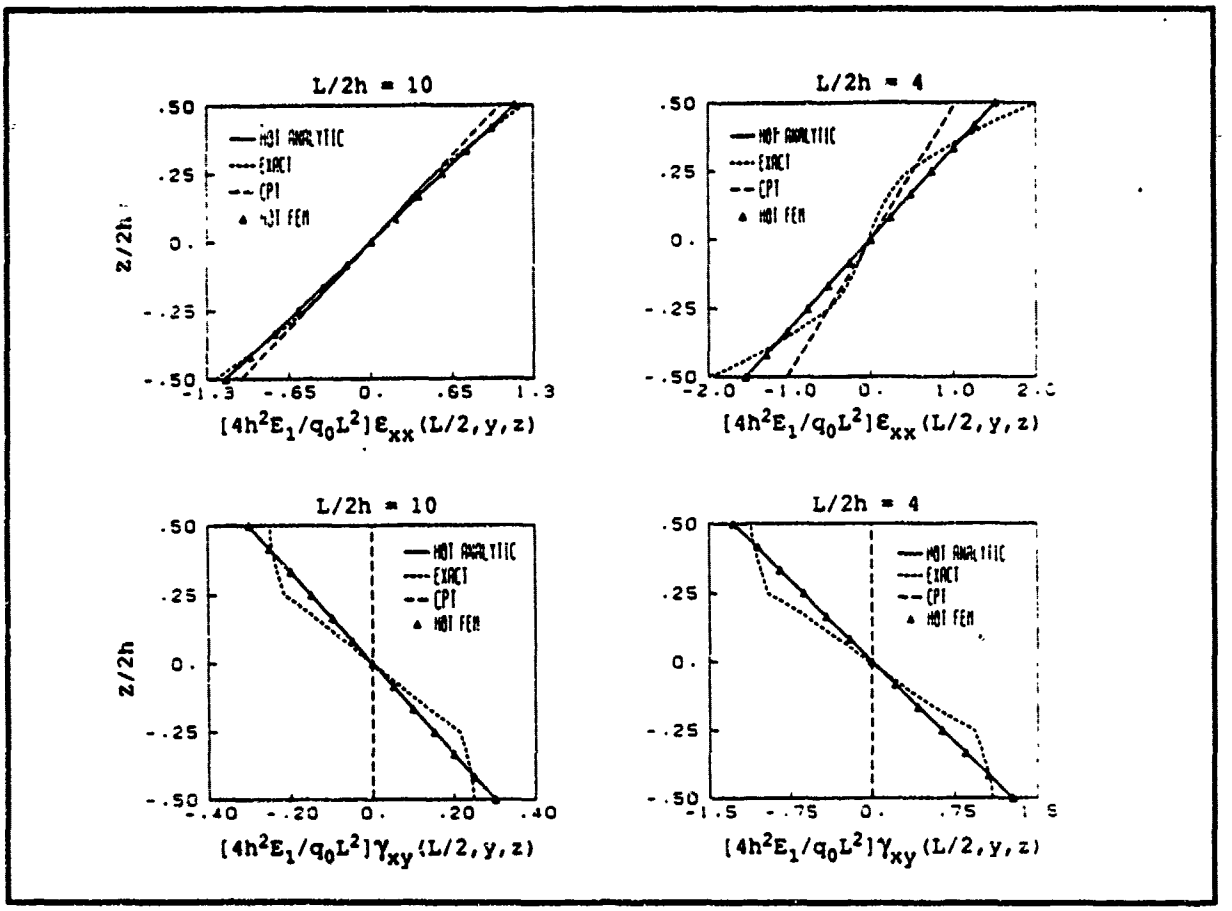

Figure 7. Distribution of inplane strains across thickness for L/2h = 10 and 4.

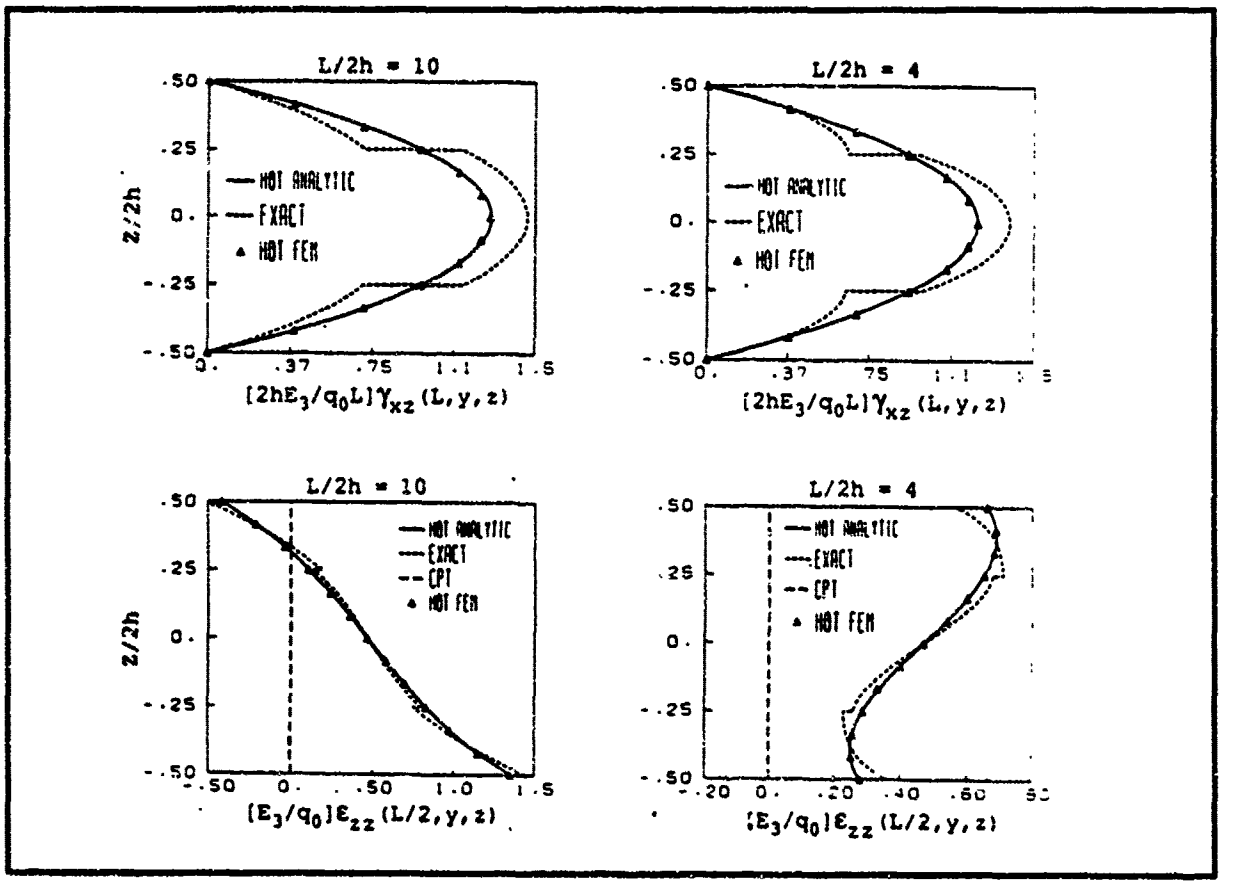

Figure 8. Distribution of transverse strains across thickness for L/2h = 10 and 4.

Demonstation Problem II - Dynamic Analysis

The problem of determining the natural vibrational frequencies of a simply supported plate as shown in Figure 9 is considered. Modes associated with the free vibration of plates may be designated as symmetric or antisymmetric depending on the relative motion of the inplane displacements with respect to the midplane. Specific modes are identified as combinations of bending, stretching, and shearing motions and are ranked with an order of harmonics denoted here as modal sets (m,n). In order to demonstrate the accuracy of the higher-order theory, analytic results from Ref. [7] for natural frequencies of orthotropic and laminated composite plates are presented followed by a discussion on the use of HOT3 with ABAQUS in dynamic analysis. Figure 10 shows the six lowest modes associated with symmetric and antisymmetric vibration corresponding to the lowest harmonic, (m,n) = (1,1) which are closely correlated with exact modes.

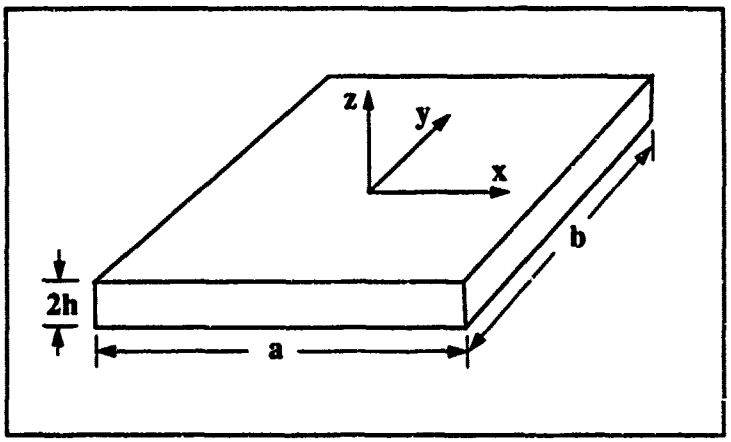

Figure 9. Simply-supported rectangular plate used in vibration analysis.

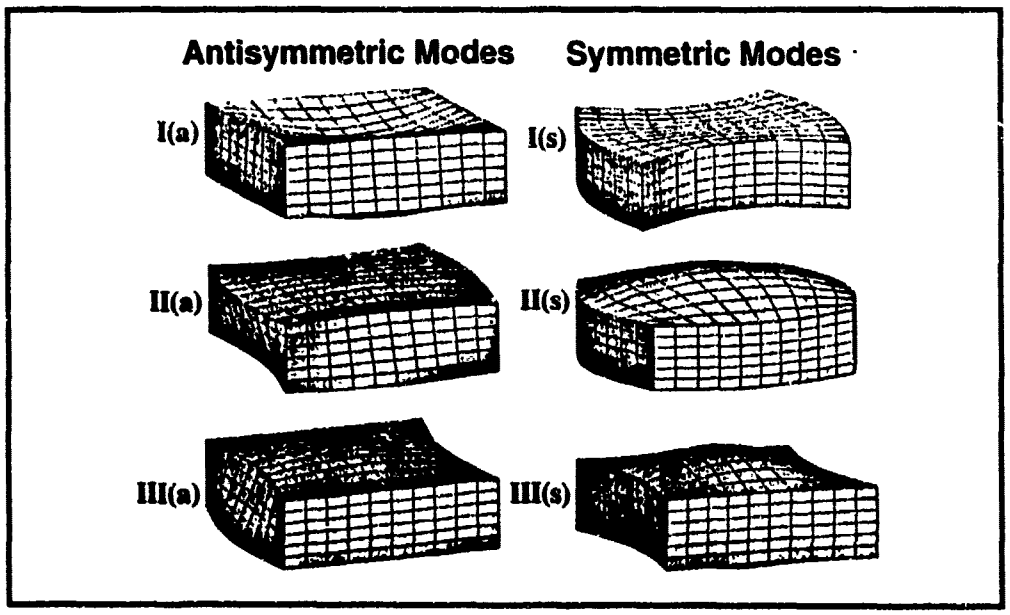

Figure 10. Mode shapes for thick, simply-supported square plate, (m,n) = (1,1).

Tables 1 and 2 show normalized antisymmetric and symmetric natural frequencies of an orthotropic plate for various modal numbers. HOT3 is capable of predicting seven thickness modes - three symmetric and four antisymmetric modes. The symmetric modes are the lowest frequency inplane stretch and thickness stretch modes denoted by I(s), II(s), and III(s) respectively. The antisymmetric modes are the fundamental bending mode I(a), the two lowest frequency thickness shear modes, II(a) and III(a), and the second thickness-stretch mode V(a). For each mode, the higher-order theory results, labeled 'HOT', are compared with exact solutions derived from three-dimensional elasticity. The material properties used are given by the following ratios of the elastic stiffness coefficients:

$$C_{22}/C_{11} = 0.543103, \quad C_{33}/C_{11} = 0.530172, \quad C_{12}/C_{11} = 0.233190$$

$$C_{13}/C_{11} = 0.010776, \quad C_{23}/C_{11} = 0.098276, \quad C_{44}/C_{11} = 0.266810$$

$$C_{55}/C_{11} = 0.159914, \quad C_{66}/C_{11} = 0.262931.$$

Table 1. Comparision of nondimensionalized natural frequencies, $\lambda = \omega(2h)(\rho/C_{11})^{1/2}$ of orthotropic square plate (2h/L=0.1) for antisymmetric modes I(a), II(a), III(a), V(a).

Wave Number		Mode I(a)		Mode II(a)		Mode III(a)		Mode V(a)	
m	n	Exact	HOT	Exact	HOT	Exact	HOT	Exact	HOT
1	1	0.0474	0.0474	1.3077	1.3078	1.6530	1.6530	4.5625	4.5763
3	1	0.2180	0.2173	1.5777	1.5780	1.7334	1.7336	4.5907	4.5790
3	3	0.3320	0.3309	1.5737	1.5737	1.9289	1.9296	4.5064	5.5874
4	1	0.3319	0.3301	1.7179	1.7179	1.8458	1.8554	4.6178	4.5813
4	2	0.3707	0.3687	1.6940	1.6940	1.9447	1.9453	4.5775	4.5845

Table 2. Comparision of nondimensionalized natural frequencies, $\lambda = \omega(2h)(\rho/C_{11})^{1/2}$ of orthotropic square plate (2h/L=0.1) for symmetric modes I(s), II(s), III(s).

Wave Number		Mode I(s)		Mode II(s)		Mode III(s)	
m	n	Exact	HOT	Exact	HOT	Exact	HOT
1	1	0.2170	0.2170	0.3941	0.3941	2.2722	2.2879
3	1	0.5029	0.5029	0.9731	0.9728	2.2396	2.2880
3	3	0.6504	0.6503	1.1814	1.1813	2.2273	2.2918
4	1	0.6591	0.6591	1.2795	1.2794	2.2346	2.2880
4	2	0.7028	0.7027	1.3453	1.3452	2.2319	2.2895

Tables 3, 4 and 5 compare the fundamental bending frequencies for various thickness ratios in a symmetric [0/90/90/0], unbalanced [0/90], and an antisymmetric angle-ply [45/-45] laminates. Several analytic plate theories and the exact elasticity solution when available are included for comparison. These theories are: The higher-order theory as described in Reference [7] designated 'HOT', higher-order shear-deformable theory of Reddy and Phan [8] designated 'HSDPT', first-order shear-deformable theory designated 'FSDPT', and the classical plate theory designated 'CPT'. The ply material properties are taken as:

$$E_l = 1.0\text{x}10^6 \text{ psi}, \qquad E_t = 0.025\text{x}10^6 \text{ psi}$$

$$G_{lt} = .0125\text{x}10^6 \text{ psi}, \quad G_{tt} = .015\text{x}10^6 \text{ psi}$$

$$\nu_{lt} = \nu_{tt} = .25$$

Table 3. Comparision of nondimensionalized fundamental frequencies, $\lambda = \omega(L^2/2h)(\rho/E_2)^{1/2}$, for symmetric, cross-ply, $[0/90]_s$, square plates.

2h/L	Exact	HOT	HSDPT	FSDPT	CPT
0.50	5.2600	5.4914	5.576	5.4926	15.830
0.25	9.2242	9.4401	9.497	9.4231	17.907
0.20	10.748	10.910	10.989	10.892	18.215
0.10	15.149	15.173	15.270	15.161	18.652
0.08	16.187	16.194	16.276	16.185	18.707
0.05	17.626	17.623	17.668	17.618	18.767
0.04	18.023	18.019	18.050	18.016	18.780
0.02	18.605	18.598	18.606	18.597	18.799
0.01	18.753	18.753	18.755	18.752	18.804

Table 4. Comparision of nondimensionalized fundamental frequencies, $\lambda = \omega(L^2/2h)(\rho/E_2)^{1/2}$, for antisymmetric, cross-ply, [0/90], square plates.

2h/L	Exact	HOT	HSDPT	FSDPT	CPT
0.50	4.7040	5.1798	5.699	5.1672	8.4987
0.25	7.3447	7.9764	8.294	7.9534	10.292
0.20	8.1859	8.7590	9.010	8.7385	10.584
0.10	10.088	10.356	10.449	10.347	11.011
0.08	10.435	10.623	10.686	10.617	11.066
0.05	10.859	10.942	10.968	10.939	11.125
0.04	10.966	11.020	11.037	11.018	11.139
0.02	11.113	11.128	11.132	11.127	11.158
0.01	11.151	11.155	11.156	11.155	11.163

Table 5. Comparision of nondimensionalized fundamental frequencies, $\lambda = \omega(L^2/2h)(\rho/E_2)^{1/2}$, for antisymmetric, angle-ply, [45/-45], square plates.

2h/L	HOT	HSDPT	FSDPT	CPT
0.50	5.4871	6.283	5.4900	6.2832
0.25	9.1439	9.759	9.1270	12.566
0.20	10.322	10.840	10.304	13.885
0.10	13.038	13.263	13.028	14.439
0.08	13.546	13.704	13.539	14.510
0.05	14.177	14.246	14.174	14.587
0.04	14.338	14.383	14.335	14.605
0.02	14.561	14.572	14.560	14.630
0.01	14.618	14.621	14.618	14.636

To demonstrate the use of HOT3 for dynamic analysis, a particular example from Table 3 for a $[0/90]_s$ composite plate with a thickness ratio equal to 0.2 is examined. The results using HOT3 are compared with those obtained using ABAQUS STRI3 discrete Kirchhoff and S4R first-order shear-deformable elements. The thickness ratio used was selected to demonstrate element behavior in the thick regime. Figure 11 shows typical discretizations of the plate using 3-noded HOT3 elements. Figure 12 depicts convergence of the fundamental bending frequency with mesh refinement. The STRI3 element converges to a frequency approximately 15% above the exact solution showing the effect of neglecting transverse shear and normal deformation at this thickness ratio. The S4R shear-deformable element is seen to converge below the exact solution which is due to a combination of excluding transverse normal deformation and the use of an empirically set shear stiffness relaxation parameter in the element formulation. A sample ABAQUS input and output file corresponding to the 8-element model shown in Figure 11 is included in Appendix C to show the required input format and typical output.

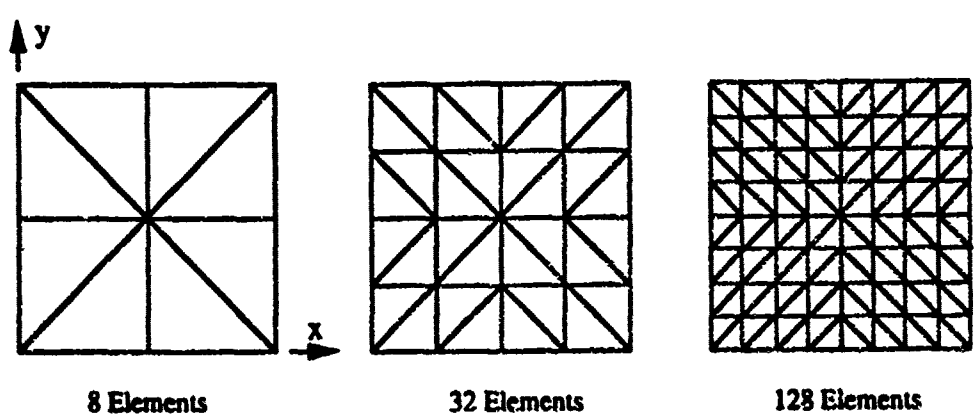

Figure 11. Typical finite element discretizations of a square plate with triangular elements.

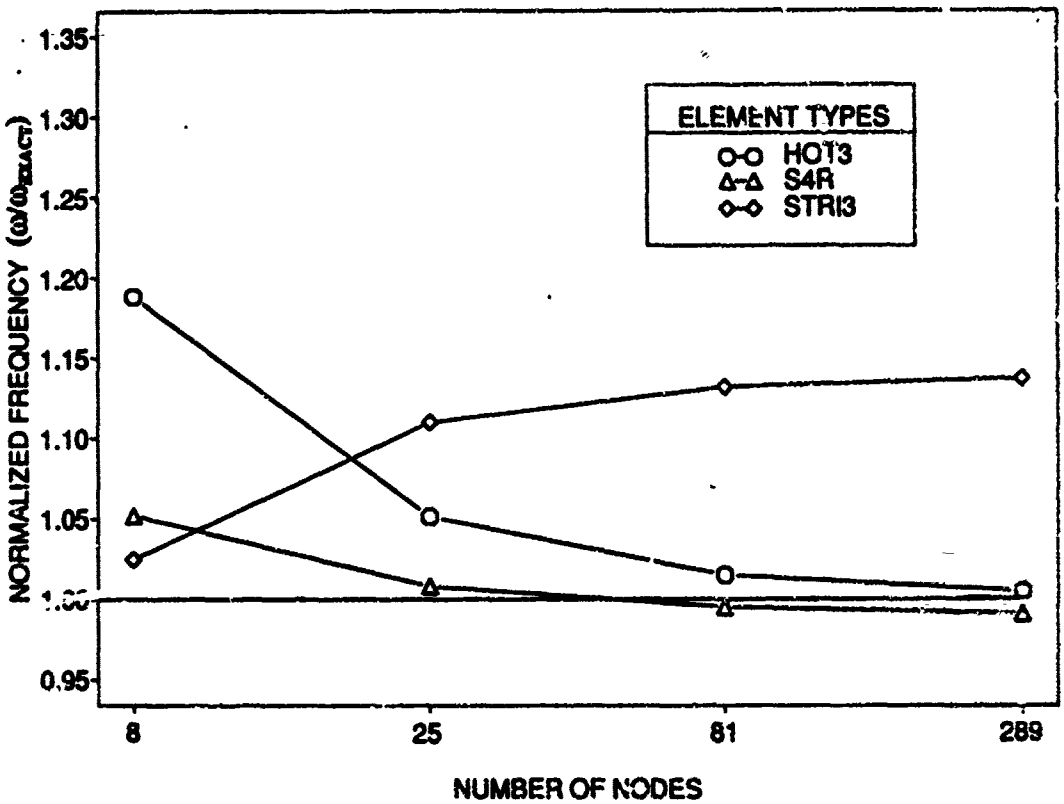

Figure 12. Convergence of the fundamental bending frequency of HOT3 , ABAQUS S4R and STRI3 elements for a square [0/90]s laminated composite plate.

Concluding Remarks

HOT3 is a higher-order three-node facet shell element incorporating both transverse shear and transverse normal strain and stress effects. The element provides improved accuracy when compared with conventional shear-deformable finite elements with virtually the same computational cost. A user-defined subroutine has been developed allowing the user of the commercial finite element code ABAQUS to obtain the computational benefits of the HOT3 element. The complete source code which performs all computations for HOT3 in ABAQUS is contained in this report.

Aknowledgement

The authors wish to thank Colin Freese for restructuring the user-defined subroutine UEL presented in appendix A for more efficient operation within ABAQUS.

REFERENCES

1. Tessler, A. *"A Higher-Order Plate Theory with Ideal Finite Element Suitability."* Comput. Meths. Appl. Mech. Engrg., **85(2)**, 1991 pp. 183-205.

2. Tessler, A. and Saether, E. *"A Computationally Viable Higher-Order Theory for laminated Composite Plates."* Int. J. Num. Meth. Engr., **31**, 1991 pp. 1069-1086.

3. Tessler, A. and Hughes, T. J. R. *"A Three-Node Mindlin Plate Element with Improved Transverse Shear."* Comput. Meths. Appl. Mech. Engrg., **50**, 1985, pp. 71-101.

4. Tessler, A. *"Shear-deformable Bending Elements with Penalty Relaxation in Finite Element Methods for Plate and Shell Structures,"* Vol. 1: Element Technology, (eds. T. J. R. Hughes and E. Hinton), Pineridge Press, Swansea, U.K., 1986, pp. 266-290.

5. Hibbitt, Karlsson and Sorensen, Inc., ABAQUS USER'S MANUAL, Version 5.2, 1992.

6. Pagano, N. J., *"Influence of Shear Coupling in Cylindrical Bending of Anisotropic Laminates."* J. Comp. Mats., **4**, 1970, pp. 330-343.

7. Tessler, A. Saether, E. and Tsui, T., *" Vibration of Thick Laminated Composite Plates,"* J. of Sound and Vibration (to appear).

8. Reddy, J.N., and Phan, D.N., *" Stability and Vibration of Isotropic, Orthotropic and Laminated plates According to a Higher-Order Shear Deformable Theory,"* J. of Sound and Vibration, **98(2)**, 1985, pp. 157-170.

APPENDIX A

Source code listing of subroutine UEL supporting the HOT3 element in ABAQUS.

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                       *
      SUBROUTINE UEL ( RHS,    AMATRX, SVARS,  ENERGY, NDOFEL, NRHS,    *
     &                 NSVARS, PROPS,  NPROPS, COORDS, MCRD,  NNODE,   *
     &                 U,      DU,     V,      AG,     JTYPE, TIME,    *
     &                 DTIME,  KSTEP,  KINC,   JELEM,  PARAMS, NDLOAD, *
     &                 JDLTYP, ADLMAG, PREDEF, NPREDF, LFLAGS, MLVARX, *
     &                 DDLMAG, MDLOAD, PNEWDT                        ) *
C                                                                       *
C     USER DEFINED ELEMENT SUBROUTINE FOR SUPPORTING THE                *
C     HOT3 HIGHER-ORDER TRIANGULAR SHELL ELEMENT IN THE                 *
C     ABAQUS CODE. LINEAR STATIC AND DYNAMIC CAPABILITIES               *
C     ARE SUPPORTED                                                     *
C                                                                       *
C     HOT3  SOFTWARE  FLOW :                                            *
C                                                                       *
C            UEL                    ABAQUS user element                 *
C              UHOTTR                  coordinate transformation        *
C              UHOTMT                  hot3 material                    *
C                 UMTCMX                  c-matrix                      *
C                    UMTDMX                 d-matrix                    *
C                    UMTTMX                 t-matrix                    *
C              UHOTSF                  hot3 stiffness                   *
C                 USFMEM                  membrane stiffness            *
C                    UBXMEM                 membrane b-matrix           *
C                 USFBND                  bending stiffness             *
C                    UBXBND                 bending b-matrix            *
C                 USFCPL                  coupling stiffness            *
C                    UBXMEM                 membrane b-matrix           *
C                    UBXBND                 bending b-matrix            *
C                 USFSHR                  shear stiffness               *
C                    UINTPT                 integration points          *
C                    UBXSHR                 shear b-matrix              *
C                 UFCSHR                  shear factor                  *
C              UHOTMM                  hot3 mass matrix                 *
C              UHOTLD                  hot3 load                        *
C              UHOTCD                  hot3 condensation                *
C              UHOTFX                  additional element computation   *
C              UHOTIO                  hot3 output                      *
C                 UHOTBX                  hot3 b-matrix                 *
C                    UBXMEM                  membrane b-matrix          *
C                    UBXBND                  bending b-matrix           *
C                    UBXSHR                  shear b-matrix             *
C                                                                       *
C                                                                       *
C    NOTE: ON THE *USER ELEMENT DATA ENTRY THE FOLLOWING                *
C          PARAMETERS VALUES ARE MANDITORY:                             *
```

```
C                                                                       *
C       * SET NODES=3 FOR STATIC ANALYSIS OR FOR DYNAMIC                *
C         ANALYSIS IN WHICH THE INERTIA ASSOCIATED WITH                 *
C         THE HIGHER-ORDER KINEMATIC TERMS MAY BE NEGLECTED.            *
C                                                                       *
C       * SET NODES=4 FOR DYNAMIC ANALYSIS IN WHICH THE                 *
C         DEGREES OF FREEDOM OF THE INTERNAL NODE ARE                   *
C         THE TWO HIGHER-ORDER KINEMATIC VARIABLES.                     *
C                                                                       *
C       * SET COORDINATES=6 TO ALLOW FOR POSSIBLE DIRECTION             *
C         COSINES DESCRIBING THE UNIT NORMAL OF THE ELEMENT.            *
C                                                                       *
C       * DEFAULT MATERIAL COORDINATE SYSTEM IS THE GLOBAL SYSTEM.      *
C                                                                       *
C                                                                       *
C       HOT3 PROPERTY LIST FORMAT:                                      *
C                                                                       *
C     1) NLAY,RHO,Q1,Q2,Q3,NOTYPE,NOUT,NMAT                             *
C     2) E1,E2,E3,G12,G23,G31                                           *
C     3) V12,V23,V31,PLY,FIB                                            *
C        (REPEAT LINES 2 AND 3 FOR EACH LAYER)                          *
C     M) PSI(1),PSI(2),PSI(3),PSU(1),PSU(2),PSU(3)                      *
C   M+1) NSOL,NSYS,MVC(1),MVC(2),MVC(3)                                 *
C                                                                       *
C       WHERE                                                           *
C                                                                       *
C       NLAY = NUMBER OF LAYERS IN LAMINATE                             *
C       RHO  = MATERIAL DENSITY                                         *
C       Qi   = DISTRIBUTED LOAD INTENSITIES AT NODES                    *
C       NOTYPE = ELEMENT OUTPUT REQUEST;                                *
C              = 0 SURPRESS ALL ELEMENT OUTPUT                          *
C              = 1 OUTPUT LAYER STRESSES AT USER SELECTED               *
C                  NUMBER OF THICKNESS COORDINATES AND                  *
C                  ELEMENT FORCES                                       *
C              = 2 OUTPUT ELEMENT STIFFNESS MATRIX AND                  *
C                  CONSISTENT LOAD VECTOR ONLY                          *
C              = 3 OUTPUT ELEMENT LAYER STRESSES, STRAINS               *
C                  AND INCLUDE STIFFNESS MATRIX AND LOAD                *
C                  VECTOR                                               *
C       NOUT   = NUMBER OF EQUALLY SPACED OUTPUT POINTS                 *
C                THROUGH THE ELEMENT THICKNESS                          *
C       NMAT   = HIGHER-ORDER THEORY SELECTION FLAG;                    *
C              = 1 FOR STRAIN-BASED                                     *
C              = 2 FOR STRESS-BASED (VER I)                             *
C              = 3 FOR STRESS-BASED (VER II)                            *
C       Ei     = YOUNGS MODULI                                          *
C       Gij    = SHEAR MODULI                                           *
C       Vij    = POISSON RATIOS                                         *
```

```
C        PLY    = LAYER THICKNESS                                        *
C        FIB    = LAYER ORIENTATION                                      *
C        PSI(I) = LOCATION FOR STRESS CALCULATION                        *
C        PSU(I) = LOCATION FOR DISPLACEMENT CALCULATION                  *
C        NSOL   = 1 FOR STATIC ANALYSIS                                  *
C               = 2 FOR DYNAMIC ANALYSIS                                 *
C               = 3 FOR DYNAMIC ANALYSIS NEGLECTING INERTIA              *
C                   ASSOCIATED WITH THE HIGHER-ORDER                     *
C                   KINEMATIC FUNCTIONS                                  *
C        NSYS   = COORDINATE SYSTEM FOR ELEMENT OUTPUT                   *
C               = 1 FOR GLOBAL COORDINATE SYSTEM                         *
C               = 2 FOR LOCAL  COORDINATE SYSTEM                         *
C       MVC1(I) = MATERIAL ORIENTATION VECTOR FOR DIRECTION 1            *
C       MVC2(I) = MATERIAL ORIENTATION VECTOR FOR DIRECTION 2            *
C                                                                        *
C                                                                        *
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                *
      PARAMETER(NPRECD=2)                                                *
C                                                                        *
C     VARIABLE AND ARRAY DECLARATIONS FOR UEL/ABAQUS INTERFACE           *
C                                                                        *
      DIMENSION RHS(MLVARX,*), AMATRX(NDOFEL,NDOFEL), PROPS(*),          *
     &          SVARS(1), ENERGY(8), COORDS(MCRD,NNODE), U(NDOFEL),      *
     &          DU(MLVARX), V(NDOFEL), AG(NDOFEL), TIME(2),              *
     &          PARAMS(*), JDLTYP(MDLOAD,*), ADLMAG(MDLOAD,*),           *
     &          DDLMAG(MDLOAD,*), PREDEF(2,NPREDF,NNODE), LFLAGS(5)      *
C                                                                        *
C     VARIABLE AND ARRAY DECLARATIONS FOR THE HOT3 ELEMENT               *
C                                                                        *
      DIMENSION ELTLOD(17), ELTSTF(17,17), AMASS(17,17), AMAT(4,4),      *
     &          BMAT(4,4), DMAT(4,4), GMAT(2,2), PLY(125), ZL(126),      *
     &          E1(125), E2(125), E3(125), V12(125), V23(125),           *
     &          V13(125), G12(125),G23(125), G31(125), X(3), Y(3),       *
     &          Z(3), C(6,6,125), Q(3), FIB(125), PSI(3), PSU(3),        *
     &          A(3), B(3), TRI(3,3), TRANS(24,24), RETN(24),            *
     &          PETN(24,24), STRN(6,6), ETRN(6,6), SMTN(6,6),            *
     &          EMTN(6,6), AMV1(3), AMV2(3)                              *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IF ( JTYPE .EQ. 1 ) THEN
C
C        COMPUTE THE ELEMENT STIFFNESS AND MASS MATRIX AND
C        CONSISTENT LOAD VECTOR FOR THE HOT3 TRIANGULAR
C        FACET SHELL ELEMENT BASED ON HIGHER-ORDER TESSLER
C        PLATE THEORY
C
         DO I = 1, 3
            X(I) = COORDS(1,I)
```

```
         Y(I) = COORDS(2,I)
         Z(I) = COORDS(3,I)
      END DO
C
C     READ ELEMENT DATA OFF PROPERTY LIST
C
      THICK  = 0.0
      NLAY   = INT(PROPS(1))
      RHO    = PROPS(2)
      Q(1)   = PROPS(3)
      Q(2)   = PROPS(4)
      Q(3)   = PROPS(5)
      NOTYPE = INT(PROPS(6))
      NOUT   = INT(PROPS(7))
      NMAT   = INT(PROPS(8))
C
      DO I = 1, NLAY
         E1(I)     = PROPS(16*(I-1)+9)
         E2(I)     = PROPS(16*(I-1)+10)
         E3(I)     = PROPS(16*(I-1)+11)
         G12(I)    = PROPS(16*(I-1)+12)
         G23(I)    = PROPS(16*(I-1)+13)
         G31(I)    = PROPS(16*(I-1)+14)
         V12(I)    = PROPS(16*(I-1)+17)
         V23(I)    = PROPS(16*(I-1)+18)
         V13(I)    = PROPS(16*(I-1)+19)
         PLY(I)    = PROPS(16*(I-1)+20)
         FIB(I)    = PROPS(16*(I-1)+21)
         THICK     = THICK + PLY(I)
      END DO
C
      DO I = 1, 3
         PSI(I) = PROPS(16*NLAY+ 8+I)
         PSU(I) = PROPS(16*NLAY+11+I)
      END DO
      NSOL = DINT(PROPS(16*NLAY+17))
      NSYS = DINT(PROPS(16*NLAY+18))
C
      DO I = 1, 3
         AMV1(I) = PROPS(16*NLAY+15+I)
         AMV2(I) = PROPS(16*NLAY+21+I)
      END DO
      IF ( NSYS .EQ. 0 ) NSYS = 2
C
C     SET DEFAULTS ON MATERIAL ORIENTATION VECTOR
C     IF NONE HAVE BEEN SPECIFIED
C
      IF ( AMV1(1) .EQ. 0.0 .AND. AMV1(2) .EQ. 0.0 .AND.
```

```
     &        AMV1(3)..EQ. 0.0 ) THEN
              AMV1(1) = 1.0
         END IF
         IF ( AMV2(1) .EQ. 0.0 .AND. AMV2(2) .EQ. 0.0 .AND.
     &        AMV2(3) .EQ. 0.0 ) THEN
              AMV2(2) = 1.0
         END IF
C
         ZL(1) = -THICK / 2.0
C
         DO  K = 1, NLAY
            ZL(K+1) = ZL(K) + PLY(K)
         END DO
C
C        COMPUTE ELEMENT COORDINATE TRANSFORMATION MATRICES
C
         CALL UHOTTR ( X, Y, Z, AMV1, AMV2, A, B, TRI, STRN, ETRN,
     &                 SMTN, EMTN, TRANS                          )
C
C        COMPUTE ELEMENT MATERIAL PROPERTY MATRICES
C
         CALL UHOTMT (  E1, E2, E3,  G12,  G23,  G31,  V12,  V23, V13,
     &                 FIB,  ZL, C, SMTN, AMAT, BMAT, DMAT, GMAT, NLAY,
     &                 NMAT                                           )
C
         IF ( NSOL .EQ. 1 .OR. LFLAGS(3) .EQ. 2 ) THEN
C
C           COMPUTE ELEMENT STIFFNESS MATRIX
C
            CALL UHOTSF ( ELTSTF, AMAT, BMAT, DMAT, GMAT, A, B, THICK,
     &                    RELAX                                       )
C
         ELSE IF ( LFLAGS(3) .EQ. 4 ) THEN
C
C           COMPUTE ELEMENT MASS MATRIX
C
            CALL UHOTMM ( AMASS, A, B, THICK, NSOL, RHO )
C
         END IF
C
C        TEST IF A DISTRIBUTED LOAD HAS BEEN APPLIED
C
         CALL UHOTLD ( Q, A, B, ELTLOD )
C
         IF (NSOL .EQ. 1 .OR. (NSOL .EQ. 3 .AND. LFLAGS(3) .EQ. 2)) THEN
C
C           PERFORM STATIC CONDENSATION
C
```

```
      CALL UHOTCD ( ELTSTF, ELTLOD )
C
      END IF
C
C     RESTORE STIFFNESS AND LOAD VECTORS TO ACCOUNT FOR
C     DIFFERENT DOF ARRANGEMENT IN ABAQUS.
C
      CALL DSCPY  ( 0.0D0,    RHS, NDOFEL,      1 )
      CALL DSCPY  ( 0.0D0, AMATRX, NDOFEL, NDOFEL )
C
      NBC = 0
C
      DO I = 1, 5
        DO J = 1, 3
          NAC        = I + 6 * ( J - 1 )
          NBC        = NBC + 1
          RHS(NAC,1) = ELTLOD(NBC)
          NBR        = 0
          DO K = 1, 5
            DO L = 1, 3
              NAR = K + 6 * ( L - 1 )
              NBR = NBR + 1
C
C             TEST FOR STIFFNESS OR MASS STORAGE IN AMATRX
C
              IF ( NSOL .EQ. 1 ) AMATRX(NAR,NAC) = ELTSTF(NBR,NBC)
              IF ( NSOL .EQ. 2 .OR. NSOL .EQ. 3 ) THEN
                IF ( LFLAGS(3) .EQ. 2 ) THEN
                   AMATRX(NAR,NAC) = ELTSTF(NBR,NBC)
                ELSE IF ( LFLAGS(3) .EQ. 4 ) THEN
                   AMATRX(NAR,NAC) = AMASS(NBR,NBC)
                END IF
              END IF
            END DO
          END DO
        END DO
      END DO
C
C     FOR DYNAMIC ANALYSIS APPEND W1 AND W2 STIFFNESS
C     OR MASS MATRIX PARTITIONS TO AMATRX
C
      IF ( NSOL .EQ. 2 ) THEN
        NB = 0
        DO  I = 1, 5
          DO  J = 1, 3
            NA = I + 6 * ( J - 1 )
            NB = NB + 1
            DO II = 16, 17
```

```
            NII = II + 3
C
C           TEST FOR STIFFNESS OR MASS STORAGE IN AMATRX
C
            IF (LFLAGS(3) .EQ. 2 ) THEN
              AMATRX(NII,NA) = ELTSTF(II,NB)
              AMATRX(NA,NII) = ELTSTF(NB,II)
            END IF
            IF( LFLAGS(3) .EQ. 4 ) THEN
              AMATRX(NII,NA) = AMASS(II,NB)
              AMATRX(NA,NII) = AMASS(NB,II)
            END IF
          END DO
        END DO
      END DO
C
      DO I = 16, 17
        NI = I + 3
        DO J = 16, 17
          NJ = J + 3
          IF ( LFLAGS(3) .EQ. 2 ) AMATRX(NI,NJ) = ELTSTF(I,J)
          IF ( LFLAGS(3) .EQ. 4 ) AMATRX(NI,NJ) = AMASS(I,J)
        END DO
      END DO
    END IF
C
C   ACCOUNT FOR DIFFERENT SIGN CONVENTION IN ABAQUS FOR THETA X
C   AND ADD ARTIFICIAL STIFFNESS OR MASS TO BE ASSOCIATED WITH
C   THETA Z
C
    CALL UHOTFX ( AMATRX, NDOFEL )
C
C   TRANSFORM STIFFNESS OR MASS MATRICES AND ELEMENT
C   LOAD VECTOR TO GLOBAL COORDINATES
C
C             T                 T                 T
C   [K] = [T][K'][T] ; [M] = [T][M'][T] ; {R} = [T]{R'}
C
C
    NDOF = NDOFEL
    DO I = 1, NDOF
       RETN(I) = RHS(I,1)
    END DO
    CALL DGMMUL ( AMATRX,NDOF,TRANS,24,PETN,24,NDOF,NDOF,NDOF )
    CALL DGMATB ( TRANS,24,PETN,24,AMATRX,NDOF,NDOF,NDOF,NDOF )
    CALL DGMMUL ( TRANS,24,RETN,24,RHS,NDOF,NDOF,NDOF,1 )
C
    IF ( NSOL .EQ. 1 ) THEN
```

```
C
C           PERFORM REQUESTED ELEMENT DATA RECOVERY
C
C           TRANSFORM GLOBAL DISPLACEMENTS INTO ELEMENT COORDINATES
C
            CALL DGMMUL ( TRANS,24,DU,MLVARX,PETN,24,NDOF,NDOF,1)
            CALL DSTRN  ( PETN, DU, NDOF )
C
C           REINSTATE ORIGINAL SIGN OF THETA X FOR DATA RECOVERY
C
            DO I = 1, 3
               NN = (I-1)*6 + 4
               DU(NN) = -DU(NN)
            END DO
C
            CALL UHOTIO ( ELTSTF, ELTLOD, C, LFLAGS, ZL, DU, ENERGY,
     &                    A, B, PSI, PSU, AMAT, BMAT,  DMAT,   GMAT,
     &                    RELAX, NLAY, NOTYPE, NOUT , JELEM,   STRN,
     &                    ETRN, TRI, NSYS, NMAT, MLVARX             )
C
         END IF
C
C        HOT3 ELEMENT PROCESSING COMPLETED
C
      END IF
C
C     IF ( JTYPE .EQ. NUEL ) THEN
C
C        ******************************************************
C        *  INSERT MAIN PROGRAMS FOR OTHER USER DEFINED ELEMENTS *
C        *  SELECTED BY JTYPE HERE                              *
C        ******************************************************
C
C     END IF
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                      *
      SUBROUTINE UHOTTR ( X, Y, Z, AMV1, AMV2, A, B, TRI, STRN, ETRN,  *
     &                    SMTN, EMTN, TRANS )                          *
C                                                                      *
      IMPLICIT REAL*8 (A-H,O-Z)                                        *
C                                                                      *
C     CALCULATE TRANSFORMATION MATRICES FOR CONVERTING QUANTITIES      *
C     BETWEEN LOCAL AND GLOBAL COORDINATE SYSTEMS                      *
C                                                                      *
```

```
      DIMENSION X(3), Y(3), Z(3), XP(3), YP(3), ZP(3), EO(3,3),          *
     &          EP(3,3), VC12(3), VC13(3), A(3), B(3), AMV1(3),          *
     &          AMV2(3), TRI(3,3), TMI(3,3), EM(3,3), STRN(6,6),         *
     &          ETRN(6,6), SMTN(6,6), EMTN(6,6), TRANS(24,24)            *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     UNIT VECTORS IN GLOBAL SYSTEM
C
      EO(1,1) = 1.0
      EO(1,2) = 0.0
      EO(1,3) = 0.0
      EO(2,1) = 0.0
      EO(2,2) = 1.0
      EO(2,3) = 0.0
      EO(3,1) = 0.0
      EO(3,2) = 0.0
      EO(3,3) = 1.0
C
C     LOCAL ELEMENT COORDINATE VECTORS
C
      AL = SQRT( (X(2)-X(1))**2+(Y(2)-Y(1))**2+(Z(2)-Z(1))**2 )
      VC12(1) = (X(2)-X(1))/AL
      VC12(2) = (Y(2)-Y(1))/AL
      VC12(3) = (Z(2)-Z(1))/AL
      AL = SQRT( (X(3)-X(1))**2+(Y(3)-Y(1))**2+(Z(3)-Z(1))**2 )
      VC13(1) = (X(3)-X(1))/AL
      VC13(2) = (Y(3)-Y(1))/AL
      VC13(3) = (Z(3)-Z(1))/AL
C
      EP(1,1) = VC12(1)
      EP(1,2) = VC12(2)
      EP(1,3) = VC12(3)
C
      AI = VC12(2)*VC13(3)-VC12(3)*VC13(2)
      AJ = VC12(3)*VC13(1)-VC12(1)*VC13(3)
      AK = VC12(1)*VC13(2)-VC12(2)*VC13(1)
      AL = SQRT( AI**2 + AJ**2 + AK**2 )
      EP(3,1) = AI/AL
      EP(3,2) = AJ/AL
      EP(3,3) = AK/AL
C
      AI = EP(3,2)*EP(1,3)-EP(3,3)*EP(1,2)
      AJ = EP(3,3)*EP(1,1)-EP(3,1)*EP(1,3)
      AK = EP(3,1)*EP(1,2)-EP(3,2)*EP(1,1)
      AL = SQRT( AI**2 + AJ**2 + AK**2 )
      EP(2,1) = AI/AL
      EP(2,2) = AJ/AL
```

```
      EP(2,3) = AK/AL
C
C     CONSTRUCT COORDINATE TRANSFORMATION MATRIX
C
C                                        T
C     {X'} = [TRI]{X};    {X} = [TRI]{X'}
C
C
      DO I = 1, 3
         DO J = 1, 3
            TRI(I,J) = 0.0
            DO K = 1, 3
               TRI(I,J) = TRI(I,J) + EP(I,K)*EO(J,K)
            END DO
         END DO
      END DO
C
      CALL DSCPY ( 0.0D0, TRANS, 24, 24 )
C
      DO I = 1, 6
         NF = 3*(I-1)
         DO J = 1, 3
            DO K = 1, 3
               TRANS(NF+J,NF+K) = TRI(J,K)
            END DO
         END DO
         TRANS(18+I,18+I) = 1.0
      END DO
C
      DO I = 1, 3
         XP(I) = TRI(1,1)*X(I)+TRI(1,2)*Y(I)+TRI(1,3)*Z(I)
         YP(I) = TRI(2,1)*X(I)+TRI(2,2)*Y(I)+TRI(2,3)*Z(I)
         ZP(I) = TRI(3,1)*X(I)+TRI(3,2)*Y(I)+TRI(3,3)*Z(I)
      END DO
C
      A(1) = XP(3) - XP(2)
      A(2) = XP(1) - XP(3)
      A(3) = XP(2) - XP(1)
C
      B(1) = YP(2) - YP(3)
      B(2) = YP(3) - YP(1)
      B(3) = YP(1) - YP(2)
C
C     CONSTRUCT STRESS AND STRAIN TRANSFORMATION MATRICES
C     TO CONVERT BETWEEN LOCAL ELEMENT COORDINATE SYSTEM
C     AND GLOBAL SYSTEM
C
C     [Ts] = STRESS TRANSFORMATION
```

```
C     [Te] = STRAIN TRANSFORMATION
C
C     {E'} = [Te]{E};   {S'} = [Ts]{S}
c
C                T                 T
C     {E}  = [Ts]{E'};  {S}  = [Te]{S'}
C
C     ( CONVENTION: [Ex,Ey,Ez,Gyz,Gxz,Gxy] )
C
      CALL TENSRT ( TRI, STRN, ETRN )
C
C     CONSTRUCT TRANSFORMATION MATRIX TO ORIENT THE MATERIAL
C     PROPERTIES DEFINED IN THE GLOBAL SYSTEM TO THE LOCAL
C     ELEMENT COORDINATE SYSTEM
C
C     [Ts] = STRESS TRANSFORMATION FOR MATERIAL PROPERTIES
C     [Te] = STRAIN TRANSFORMATION FOR MATERIAL PROPERTIES
C
C                        T                        T
C     [C'] = [Ts][C][Ts]       [C] = [Te][C'][Te]
C
C     CONVENTION: [Ex,Ey,Ez,Gyz,Gxz,Gxy]
C
      DO I = 1, 3
         EM(1,I) = AMV1(I)/SQRT(AMV1(1)**2+AMV1(2)**2+AMV1(3)**2)
         EM(2,I) = AMV2(I)/SQRT(AMV2(1)**2+AMV2(2)**2+AMV2(3)**2)
      END DO
      AI = EM(1,2)*EM(2,3)-EM(1,3)*EM(2,2)
      AJ = EM(1,3)*EM(2,1)-EM(1,1)*EM(2,3)
      AK = EM(1,1)*EM(2,2)-EM(1,2)*EM(2,1)
      AL = SQRT( AI**2 + AJ**2 + AK**2 )
      EM(3,1) = AI/AL
      EM(3,2) = AJ/AL
      EM(3,3) = AK/AL
C
      DO I = 1, 3
         DO J = 1, 3
            TMI(I,J) = 0.0
            DO K = 1, 3
               TMI(I,J) = TMI(I,J) + EP(I,K)*EM(J,K)
            END DO
         END DO
      END DO
C
      CALL TENSRT ( TMI, SMTN, EMTN )
C
      RETURN
      END
```

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
      SUBROUTINE TENSRT ( T, TS, TE )                                 *
C                                                                     *
      IMPLICIT REAL*8 (A-H,O-Z)                                       *
C                                                                     *
C     COMPUTE STRESS AND STRAIN TRANSFORMATION MATRICES               *
C                                                                     *
      DIMENSION T(3,3), TS(6,6), TE(6,6)                              *
C                                                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     CONSTRUCT TRANSFORMATION MATRICES TO CONVERT
C     BETWEEN REFERENCE COORDINATE SYSTEMS
C
C     [Ts] = STRESS TRANSFORMATION
C     [Te] = STRAIN TRANSFORMATION
C
C     {E'} = [Te]{E};   {S'} = [Ts]{S}
C
C                T                 T
C     {E}  = [Ts]{E'};  {S}  = [Te]{S'}
C
C       CONVENTION: [Ex,Ey,Ez,Gyz,Gxz,Gxy] )
C
C     STRESS TRANSFORMATION MATRIX, [Ts]:
C
      TS(1,1) =  T(1,1)**2
      TS(1,2) =  T(1,2)**2
      TS(1,3) =  T(1,3)**2
      TS(1,4) =  2*T(1,2)*T(1,3)
      TS(1,5) =  2*T(1,3)*T(1,1)
      TS(1,6) =  2*T(1,1)*T(1,2)
      TS(2,1) =  T(2,1)**2
      TS(2,2) =  T(2,2)**2
      TS(2,3) =  T(2,3)**2
      TS(2,4) =  2*T(2,2)*T(2,3)
      TS(2,5) =  2*T(2,3)*T(2,1)
      TS(2,6) =  2*T(2,1)*T(2,2)
      TS(3,1) =  T(3,1)**2
      TS(3,2) =  T(3,2)**2
      TS(3,3) =  T(3,3)**2
      TS(3,4) =  2*T(3,2)*T(3,3)
      TS(3,5) =  2*T(3,3)*T(3,1)
      TS(3,6) =  2*T(3,1)*T(3,2)
      TS(4,1) =  T(2,1)*T(3,1)
      TS(4,2) =  T(2,2)*T(3,2)
```

```
      TS(4,3) =  T(2,3)*T(3,3)
      TS(4,4) =  T(2,2)*T(3,3) + T(3,2)*T(2,3)
      TS(4,5) =  T(2,3)*T(3,1) + T(3,3)*T(2,1)
      TS(4,6) =  T(2,1)*T(3,2) + T(3,1)*T(2,2)
      TS(5,1) =  T(3,1)*T(1,1)
      TS(5,2) =  T(3,2)*T(1,2)
      TS(5,3) =  T(3,3)*T(1,3)
      TS(5,4) =  T(3,2)*T(1,3) + T(1,2)*T(3,3)
      TS(5,5) =  T(3,3)*T(1,1) + T(1,3)*T(3,1)
      TS(5,6) =  T(3,1)*T(1,2) + T(1,1)*T(3,2)
      TS(6,1) =  T(1,1)*T(2,1)
      TS(6,2) =  T(1,2)*T(2,2)
      TS(6,3) =  T(1,3)*T(2,3)
      TS(6,4) =  T(1,2)*T(2,3) + T(2,2)*T(1,3)
      TS(6,5) =  T(1,3)*T(2,1) + T(2,3)*T(1,1)
      TS(6,6) =  T(1,1)*T(2,2) + T(2,1)*T(1,2)
C
C     STRAIN TRANSFORMATION MATRIX, [Te]:
C
      DO I = 1, 3
         DO J = 1, 3
            TE(I,J)     = TS(I,J)
            TE(I,J+3)   = TS(I,J+3)/2.0
            TE(I+3,J)   = TS(I+3,J)/0.5
            TE(I+3,J+3) = TS(I+3,J+3)
         END DO
      END DO
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                       *
      SUBROUTINE USFMEM ( SMX, A, B, THICK, DMX )                       *
C                                                                       *
      IMPLICIT REAL*8 (A-H,O-Z)                                         *
C                                                                       *
C     CALCULATE MEMBRANE EFFECTS OF THE HOT3 TRIANGULAR PLATE           *
C     ELEMENT INCORPORATING HIGHER ORDER KINEMATIC EXPANSIONS           *
C                                                                       *
      DIMENSION DMX(4,4), BMX(4,17), SMX(17,17), BTD(17,4), A(3), B(3)  *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      AREA = ( A(3) * B(2) - A(2) * B(3) ) / 2.0
C
      CALL DSCPY  ( 0.D0, BMX, 4, 17)
C
```

```
      CALL UBXMEM (  BMX, 4, A, B, THICK )
      CALL DMATB  (  BMX, DMX, BTD, 17,  4, 4 )
      CALL DMMUL  (  BTD, BMX, SMX, 17,  4, 17 )
      CALL DSMUL  ( AREA, SMX, SMX, 17, 17)
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
      SUBROUTINE USFBND ( SMX, A, B, THICK, DMX )                     *
C                                                                     *
      IMPLICIT REAL*8 (A-H,O-Z)                                       *
C                                                                     *
C     CALCULATE BENDING STIFFNESS MATRIX OF THE HOT3 TRIANGULAR PLATE *
C     ELEMENT INCORPORATING HIGHER ORDER KINEMATIC EXPANSIONS         *
C                                                                     *
      DIMENSION DMX(4,4), BMX(4,17), SMX(17,17), BTD(17,4), A(3), B(3) *
C                                                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      AREA = ( A(3) * B(2) - A(2) * B(3) ) / 2.0
C
      CALL DSCPY  ( 0.D0, BMX, 4, 17)
C
      CALL UBXBND (  BMX, 4, A, B, THICK )
      CALL DMATB  (  BMX, DMX, BTD, 17,  4,  4 )
      CALL DMMUL  (  BTD, BMX, SMX, 17,  4, 17 )
      CALL DSMUL  ( AREA, SMX, SMX, 17, 17)
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
      SUBROUTINE USFCPL ( SMX, A, B, THICK, DMX )                     *
C                                                                     *
      IMPLICIT REAL*8 (A-H,O-Z)                                       *
C                                                                     *
C     CALCULATE COUPLING EFFECTS OF THE HOT3 TRIANGULAR PLATE         *
C     ELEMENT INCORPORATING HIGHER ORDER KINEMATIC EXPANSIONS         *
C                                                                     *
      DIMENSION DMX(4,4), BMX(4,17), SMX(17,17), BTD(17,4), BTDB(17,17),*
     &          BTDTB(17,17), A(3), B(3)                              *
C                                                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      AREA = ( A(3) * B(2) - A(2) * B(3) ) / 2.0
```

```
C
      CALL DSCPY  ( 0.0D0,   BMX,    4, 17)
      CALL UBXMEM (   BMX,     4,    A,  B, THICK )
      CALL DMATB  (   BMX,   DMX,  BTD, 17, 4, 4  )
      CALL DSCPY  ( 0.0D0,   BMX,    4, 17)
      CALL UBXBND (   BMX,     4,    A,  B, THICK )
      CALL DMMUL  (   BTD,   BMX, BTDB, 17,  4, 17 )
      CALL DMTRN  (  BTDB, BTDTB,   17, 17)
      CALL DMADD  (  BTDB, BTDTB,  SMX, 17, 17 )
      CALL DSMUL  (  AREA,   SMX,  SMX, 17, 17 )
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                       *
      SUBROUTINE USFSHR ( SMX, A, B, DMX )                              *
C                                                                       *
      PARAMETER ( KORD = 7 )                                            *
C                                                                       *
      IMPLICIT REAL*8 (A-H,O-Z)                                         *
C                                                                       *
C     CALCULATE SHEAR STIFFNESS OF THE HOT3 TRIANGULAR PLATE            *
C     ELEMENT INCORPORATING HIGHER ORDER KINEMATIC EXPANSIONS           *
C                                                                       *
      DIMENSION DMX(2,2), BMX(2,17), SMX(17,17), BTD(17,2), BTDB(17,17),*
     &          A(3), B(3)                                              *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      AREA = ( A(3) * B(2) - A(2) * B(3) ) / 2.0
C
      CALL DSCPY ( 0.D0, SMX, 17, 17 )
      CALL DSCPY ( 0.D0, BMX, 2, 17 )
C
      DO INT= 1, KORD
         CALL UINTPT ( X1, X2, X3, WEIGHT, KORD, INT )
         CALL UBXSHR ( BMX, 2, A, B, X1, X2, X3 )
         CALL DMATB  ( BMX, DMX, BTD, 17, 2, 2 )
         CALL DMMUL  ( BTD, BMX, BTDB, 17, 2, 17)
         CALL DMASB  ( SMX, AREA*WEIGHT, BTDB, SMX, 17, 17 )
      END DO
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                       *
```

```
      SUBROUTINE UBXMEM ( BMX, LR, A, B, THICK )                        *
C                                                                       *
      IMPLICIT REAL*8 (A-H,O-Z)                                         *
C                                                                       *
C     CALCULATE MEMBRANE B-MATRIX                                       *
C                                                                       *
      DIMENSION BMX(LR,17), A(3), B(3)                                  *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      AREA = ( A(3) * B(2) - A(2) * B(3) ) / 2.0
C
      BMX( 1, 1) = B(1) / ( 2. * AREA )
      BMX( 1, 2) = B(2) / ( 2. * AREA )
      BMX( 1, 3) = B(3) / ( 2. * AREA )
C
      BMX( 2, 4) = A(1) / ( 2. * AREA )
      BMX( 2, 5) = A(2) / ( 2. * AREA )
      BMX( 2, 6) = A(3) / ( 2. * AREA )
C
      BMX( 3,16) = 2.0 / THICK
C
      BMX( 4, 1) = A(1) / ( 2. * AREA )
      BMX( 4, 2) = A(2) / ( 2. * AREA )
      BMX( 4, 3) = A(3) / ( 2. * AREA )
      BMX( 4, 4) = B(1) / ( 2. * AREA )
      BMX( 4, 5) = B(2) / ( 2. * AREA )
      BMX( 4, 6) = B(3) / ( 2. * AREA )
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                       *
      SUBROUTINE UBXBND ( BMX, LR, A, B, THICK )                        *
C                                                                       *
      IMPLICIT REAL*8 (A-H,O-Z)                                         *
C                                                                       *
C     CALCULATE BENDING B-MATRIX                                        *
C                                                                       *
      DIMENSION BMX(LR,17), A(3), B(3)                                  *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      AREA = ( A(3) * B(2) - A(2) * B(3) ) / 2.0
C
      BMX( 1,13) = B(1) / ( 2. * AREA )
      BMX( 1,14) = B(2) / ( 2. * AREA )
```

```
      BMX( 1,15) = B(3) / ( 2. * AREA )
C
      BMX( 2,10) = A(1) / ( 2. * AREA )
      BMX( 2,11) = A(2) / ( 2. * AREA )
      BMX( 2,12) = A(3) / ( 2. * AREA )
C
      BMX( 3,17) = 2.0 / THICK
C
      BMX( 4,10) = B(1) / ( 2. * AREA )
      BMX( 4,11) = B(2) / ( 2. * AREA )
      BMX( 4,12) = B(3) / ( 2. * AREA )
      BMX( 4,13) = A(1) / ( 2. * AREA )
      BMX( 4,14) = A(2) / ( 2. * AREA )
      BMX( 4,15) = A(3) / ( 2. * AREA )
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                       *
      SUBROUTINE UBXSHR ( BMX, LR, A, B, X1, X2, X3 )                   *
C                                                                       *
      IMPLICIT REAL*8 (A-H,O-Z)                                         *
C                                                                       *
C     CALCULATE SHEAR B-MATRIX                                          *
C                                                                       *
      DIMENSION BMX(LR,17), A(3), B(3)                                  *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      AREA = ( A(3) * B(2) - A(2) * B(3) ) / 2.0
C
      BMX(1, 7) = A(1) / ( 2. * AREA )
      BMX(1, 8) = A(2) / ( 2. * AREA )
      BMX(1, 9) = A(3) / ( 2. * AREA )
      BMX(1,10) = (  B(3) * ( X2 * A(1) + X1 * A(2) )
     &             - B(2) * ( X3 * A(1) + X1 * A(3) ))/(4.*AREA) + X1
      BMX(1,11) = (  B(1) * ( X3 * A(2) + X2 * A(3) )
     &             - B(3) * ( X2 * A(1) + X1 * A(2) ))/(4.*AREA) + X2
      BMX(1,12) = (- B(1) * ( X3 * A(2) + X2 * A(3) )
     &             + B(2) * ( X3 * A(1) + X1 * A(3) ))/(4.*AREA) + X3
      BMX(1,13) = (  A(2) * ( X3 * A(1) + X1 * A(3) )
     &             - A(3) * ( X2 * A(1) + X1 * A(2) ))/(4.*AREA)
      BMX(1,14) = (- A(1) * ( X3 * A(2) + X2 * A(3) )
     &             + A(3) * ( X2 * A(1) + X1 * A(2) ))/(4.*AREA)
      BMX(1,15) = (  A(1) * ( X3 * A(2) + X2 * A(3) )
     &             - A(2) * ( X3 * A(1) + X1 * A(3) ))/(4.*AREA)
C
```

```
      BMX(2, 7) = B(1) / ( 2. * AREA )
      BMX(2, 8) = B(2) / ( 2. * AREA )
      BMX(2, 9) = B(3) / ( 2. * AREA )
      BMX(2,10) = (  B(3) * ( X2 * B(1) + X1 * B(2) )
     &            - B(2) * ( X3 * B(1) + X1 * B(3) ))/(4.*AREA)
      BMX(2,11) = (  B(1) * ( X3 * B(2) + X2 * B(3) )
     &            - B(3) * ( X2 * B(1) + X1 * B(2) ))/(4.*AREA)
      BMX(2,12) = (- B(1) * ( X3 * B(2) + X2 * B(3) )
     &            + B(2) * ( X3 * B(1) + X1 * B(3) ))/(4.*AREA)
      BMX(2,13) = (  A(2) * ( X3 * B(1) + X1 * B(3) )
     &            - A(3) * ( X2 * B(1) + X1 * B(2) ))/(4.*AREA) + X1
      BMX(2,14) = (- A(1) * ( X3 * B(2) + X2 * B(3) )
     &            + A(3) * ( X2 * B(1) + X1 * B(2) ))/(4.*AREA) + X2
      BMX(2,15) = (  A(1) * ( X3 * B(2) + X2 * B(3) )
     &            - A(2) * ( X3 * B(1) + X1 * B(3) ))/(4.*AREA) + X3
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE UFCSHR ( STFSHR, STFBND, RELAX )                        *
C                                                                        *
C     FCSHR COMPUTES THE PENALTY FACTOR RELAX FOR THE SHEAR STIFFNESS    *
C     COMPONENTS TO ENFORCE KIRCHHOFF BEHAVIOR IN THE THIN LIMIT         *
C                                                                        *
      IMPLICIT REAL*8 (A-H,O-Z)                                          *
C                                                                        *
      DIMENSION STFBND(17,17), STFSHR(17,17)                             *
C                                                                        *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SBEND = 0.0D0
      SHEAR = 0.0D0
C
      DO I = 10, 15
         SBEND = SBEND + STFBND(I,I)
         SHEAR = SHEAR + STFSHR(I,I)
      END DO
C
      IF ( SBEND .EQ. 0.0 ) THEN
         RELAX = 0.0
      ELSE
         SCALE = 2.0D0
         RATIO = SBEND / SHEAR
         RELAX = SCALE * RATIO / ( 1.0D0 + SCALE * RATIO )
      END IF
C
```

```
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
      SUBROUTINE UHOTBX ( BMX, A, B, X1, X2, X3, THICK )              *
C                                                                     *
      IMPLICIT REAL*8 (A-H,O-Z)                                       *
C                                                                     *
C     CALCULATE THE STRAIN-DISPLACEMENT MATRIX                        *
C     FOR THE HOT3 ELEMENT                                            *
C                                                                     *
      DIMENSION BMX(10,17), A(3), B(3)                                *
C                                                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      CALL DSCPY  ( 0.0D+0, BMX, 10, 17 )
C
      CALL UBXMEM ( BMX( 1,1), 10, A, B, THICK )
      CALL UBXBND ( BMX( 5,1), 10, A, B, THICK )
      CALL UBXSHR ( BMX( 9,1), 10, A, B, X1, X2, X3 )
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
      SUBROUTINE UHOTSF ( ELTSTF, AMAT, BMAT, DMAT, GMAT, A, B, THICK,  *
     &                    RELAX                                      ) *
C                                                                     *
      IMPLICIT REAL*8 (A-H,O-Z)                                       *
C                                                                     *
C     CALCULATE THE ELASTIC STIFFNESS MATRIX FOR THE HOT3 ELEMENT     *
C                                                                     *
      DIMENSION ELTSTF(17,17), STFMEM(17,17), STFBND(17,17),          *
     &          STFCPL(17,17), STFSHR(17,17), AMAT(4,4), BMAT(4,4),   *
     &          DMAT(4,4), GMAT(2,2), A(3), B(3)                      *
C                                                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      CALL DSCPY  ( 0.0D0, ELTSTF, 17, 17 )
C
      CALL USFMEM ( STFMEM, A, B, THICK, AMAT )
      CALL DMADD  ( STFMEM, ELTSTF, ELTSTF, 17, 17 )
      CALL USFBND ( STFBND, A, B, THICK, DMAT )
      CALL DMADD  ( STFBND, ELTSTF, ELTSTF, 17, 17 )
      CALL USFCPL ( STFCPL, A, B, THICK, BMAT )
      CALL DMADD  ( STFCPL, ELTSTF, ELTSTF, 17, 17 )
```

```
      CALL USFSHR ( STFSHR, A, B, GMAT )
C
      CALL UFCSHR ( STFSHR, STFBND, RELAX )
      CALL DMASB  ( ELTSTF, RELAX, STFSHR, ELTSTF, 17, 17 )
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
      SUBROUTINE UHOTMM ( AMASS, A, B, THK, NSOL, RHO )               *
C                                                                     *
      IMPLICIT REAL*8 (A-H,O-Z)                                       *
C                                                                     *
C     CALCULATE HOT3 ELEMENT CONSISTENT MASS MATRIX                   *
C                                                                     *
      DIMENSION AMASS(17,17), SHAPE(3,17),    A(3),   B(3),           *
     &          TEMP(17,17),  PSI(3), XML(3), XMM(3), XMN(3)          *
C                                                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      CALL DSCPY  ( 0.0D0, AMASS, 17, 17 )
      CALL DSCPY  ( 0.0D0, SHAPE,  3, 17 )
C
      H = THK/2.0
C
      AREA = ( A(3) * B(2) - A(2) * B(3) ) / 2.0
C
C     NSOL FLAG USED TO INCLUDE/EXCLUDE MASS ASSOCIATED
C     WITH HIGHER-ORDER FUNCTIONS W1 & W2;
C
      BETA = 1.0
      IF ( NSOL .EQ. 3 ) BETA = 0.0
C
C     SET ORDER OF INTEGRATION ACROSS THICKNESS
C
      NORD = 4
C
C     SET ORDER OF INTEGRATION OVER AREA
C
      KORD = 7
C
      DO 100 INT = 1, KORD
C
C        INVOKE GUASS INTEGRATION FOR A TRIANGLE
C
         CALL UINTPT ( X1, X2, X3, WGHT1, KORD, INT )
C
```

```
      PSI(1) = X1
      PSI(2) = X2
      PSI(3) = X3
C
C     ANISOPARAMETRIC SHAPE FUNCTIONS
C
      XMN(1) = 4.0*X1*X2
      XMN(2) = 4.0*X2*X3
      XMN(3) = 4.0*X3*X1
C
      XML(1) =  B(3)/8.0*XMN(1)-B(2)/8.0*XMN(3)
      XML(2) =  B(1)/8.0*XMN(2)-B(3)/8.0*XMN(1)
      XML(3) = -B(1)/8.0*XMN(2)+B(2)/8.0*XMN(3)
C
      XMM(1) =  A(2)/8.0*XMN(3)-A(3)/8.0*XMN(1)
      XMM(2) = -A(1)/8.0*XMN(2)+A(3)/8.0*XMN(1)
      XMM(3) =  A(1)/8.0*XMN(2)-A(2)/8.0*XMN(3)
C
      DO 110 IZ = 1, NORD
C
         CALL UIGSS ( NORD, IZ, CEE, WGHT2 )
C
         DO I = 1, 3
            SHAPE(1,I)    =  PSI(I)
            SHAPE(1,I+12) =  H*CEE*PSI(I)
            SHAPE(2,I+3)  =  PSI(I)
            SHAPE(2,I+9)  =  H*CEE*PSI(I)
            SHAPE(3,I+6)  =  PSI(I)
            SHAPE(3,I+9)  =  XML(I)
            SHAPE(3,I+12) =  XMM(I)
         END DO
         SHAPE(3,16) = BETA*CEE
         SHAPE(3,17) = BETA*(CEE**2 - 0.2)
C
C        NUMERICALLY INTEGRATE MASS MATRIX
C
         CALL DMATB ( SHAPE, SHAPE, TEMP, 17, 3, 17 )
         FAC = RHO*H*AREA*WGHT1*WGHT2
C
         DO I = 1, 17
            DO J = 1, 17
               AMASS(I,J) = AMASS(I,J) + FAC*TEMP(I,J)
            END DO
         END DO
C
  110    CONTINUE
C
  100 CONTINUE
```

```
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
      SUBROUTINE UHOTFX ( AMATRX, NDOF )                              *
C                                                                     *
      IMPLICIT REAL*8 (A-H,O-Z)                                       *
C                                                                     *
C     CHANGE SIGN OF THETA X TO ACCOUNT FOR DIFFERENT SIGN            *
C     CONVENTION IN ABAQUS AND ADD ARTIFICIAL STIFFNESS OR            *
C     MASS TO BE ASSOCIATED WITH THETA Z                              *
C                                                                     *
      DIMENSION AMATRX(NDOF,NDOF)                                     *
C                                                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DO I = 1, 3
         NN = (I-1)*6 + 4
         DO J = 1, NDOF
            AMATRX(NN,J) = -AMATRX(NN,J)
            AMATRX(J,NN) = -AMATRX(J,NN)
         END DO
      END DO
C
C     SET SCALE FACTOR FOR MINIMUM DIAGONAL TERM
C
      SCALE = 1.0E-5
C
      SMIN = AMATRX(1,1)
      DO I = 1, 3
         DO J = 1, 5
            N = 6*(I-1) + J
            SMIN = MIN( AMATRX(N,N), SMIN )
         END DO
      END DO
      DO I = 1, 3
         N = 6*I
         AMATRX(N,N) = SCALE * SMIN
      END DO
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
      SUBROUTINE UHOTLD ( Q, A, B, ELTLOD )                           *
```

```
C                                                                       *
C     HTLOD COMPUTES THE CONSISTENT LOAD VECTOR FOR                     *
C     THE HOT3 ELEMENT BASED ON HIGHER ORDER TESSLER                    *
C     PLATE THEORY                                                      *
C                                                                       *
      IMPLICIT REAL*8 (A-H,O-Z)                                         *
C                                                                       *
C     HOT3 IS FORMULATED TO ACCEPT LINEARLY VARYING LOAD                *
C     DISTRIBUTIONS TO BE APPLIED ON BOTH THE UPPER AND                 *
C     LOWER SURFACES. CURRENT RESTRICTION IN THE INPUT                  *
C     MODULE LIMITS DISTRIBUTED LOAD INPUT TO ONLY ONE                  *
C     SURFACE WHICH IS ASSUMED TO BE THE TOP SURFACE                    *
C                                                                       *
      DIMENSION ELTLOD(17), A(3), B(3), Q(3)                            *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      CALL DSCPY ( 0.0D+0, ELTLOD, 17, 1)
C
      IF (( Q(1).NE.0.0 ).OR.( Q(2).NE.0.0 ).OR.( Q(3).NE.0.0 )) THEN
C
         AREA       = ( A(3) * B(2) - A(2) * B(3) ) / 2.0
C
         ELTLOD( 7) = AREA * (   2.*Q(1) +    Q(2) +    Q(3) ) / 12.0D0
         ELTLOD( 8) = AREA * (      Q(1) + 2.*Q(2) +    Q(3) ) / 12.0D0
         ELTLOD( 9) = AREA * (      Q(1) +    Q(2) + 2.*Q(3) ) / 12.0D0
         ELTLOD(10) = AREA * ( ( 2.*Q(1) + 2.*Q(2) +    Q(3) ) * B(3)
     &                        -( 2.*Q(1) +    Q(2) + 2.*Q(3) ) * B(2) )/
     &                         120.0D0
         ELTLOD(11) = AREA * ( (    Q(1) + 2.*Q(2) + 2.*Q(3) ) * B(1)
     &                        -( 2.*Q(1) + 2.*Q(2) +    Q(3) ) * B(3) )/
     &                         120.0D0
         ELTLOD(12) = AREA * ( ( 2.*Q(1) +    Q(2) + 2.*Q(3) ) * B(2)
     &                        -(    Q(1) + 2.*Q(2) + 2.*Q(3) ) * B(1) )/
     &                         120.0D0
         ELTLOD(13) = AREA * ( ( 2.*Q(1) +    Q(2) + 2.*Q(3) ) * A(2)
     &                        -( 2.*Q(1) + 2.*Q(2) +    Q(3) ) * A(3) )/
     &                         120.0D0
         ELTLOD(14) = AREA * ( ( 2.*Q(1) + 2.*Q(2) +    Q(3) ) * A(3)
     &                        -(    Q(1) + 2.*Q(2) + 2.*Q(3) ) * A(1) )/
     &                         120.0D0
         ELTLOD(15) = AREA * ( (    Q(1) + 2.*Q(2) + 2.*Q(3) ) * A(1)
     &                        -( 2.*Q(1) +    Q(2) + 2.*Q(3) ) * A(2) )/
     &                         120.0D0
         ELTLOD(16) = AREA * ( Q(1) + Q(2) + Q(3) ) / 3.0D0
         ELTLOD(17) = AREA * ( Q(1) + Q(2) + Q(3) ) / 2.5D0
      END IF
C
```

```
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
      SUBROUTINE UHOTMT( E1, E2, E3,  G12,  G23,  G31,  V12,  V23, V13,  *
     &                   FIB,  Z,  C, STRN, AMAT, BMAT, DMAT, GMAT,     *
     &                   NLAY, NMAT                                  )  *
C                                                                     *
C     MATPRP COMPUTES THE EQUIVALENT COMPOSITE A,B,D AND              *
C     G MATRICES FOR COMPOSITE PLATE PROPERTIES IN THE                *
C     HOT3 ELEMENT                                                    *
C                                                                     *
      IMPLICIT REAL*8 (A-H,O-Z)                                       *
C                                                                     *
      DIMENSION AMAT(4,4), BMAT(4,4), DMAT(4,4), GMAT(2,2),           *
     &          ABAR(6,6), BBAR(6,6), DBAR(6,6), C(6,6,125),          *
     &          E1(1), E2(1), E3(1), V12(1), V23(1), V13(1),          *
     &          G12(1), G23(1), G31(1), FIB(1), Z(1), TEMP(6,6),      *
     &          PII(6), PIZ(6), PIP(6,6), STRN(6,6), C1(6,6,125),     *
     &          S33(25),RA13(25),RA23(25),RA63(25),CHH(25,3),         *
     &          AAK(25,4),CH(25,7),BK(25,3),D3(25),ABC(25,4)          *
C                                                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     OPTION FLAG TO SIMULATE FIRST ORDER SHEAR-DEFORMABLE THEORY
C
      NSDF = 0
C
      DO K = 1, NLAY
C
         IF ( NSDF .EQ. 1 ) THEN
C
C           SIMULATE FIRST-ORDER SHEAR DEFORMABLE THEORY:
C
            V13(K) = 0.0
            V23(K) = 0.0
            E3(K)  = 1.0E6*E1(K)
C
         END IF
C
         CALL UMTCMX ( E1(K),  E2(K),  E3(K), V12(K), V23(K), V13(K),
     &                 G12(K), G23(K), G31(K), FIB(K), C1(1,1,K) )

C
C        TRANSFORM MATERIAL PROPERTY MATRIX TO LOCAL COORDINATE SYSTEM
C
C                          T       T       T
```

```
C        [C'] = [Ts][C][Ts]   ; [C'] = [Ts][C][Ts]
C
         CALL DMMUL( STRN, C1(1,1,K), TEMP, 6, 6, 6 )
         CALL DMABT( TEMP, STRN, C(1,1,K), 6, 6, 6 )
C
      END DO
C
      PI = ACOS(-1.0D0)
C
      HALF = ( Z(NLAY+1) - Z(1) ) / 2.0
      THK  = 2*HALF
      AK   = SQRT(5.D0/6.D0)
      AKZ  = SQRT(42.D0/85.D0)
C
C     INITIALIZE MATRICES
C
      CALL DSCPY ( 0.0D0, ABAR, 6, 6 )
      CALL DSCPY ( 0.0D0, BBAR, 6, 6 )
      CALL DSCPY ( 0.0D0, DBAR, 6, 6 )
      CALL DSCPY ( 0.0D0, GMAT, 2, 2 )
C
      IF ( NMAT .EQ. 1 ) THEN
C
C       COMPUTE LAMINATE CONSTITUTIVE MATRICES USING
C       STRAIN-BASED VERSION OF HIGHER-ORDER THEORY
C
      H1  = ( HALF    )
      H2  = ( HALF**2 )
      H3  = ( HALF**3 )
      H4  = ( HALF**4 )
      H5  = ( HALF**5 )
      H6  = ( HALF**6 )
C
C     FORM CONSTANTS TO ENFORCE ZERO SHEAR BOUNDARY
C     CONDITION AT THE TOP AND BOTTOM SURFACES
C
      R13 = C(1,3,1) / C(3,3,1) - C(1,3,NLAY) / C(3,3,NLAY)
      R23 = C(2,3,1) / C(3,3,1) - C(2,3,NLAY) / C(3,3,NLAY)
      R36 = C(3,6,1) / C(3,3,1) - C(3,6,NLAY) / C(3,3,NLAY)
      S13 = C(1,3,1) / C(3,3,1) + C(1,3,NLAY) / C(3,3,NLAY)
      S23 = C(2,3,1) / C(3,3,1) + C(2,3,NLAY) / C(3,3,NLAY)
      S36 = C(3,6,1) / C(3,3,1) + C(3,6,NLAY) / C(3,3,NLAY)
C
      A1  = - ( HALF / 510.0 ) * S13
      B1  = - ( HALF / 510.0 ) * 85. * R13
      A2  = - ( HALF / 510.0 ) * S23
      B2  = - ( HALF / 510.0 ) * 85. * R23
      A6  = - ( HALF / 510.0 ) * S36
```

```
      B6  = - ( HALF / 510.0 ) * 85. * R36
      A3  =   42.D0/85.D0
C
C     COMPUTE LAMINATE PROPERTY MATRICES
C
      DO K= 1, NLAY
C
         Z1  = ( Z(K+1)    - Z(K)    )
         Z2  = ( Z(K+1)**2 - Z(K)**2 )
         Z3  = ( Z(K+1)**3 - Z(K)**3 )
         Z4  = ( Z(K+1)**4 - Z(K)**4 )
         Z5  = ( Z(K+1)**5 - Z(K)**5 )
         Z6  = ( Z(K+1)**6 - Z(K)**6 )
         Z7  = ( Z(K+1)**7 - Z(K)**7 )
C
C        EXPLICIT INTEGRATION OF PHI TERMS ( OBTAINED FROM MACSYMA ) :
C
C        INT[PHI(I)]DZ
C
      PII(1) = ( 105*A1*Z4-2*B1*H1*Z3-120*A1*H2*Z2+2*B1*H3*Z1 ) / (4*H3)
      PII(2) = ( 105*A2*Z4-2*B2*H1*Z3-120*A2*H2*Z2+2*B2*H3*Z1 ) / (4*H3)
      PII(3) = (-  5*A3*Z4           + 30*A3*H2*Z2            ) / (8*H3)
      PII(6) = ( 105*A6*Z4-2*B6*H1*Z3-120*A6*H2*Z2+2*B6*H3*Z1 ) / (4*H3)
C
C        INT[PHI(I)Z]DZ
C
      PIZ(1) = ( 168*A1*Z5-3*B1*H1*Z4-160*A1*H2*Z3+2*B1*H3*Z2 ) / (8*H3)
      PIZ(2) = ( 168*A2*Z5-3*B2*H1*Z4-160*A2*H2*Z3+2*B2*H3*Z2 ) / (8*H3)
      PIZ(3) = (-    A3*Z5           +  5*A3*H2*Z3            ) / (2*H3)
      PIZ(6) = ( 168*A6*Z5-3*B6*H1*Z4-160*A6*H2*Z3+2*B6*H3*Z2 ) / (8*H3)
C
C        INT[PHI(I)PHI(J)]DZ
C
      PIP(1,1) = (( 31500*A1*A1                      )   *Z7 +
     &           (            -1050*A1*B1            )*H1*Z6 +
     &           (-50400*A1*A1            + 9*B1*B1  )*H2*Z5 +
     &           (              1425*A1*B1           )*H3*Z4 +
     &           ( 24000*A1*A1            -10*B1*B1  )*H4*Z3 +
     &           (            - 600*A1*B1            )*H5*Z2 +
     &           (                          5*B1*B1  )*H6*Z1 ) / (20*H6)
      PIP(1,2) = (( 63000*A1*A2                      )*   Z7 +
     &           (          -1050*(A1*B2+B1*A2)      )*H1*Z6 +
     &           (-100800*A1*A2            +18*B1*B2)*H2*Z5 +
     &           (           1425*(A1*B2+B1*A2)      )*H3*Z4 +
     &           ( 48000*A1*A2             -20*B1*B2)*H4*Z3 +
     &           (           -600*(A1*B2+B1*A2)      )*H5*Z2 +
     &           (                         10*B1*B2)*H6*Z1 ) / (40*H6)
      PIP(1,3) = (( -300*A3*A1                       )*   Z7 +
```

```
&            (                          5*A3*B1)*H1*Z6 +
&            (   1500*A3*A1                    )*H2*Z5 +
&            (                        -25*A3*B1)*H3*Z4 +
&            (  -1200*A3*A1                    )*H4*Z3 +
&            (                         15*A3*B1)*H5*Z2 ) / ( 8*H6)
 PIP(1,6) = ((  63000*A1*A6                    )*   Z7 +
&            (          -1050*(A1*B6+B1*A6)    )*H1*Z6 +
&            (-100800*A1*A6            +18*B1*B6)*H2*Z5 +
&            (           1425*(A1*B6+B1*A6)    )*H3*Z4 +
&            (  48000*A1*A6            -20*B1*B6)*H4*Z3 +
&            (           -600*(A1*B6+B1*A6)    )*H5*Z2 +
&            (                         10*B1*B6)*H6*Z1 ) / (40*H6)
 PIP(2,2) = (( 31500*A2*A2                     )   *Z7 +
&            (            -1050*A2*B2          )*H1*Z6 +
&            (-50400*A2*A2           + 9*B2*B2  )*H2*Z5 +
&            (             1425*A2*B2          )*H3*Z4 +
&            ( 24000*A2*A2           -10*B2*B2  )*H4*Z3 +
&            (            - 600*A2*B2          )*H5*Z2 +
&            (                        5*B2*B2  )*H6*Z1 ) / (20*H6)
 PIP(2,3) = ((   -300*A3*A2                    )*   Z7 +
&            (                          5*A3*B2)*H1*Z6 +
&            (   1500*A3*A2                    )*H2*Z5 +
&            (                        -25*A3*B2)*H3*Z4 +
&            (  -1200*A3*A2                    )*H4*Z3 +
&            (                         15*A3*B2)*H5*Z2 ) / ( 8*H6)
 PIP(2,6) = ((  63000*A2*A6                    )*   Z7 +
&            (          -1050*(A2*B6+B2*A6)    )*H1*Z6 +
&            (-100800*A2*A6            +18*B2*B6)*H2*Z5 +
&            (           1425*(A2*B6+B2*A6)    )*H3*Z4 +
&            (  48000*A2*A6            -20*B2*B6)*H4*Z3 +
&            (           -600*(A2*B6+B2*A6)    )*H5*Z2 +
&            (                         10*B2*B6)*H6*Z1 ) / (40*H6)
 PIP(3,3) = ((    25*A3*A3                     )   *Z7 +
&            (  -210*A3*A3                     )*H2*Z5 +
&            (   525*A3*A3                     )*H4*Z3 ) / (28*H6)
 PIP(3,6) = ((  -300*A3*A6                     )*   Z7 +
&            (                          5*A3*B6)*H1*Z6 +
&            (   1500*A3*A6                    )*H2*Z5 +
&            (                        -25*A3*B6)*H3*Z4 +
&            (  -1200*A3*A6                    )*H4*Z3 +
&            (                         15*A3*B6)*H5*Z2 ) / ( 8*H6)
 PIP(6,6) = (( 31500*A6*A6                     )   *Z7 +
&            (            -1050*A6*B6          )*H1*Z6 +
&            (-50400*A6*A6           + 9*B6*B6  )*H2*Z5 +
&            (             1425*A6*B6          )*H3*Z4 +
&            ( 24000*A6*A6           -10*B6*B6  )*H4*Z3 +
&            (            - 600*A6*B6          )*H5*Z2 +
&            (                        5*B6*B6  )*H6*Z1 ) / (20*H6)
```

```
C
C       COMPUTE THE LAMINATE ABAR, BBAR AND DBAR MATRICES
C
        CALL DMASB ( ABAR, Z1, C(1,1,K), ABAR, 6, 6 )
C
        BBAR(1,1) = BBAR(1,1) + C(1,1,K) * ( Z2 / 2.0 + PII(1) )
        BBAR(1,2) = BBAR(1,2) + C(1,2,K) * ( Z2 / 2.0 + PII(2) )
        BBAR(1,3) = BBAR(1,3) + C(1,3,K) * (            PII(3) )
        BBAR(1,6) = BBAR(1,6) + C(1,6,K) * ( Z2 / 2.0 + PII(6) )
        BBAR(2,1) = BBAR(2,1) + C(2,1,K) * ( Z2 / 2.0 + PII(1) )
        BBAR(2,2) = BBAR(2,2) + C(2,2,K) * ( Z2 / 2.0 + PII(2) )
        BBAR(2,3) = BBAR(2,3) + C(2,3,K) * (            PII(3) )
        BBAR(2,6) = BBAR(2,6) + C(2,6,K) * ( Z2 / 2.0 + PII(6) )
        BBAR(3,1) = BBAR(3,1) + C(3,1,K) * ( Z2 / 2.0 + PII(1) )
        BBAR(3,2) = BBAR(3,2) + C(3,2,K) * ( Z2 / 2.0 + PII(2) )
        BBAR(3,3) = BBAR(3,3) + C(3,3,K) * (            PII(3) )
        BBAR(3,6) = BBAR(3,6) + C(3,6,K) * ( Z2 / 2.0 + PII(6) )
        BBAR(6,1) = BBAR(6,1) + C(6,1,K) * ( Z2 / 2.0 + PII(1) )
        BBAR(6,2) = BBAR(6,2) + C(6,2,K) * ( Z2 / 2.0 + PII(2) )
        BBAR(6,3) = BBAR(6,3) + C(6,3,K) * (            PII(3) )
        BBAR(6,6) = BBAR(6,6) + C(6,6,K) * ( Z2 / 2.0 + PII(6) )
C
        DBAR(1,1) = DBAR(1,1) + C(3,3,K) * PIP(1,1) + C(1,1,K) * Z3/3.0
     &                        + C(1,3,K) * PIZ(1)   + C(1,3,K) * PIZ(1)
        DBAR(1,2) = DBAR(1,2) + C(3,3,K) * PIP(1,2) + C(1,2,K) * Z3/3.0
     &                        + C(2,3,K) * PIZ(1)   + C(1,3,K) * PIZ(2)
        DBAR(1,3) = DBAR(1,3) + C(3,3,K) * PIP(1,3) + C(1,3,K) * PIZ(3)
        DBAR(1,6) = DBAR(1,6) + C(3,3,K) * PIP(1,6) + C(1,6,K) * Z3/3.0
     &                        + C(3,6,K) * PIZ(1)   + C(1,3,K) * PIZ(6)
        DBAR(2,2) = DBAR(2,2) + C(3,3,K) * PIP(2,2) + C(2,2,K) * Z3/3.0
     &                        + C(2,3,K) * PIZ(2)   + C(2,3,K) * PIZ(2)
        DBAR(2,3) = DBAR(2,3) + C(3,3,K) * PIP(2,3) + C(2,3,K) * PIZ(3)
        DBAR(2,6) = DBAR(2,6) + C(3,3,K) * PIP(2,6) + C(2,6,K) * Z3/3.0
     &                        + C(3,6,K) * PIZ(2)   + C(2,3,K) * PIZ(6)
        DBAR(3,3) = DBAR(3,3) + C(3,3,K) * PIP(3,3)
        DBAR(3,6) = DBAR(3,6) + C(3,3,K) * PIP(3,6) + C(3,6,K) * PIZ(3)
        DBAR(6,6) = DBAR(6,6) + C(3,3,K) * PIP(6,6) + C(6,6,K) * Z3/3.0
     &                        + C(6,3,K) * PIZ(6)   + C(6,3,K) * PIZ(6)
C
C       INT[P2BAR]DZ
C
        P2B2I = (45 * Z5 - 150 * H2 * Z3 + 225 * H4 * Z1) / (144 * H4)
C
        GMAT(1,1) = GMAT(1,1) + C(4,4,K) * P2B2I
        GMAT(1,2) = GMAT(1,2) + C(4,5,K) * P2B2I
        GMAT(2,2) = GMAT(2,2) + C(5,5,K) * P2B2I
        GMAT(2,1) = GMAT(1,2)
C
```

```
      END DO
C
      ELSE IF ( NMAT .EQ. 2 ) THEN
C
C       COMPUTE LAMINATE CONSTITUTIVE MATRICES USING
C       STRESS-BASED VERSION OF HIGHER-ORDER THEORY
C       (VERSION I)
C
C        INITIALIZE VARIABLES
C
         AT1  = 0.0
         AT2  = 0.0
         AT12 = 0.0
         AT3  = 0.0
C
         A44  = 0.0
         A45  = 0.0
         A55  = 0.0
c
      DO 100 K=1, NLAY
C
         HK   = Z(K)
         HKP1 = Z(K+1)
C
         ZOI  = HKP1 - HK
C
         AT1 = -(ZOI*C(1,3,K)/C(3,3,K))/THK + AT1
         AT2 = -(ZOI*C(2,3,K)/C(3,3,K))/THK + AT2
         AT12= -(ZOI*C(3,6,K)/C(3,3,K))/THK + AT12
         AT3 =  (ZOI/C(3,3,K))/THK + AT3
C
         AAK(K,1) = -C(1,3,K)/C(3,3,K)
         AAK(K,2) = -C(2,3,K)/C(3,3,K)
         AAK(K,3) =      1.00/C(3,3,K)
         AAK(K,4) = -C(3,6,K)/C(3,3,K)
C
         CH(K,1) = C(1,1,K) + C(1,3,K)*AAK(K,1)
         CH(K,2) = C(1,2,K) + C(1,3,K)*AAK(K,2)
         CH(K,3) = C(1,6,K) + C(1,3,K)*AAK(K,4)
         CH(K,4) = C(1,3,K)*AAK(K,3)
         CH(K,5) = C(2,2,K) + C(2,3,K)*AAK(K,2)
         CH(K,6) = C(2,6,K) + C(3,6,K)*AAK(K,2)
         CH(K,7) = C(6,6,K) - C(3,6,K)**2/C(3,3,K)
C
         C44 = C(K,4,4)
         C45 = C(K,4,5)
         C55 = C(K,5,5)
c
```

```
      SS44 =  C55/(C44*C55-C45**2)
      SS55 =  C44/(C44*C55-C45**2)
      SS45 = -C45/(C44*C55-C45**2)
C
      AFACTOR = -(4*HKP1**3-3*THK**2*HKP1 -
     1              4*HK**3+3*THK**2*HK)/THK**3/2.0

      A44 = A44 + AFACTOR*SS44
      A45 = A45 + AFACTOR*SS45
      A55 = A55 + AFACTOR*SS55
c
  100 CONTINUE
c
      DENN= 1.0/(A44*A55-A45**2)
      B44 = (5.0/4.0)*A44*DENN
      B45 =-(5.0/4.0)*A45*DENN
      B55 = (5.0/4.0)*A55*DENN
C
      AR13 = AT1/AT3
      AR23 = AT2/AT3
      AR123= AT12/AT3
C
     DO 200 K = 1, NLAY
C
      HK   = Z(K)
      HKP1 = Z(K+1)
C
      ZOI = HKP1 - HK
C
      Z1I = (HKP1**2 - HK**2) / 2.0
      Z2I = (HKP1**3 - HK**3) / 3.0
      Z3I = -.02941176470588236*(14.0*HKP1**4-21.0*THK**2*HKP1**2
     1        -14.0*HK**4 + 21.0*THK**2*HK**2)/THK**2
C
      Z4I = (560.0*HKP1**7-1176.0*THK**2*HKP1**5
     1         +735.0*THK**4*HKP1**3-560.0*HK**7
     2         +1176.0*THK**2*HK**5-735.0*THK**4*HK**3)
     3         /THK**4/1445.0
C
      BK(K,1) = AAK(K,1) - AAK(K,3)*AR13
      BK(K,2) = AAK(K,2) - AAK(K,3)*AR23
      BK(K,3) = AAK(K,4) - AAK(K,3)*AR123
C
      D3(K)   = AAK(K,3)/AT3
C
      TB      = CH(K,4) + BK(K,1)
      ABC(K,1)= TB*AR13
      ABC(K,2)= TB*AR23
```

```
      ABC(K,4)= TB*AR123
      ABC(K,3)= TB/AT3
C
      CHH(K,1) = CH(K,1)-ABC(K,1)
      CHH(K,2) = CH(K,2)-ABC(K,2)
      CHH(K,3) = CH(K,3)-ABC(K,4)
C
C     COMPUTE THE LAMINATE ABAR, BBAR AND DBAR MATRICES
C
      ABAR(1,1) = ABAR(1,1) + CHH(K,1) * Z0I
      ABAR(1,2) = ABAR(1,2) + CHH(K,2) * Z0I
      ABAR(1,3) = ABAR(1,3) + ABC(K,3) * Z0I
      ABAR(1,6) = ABAR(1,6) + CHH(K,3) * Z0I
C
      ABAR(2,2) = ABAR(2,2)+(CH(K,5)-AR23*(BK(K,2)-AAK(K,2)))*Z0I
      ABAR(2,3) = ABAR(2,3) - AR23 * Z0I/(C(K,3,3)*AT3)
      ABAR(2,6) = ABAR(2,6) + (C(K,2,6)
     1            + (AR123*AR23-C(K,2,3)*C(K,3,6))/C(K,3,3))* Z0I
C
      BBAR(1,1) = BBAR(1,1) + CH(K,1) * Z1I - ABC(K,1)*Z3I
      BBAR(1,2) = BBAR(1,2) + CH(K,2) * Z1I - ABC(K,2)*Z3I
      BBAR(1,3) = BBAR(1,3) + 2.0*ABC(K,3)* Z3I
      BBAR(1,6) = BBAR(1,6) + CH(K,3) * Z1I - ABC(K,4)*Z3I
C
      BBAR(2,1) = BBAR(2,1) + CH(K,2) * Z1I + AR13*AR23*Z3I/C(K,3,3)
      BBAR(2,2) = BBAR(2,2) + CH(K,5) * Z1I + AR23*AR23*Z3I/C(K,3,3)
      BBAR(2,3) = BBAR(2,3) - 2.0*AR23* Z3I/(AT3*C(K,3,3))
      BBAR(2,6) = BBAR(2,6) + CH(K,6) * Z1I+ AR123*AR23*Z3I/C(K,3,3)
C
      FC1 = 1.0/(C(K,3,3)*AT3**2)
C
      ABAR(3,3) = ABAR(3,3) +  Z0I*FC1
      ABAR(3,6) = ABAR(3,6) -  Z0I*AT12*FC1
C
      BBAR(3,1) = BBAR(3,1) -  AT1*Z3I*FC1
      BBAR(3,2) = BBAR(3,2) -  AT2*Z3I*FC1
      BBAR(3,3) = BBAR(3,3) +  2.0*Z3I*FC1
      BBAR(3,6) = BBAR(3,6) - AT12*Z3I*FC1
C
      FC2 = - AR123/C(K,3,3)
C
      ABAR(6,6) = ABAR(6,6)  + (CH(K,7) - AR123*FC2)*Z0I
C
      BBAR(6,1) = BBAR(6,1) + CH(K,3) * Z1I - AR13*Z3I*FC2
      BBAR(6,2) = BBAR(6,2) + CH(K,6) * Z1I - AR23*Z3I*FC2
      BBAR(6,3) = BBAR(6,3) + 2.0* FC2* Z3I/AT3
      BBAR(6,6) = BBAR(6,6) + CH(K,7) * Z1I - AR123*Z3I*FC2
C
```

```
      FC3 = AR13/C(K,3,3)
      DBAR(1,1) = DBAR(1,1) + CH(K,1) * Z2I + AR13*FC3*Z4I
      DBAR(1,2) = DBAR(1,2) + CH(K,2) * Z2I + AR23*FC3*Z4I
      DBAR(1,3) = DBAR(1,3) - 2.0*FC3*Z4I/AT3
      DBAR(1,6) = DBAR(1,6) + CH(K,3) * Z2I + AR123*FC3*Z4I
C
      FC4 = AR23/C(K,3,3)
C
      DBAR(2,2) = DBAR(2,2) + CH(K,5) * Z2I + AR23*FC4*Z4I
      DBAR(2,3) = DBAR(2,3) - 2.0*FC4*Z4I/AT3
      DBAR(2,6) = DBAR(2,6) + CH(K,6) * Z2I + AR123*FC4*Z4I
C
      DBAR(3,3) = DBAR(3,3) + 4.0*D3(K)*Z4I/AT3
      DBAR(3,6) = DBAR(3,6) - 2.0*AR123*Z4I/(AT3*C(K,3,3))
C
      DBAR(6,6) = DBAR(6,6) + Z2I*CH(K,7) + AR123**2*Z4I*AAK(K,3)
C
      ABAR(2,1) = ABAR(1,2)
      ABAR(3,1) = ABAR(1,3)
      ABAR(3,2) = ABAR(2,3)
      ABAR(6,1) = ABAR(1,6)
      ABAR(6,2) = ABAR(2,6)
      ABAR(6,3) = ABAR(3,6)
C
      FACTOR = (48*HKP1**5-40*tHK**2*HKP1**3+15*tHK**4*HKP1-
     1          48*HK**5+40*tHK**2*HK**3-15*tHK**4*HK)/tHK**4/15.0
C
      C44 = C(K,4,4)
      C45 = C(K,4,5)
      C55 = C(K,5,5)
C
      GCAP11 = (25*A45**2*C55+50*A44*A45*C45+25*A44**2*C44)/(C44*
     1          (16*A44**2*A55**2-32*A44*A45**2*A55+16*A45**4)*C55+
     2          C45**2*(-16*A44**2*A55**2+32*A44*A45**2*A55-
     3          16*A45**4))
C
      GCAP12 = -(25*A45*A55*C55+C45*(25*A44*A55+25*A45**2)+
     1            25*A44*A45*C44)/(C44*(16*A44**2*A55**2-32*A44*
     2            A45**2*A55+16*A45**4)*C55+C45**2*(-16*A44**2*
     3            A55**2+32*A44*A45**2*A55-16*A45**4))
C
     GCAP22 = (25*A55**2*C55+50*A45*A55*C45+25*A45**2*C44)/(C44*
     1          (16*A44**2*A55**2-32*A44*A45**2*A55+16*A45**4)*C55+
     2          C45**2*(-16*A44**2*A55**2+32*A44*A45**2*A55-16*A45**4))
C
      GMAT(2,2) = GMAT(2,2) + GCAP11*FACTOR
      GMAT(1,2) = GMAT(1,2) + GCAP12*FACTOR
      GMAT(1,1) = GMAT(1,1) + GCAP22*FACTOR
```

```
      GMAT(2,1) = GMAT(1,2)
C
  200 CONTINUE
C
C     SCALING TO THE ORIGINAL LEGENDRE FORMULATION
C     WITH THE USE OF (3/2) FACTOR AND HALF THICKNESS
C
      SCALE = (3.0/2.0)/(tHk/2.0)
C
      DO 99 I = 1, 3
         BBAR(I,3) = BBAR(I,3)*SCALE
         DBAR(I,3) = DBAR(I,3)*SCALE
 99   CONTINUE
C
      DBAR(3,6) = DBAR(3,6)*SCALE
      BBAR(6,3) = BBAR(6,3)*SCALE
      DBAR(3,3) = DBAR(3,3)*SCALE
C
      DBAR(2,1) = DBAR(1,2)
      DBAR(3,1) = DBAR(1,3)
      DBAR(3,2) = DBAR(2,3)
      DBAR(6,1) = DBAR(1,6)
      DBAR(6,2) = DBAR(2,6)
      DBAR(6,3) = DBAR(3,6)
C
    ELSE IF ( NMAT .EQ. 3 ) THEN
C
C     COMPUTE LAMINATE CONSTITUTIVE MATRICES USING
C     STRESS-BASED VERSION OF HIGHER-ORDER THEORY
C     (VERSION II)
C
C     INITIALIZE VARIABLES
C
      S11  = 0.0
      S12  = 0.0
      S22  = 0.0
      T11  = 0.0
      T12  = 0.0
      T16  = 0.0
      T13  = 0.0
      TH11 = 0.0
      TH12 = 0.0
      TH16 = 0.0
      TH13 = 0.0
      T21  = 0.0
      T22  = 0.0
      T26  = 0.0
      T23  = 0.0
```

```
      TH21 = 0.0
      TH22 = 0.0
      TH26 = 0.0
      TH23 = 0.0
C
      A11 = 0.0
      A12 = 0.0
      A21 = 0.0
      A22 = 0.0
      B11 = 0.0
      B22 = 0.0
      A44 = 0.0
      A45 = 0.0
      A55 = 0.0
C
      HL  = HALF
C
      DO K = 1, NLAY
C
         HK   =  Z(K)
         HKP1 =  Z(K+1)
         ZOI  =  HKP1    - HK
         Z1I  = (HKP1**2 - HK**2) / 2.0
C
         FINT  = -(HKP1-HK)*(HKP1+HK)*(-3*THK**2+2*HKP1**2+
     1             2*HK**2)/THK**3/3.0
         FINT2 = (320*HKP1**7-672*THK**2*HKP1**5+420*THK**4*HKP1**3-
     1            320*HK**7+672*THK**2*HK**5-420*THK**4*HK**3)/
     2            THK**6/315.0
         ZFINT = -(8*HKP1**5-10*THK**2*HKP1**3-8*HK**5+
     1             10*THK**2*HK**3)/THK**3/15.0
         PINT  = -(4*HKP1**3-3*THK**2*HKP1-4*HK**3+3*THK**2*HK)/
     1             THK**2/3.0
         PINT2 = (48*HKP1**5-40*THK**2*HKP1**3+15*THK**4*HKP1-
     1            48*HK**5+40*THK**2*HK**3-15*THK**4*HK)/THK**4/15.0
C
         S33(K) =       1.0  / C(3,3,K)
         RA13(K) =  C(1,3,K) / C(3,3,K)
         RA23(K) =  C(2,3,K) / C(3,3,K)
         RA63(K) =  C(6,3,K) / C(3,3,K)
C
         C1(1,1,K) = C(1,1,K) - RA13(K) * C(1,3,K)
         C1(1,2,K) = C(1,2,K) - RA13(K) * C(2,3,K)
         C1(1,6,K) = C(1,6,K) - RA13(K) * C(6,3,K)
         C1(2,2,K) = C(2,2,K) - RA23(K) * C(2,3,K)
         C1(2,6,K) = C(2,6,K) - RA23(K) * C(6,3,K)
         C1(6,6,K) = C(6,6,K) - RA63(K) * C(6,3,K)
C
```

```
      S11 =  S11 +   ZOI  * S33(K)**2
      S12 =  S12 +   FINT * S33(K)**2
      S22 =  S22 +  FINT2 * S33(K)**2
C
      T11 =  T11 +   ZOI  * C(1,3,K)* S33(K)**2
      T12 =  T12 +   ZOI  * C(2,3,K)* S33(K)**2
      T16 =  T16 +   ZOI  * C(3,6,K)* S33(K)**2
      T13 =  T13 +   ZOI  * S33(K)
C
      T21 =  T21 +  FINT  * C(1,3,K)* S33(K)**2
      T22 =  T22 +  FINT  * C(2,3,K)* S33(K)**2
      T26 =  T26 +  FINT  * C(3,6,K)* S33(K)**2
      T23 =  T23 +  FINT  * S33(K)
C
      TH11 =  TH11 +  Z1I * C(1,3,K)* S33(K)**2
      TH12 =  TH12 +  Z1I * C(2,3,K)* S33(K)**2
      TH16 =  TH16 +  Z1I * C(3,6,K)* S33(K)**2
      TH13 =  TH13 +  2.0 * Z1I * S33(K)
C
      TH21 =  TH21 + ZFINT * C(1,3,K)* S33(K)**2
      TH22 =  TH22 + ZFINT * C(2,3,K)* S33(K)**2
      TH26 =  TH26 + ZFINT * C(3,6,K)* S33(K)**2
      TH23 =  TH23 + 2.0 * ZFINT * S33(K)
C
      C44 =  C(4,4,K)
      C45 =  C(4,5,K)
      C55 =  C(5,5,K)
C
      S44 =  C55 / (C44*C55-C45**2)
      S55 =  C44 / (C44*C55-C45**2)
      S45 = -C45 / (C44*C55-C45**2)
C
      A11 =  A11 + PINT2 * S55**2
      A12 =  A12 + PINT2 * S45*S55
      A21 =  A21 + PINT2 * S45*S44
      A22 =  A22 + PINT2 * S44**2
      B11 =  B11 + PINT  * S55
      B22 =  B22 + PINT  * S44
C
      AFACTOR = -(4*HKP1**3-3*THK**2*HKP1-4*HK**3+3*THK**2*HK)/
     1                THK**3/2.0
C
      A44 = A44 + AfACTOR * S44
      A45 = A45 + AfACTOR * S45
      A55 = A55 + AfACTOR * S55
C
    END DO
C
```

```
      B44 = (5.0/4.0) * A44 / (A44*A55-A45**2)
      B45 =-(5.0/4.0) * A45 / (A44*A55-A45**2)
      B55 = (5.0/4.0) * A55 / (A44*A55-A45**2)
C
C     AVERAGE COEFFICIENTS
C
      R11 =  S22 / (S11*S22 - S12**2)
      R22 =  S11 / (S11*S22 - S12**2)
      R12 = -S12 / (S11*S22 - S12**2)
C
      Q11 = R11 * T11 + R12 * T21
      Q21 = R12 * T11 + R22 * T21
      Q12 = R11 * T12 + R12 * T22
      Q22 = R12 * T12 + R22 * T22
      Q13 = R11 * T13 + R12 * T23
      Q23 = R12 * T13 + R22 * T23
      Q16 = R11 * T16 + R12 * T26
      Q26 = R12 * T16 + R22 * T26
C
      QH11 = R11 * TH11 + R12 * TH21
      QH21 = R12 * TH11 + R22 * TH21
      QH12 = R11 * TH12 + R12 * TH22
      QH22 = R12 * TH12 + R22 * TH22
      QH16 = R11 * TH16 + R12 * TH26
      QH26 = R12 * TH16 + R22 * TH26
      QH13 = R11 * TH13 + R12 * TH23
      QH23 = R12 * TH13 + R22 * TH23
C
      D11 =  A22 * B11 / (A11*A22 - A12*A21)
      D12 = -A12 * B22 / (A11*A22 - A12*A21)
      D21 = -A21 * B11 / (A11*A22 - A12*A21)
      D22 =  A11 * B22 / (A11*A22 - A12*A21)
C
      SCALE = (3.0 / 2.0) / HL
C
      DO K=1, NLAY
C
         HK   = Z(K)
         HKP1 = Z(K+1)
         Z0I  =  HKP1    - HK
         Z1I  = (HKP1**2 - HK**2) / 2.0
         Z2I  = (HKP1**3 - HK**3) / 3.0
C
         PINT2 = (48*HKP1**5-40*THK**2*HKP1**3+15*THK**4*HKP1-
     1             48*HK**5+40*THK**2*HK**3-15*THK**4*HK)/
     2             THK**4/15.0
C
         FINT = -(HKP1-HK)*(HKP1+HK)*(-3*THK**2+2*HKP1**2+
```

```
     1            2*HK**2)/THK**3/3.0
C
      FINT2 = (320*HKP1**7-672*THK**2*HKP1**5+420*THK**4*HKP1**3-
     1         320*HK**7+672*THK**2*HK**5-420*THK**4*HK**3)/
     2         THK**6/315.0
C
      C44 = C(4,4,K)
      C45 = C(4,5,K)
      C55 = C(5,5,K)
C
      S44 =  C55/(C44*C55-C45**2)
      S55 =  C44/(C44*C55-C45**2)
      S45 = -C45/(C44*C55-C45**2)
C
C     COMPUTE THE LAMINATE ABAR, BBAR AND DBAR MATRICES
C
C     EXPLICIT INTEGRATION OF P TERMS ( OBTAINED FROM MACSYMA ) :
C
      P1 = (320*Q21**2*HKP1**7-672*Q21**2*THK**2*HKP1**5-
     1      420*Q11*Q21*THK**3*HKP1**4+420*Q21**2*THK**4*
     2      HKP1**3+630*Q11*Q21*THK**5*HKP1**2+315*Q11**2*
     3      THK**6*HKP1+Q21**2*(-320*HK**7+672*THK**2*HK**5-
     4      420*THK**4*HK**3)+Q11*Q21*(420*THK**3*HK**4-
     5      630*THK**5*HK**2)-315*Q11**2*THK**6*HK)/
     6      THK**6/315.0
C
      P2 = (HKP1**2*(315*Q11*Q22*THK**5+315*Q12*Q21*THK**5)+
     1      HKP1**4*(-210*Q11*Q22*THK**3-210*Q12*Q21*THK**3)+
     2      320*Q21*Q22*HKP1**7-672*Q21*Q22*THK**2*HKP1**5+
     3      420*Q21*Q22*THK**4*HKP1**3+315*Q11*Q12*THK**6*HKP1+
     4      Q22*(Q21*(-320*HK**7+672*THK**2*HK**5-420*THK**4*
     5      HK**3)+210*Q11*THK**3*HK**4-315*Q11*THK**5*HK**2)+
     6      Q21*(210*Q12*THK**3*HK**4-315*Q12*THK**5*HK**2)-
     7      315*Q11*Q12*THK**6*HK)/THK**6/315.0
C
      P3 = (HKP1**2*(315*Q11*Q23*THK**5+315*Q13*Q21*THK**5)+
     1      HKP1**4*(-210*Q11*Q23*THK**3-210*Q13*Q21*THK**3)+
     2      320*Q21*Q23*HKP1**7-672*Q21*Q23*THK**2*HKP1**5+
     3      420*Q21*Q23*THK**4*HKP1**3+315*Q11*Q13*THK**6*HKP1+
     4      Q23*(Q21*(-320*HK**7+672*THK**2*HK**5-420*THK**4*
     5      HK**3)+210*Q11*THK**3*HK**4-315*Q11*THK**5*HK**2)+
     6      Q21*(210*Q13*THK**3*HK**4-315*Q13*THK**5*HK**2)-
     7      315*Q11*Q13*THK**6*HK)/THK**6/315.0
C
      P4 = (HKP1**2*(315*Q11*Q26*THK**5+315*Q16*Q21*THK**5)+
     1      HKP1**4*(-210*Q11*Q26*THK**3-210*Q16*Q21*THK**3)+
     2      320*Q21*Q26*HKP1**7-672*Q21*Q26*THK**2*HKP1**5+
     3      420*Q21*Q26*THK**4*HKP1**3+315*Q11*Q16*THK**6*HKP1+
```

```
     4          Q26*(Q21*(-320*HK**7+672*THK**2*HK**5-420*THK**4*
     5          HK**3)+210*Q11*THK**3*HK**4-315*Q11*THK**5*HK**2)+
     6          Q21*(210*Q16*THK**3*HK**4-315*Q16*THK**5*HK**2)-
     7          315*Q11*Q16*THK**6*HK)/THK**6/315.0
C
          P5 = (320*Q22**2*HKP1**7-672*Q22**2*THK**2*HKP1**5-
     1          420*Q12*Q22*THK**3*HKP1**4+420*Q22**2*THK**4*
     2          HKP1**3+630*Q12*Q22*THK**5*HKP1**2+315*Q12**2*
     3          THK**6*HKP1+Q22**2*(-320*HK**7+672*THK**2*HK**5-
     4          420*THK**4*HK**3)+Q12*Q22*(420*THK**3*HK**4-
     5          630*THK**5*HK**2)-315*Q12**2*THK**6*HK)/
     6          THK**6/315.0
C
          P6 = (HKP1**2*(315*Q12*Q23*THK**5+315*Q13*Q22*THK**5)+
     1          HKP1**4*(-210*Q12*Q23*THK**3-210*Q13*Q22*THK**3)+
     2          320*Q22*Q23*HKP1**7-672*Q22*Q23*THK**2*HKP1**5+
     3          420*Q22*Q23*THK**4*HKP1**3+315*Q12*Q13*THK**6*HKP1+
     4          Q23*(Q22*(-320*HK**7+672*THK**2*HK**5-420*THK**4*
     5          HK**3)+210*Q12*THK**3*HK**4-315*Q12*THK**5*HK**2)+
     6          Q22*(210*Q13*THK**3*HK**4-315*Q13*THK**5*HK**2)-
     7          315*Q12*Q13*THK**6*HK)/THK**6/315.0
C
          P7 = (HKP1**2*(315*Q12*Q26*THK**5+315*Q16*Q22*THK**5)+
     1          HKP1**4*(-210*Q12*Q26*THK**3-210*Q16*Q22*THK**3)+
     2          320*Q22*Q26*HKP1**7-672*Q22*Q26*THK**2*HKP1**5+
     3          420*Q22*Q26*THK**4*HKP1**3+315*Q12*Q16*THK**6*HKP1+
     4          Q26*(Q22*(-320*HK**7+672*THK**2*HK**5-420*THK**4*
     5          HK**3)+210*Q12*THK**3*HK**4-315*Q12*THK**5*HK**2)+
     6          Q22*(210*Q16*THK**3*HK**4-315*Q16*THK**5*HK**2)-
     7          315*Q12*Q16*THK**6*HK)/THK**6/315.0
C
          P8 = (320*Q23**2*HKP1**7-672*Q23**2*THK**2*HKP1**5-
     1          420*Q13*Q23*THK**3*HKP1**4+420*Q23**2*THK**4*
     2          HKP1**3+630*Q13*Q23*THK**5*HKP1**2+315*Q13**2*
     3          THK**6*HKP1+Q23**2*(-320*HK**7+672*THK**2*HK**5-
     4          420*THK**4*HK**3)+Q13*Q23*(420*THK**3*HK**4-
     5          630*THK**5*HK**2)-315*Q13**2*THK**6*HK)/
     6          THK**6/315.0
C
          P9 = (HKP1**2*(315*Q13*Q26*THK**5+315*Q16*Q23*THK**5)+
     1          HKP1**4*(-210*Q13*Q26*THK**3-210*Q16*Q23*THK**3)+
     2          320*Q23*Q26*HKP1**7-672*Q23*Q26*THK**2*HKP1**5+
     3          420*Q23*Q26*THK**4*HKP1**3+315*Q13*Q16*THK**6*HKP1+
     4          Q26*(Q23*(-320*HK**7+672*THK**2*HK**5-420*THK**4*
     5          HK**3)+210*Q13*THK**3*HK**4-315*Q13*THK**5*HK**2)+
     6          Q23*(210*Q16*THK**3*HK**4-315*Q16*THK**5*HK**2)-
     7          315*Q13*Q16*THK**6*HK)/THK**6/315.0
C
```

```
      P10 = (320*Q26**2*HKP1**7-672*Q26**2*THK**2*HKP1**5-
     1       420*Q16*Q26*THK**3*HKP1**4+420*Q26**2*THK**4*
     2       HKP1**3+630*Q16*Q26*THK**5*HKP1**2+315*Q16**2*
     3       THK**6*HKP1+Q26**2*(-320*HK**7+672*THK**2*HK**5-
     4       420*THK**4*HK**3)+Q16*Q26*(420*THK**3*HK**4-
     5       630*THK**5*HK**2)-315*Q16**2*THK**6*HK)/THK**6/315.0
C
      PH1 = (HKP1**2*(315*Q11*QH23*THK**5+315*Q21*QH13*THK**5)+
     1       HKP1**4*(-210*Q11*QH23*THK**3-210*Q21*QH13*THK**3)+
     2       320*Q21*QH23*HKP1**7-672*Q21*QH23*THK**2*HKP1**5+
     3       420*Q21*QH23*THK**4*HKP1**3+315*Q11*QH13*THK**6*
     4       HKP1+QH23*(Q21*(-320*HK**7+672*THK**2*HK**5-
     5       420*THK**4*HK**3)+210*Q11*THK**3*HK**4-315*Q11*
     6       THK**5*HK**2)+Q21*(210*QH13*THK**3*HK**4-
     7       315*QH13*THK**5*HK**2)-315*Q11*QH13*THK**6*HK)/
     8       THK**6/315.0
C
      PH2 = (HKP1**2*(315*Q12*QH23*THK**5+315*QH13*Q22*THK**5)+
     1       HKP1**4*(-210*Q12*QH23*THK**3-210*QH13*Q22*THK**3)+
     2       320*Q22*QH23*HKP1**7-672*Q22*QH23*THK**2*HKP1**5+
     3       420*Q22*QH23*THK**4*HKP1**3+315*Q12*QH13*THK**6*
     4       HKP1+QH23*(Q22*(-320*HK**7+672*THK**2*HK**5-
     5       420*THK**4*HK**3)+210*Q12*THK**3*HK**4-315*Q12*
     6       THK**5*HK**2)+Q22*(210*QH13*THK**3*HK**4-
     7       315*QH13*THK**5*HK**2)-315*Q12*QH13*THK**6*HK)/
     8       THK**6/315.0
C
      PH3 = (HKP1**2*(315*QH11*Q23*THK**5+315*Q13*QH21*THK**5)+
     1       HKP1**4*(-210*QH11*Q23*THK**3-210*Q13*QH21*THK**3)+
     2       320*Q23*QH21*HKP1**7-672*Q23*QH21*THK**2*HKP1**5+
     3       420*Q23*QH21*THK**4*HKP1**3+315*Q13*QH11*THK**6*
     4       HKP1+QH21*(Q23*(-320*HK**7+672*THK**2*HK**5-
     5       420*THK**4*HK**3)+210*Q13*THK**3*HK**4-315*Q13*
     6       THK**5*HK**2)+Q23*(210*QH11*THK**3*HK**4-
     7       315*QH11*THK**5*HK**2)-315*Q13*QH11*THK**6*HK)/
     8       THK**6/315.0
C
      PH4 = (HKP1**2*(315*QH12*Q23*THK**5+315*Q13*QH22*THK**5)+
     1       HKP1**4*(-210*QH12*Q23*THK**3-210*Q13*QH22*THK**3)+
     2       320*Q23*QH22*HKP1**7-672*Q23*QH22*THK**2*HKP1**5+
     3       420*Q23*QH22*THK**4*HKP1**3+315*Q13*QH12*THK**6*
     4       HKP1+QH22*(Q23*(-320*HK**7+672*THK**2*HK**5-
     5       420*THK**4*HK**3)+210*Q13*THK**3*HK**4-315*Q13*
     6       THK**5*HK**2)+Q23*(210*QH12*THK**3*HK**4-
     7       315*QH12*THK**5*HK**2)-315*Q13*QH12*THK**6*HK)/
     8       THK**6/315.0
C
      PH5 = (HKP1**2*(315*QH13*Q23*THK**5+315*Q13*QH23*THK**5)+
```

```
     1          HKP1**4*(-210*QH13*Q23*THK**3-210*Q13*QH23*THK**3)+
     2          320*Q23*QH23*HKP1**7-672*Q23*QH23*THK**2*HKP1**5+
     3          420*Q23*QH23*THK**4*HKP1**3+315*Q13*QH13*THK**6*.
     4          HKP1+QH23*(Q23*(-320*HK**7+672*THK**2*HK**5-
     5          420*THK**4*HK**3)+210*Q13*THK**3*HK**4-315*Q13*
     6          THK**5*HK**2)+Q23*(210*QH13*THK**3*HK**4-
     7          315*QH13*THK**5*HK**2)-315*Q13*QH13*THK**6*HK)/
     8          THK**6/315.0
C
          PH6 = (HKP1**2*(315*QH16*Q23*THK**5+315*Q13*QH26*THK**5)+
     1          HKP1**4*(-210*QH16*Q23*THK**3-210*Q13*QH26*THK**3)+
     2          320*Q23*QH26*HKP1**7-672*Q23*QH26*THK**2*HKP1**5+
     3          420*Q23*QH26*THK**4*HKP1**3+315*Q13*QH16*THK**6*
     4          HKP1+QH26*(Q23*(-320*HK**7+672*THK**2*HK**5-
     5          420*THK**4*HK**3)+210*Q13*THK**3*HK**4-315*Q13*
     6          THK**5*HK**2)+Q23*(210*QH16*THK**3*HK**4-
     7          315*QH16*THK**5*HK**2)-315*Q13*QH16*THK**6*HK)/
     8          THK**6/315.0
C
          PH7 = (HKP1**2*(315*QH13*Q26*THK**5+315*Q16*QH23*THK**5)+
     1          HKP1**4*(-210*QH13*Q26*THK**3-210*Q16*QH23*THK**3)+
     2          320*Q26*QH23*HKP1**7-672*Q26*QH23*THK**2*HKP1**5+
     3          420*Q26*QH23*THK**4*HKP1**3+315*Q16*QH13*THK**6*
     4          HKP1+QH23*(Q26*(-320*HK**7+672*THK**2*HK**5-
     5          420*THK**4*HK**3)+210*Q16*THK**3*HK**4-315*Q16*
     6          THK**5*HK**2)+Q26*(210*QH13*THK**3*HK**4-
     7          315*QH13*THK**5*HK**2)-315*Q16*QH13*THK**6*HK)/
     8          THK**6/315.0
C
          ABAR(1,1) = ABAR(1,1) + ZOI * C1(1,1,K) + P1 * S33(K)
          ABAR(1,2) = ABAR(1,2) + ZOI * C1(1,2,K) + P2 * S33(K)
          ABAR(1,3) = ABAR(1,3)                   + P3 * S33(K)
          ABAR(1,6) = ABAR(1,6) + ZOI * C1(1,6,K) + P4 * S33(K)
          ABAR(2,2) = ABAR(2,2) + ZOI * C1(2,2,K) + P5 * S33(K)
          ABAR(2,3) = ABAR(2,3)                   + P6 * S33(K)
          ABAR(2,6) = ABAR(2,6) + ZOI * C1(2,6,K) + P7 * S33(K)
          ABAR(3,3) = ABAR(3,3)                   + P8 * S33(K)
          ABAR(3,6) = ABAR(3,6)                   + P9 * S33(K)
          ABAR(6,6) = ABAR(6,6) + ZOI * C1(6,6,K) + P10* S33(K)
C
          PG1 = ZOI * Q11 * QH11 + FINT  * (QH11 * Q21 + Q11 * QH21)
     &                           + FINT2 *  QH21 * Q21
          PG2 = ZOI * Q11 * QH12 + FINT  * (QH12 * Q21 + Q11 * QH22)
     &                           + FINT2 *  QH22 * Q21
          PG3 = ZOI * Q11 * QH13 + FINT  * (QH13 * Q21 + Q11 * QH23)
     &                           + FINT2 *  QH23 * Q21
          PG4 = ZOI * Q11 * QH16 + FINT  * (QH16 * Q21 + Q11 * QH26)
     &                           + FINT2 *  QH26 * Q21
```

```
      PG5 = ZOI * Q12 * QH11 + FINT  * (QH11 * Q22 + Q12 * QH21)
     &                       + FINT2 *  QH21 * Q22
      PG6 = ZOI * Q12 * QH12 + FINT  * (QH12 * Q22 + Q12 * QH22)
     &                       + FINT2 *  QH22 * Q22
      PG7 = ZOI * Q12 * QH13 + FINT  * (QH13 * Q22 + Q12 * QH23)
     &                       + FINT2 *  QH23 * Q22
      PG8 = ZOI * Q12 * QH16 + FINT  * (QH16 * Q22 + Q12 * QH26)
     &                       + FINT2 *  QH26 * Q22
      PG9 = ZOI * Q13 * QH11 + FINT  * (QH11 * Q23 + Q13 * QH21)
     &                       + FINT2 *  QH21 * Q23
      PG10= ZOI * Q13 * QH12 + FINT  * (QH12 * Q23 + Q13 * QH22)
     &                       + FINT2 *  QH22 * Q23
      PG11= ZOI * Q13 * QH13 + FINT  * (QH13 * Q23 + Q13 * QH23)
     &                       + FINT2 *  QH23 * Q23
      PG12= ZOI * Q13 * QH16 + FINT  * (QH16 * Q23 + Q13 * QH26)
     &                       + FINT2 *  QH26 * Q23
      PG13= ZOI * Q16 * QH11 + FINT  * (QH11 * Q26 + Q16 * QH21)
     &                       + FINT2 *  QH21 * Q26
      PG14= ZOI * Q16 * QH12 + FINT  * (QH12 * Q26 + Q16 * QH22)
     &                       + FINT2 *  QH22 * Q26
      PG15= ZOI * Q16 * QH13 + FINT  * (QH13 * Q26 + Q16 * QH23)
     &                       + FINT2 *  QH23 * Q26
      PG16= ZOI * Q16 * QH16 + FINT  * (QH16 * Q26 + Q16 * QH26)
     &                       + FINT2 *  QH26 * Q26
C
      BBAR(1,3) = BBAR(1,3) + SCALE * PG3  * S33(K)
      BBAR(2,3) = BBAR(2,3) + SCALE * PG7  * S33(K)
      BBAR(3,1) = BBAR(3,1) +         PG9  * S33(K)
      BBAR(3,2) = BBAR(3,2) +         PG10 * S33(K)
      BBAR(3,3) = BBAR(3,3) + SCALE * PG11 * S33(K)
      BBAR(3,6) = BBAR(3,6) +         PG12 * S33(K)
      BBAR(6,3) = BBAR(6,3) + SCALE * PG15 * S33(K)
C
      BBAR(1,1) = BBAR(1,1) + Z1I * C1(1,1,K) + PG1 * S33(K)
      BBAR(1,2) = BBAR(1,2) + Z1I * C1(1,2,K) + PG2 * S33(K)
      BBAR(1,6) = BBAR(1,6) + Z1I * C1(1,6,K) + PG4 * S33(K)
      BBAR(2,1) = BBAR(2,1) + Z1I * C1(1,2,K) + PG5 * S33(K)
      BBAR(2,2) = BBAR(2,2) + Z1I * C1(2,2,K) + PG6 * S33(K)
      BBAR(2,6) = BBAR(2,6) + Z1I * C1(2,6,K) + PG8 * S33(K)
      BBAR(6,1) = BBAR(6,1) + Z1I * C1(1,6,K) + PG13* S33(K)
      BBAR(6,2) = BBAR(6,2) + Z1I * C1(2,6,K) + PG14* S33(K)
      BBAR(6,6) = BBAR(6,6) + Z1I * C1(6,6,K) + PG16* S33(K)
C
      PF1 = ZOI * QH11**2 + 2.0 * FINT  * QH11 * QH21 +
     &                          FINT2 * QH21**2
C
      PF2 = ZOI * QH11 * QH12 + FINT  * (QH11 * QH22 + QH21 * QH12)
     &                        + FINT2 *  QH21 * QH22
```

```
C
      PF3 = ZOI * QH11 * QH13 + FINT  * (QH21 * QH13 + QH11 * QH23)
     &                        + FINT2 *  QH21 * QH23
C
      PF4 = ZOI * QH11 * QH16 + FINT  * (QH11 * QH26 + QH21 * QH16)
     &                        + FINT2 *  QH21 * QH26
C
      PF5 = ZOI * QH12**2 + 2.0 * FINT  * QH12 * QH22 +
     &                            FINT2 * QH22**2
C
      PF6 = ZOI * QH12 * QH13 + FINT  * (QH12 * QH23 + QH22 * QH13)
     &                        + FINT2 *  QH22 * QH23
C
      PF7 = ZOI * QH12 * QH16 + FINT  * (QH12 * QH26 + QH22 * QH16)
     &                        + FINT2 *  QH22 * QH26
C
      PF8 = ZOI * QH13**2 + 2.0 * FINT  * QH13 * QH23 +
     &                            FINT2 * QH23**2
C
      PF9 = ZOI * QH13 * QH16 + FINT  * (QH13 * QH26 + QH23 * QH16)
     &                        + FINT2 *  QH23 * QH26
C
      PF10= ZOI * QH16**2 + 2.0 * FINT  * QH16 * QH26 +
     &                            FINT2 * QH26**2
C
      DBAR(1,1) = DBAR(1,1) + Z2I * C1(1,1,K) + PF1 * S33(K)
      DBAR(1,2) = DBAR(1,2) + Z2I * C1(1,2,K) + PF2 * S33(K)
      DBAR(1,3) = DBAR(1,3) +           SCALE * PF3 * S33(K)
      DBAR(1,6) = DBAR(1,6) + Z2I * C1(1,6,K) + PF4 * S33(K)
      DBAR(2,2) = DBAR(2,2) + Z2I * C1(2,2,K) + PF5 * S33(K)
      DBAR(2,3) = DBAR(2,3) +           SCALE * PF6 * S33(K)
      DBAR(2,6) = DBAR(2,6) + Z2I * C1(2,6,K) + PF7 * S33(K)
      DBAR(3,3) = DBAR(3,3) +        SCALE**2 * PF8 * S33(K)
      DBAR(3,6) = DBAR(3,6) +           SCALE * PF9 * S33(K)
      DBAR(6,6) = DBAR(6,6) + Z2I * C1(6,6,K) + PF10* S33(K)
C
C     INT[P2BAR]DZ
C
      P2B2I = (AK**4)*(9.*(HKP1**5-HK**5)-30.*HL**2*(HKP1**3-
     1          HK**3)+45.*HL**4*(HKP1-HK))/(20.*HL**4)
C
      GMAT(1,1) = GMAT(1,1) + C(4,4,K) * P2B2I
      GMAT(1,2) = GMAT(1,2) + C(4,5,K) * P2B2I
      GMAT(2,2) = GMAT(2,2) + C(5,5,K) * P2B2I
      GMAT(2,1) = GMAT(1,2)
C
    END DO
C
```

```
        ABAR(2,1) = ABAR(1,2)
        ABAR(3,1) = ABAR(1,3)
        ABAR(3,2) = ABAR(2,3)
        ABAR(6,1) = ABAR(1,6)
        ABAR(6,2) = ABAR(2,6)
        ABAR(6,3) = ABAR(3,6)
        DBAR(2,1) = DBAR(1,2)
        DBAR(3,1) = DBAR(1,3)
        DBAR(3,2) = DBAR(2,3)
        DBAR(6,1) = DBAR(1,6)
        DBAR(6,2) = DBAR(2,6)
        DBAR(6,3) = DBAR(3,6)
C
      END IF
C
C     APPLY CORRECTION FACTORS TO LAMINATE MATRICES
C
      ACSHR = PI**2/10.0
      ACTHK = PI/SQRT(12.0)
      ACHMD = PI*SQRT(17.0/252.0)
C
      DO I = 1, 6
         ABAR(I,3) = ABAR(I,3) * ACTHK
         ABAR(3,I) = ABAR(3,I) * ACTHK
         BBAR(3,I) = BBAR(3,I) * ACTHK
         BBAR(I,3) = BBAR(I,3) * ACHMD
         DBAR(3,I) = DBAR(3,I) * ACHMD
         DBAR(I,3) = DBAR(I,3) * ACHMD
      END DO
C
      DO I = 1, 2
         DO J = 1, 2
            GMAT(I,J) = GMAT(I,J) * ACSHR
         END DO
      END DO
C
C     REORDER STORAGE OF A, B AND D MATRICES FOR CALLING PROGRAM
C
      DO I = 1, 4
         DO J = I, 4
            N = I
            M = J
            IF ( I .EQ. 4 ) N = 6
            IF ( J .EQ. 4 ) M = 6
            AMAT(I,J) = ABAR(N,M)
            AMAT(J,I) = ABAR(N,M)
            BMAT(I,J) = BBAR(N,M)
            BMAT(J,I) = BBAR(M,N)
```

```
            DMAT(I,J) = DBAR(N,M)
            DMAT(J,I) = DBAR(N,M)
         END DO
      END DO
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
      SUBROUTINE UINTPT ( X1, X2, X3, WEIGHT, KORD, I )               *
C                                                                     *
      IMPLICIT REAL*8 (A-H,O-Z)                                       *
C                                                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IF      (KORD .EQ. 1) THEN
            X1     = 1.0D0 / 3.0D0
            X2     = 1.0D0 / 3.0D0
            WEIGHT = 1.0
      ELSE IF (KORD .EQ. 3) THEN
         IF      (I .EQ. 1) THEN
            X1     = 1.0D0 / 2.0D0
            X2     = 1.0D0 / 2.0D0
            WEIGHT = 1.0D0 / 3.0D0
         ELSE IF (I .EQ. 2) THEN
            X1     = 0.0D0
            X2     = 1.0D0 / 2.0D0
            WEIGHT = 1.0D0 / 3.0D0
         ELSE IF (I .EQ. 3) THEN
            X1     = 1.0D0 / 2.0D0
            X2     = 0.0D0
            WEIGHT = 1.0D0 / 3.0D0
         END IF
      ELSE IF (KORD .EQ. 4) THEN
         IF      (I .EQ. 1) THEN
            X1     = 1.0D0 / 3.0D0
            X2     = 1.0D0 / 3.0D0
            WEIGHT = -27.0D0  / 48.0D0
         ELSE IF (I .EQ. 2) THEN
            X1     = 0.6D0
            X2     = 0.2D0
            WEIGHT =  25.0D0 / 48.0D0
         ELSE IF (I .EQ. 3) THEN
            X1     = 0.2D0
            X2     = 0.6D0
            WEIGHT =  25.0D0 / 48.0D0
         ELSE IF (I .EQ. 4) THEN
```

```
            X1     = 0.2D0
            X2     = 0.2D0
            WEIGHT =  25.0D0 / 48.0D0
         END IF
      ELSE IF (KORD .EQ. 7) THEN
         IF      (I .EQ. 1) THEN
            X1     = 1.0D0 / 3.0D0
            X2     = 1.0D0 / 3.0D0
            WEIGHT = 0.225D0
         ELSE IF (I .EQ. 2) THEN
            X1     = 0.0597158717D0
            X2     = 0.4701420641D0
            WEIGHT = 0.1323941527D0
         ELSE IF (I .EQ. 3) THEN
            X1     = 0.4701420641D0
            X2     = 0.0597158717D0
            WEIGHT = 0.1323941527D0
         ELSE IF (I .EQ. 4) THEN
            X1     = 0.4701420641D0
            X2     = 0.4701420641D0
            WEIGHT = 0.1323941527D0
         ELSE IF (I .EQ. 5) THEN
            X1     = 0.7974269853D0
            X2     = 0.1012865073D0
            WEIGHT = 0.1259391805D0
         ELSE IF (I .EQ. 6) THEN
            X1     = 0.1012865073D0
            X2     = 0.7974269853D0
            WEIGHT = 0.1259391805D0
         ELSE IF (I .EQ. 7) THEN
            X1     = 0.1012865073D0
            X2     = 0.1012865073D0
            WEIGHT = 0.1259391805D0
         END IF
      END IF
C
      X3 = 1. - X1 - X2
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
      SUBROUTINE UIGSS(NORD,KCEE,CEE,WEIGHT)                          *
C                                                                     *
C     GAUSS POINTS FOR NUMERICAL QUADRATURE OF A                      *
C     SQUARE AREA                                                     *
C                                                                     *
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IF ( NORD .EQ. 2 ) THEN
C
         WEIGHT = 1.000
         CEE    = 0.577350269189626
         IF ( KCEE .EQ. 2 ) CEE = -.577350269189626
C
      ELSE IF ( NORD .EQ. 3 ) THEN
C
         IF ( KCEE .EQ. 1 ) THEN
             CEE = 0.774596669241483
             WC  = 0.555555555555556
         ELSE IF ( KCEE .EQ. 2 ) THEN
             CEE = 0.000000000000000
             WC  = 0.888888888888889
         ELSE IF ( KCEE .EQ. 3 ) THEN
             CEE = -.774596669241483
             WC  = 0.555555555555556
         END IF
         WEIGHT = WC
C
      ELSE IF ( NORD .EQ. 4 ) THEN
C
         IF ( KCEE .EQ. 1 ) THEN
             CEE = 0.861136311594053
             WC  = 0.347854845137454
         ELSE IF ( KCEE .EQ. 2 ) THEN
             CEE = 0.339981043584856
             WC  = 0.652145154862546
         ELSE IF ( KCEE .EQ. 3 ) THEN
             CEE = -.339981043584856
             WC  = 0.652145154862546
         ELSE IF ( KCEE .EQ. 4 ) THEN
             CEE = -.861136311594053
             WC  = 0.347854845137454
         END IF
         WEIGHT = WC
C
      ELSE IF ( NORD .EQ. 5 ) THEN
C
         IF ( KCEE .EQ. 1 ) THEN
             CEE = 0.906179845938664
             WC  = 0.236926885056189
         ELSE IF ( KCEE .EQ. 2 ) THEN
             CEE = 0.538469310105683
```

```
            WC  = 0.478628670499366
         ELSE IF ( KCEE .EQ. 3 ) THEN
            CEE = 0.000000000000000
            WC  = 0.568888888888889
         ELSE IF ( KCEE .EQ. 4 ) THEN
            CEE = -.538469310105683
            WC  = 0.478628670499366
         ELSE IF ( KCEE .EQ. 5 ) THEN
            CEE = -.906179845938664
            WC  = 0.236926885056189
         END IF
         WEIGHT = WC
C
      ELSE IF ( NORD .EQ. 6 ) THEN
C
         IF ( KCEE .EQ. 1 ) THEN
            CEE = 0.932469514203152
            WC  = 0.171324492379170
         ELSE IF ( KCEE .EQ. 2 ) THEN
            CEE = 0.661209386466265
            WC  = 0.360761573048139
         ELSE IF ( KCEE .EQ. 3 ) THEN
            CEE = 0.238619186083197
            WC  = 0.467913934572691
         ELSE IF ( KCEE .EQ. 4 ) THEN
            CEE = -.238619186083197
            WC  = 0.467913934572691
         ELSE IF ( KCEE .EQ. 5 ) THEN
            CEE = -.661209386466265
            WC  = 0.360761573048139
         ELSE IF ( KCEE .EQ. 6 ) THEN
            CEE = -.932469514203152
            WC  = 0.171324492379170
         END IF
         WEIGHT = WC
C
      END IF
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                       *
      SUBROUTINE UHOTCD ( ELTSTF, ELTLOD )                              *
C                                                                       *
C     STATC PERFORMES STATIC CONDENSATION OF THE HOT3                   *
C     ELEMENT STIFFNESS MATRIX AND LOAD VECTOR                          *
C                                                                       *
```

```
      IMPLICIT REAL*8 (A-H,O-Z)                                         *
C                                                                       *
      DIMENSION ELTSTF(17,17), ELTLOD(17)                               *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DO M = 1, 2
        I = 18 - M
        P = 1.0 / ELTSTF(I,I)
        DO J = 1, I
          ELTSTF(I,J) = P * ELTSTF(I,J)
        END DO
        ELTLOD(I) = P * ELTLOD(I)
        DO K = 1, I-1
          DO J = 1, I-1
            ELTSTF(K,J) = ELTSTF(K,J) - ELTSTF(K,I) * ELTSTF(I,J)
          END DO
          ELTLOD(K) = ELTLOD(K) - ELTSTF(K,I) * ELTLOD(I)
        END DO
      END DO
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                       *
      SUBROUTINE UHOTIO(ELTSTF, ELTLOD, C, LFLAGS, Z, DU, ENERGY, A, B, *
     &                  PSI, PSU, AMAT, BMAT, DMAT, GMAT, RELAX, NLAY,  *
     &                  NOTYPE, NOUT, NELID, STRN, ETRN, TRI, NSYS,     *
     &                  NMAT, MLVARX                                  ) *
C                                                                       *
C     THIS ROUTINE COMPUTES ELEMENT OUTPUT DATA FOR THE HOT3            *
C     HIGHER ORDER PLATE ELEMENT.  THIS ROUTINE SUPPORTS LAYER          *
C     OUTPUT REQUESTS FOR COMPOSITE MATERIALS.                          *
C                                                                       *
C     NMAT   = HIGHER-ORDER THEORY SELECTION FLAG;                      *
C            = 1 FOR STRAIN-BASED                                       *
C            = 2 FOR STRESS-BASED (VER I)                               *
C            = 3 FOR STRESS-BASED (VER II)                              *
C                                                                       *
C     OUTPUT OPTIONS DEFINED BY NOTYPE ARE:                             *
C                                                                       *
C     NOTYPE = 0 SUPPRESS ALL ELEMENT OUTPUT                            *
C            = 1 OUTPUT LAYER STRESSES AT USER SELECTED                 *
C                NUMBER OF THICKNESS COORDINATES AND                    *
C                ELEMENT FORCES                                         *
C            = 2 OUTPUT ELEMENT STIFFNESS MATRIX AND                    *
C                CONSISTENT LOAD VECTOR ONLY                            *
```

```
C            = 3 OUTPUT ELEMENT LAYER STRESSES, STRAINS AND              *
C                INCLUDE STIFFNESS MATRIX AND LOAD VECTOR                *
C                                                                        *
C     NSYS   = 1 TO OUTPUT ELEMENT DATA IN GLOBAL COORDINATE SYSTEM      *
C            = 2 TO OUTPUT ELEMENT DATA IN LOCAL  COORDINATE SYSTEM      *
C                                                                        *
      IMPLICIT REAL*8 (A-H,O-Z)                                          *
C                                                                        *
      REAL*8    KAPXO, KAPYO, KAPZO, KAPXYO                              *
C                                                                        *
      DIMENSION ELTSTF(17,17), ELTLOD(17), C(6,6,125), LFLAGS(5),        *
     &          Z(126), U(18), ENERGY(8), A(3), B(3), PSI(3),            *
     &          PSU(3), RHS(18), DU(MLVARX)                              *
      DIMENSION BMTRX(10,17), EVEC(6), SVEC(6), STRAIN(10), XML(3),      *
     &          XMM(3), UL(17), ELTFOR(15), AMAT(4,4), BMAT(4,4),        *
     &          DMAT(4,4), GMAT(2,2), RES(18), STRN(6,6), ETRN(6,6),     *
     &          TRI(3,3), TMP1(3), OEVC(6), OSVC(6)                      *
C                                                                        *
      LOGICAL   PHASE1                                                   *
C                                                                        *
      DATA  NSTART /   0     /                                           *
      DATA  EPS    / 1.0D-20 /                                           *
C                                                                        *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IF ( NOTYPE .LE. 0 ) RETURN
C
C        TEST IF ABAQUS IS IN THE ELEMENT DATA
C        RECOVERY PHASE BY CHECKING DISPLACEMENTS
C
      PHASE1 = .TRUE.
C
      DO I = 1, 18
         IF ( DABS(DU(I)) .GT. EPS ) PHASE1 = .FALSE.
      END DO
C
      IF ( PHASE1 ) RETURN
C
      PI = ACOS(-1.0D0)
C
      DO I = 1, 18
         U(I) = DU(I)
      END DO
      CALL DGMTRN ( U, 6, UL, 3, 3, 5 )
C
      DO I = 16, 17
        UL(I) = ELTLOD(I)
        DO J = 1, I-1
```

```
         UL(I) = UL(I) - ELTSTF(I,J) * UL(J)
       END DO
     END DO
C
     IF ( NSYS .EQ. 2 ) THEN
        CALL DSCPY ( 0.D0, TRI,  3, 3)
        CALL DSCPY ( 0.D0, ETRN, 6, 6)
        CALL DSCPY ( 0.D0, STRN, 6, 6)
        DO I = 1, 6
           IF ( I .LE. 3 ) TRI(I,I) = 1.0
           ETRN(I,I) = 1.0
           STRN(I,I) = 1.0
        END DO
     END IF
C
C    WRITE HEADER TO ABAQUS OUTPUT FILE
C
     IF ( NSTART .EQ. 0 ) THEN
         WRITE(6,1001)
         IF ( NSYS .EQ. 1 ) WRITE(6,2001)
         IF ( NSYS .EQ. 2 ) WRITE(6,2002)
         NSTART = 1
     END IF
C
     WRITE(6,1002) NELID, UL(16), UL(17)
C
     IF ( NOTYPE .EQ. 2 .OR. NOTYPE .EQ. 3 ) THEN
C
C       ELEMENT STIFFNESS MATRIX AND LOAD VECTOR
C       IS OUTPUT IN LOCAL COORDINATES
C
        WRITE(6,2003)
        WRITE(6,1003) (N,N=1,10)
C
        DO I = 1, 15
           WRITE(6,1004) I,(ELTSTF(I,J),J=1,10)
        END DO
C
        WRITE(6,1005) (N,N=11,15)
C
        DO I = 1, 15
           WRITE(6,1004) I,(ELTSTF(I,J),J=11,15)
        END DO
C
        WRITE(6,1006)
C
        DO I = 1, 15
           WRITE(6,1007) I,ELTLOD(I)
```

```
         END DO
C
         WRITE(6,1008)
C
         IF ( NOTYPE .EQ. 2 ) RETURN
      END IF
C
C     COMPUTE ELEMENT STRESSES, STRAINS AND DISPLACEMENTS
C     THROUGH THE THICKNESS
C
      X1 = PSU(1)
      X2 = PSU(2)
      X3 = PSU(3)
C
      XML(1) =  B(3) / 2.0 * X1 * X2 - B(2) / 2.0 * X1 * X3
      XML(2) =  B(1) / 2.0 * X2 * X3 - B(3) / 2.0 * X1 * X2
      XML(3) = -B(1) / 2.0 * X2 * X3 + B(2) / 2.0 * X1 * X3
C
      XMM(1) =  A(2) / 2.0 * X1 * X3 - A(3) / 2.0 * X1 * X2
      XMM(2) = -A(1) / 2.0 * X2 * X3 + A(3) / 2.0 * X1 * X2
      XMM(3) =  A(1) / 2.0 * X2 * X3 - A(2) / 2.0 * X1 * X3
C
C     FORM CONSTANTS TO ENFORCE ZERO SHEAR BOUNDARY
C     CONDITION AT THE TOP AND BOTTOM SURFACES
C
      THICK = Z(NLAY+1) - Z(1)
      HALF  = THICK / 2.0
C
      R13 = C(1,3,1) / C(3,3,1) - C(1,3,NLAY) / C(3,3,NLAY)
      R23 = C(2,3,1) / C(3,3,1) - C(2,3,NLAY) / C(3,3,NLAY)
      R36 = C(3,6,1) / C(3,3,1) - C(3,6,NLAY) / C(3,3,NLAY)
      S13 = C(1,3,1) / C(3,3,1) + C(1,3,NLAY) / C(3,3,NLAY)
      S23 = C(2,3,1) / C(3,3,1) + C(2,3,NLAY) / C(3,3,NLAY)
      S36 = C(3,6,1) / C(3,3,1) + C(3,6,NLAY) / C(3,3,NLAY)
C
C     COMPUTE WEIGTED AVERAGE DISPLACEMENTS AND
C     ROTATIONS AT SPECIFIED RECOVERY POINT
C
      U0 = PSU(1)*UL(1)  + PSU(2)*UL(2)  + PSU(3)*UL(3)
      V0 = PSU(1)*UL(4)  + PSU(2)*UL(5)  + PSU(3)*UL(6)
C
      W0 = PSU(1)*UL(7)  + PSU(2)*UL(8)  + PSU(3)*UL(9)  +
     &     XML(1)*UL(10) + XML(2)*UL(11) + XML(3)*UL(12) +
     &     XMM(1)*UL(13) + XMM(2)*UL(14) + XMM(3)*UL(15)
C
      TX = PSU(1)*UL(10) + PSU(2)*UL(11) + PSU(3)*UL(12)
      TY = PSU(1)*UL(13) + PSU(2)*UL(14) + PSU(3)*UL(15)
C
```

```
C     COMPUTE STRAIN-DISPLACEMENT MATRIX AT SPECIFIED
C     RECOVERY POINT FOR STRESSES
C
      CALL UHOTBX ( BMTRX, A, B, PSI(1), PSI(2), PSI(3), THICK )
      CALL DMMUL  ( BMTRX, UL, STRAIN ,10, 17, 1 )
C
      EPSXO  = STRAIN( 1)
      EPSYO  = STRAIN( 2)
      EPSZO  = STRAIN( 3)
      GAMXYO = STRAIN( 4)
      KAPXO  = STRAIN( 5)
      KAPYO  = STRAIN( 6)
      KAPZO  = STRAIN( 7)
      KAPXYO = STRAIN( 8)
      GAMYZO = STRAIN( 9)
      GAMXZO = STRAIN(10)
C
C     APPLY APPROPRIATE CORRECTION FACTORS TO TRANSVERSE QUANTITIES
C
      GAMXZO = GAMXZO * PI/SQRT(10.0)
      GAMYZO = GAMYZO * PI/SQRT(10.0)
      EPSZO  = EPSZO  * PI/SQRT(12.0)
      KAPZO  = KAPZO  * PI*SQRT(17.0/252.0)
C
C     COMPUTE DISPLACEMENTS, STRESSES AND STRAINS THROUGH
C     THE ELEMENT THICKNESS
C
      WRITE(6,1009)
C
      IF ( NOUT .LE. 1 ) THEN
         LOUT =   1
         ZBEG =   0.0
         ZINC =   0.0
      ELSE
         LOUT =   NOUT
         ZBEG = - HALF
         ZINC =   THICK / ( LOUT - 1 )
      END IF
C
      IF ( NMAT .NE. 1 ) THEN
C
C        COMPUTE CONSTANTS REQUIRED IN STRESS-BASED THEORY
C
         CALL UMTSV2( C, Z, NLAY, THICK, Q11, Q12, Q13, Q16, Q21, Q22,
     1                Q23, Q26, QH11, QH12, QH13, QH16, QH21, QH22,
     2                QH23, QH26 )
C
      END IF
```

```
C
      DO IL = 1, LOUT
C
         ZL = ZBEG + ( IL - 1 ) * ZINC
C
         DO I= 1, NLAY
            IF ( ZL .GE. Z(I) ) NPL = I
         END DO
C
         PO  = 1.0
         P1  = ZL / HALF
         P2  = 0.5 * (3.*(ZL/HALF)**2 - 1.0)
         P3  = 0.5 * (ZL/HALF) * (5.*(ZL/HALF)**2 - 3.0)
         FI  = P1 - P1**3 / 3.0
C
         PH1 = -(HALF/510.)*(S13 * (42. * P3 + 3. * P1) -85. * P2 * R13)
         PH2 = -(HALF/510.)*(S23 * (42. * P3 + 3. * P1) -85. * P2 * R23)
         PH6 = -(HALF/510.)*(S36 * (42. * P3 + 3. * P1) -85. * P2 * R36)
         PH3 = 42.DO / 85.DO * (6. * P1 - P3 )
         PHT =  5.DO /  6.DO * ( PO - P2 )
C
C        COMPUTATION OF DISPLACEMENTS:
C
         UXX = UO + ZL * TY
         UYY = VO + ZL * TX
         UZZ = WO + P1 * UL(16) + ( .20 + P2 ) * UL(17)
C
         OUX = UXX*TRI(1,1) + UYY*TRI(2,1) + UZZ*TRI(3,1)
         OUY = UXX*TRI(1,2) + UYY*TRI(2,2) + UZZ*TRI(3,2)
         OUZ = UXX*TRI(1,3) + UYY*TRI(2,3) + UZZ*TRI(3,3)

C        COMPUTATION OF STRAINS:
C
         EPSXX = EPSXO + ZL  * KAPXO
         EPSYY = EPSYO + ZL  * KAPYO
         EPSZZ = EPSZO + KAPZO * PH3 + KAPXO  * PH1 +
     &                   KAPYO * PH2 + KAPXYO * PH6
         GAMXY = GAMXYO + ZL * KAPXYO
         GAMXZ = PHT * GAMXZO
         GAMYZ = PHT * GAMYZO
C
C        COMPUTATION OF STRESSES:
C
         EVEC(1) = EPSXX
         EVEC(2) = EPSYY
         EVEC(3) = EPSZZ
         EVEC(4) = GAMYZ
         EVEC(5) = GAMXZ
```

```
      EVEC(6) = GAMXY
C
      CALL DMMUL ( C(1,1,NPL), EVEC, SVEC, 6, 6, 1 )
C
      IF ( NMAT .NE. 1 ) THEN
C
C        RECOMPUTE EPSZZ AND SIGZZ USING STRESS-BASED THEORY
C
         SIGZZ = ( Q11  + FI * Q21  ) * EPSXO
     &         + ( Q12  + FI * Q22  ) * EPSYO
     &         + ( Q13  + FI * Q23  ) * EPSZO
     &         + ( Q16  + FI * Q26  ) * GAMXYO
     &         + ( QH11 + FI * QH21 ) * KAPXO
     &         + ( QH12 + FI * QH22 ) * KAPYO
     &         + ( QH13 + FI * QH23 ) * KAPZO
     &         + ( QH16 + FI * QH26 ) * KAPXYO
C
         EPSZZ = ( SIGZZ - C(1,3,NPL) * EPSXX - C(2,3,NPL) * EPSYY -
     &             C(3,6,NPL) * GAMXY ) / C(3,3,NPL)
         EVEC(3) = EPSZZ
         SVEC(3) = SIGZZ
C
      END IF
C
C     TRANSFORM STRESSES AND STRAINS TO SELECTED OUTPUT COORDINATES
C
      CALL DMATB( STRN, EVEC, OEVC, 6, 6, 1 )
      CALL DMATB( ETRN, SVEC, OSVC, 6, 6, 1 )
C
      WRITE(6,1010)  ZL,OUX,OUY,OUZ,( OEVC(K),K=1,6),
     &                              ( OSVC(L),L=1,6)
C
    END DO
C
C   COMPUTE RESULTANTS
C
    CALL DSCPY ( 0.D0, RES, 18, 1)
    DO I = 1, 10
      IF ( I .LE. 3 ) THEN
         DO J = 1, 4
            RES(I) = RES(I) + AMAT(I,J)*STRAIN(J) +
     &                        BMAT(I,J)*STRAIN(J+4)
         END DO
      ELSE IF ( I .EQ. 4 ) THEN
         DO J = 1, 4
            RES(6) = RES(6) + AMAT(I,J)*STRAIN(J) +
     &                        BMAT(I,J)*STRAIN(J+4)
         END DO
```

```
      ELSE IF ( I .GT. 4 .AND. I .LE. 7 ) THEN
         DO J = 1, 4
            RES(I+2) = RES(I+2) + BMAT(J,I-4)*STRAIN(J) +
     &                            DMAT(I-4,J)*STRAIN(J+4)
         END DO
      ELSE IF ( I .EQ. 8 ) THEN
         DO J = 1, 4
            RES(12) = RES(12) + BMAT(J,I-4)*STRAIN(J) +
     &                          DMAT(I-4,J)*STRAIN(J+4)
         END DO
      ELSE IF ( I .GT. 8 ) THEN
         DO J = 1, 2
            RES(I+7) = RES(I+7) + RELAX*GMAT(I-8,J)*STRAIN(11-J)
         END DO
      END IF
   END DO
C
   DO I = 1, 3
      CALL DSTRN ( RES(6*I-5), EVEC, 6 )
      CALL DMATB ( ETRN, EVEC, RES(6*I-5), 6, 6, 1 )
   END DO
   IF ( LOUT .GT. 1 ) THEN
     IF (NSYS .EQ. 1) WRITE(6,2011) (RES(K),K=1,12),(RES(J),J=16,18)
     IF (NSYS .EQ. 2) WRITE(6,1011) (RES(K),K=1, 3),(RES(J),J=6 , 9),
     &                              RES(12),RES(16),RES(17)
   END IF
C
C  COMPUTE ELEMENT GRID POINT FORCES,  {F} = [K]{U}
C
   CALL DGMMUL ( ELTSTF,  17,  UL, 17, ELTFOR, 15, 15, 15, 1 )
   CALL DSCPY  (   0.D0, RHS,  18,  1)
   CALL DGMTRN ( ELTFOR,   3, RHS,  6, 5, 3 )
C
   DO I = 1, 3
      N1 = I*6-5
      N2 = N1+2
      N3 = N2+1
      N4 = N3+2
      NC = 0
      DO J = N1, N2
         NC = NC + 1
         TMP1(NC) = RHS(J)
      END DO
      CALL DMATB(TRI,TMP1,RHS(N1),3,3,1)
      NC = 0
      DO J = N3, N4
         NC = NC + 1
         TMP1(NC) = RHS(J)
```

```
         END DO
C
C        TOGGLE SIGN ON MX DUE TO SIGN CONVENTION OF THETA X
C
         TMP1(1) = -TMP1(1)
         CALL DMATB(TRI,TMP1,RHS(N3),3,3,1)
         RHS(N3) = -RHS(N3)
      END DO
C
      WRITE(6,1012)
C
      DO I = 1, 3
         WRITE(6,1013) I,(RHS(J),J=I*6-5,I*6)
      END DO
C
C     COMPUTE ELEMENT ELASTIC STRAIN ENERGY, E = 0.5 {U}[KEE]{U}
C
      CALL DMATB  ( UL, ELTFOR, STNE, 1, 15, 1 )
C
      STNE        = 0.5 * STNE
      ENERGY(2)   = STNE
      WRITE(6,1014) STNE
C
      CALL DSCPY  (   0.D0, RHS,  18,  1)
C
 1001 FORMAT(///, 30X,'H O T   E L E M E N T  D A T A  ',//)
 1002 FORMAT(// ,' ELEMENT ID: ',I5,//,
     &          ' HIGHER ORDER AUXILIARY FUNCTIONS: ',/,
     &          ' W1 = ',E9.3,' W2 = ',E9.3,/)
 1003 FORMAT(/  , 45X,'HOT3 STIFFNESS MATRIX:',/,10I11)
 1004 FORMAT(     I4,2X,10(E9.3,2X))
 1005 FORMAT(// , 5I11)
 1006 FORMAT(// , 40X,'HOT3 CONSISTENT LOAD VECTOR:',/)
 1007 FORMAT(     44X,I5,2X,E9.3)
 1008 FORMAT(//   )
 1009 FORMAT(/  ,' HOT3 DISPLACEMENTS, STRAINS AND STRESSES'
     &          ,' THROUGH ELEMENT THICKNESS:',///,51X
     &          ,'XX',9X,'YY',9X,'ZZ',9X,'YZ',9X,'XZ',9X,'XY',/
     &          ,'       Z         UX         UY         UZ   '
     &          , 3('    EPS/SIG'),3('    GAM/TAU'))
 1010 FORMAT(/  ,  10D11.3,/,44X,6D11.3)
 1011 FORMAT(/  ,' FORCE AND MOMENT RESULTANTS:',//
     &          ,' NX  = ',D9.3,'   NY  = ',D9.3,'   NZ = ',D9.3
     &          ,'   NXY = ',D9.3,/
     &          ,' MX  = ',D9.3,'   MY  = ',D9.3,'   MZ = ',D9.3
     &          ,'   MXY = ',D9.3,/
     &          ,' QX  = ',D9.3,'   QY  = ',D9.3,/)
 2011 FORMAT(/  ,' FORCE AND MOMENT RESULTANTS:',//
```

```
     &          ,' NX  = ',D9.3,'   NY  = ',D9.3,'   NZ = ',D9.3
     &          ,'   NYZ = ',D9.3,'   NZX = ',D9.3,'   NXY = ',D9.3,/
     &          ,' MX  = ',D9.3,'   MY  = ',D9.3,'   MZ = ',D9.3
     &          ,'   MYZ = ',D9.3,'   MZX = ',D9.3,'   MXY = ',D9.3,/
     &          ,' QX  = ',D9.3,'   QY  = ',D9.3,'   QZ = ',D9.3/)
 1012 FORMAT(/  ,' ELEMENT FORCES',//
     &          ,'  NODE',4X,'R1',9X,'R2',9X,'R3',9X,'R4',9X,'R5'
     &          , 9X,'R6',/)
 1013 FORMAT(     I5,2X,6(E9.3,2X))
 1014 FORMAT(/  ,' ELEMENT STRAIN ENERGY = ',E9.3)
 2001 FORMAT(25X,' ELEMENT DATA IN GLOBAL COORDINATE SYSTEM',//)
 2002 FORMAT(25X,' ELEMENT DATA IN LOCAL COORDINATE SYSTEM',//)
 2003 FORMAT(23X,' (ELEMENT [K] & {R} ARE IN LOCAL COORDINATES)',//)
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                       *
      SUBROUTINE UMTSV2( C, NLAY, THKNS, Q11, Q12, Q13, Q16, Q21, Q22,  *
     1                   Q23, Q26, QH11, QH12, QH13, QH16, QH21, QH22,  *
     2                   QH23, QH26 )                                   *
C                                                                       *
C     THIS ROUTINE COMPUTES CONSTANTS REQUIRED FOR TRANSVERSE NORMAL    *
C     STRAIN AND STRESS COMPUTATION IN STRESS-BASED TESSLER THEORY      *
C                                                                       *
      IMPLICIT REAL*8 (A-H,O-Z)                                         *
C                                                                       *
      DIMENSION   C(6,6,125), Z(126), S33(125)                          *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     INITIALIZE VARIABLES
C
      S11  = 0.0
      S12  = 0.0
      S22  = 0.0
      T11  = 0.0
      T12  = 0.0
      T16  = 0.0
      T13  = 0.0
      TH11 = 0.0
      TH12 = 0.0
      TH16 = 0.0
      TH13 = 0.0
      T21  = 0.0
      T22  = 0.0
      T26  = 0.0
```

```
      T23  = 0.0
      TH21 = 0.0
      TH22 = 0.0
      TH26 = 0.0
      TH23 = 0.0
C
C     COMPUTE WEIGHTED-AVERAGE TERMS
C
      HL = THKNS / 2.0
C
      DO 100 K = 1, NLAY
C
         HK   = Z(K)
         HKP1 = Z(K+1)
C
         ZOI  =  HKP1    - HK
         Z1I  = (HKP1**2 - HK**2) / 2.0
C
C        INTEGRAL OF FI = PSI-PSI**3/3 ACROSS PLY THICKNESS
C
         FINT = -(HKP1-HK)*(HKP1+HK)*(-3*THKNS**2+2*HKP1**2+2*HK**2)/
     1            THKNS**3/3.0
C
C        INTEGRAL OF FI**2 = (PSI-PSI**3/3)**2 ACROSS PLY THICKNESS
C
         FINT2 =(320*HKP1**7-672*THKNS**2*HKP1**5+420*THKNS**4*HKP1**3-
     1           320*HK**7+672*THKNS**2*HK**5-420*THKNS**4*HK**3)/
     2           THKNS**6/315.0
C
C        INTEGRAL OF Z*FI ACROSS PLY THICKNESS
C
         ZFINT = -(8*HKP1**5-10*THKNS**2*HKP1**3-8*HK**5+
     1             10*THKNS**2*HK**3)/THKNS**3/15.0
C
         S33(K)  = 1.0 / C(3,3,K)
C
         S11 =  S11 + ZOI   * S33(K)**2
         S12 =  S12 + FINT  * S33(K)**2
         S22 =  S22 + FINT2 * S33(K)**2
C
         T11 =  T11 + ZOI * C(1,3,K) * S33(K)**2
         T12 =  T12 + ZOI * C(2,3,K) * S33(K)**2
         T16 =  T16 + ZOI * C(3,6,K) * S33(K)**2
         T13 =  T13 + ZOI * S33(K)
C
         T21 =  T21 + FINT * C(1,3,K) * S33(K)**2
         T22 =  T22 + FINT * C(2,3,K) * S33(K)**2
         T26 =  T26 + FINT * C(3,6,K) * S33(K)**2
```

```
      T23 =  T23 + FINT * S33(K)
C
      TH11 =  TH11 + Z1I * C(1,3,K) * S33(K)**2
      TH12 =  TH12 + Z1I * C(2,3,K) * S33(K)**2
      TH16 =  TH16 + Z1I * C(3,6,K) * S33(K)**2
      TH13 =  TH13 + 2.0 * Z1I * S33(K)
C
      TH21 =  TH21 + ZFINT * C(1,3,K)* S33(K)**2
      TH22 =  TH22 + ZFINT * C(2,3,K)* S33(K)**2
      TH26 =  TH26 + ZFINT * C(3,6,K)* S33(K)**2
      TH23 =  TH23 + 2.0 * ZFINT * S33(K)
C
  100 CONTINUE
C
C     AVERAGE COEFFICIENTS
C
      R11 =  S22 / (S11*S22 - S12**2)
      R22 =  S11 / (S11*S22 - S12**2)
      R12 = -S12 / (S11*S22 - S12**2)
C
      Q11 = R11 * T11 + R12 * T21
      Q21 = R12 * T11 + R22 * T21
      Q12 = R11 * T12 + R12 * T22
      Q22 = R12 * T12 + R22 * T22
      Q13 = R11 * T13 + R12 * T23
      Q23 = R12 * T13 + R22 * T23
      Q16 = R11 * T16 + R12 * T26
      Q26 = R12 * T16 + R22 * T26
C
      QH11 = R11 * TH11 + R12 * TH21
      QH21 = R12 * TH11 + R22 * TH21
      QH12 = R11 * TH12 + R12 * TH22
      QH22 = R12 * TH12 + R22 * TH22
      QH16 = R11 * TH16 + R12 * TH26
      QH26 = R12 * TH16 + R22 * TH26
      QH13 = R11 * TH13 + R12 * TH23
      QH23 = R12 * TH13 + R22 * TH23
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
      SUBROUTINE UMTCMX( E11,E22,E33,V12,V23,V13,G12,G23,G31,FIB,C )  *
C                                                                     *
      IMPLICIT REAL*8 (A-H,O-Z)                                       *
C                                                                     *
      DIMENSION  D(6,6), T(6,6), TD(6,6), C(6,6)                      *
```

```
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      CALL UMTDMX (  D, E11, E22, E33, V12, V23, V13, G12, G23, G31 )
      CALL UMTTMX (  T, FIB)
      CALL DMMUL  (  T, D, TD, 6, 6, 6)
      CALL DMABT  ( TD, T,  C, 6, 6, 6)
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                       *
      SUBROUTINE UMTDMX ( D, E1, E2, E3, V12, V23, V13, G12, G23, G31 ) *
C                                                                       *
C             COMPUTE FULL ELASTIC D-MATRIX                             *
C                                                                       *
      IMPLICIT REAL*8    (A-H,O-Z)                                      *
C                                                                       *
      DIMENSION D(6,6)                                                  *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      CALL DSCPY ( 0.D0, D, 6, 6 )
C
      V21 = V12 * E2 / E1
      V32 = V23 * E3 / E2
      V31 = V13 * E3 / E1
C
      FACTOR = 1. / ( 1.0 - V12 * V21 - V13 * V31 - V23 * V32
     &                    - V12 * V23 * V31 - V21 * V13 * V32 )
C
      D(1,1) = FACTOR * E1 * ( 1.0 - V23 * V32 )
      D(2,2) = FACTOR * E2 * ( 1.0 - V13 * V31 )
      D(3,3) = FACTOR * E3 * ( 1.0 - V12 * V21 )
      D(1,2) = FACTOR * E2 * ( V12 + V13 * V32 )
      D(1,3) = FACTOR * E3 * ( V13 + V12 * V23 )
      D(2,3) = FACTOR * E3 * ( V23 + V21 * V13 )
      D(2,1) = D(1,2)
      D(3,1) = D(1,3)
      D(3,2) = D(2,3)
      D(4,4) = G23
      D(5,5) = G31
      D(6,6) = G12
C
      RETURN
      END
C
```

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE UMTTMX ( T, FIB )                                      *
C                                                                       *
C                 FORM T-MATRIX                                         *
C                                                                       *
      IMPLICIT REAL*8    (A-H,O-Z)                                      *
C                                                                       *
      DIMENSION T(6,6)                                                  *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      ANG = FIB * ACOS(-1.0D0) / 180.
      C   = COS(ANG)
      S   = SIN(ANG)
C
      CALL  DSCPY ( 0.D0, T, 6, 6 )
C
      T(1,1) =  C * C
      T(1,2) =  S * S
      T(1,6) = -2. * S * C
      T(2,1) =  S * S
      T(2,2) =  C * C
      T(2,6) =  2. * S * C
      T(3,3) =  1.0
      T(6,1) =  S * C
      T(6,2) = -S * C
      T(6,6) =  C * C - S * S
      T(5,5) =  C
      T(5,4) = -S
      T(4,5) =  S
      T(4,4) =  C
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                       *
C                   [A] x [B] = [C]                                     *
C                                                                       *
      SUBROUTINE     DGMMUL ( DA, LDA, DB, LDB, DC, LDC, IRA, IRB, ICB )*
C                                                                       *
      IMPLICIT INTEGER*4  (    I-N    )                                 *
      IMPLICIT REAL*8     ( A-H , O-Z )                                 *
C                                                                       *
      DIMENSION     DA(LDA,1), DB(LDB,1), DC(LDC,1)                     *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
```

```
      DO I=1,IRA
         DO K=1,ICB
            DS = 0.0D+00
            DO J=1,IRB
               DS = DS + DA(I,J) * DB(J,K)
            END DO
            DC(I,K) = DS
         END DO
      END DO
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                      T                                              *
C                   [A] x [B] = [C]                                   *
C                                                                     *
      SUBROUTINE     DGMATB ( DA, LDA, DB, LDB, DC, LDC, IRA, IRB, ICB )*
C                                                                     *
      IMPLICIT INTEGER*4  (    I-N    )                               *
      IMPLICIT REAL*8     ( A-H , O-Z )                               *
C                                                                     *
      DIMENSION     DA(LDA,1), DB(LDB,1), DC(LDC,1)                   *
C                                                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DO I=1,IRA
         DO K=1,ICB
            DS = 0.0D+00
            DO J=1,IRB
               DS = DS + DA(J,I) * DB(J,K)
            END DO
            DC(I,K) = DS
         END DO
      END DO
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                          T                                          *
C                        [A] --> [C]                                  *
C                                                                     *
      SUBROUTINE     DGMTRN ( DA, LDA, DC, LDC, IRC, ICC )             *
C                                                                     *
      IMPLICIT INTEGER*4  (    I-N    )                               *
      IMPLICIT REAL*8     ( A-H , O-Z )                               *
C                                                                     *
```

```
      DIMENSION     DA(LDA,1), DC(LDC,1)                                *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DO J=1,ICC
         DO I=1,IRC
            DC(I,J) = DA(J,I)
         END DO
      END DO
C
      RETURN
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                            T                                          *
C                    [A] x [B] = [C]                                    *
C                                                                       *
      SUBROUTINE     DMABT ( DA, DB, DC, IRA, IRB, ICB )                *
C                                                                       *
      IMPLICIT INTEGER*4  (    I-N    )                                *
      IMPLICIT REAL*8     ( A-H , O-Z )                                *
C                                                                       *
      DIMENSION     DA(IRA,1), DB(ICB,1), DC(IRA,1)                     *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DO I=1,IRA
         DO K=1,ICB
            DS = 0.0D+00
            DO J=1,IRB
               DS = DS + DA(I,J) * DB(K,J)
            END DO
            DC(I,K) = DS
         END DO
      END DO
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                       *
C                        [A] + [B] = [C]                                *
C                                                                       *
      SUBROUTINE     DMADD ( DA, DB, DC, IRC, ICC )                     *
C                                                                       *
      IMPLICIT INTEGER*4  (    I-N    )                                *
      IMPLICIT REAL*8     ( A-H , O-Z )                                *
C                                                                       *
C                                                                       *
```

```
      DIMENSION      DA(1), DB(1), DC(1)                                *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DO I= 1, IRC * ICC
         DC(I) = DA(I) + DB(I)
      END DO
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                       *
C                        [A] + S x [B] = [C]                            *
C                                                                       *
      SUBROUTINE     DMASB ( DA, DS, DB, DC, IRC, ICC )                 *
C                                                                       *
      IMPLICIT INTEGER*4  (    I-N    )                                *
      IMPLICIT REAL*8     ( A-H , O-Z )                                *
C                                                                       *
      DIMENSION     DA(1), DB(1), DC(1)                                *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DO I= 1, IRC * ICC
         DC(I) = DA(I) + DS * DB(I)
      END DO
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                      T                                                *
C                   [A] x [B] = [C]                                     *
C                                                                       *
      SUBROUTINE     DMATB ( DA, DB, DC, IRA, IRB, ICB )                *
C                                                                       *
      IMPLICIT INTEGER*4  (    I-N    )                                *
      IMPLICIT REAL*8     ( A-H , O-Z )                                *
C                                                                       *
      DIMENSION     DA(IRB,1), DB(IRB,1), DC(IRA,1)                    *
C                                                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DO I=1,IRA
         DO K=1,ICB
            DS  = 0.0D+00
            DO J=1,IRB
```

```
            DS = DS + DA(J,I) * DB(J,K)
         END DO
         DC(I,K) = DS
      END DO
   END DO
C
   RETURN
   END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
C                  [A] x [B] = [C]                                    *
C                                                                     *
   SUBROUTINE     DMMUL ( DA, DB, DC, IRA, IRB, ICB )                 *
C
   IMPLICIT INTEGER*4  (    I-N    )                                 *
   IMPLICIT REAL*8     ( A-H , O-Z )                                 *
C                                                                     *
   DIMENSION     DA(IRA,1), DB(IRB,1), DC(IRA,1)                      *
C                                                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
   DO I=1,IRA
      DO K=1,ICB
         DS = 0.0D+00
         DO J=1,IRB
            DS = DS + DA(I,J) * DB(J,K)
         END DO
         DC(I,K) = DS
      END DO
   END DO
C
   RETURN
   END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                          T                                          *
C                       [A] --> [C]                                   *
C                                                                     *
   SUBROUTINE     DMTRN ( DA, DC, IRC, ICC )                          *
C                                                                     *
   IMPLICIT INTEGER*4  (    I-N    )                                 *
   IMPLICIT REAL*8     ( A-H , O-Z )                                 *
C                                                                     *
   DIMENSION     DA(ICC,1), DC(IRC,1)                                 *
C                                                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
```

```
      DO J=1,ICC
         DO I=1,IRC
            DC(I,J) = DA(J,I)
         END DO
      END DO
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                   *
C                    S  -->  [C]                                    *
C                                                                   *
      SUBROUTINE     DSCPY ( DS, DC, IRC, ICC )                     *
C                                                                   *
      IMPLICIT INTEGER*4  (    I-N    )                             *
      IMPLICIT REAL*8     ( A-H , O-Z )                             *
C                                                                   *
      DIMENSION      DC(1)                                          *
C                                                                   *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DO I= 1, IRC * ICC
         DC(I) = DS
      END DO
C
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                   *
C                     S x [A] = [C]                                 *
C                                                                   *
      SUBROUTINE     DSMUL ( DS, DA, DC, IRC, ICC )                 *
C                                                                   *
      IMPLICIT INTEGER*4  (    I-N    )                             *
      IMPLICIT REAL*8     ( A-H , O-Z )                             *
C                                                                   *
      DIMENSION      DA(1), DC(1)                                   *
C                                                                   *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DO I= 1, IRC * ICC
         DC(I) = DS * DA(I)
      END DO
C
      RETURN
      END
```

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
C                        [A] = [C]                                    *
C                                                                     *
      SUBROUTINE     DSTRN ( DA, DC, IRC )                            *
C                                                                     *
      IMPLICIT INTEGER*4  (    I-N    )                              *
      IMPLICIT REAL*8     ( A-H , O-Z )                              *
C                                                                     *
      DIMENSION      DA(1), DC(1)                                     *
C                                                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DO I= 1, IRC
         DC(I) = DA(I)
      END DO
C
      RETURN
      END
```

APPENDIX B

Demonstration Problem I: Static Analysis

ABAQUS INPUT DECK

```
*HEADING
 USER ELEMENT HOT3 TEST CASE; CYLINDRICAL PLATE BENDING
*PREPRINT,ECHO=YES
*NODE
1,0.0,    0.0,    0.0
2,0.3333,0.0,    0.0
3,0.6667,0.0,    0.0
4,1.0,    0.0,    0.0
5,0.0,    0.3333,0.0
6,0.3333,0.3333,0.0
7,0.6667,0.3333,0.0
8,1.0,    0.3333,0.0
**
*USER ELEMENT,NODES=3,TYPE=U1,PROPERTIES=72,COORDINATES=3
1,2,3,4,5,6
**
*ELSET,ELSET=N1
*ELSET,ELSET=N2
*ELSET,ELSET=N3
*ELSET,ELSET=N4
*ELSET,ELSET=N5
*ELSET,ELSET=N6
**
*UEL PROPERTY,ELSET=N1
3.0,0.0,1.000,.8660,1.000,1.0,13.0,1.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.025,30.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.05,-30.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.025,30.0
0.333,0.333,0.333,1.0,0.0,0.0
1.0,1.0,1.0,0.0,0.0,0.0,1.0,0.0
*UEL PROPERTY,ELSET=N2
3.0,0.0,1.000,0.8660,0.866,1.0,13.0,1.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.025,30.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.05,-30.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.025,30.0
0.333,0.333,0.333,1.0,0.0,0.0
1.0,1.0,1.0,0.0,0.0,0.0,1.0,0.0
*UEL PROPERTY,ELSET=N3
3.0,0.0,0.866,0.500,0.866,1.0,13.0,1.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.025,30.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.05,-30.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.025,30.0
0.333,0.333,0.333,1.0,0.0,0.0
1.0,1.0,1.0,0.0,0.0,0.0,1.0,0.0
*UEL PROPERTY,ELSET=N4
3.0,0.0,0.866,0.500,0.500,1.0,13.0,1.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.025,30.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.05,-30.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.025,30.0
0.333,0.333,0.333,1.0,0.0,0.0
1.0,1.0,1.0,0.0,0.0,0.0,1.0,0.0
*UEL PROPERTY,ELSET=N5
3.0,0.0,0.500,0.000,0.500,1.0,13.0,1.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.025,30.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.05,-30.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.025,30.0
0.333,0.333,0.333,1.0,0.0,0.0
1.0,1.0,1.0,0.0,0.0,0.0,1.0,0.0
*UEL PROPERTY,ELSET=N6
3.0,0.0,0.500,0.000,0.000,1.0,13.0,1.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.025,30.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.05,-30.0
25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
0.25,0.25,0.25,0.025,30.0
0.333,0.333,0.333,1.0,0.0,0.0
1.0,1.0,1.0,0.0,0.0,0.0,1.0,0.0
**
*ELEMENT,TYPE=U1,ELSET=N1
```

```
1,1,6,5
*ELEMENT,TYPE=U1,ELSET=N2
2,1,2,6
*ELEMENT,TYPE=U1,ELSET=N3
3,2,7,6
*ELEMENT,TYPE=U1,ELSET=N4
4,2,3,7
*ELEMENT,TYPE=U1,ELSET=N5
5,3,8,7
*ELEMENT,TYPE=U1,ELSET=N6
6,3,4,8
**
*USER SUBROUTINE,INPUT=uel_hot3.f
**
*BOUNDARY
1,1,2
1,4,6
2,2,2
2,6,6
3,2,2
3,6,6
4,2,3
4,6,6
5,1,2
5,4,6
6,2,2
6,6,6
7,2,2
7,6,6
8,2,3
8,6,6
**
**  INPUT EQUATIONS TO SIMULATE
**  CYLINDRICAL BENDING
**
*EQUATION
2
5,3,1.0,1,3,-1.0
2
6,1,1.0,2,1,-1.0
2
6,3,1.0,2,3,-1.0
2
6,4,1.0,2,4,-1.0
2
6,5,1.0,2,5,-1.0
2
7,1,1.0,3,1,-1.0
2
7,3,1.0,3,3,-1.0
2
7,4,1.0,3,4,-1.0
2
7,5,1.0,3,5,-1.0
2
8,1,1.0,4,1,-1.0
2
8,4,1.0,4,4,-1.0
2
8,5,1.0,4,5,-1.0
**
*STEP,PERTURBATION
*STATIC
*NODE PRINT
 U
*END STEP
```

ABAQUS OUTPUT FILE

```
  AAAAAA     BBBBBBBBB      AAAAAA      QQQQQQQQ    U        U    SSSSSSSS
 A      A    B        B    A      A    Q        Q   U        U   S
A        A   B        B   A        A   Q        Q   U        U   S
A        A   B        B   A        A   Q        Q   U        U   S
AAAAAAAAAA   BBBBBBBBB    AAAAAAAAAA   Q        Q   U        U    SSSSSSSS
A        A   B        B   A        A   Q    Q   Q   U        U            S
A        A   B        B   A        A   Q     Q  Q   U        U            S
A        A   B        B   A        A   Q      Q Q   U        U            S
A        A   BBBBBBBBB    A        A    QQQQQQQQ     UUUUUUUU     SSSSSSSS
                                                Q
```

```
<|> <|> <|> <|> <|> <|> <|> <|> <|> <|>
<|> <|> <|> <|> <|> <|> <|>  |   |   |
 |   |   |   |   |   |   |  <|> <|> <|>
----------------------------------------
----------------------------------------
 |  <|>  |  <|>  |   |  <|> <|> <|> <|>
 |  <|>  |   |   |   |   |  <|> <|> <|>
<|>  |  <|>  |  <|> <|>  |  <|>  |  <|>
<|>  |  <|> <|> <|> <|> <|> <|>  |   |
<|> <|> <|> <|> <|> <|> <|>  |  <|>  |
<|> <|> <|> <|> <|> <|> <|>  |  <|> <|>
<|> <|> <|> <|> <|> <|> <|> <|> <|> <|>
```

THIS PROGRAM HAS BEEN DEVELOPED BY

HIBBITT, KARLSSON AND SORENSEN, INC.
1080 MAIN STREET
PAWTUCKET, R.I. 02860

THIS IS A PROPRIETARY PROGRAM. IT MAY ONLY BE USED UNDER THE TERMS OF THE LICENSE AGREEMENT BETWEEN HIBBITT, KARLSSON & SORENSEN, INC. AND ARMY RESEARCH LABORATORY.

ON MACHINE 0X080009411654,
YOU ARE AUTHORIZED TO RUN
STANDARD AND POST UNTIL 12/31/93

YOUR SITE ID IS: ARMY RESEARCH

FOR ASSISTANCE OR ANY OTHER INFORMATION CALL

401-727-4200 (PAWTUCKET, RHODE ISLAND)
OR
510-794-5891 (NEWARK, CALIFORNIA)

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                           *
*                  ****************                         *
*                  *  N O T I C E  *                        *
*                  ****************                         *
*                                                           *
*                                                           *
*              THIS IS ABAQUS VERSION 5.2.                  *
*                                                           *
*                                                           *
*  PLEASE MAKE SURE YOU ARE USING VERSION 5.2 MANUALS       *
*  PLUS THE NOTES ACCOMPANYING THIS RELEASE. THESE NOTES    *
*  CAN BE OBTAINED BY USING THE INFORMATION OPTION ON THE   *
*  ABAQUS COMMAND LINE                                      *
*                                                           *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

A B A Q U S I N P U T E C H O

```
      5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
-------------------------------------------------------------------------------
*HEADING
 USER ELEMENT HOT: TEST CASE; CYLINDRICAL PLATE BENDING
*PREPRINT,ECHO=YES
*NODE
CARD     5    1,0.0,   0.0,   0.0
```

```
                2,0.3333,0.0,    0.0
                3,0.6667,0.0,    0.0
                4,1.0,    0.0,    0.0
                5,0.0,    0.3333,0.0
CARD     10     6,0.3333,0.3333,0.0
                7,0.6667,0.3333,0.0
                8,1.0,    0.3333,0.0
                **
                *USER ELEMENT,NODES=3,TYPE=U1,PROPERTIES=72,COORDINATES=3
CARD     15     1,2,3,4,5,6
                **
                *ELSET,ELSET=N1
                *ELSET,ELSET=N2
                *ELSET,ELSET=N3
CARD     20     *ELSET,ELSET=N4
                *ELSET,ELSET=N5
                *ELSET,ELSET=N6
                **
                *UEL PROPERTY,ELSET=N1
CARD     25     3.0,0.0,1.000,.8660,1.000,1.0,13.0,1.0
                25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.025,30.0
                25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.05,-30.0
CARD     30     25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.025,30.0
                0.333,0.333,0.333,1.0,0.0,0.0
                1.0,1.0,1.0,0.0,0.0,0.0,1.0,0.0
                *UEL PROPERTY,ELSET=N2
CARD     35     3.0,0.0,1.000,0.8660,0.8660,1.0,13.0,1.0
                25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.025,30.0
                25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.05,-30.0
CARD     40     25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.025,30.0
                0.333,0.333,0.333,1.0,0.0,0.0
                1.0,1.0,1.0,0.0,0.0,0.0,1.0,0.0
                *UEL PROPERTY,ELSET=N3
CARD     45     3.0,0.0,0.866,0.500,0.866,1.0,13.0,1.0
                25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.025,30.0
                25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25.0.25,0.25,0.05,-30.0
CARD     50     25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.025,30.0
                0.333,0.333,0.333,1.0,0.0,0.0
                1.0,1.0,1.0,0.0,0.0,0.0,1.0,0.0
                *UEL PROPERTY,ELSET=N4
CARD     55     3.0,0.0,0.866,0.500,0.500,1.0,13.0,1.0
                25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.025,30.0
                25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.05,-30.0
CARD     60     25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.025,30.0
                0.333,0.333,0.333,1.0,0.0,0.0
                1.0,1.0,1.0,0.0,0.0,0.0,1.0,0.0
                *UEL PROPERTY,ELSET=N5
CARD     65     3.0,0.0,0.500,0.000,0.500,1.0,13.0,1.0
                25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.025,30.0
                25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.05,-30.0
CARD     70     25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.025,30.0
                0.333,0.333,0.333,1.0,0.0,0.0
                1.0,1.0,1.0,0.0,0.0,0.0,1.0,0.0
                *UEL PROPERTY,ELSET=N6
CARD     75     3.0,0.0,0.500,0.000,0.000,1.0,13.0,1.0
                25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.025,30.0
                25.0D6,1.0D6,1.0D6,0.5D6,0.2D6,0.5D6
                0.25,0.25,0.25,0.05,-30.0
CARD     80     25.0D6,1.0D6,1.0D6,0.5D6,0.2D6.0.5D6
                0.25,0.25,0.25,0.025,30.0
                0.333,0.333,0.333,1.0,0.0,0.0
                1.0,1.0,1.0,0.0,0.0,0.0,1.0,0.0
                **
CARD     85     *ELEMENT,TYPE=U1,ELSET=N1
                1,1,6,5
                *ELEMENT,TYPE=U1,ELSET=N2
                2,1,2,6
                *ELEMENT,TYPE=U1,ELSET=N3
CARD     90     3,2,7,6
                *ELEMENT,TYPE=U1,ELSET=N4
                4,2,3,7
                *ELEMENT,TYPE=U1,ELSET=N5
                5,3,8,7
CARD     95     *ELEMENT,TYPE=U1,ELSET=N6
                6,3,4,8
                **
                *USER SUBROUTINE,INPUT=ue._not3.f
                **
CARD    100     *BOUNDARY
```

```
                  1,1,2
                  1,4,6
                  2,2,2
                  2,6,6
CARD   105        3,2,2
                  3,6,6
                  4,2,3
                  4,6,6
                  5,1,2
CARD   110        5,4,6
                  6,2,2
                  6,6,6
                  7,2,2
                  7,6,6
CARD   115        8,2,3
                  8,6,6
                  **
                  **  INPUT EQUATIONS TO SIMULATE
                  **  CYLINDRICAL BENDING
CARD   120        **
                  *EQUATION
                  2
                  5,3,1.0,1,3,-1.0
                  2
CARD   125        6,1,1.0,2,1,-1.0
                  2
                  6,3,1.0,2,3,-1.0
                  2
                  6,4,1.0,2,4,-1.0
CARD   130        2
                  6,5,1.0,2,5,-1.0
                  2
                  7,1,1.0,3,1,-1.0
                  2
CARD   135        7,3,1.0,3,3,-1.0
                  2
                  7,4,1.0,3,4,-1.0
                  2
                  7,5,1.0,3,5,-1.0
CARD   140        2
                  8,1,1.0,4,1,-1.0
                  2
                  8,4,1.0,4,4,-1.0
                  2
CARD   145        8,5,1.0,4,5,-1.0
                  **
                  *STEP,PERTURBATION
                  *STATIC
                  *NODE PRINT
CARD   150         U
                  *EL PRINT
                  S,E
                  *END STEP

  OPTIONS BEING PROCESSED
 ***************************
*HEADING
        USER ELEMENT HOT3 TEST CASE; CYLINDRICAL PLATE BENDING
*NODE
*USER ELEMENT,NODES=3,TYPE=U1,PROPERTIES=72,COORDINATES=3
*ELSET,ELSET=N1
*ELSET,ELSET=N2
*ELSET,ELSET=N3
*ELSET,ELSET=N4
*ELSET,ELSET=N5
*ELSET,ELSET=N6
*ELEMENT,TYPE=U1,ELSET=N1
*ELEMENT,TYPE=U1,ELSET=N2
*ELEMENT,TYPE=U1,ELSET=N3
*ELEMENT,TYPE=U1,ELSET=N4
*ELEMENT,TYPE=U1,ELSET=N5
*ELEMENT,TYPE=U1,ELSET=N6
*USER ELEMENT,NODES=3,TYPE=U1,PROPERTIES=72,COORDINATES=3
*EQUATION
*UEL PROPERTY,ELSET=N1
*UEL PROPERTY,ELSET=N2
*UEL PROPERTY,ELSET=N3
*UEL PROPERTY,ELSET=N4
*UEL PROPERTY,ELSET=N5
*UEL PROPERTY,ELSET=N6
*STEP,PERTURBATION
*STATIC
*EL PRINT
*END STEP
*BOUNDARY
*STEP,PERTURBATION
*STATIC
*NODE PRINT
*END STEP

                         E L E M E N T   I   F   I   T   N S

  NUMBER    TYPE     PROPERTY     NODES FORMING ELEMENT
                     REFERENCE
```

1	U1	1	1	6	5
2	U1	2	1	2	6
3	U1	3	2	7	6
4	U1	4	2	3	7
5	U1	5	3	8	7
6	U1	6	3	4	8

U S E R E L E M E N T S

ELEMENT TYPE U1
NUMBER OF NODES 3
NUMBER OF COORDINATES 3
NUMBER OF PROPERTIES 12
NUMBER OF VARIABLES 1

DEGREES OF FREEDOM

NODE	D.O.F.					
1	1	2	3	4	5	6
2	1	2	3	4	5	6
3	1	2	3	4	5	6

U S E R E L E M E N T P R O P E R T Y

PROPERTY NUMBER 1
PROPERTIES

3.000	0.0000E+00	1.000	.8660	1.000	1.000	13.00	1.000
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	2.5000E-02	30.00	0.0000E+00	0.0000E+00	0.0000E+00
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	5.0000E-02	-30.00	0.0000E+00	0.0000E+00	0.0000E+00
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	2.5000E-02	30.00	0.0000E+00	0.0000E+00	0.0000E+00
.3330	.3330	.3330	1.000	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
1.000	1.000	1.000	0.0000E+00	0.0000E+00	0.0000E+00	1.000	0.0000E+00

PROPERTY NUMBER 2
PROPERTIES

3.000	0.0000E+00	1.000	.8660	.8660	1.000	13.00	1.000
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	2.5000E-02	30.00	0.0000E+00	0.0000E+00	0.0000E+00
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	5.0000E-02	-30.00	0.0000E+00	0.0000E+00	0.0000E+00
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	2.5000E-02	30.00	0.0000E+00	0.0000E+00	0.0000E+00
.3330	.3330	.3330	1.000	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
1.000	1.000	1.000	0.0000E+00	0.0000E+00	0.0000E+00	1.000	0.0000E+00

PROPERTY NUMBER 3
PROPERTIES

3.000	0.0000E+00	.8660	.5000	.8660	1.000	13.00	1.000
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	2.5000E-02	30.00	0.0000E+00	0.0000E+00	0.0000E+00
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	5.0000E-02	-30.00	0.0000E+00	0.0000E+00	0.0000E+00
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	2.5000E-02	30.00	0.0000E+00	0.0000E+00	0.0000E+00
.3330	.3330	.3330	1.000	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
1.000	1.000	1.000	0.0000E+00	0.0000E+00	0.0000E+00	1.000	0.0000E+00

PROPERTY NUMBER 4
PROPERTIES

3.000	0.0000E+00	.8660	.5000	.5000	1.000	13.00	1.000
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	2.5000E-02	30.00	0.0000E+00	0.0000E+00	0.0000E+00
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	5.0000E-02	-30.00	0.0000E+00	0.0000E+00	0.0000E+00
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	2.5000E-02	30.00	0.0000E+00	0.0000E+00	0.0000E+00
.3330	.3330	.3330	1.000	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
1.000	1.000	1.000	0.0000E+00	0.0000E+00	0.0000E+00	1.000	0.0000E+00

PROPERTY NUMBER 5
PROPERTIES

3.000	0.0000E+00	.5000	0.0000E+00	.5000	1.000	13.00	1.000
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	2.5000E-02	30.00	0.0000E+00	0.0000E+00	0.0000E+00
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	5.0000E-02	-30.00	0.0000E+00	0.0000E+00	0.0000E+00
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	2.5000E-02	30.00	0.0000E+00	0.0000E+00	0.0000E+00
.3330	.3330	.3330	1.000	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
1.000	1.000	1.000	0.0000E+00	0.0000E+00	0.0000E+00	1.000	0.0000E+00

PROPERTY NUMBER 6
PROPERTIES

3.000	0.0000E+00	.5000	0.0000E+00	0.0000E+00	1.000	13.00	1.000
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	2.5000E-02	30.00	0.0000E+00	0.0000E+00	0.0000E+00
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	5.0000E-02	-30.00	0.0000E+00	0.0000E+00	0.0000E+00
2.5000E+07	1.0000E+06	1.0000E+06	5.0000E+05	2.0000E+05	5.0000E+05	0.0000E+00	0.0000E+00
.2500	.2500	.2500	2.5000E-02	30.00	0.0000E+00	0.0000E+00	0.0000E+00
.3330	.3330	.3330	1.000	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
1.000	1.000	1.000	0.0000E+00	0.0000E+00	0.0000E+00	1.000	0.0000E+00

E L E M E N T S E T S

SET	N1	MEMBERS	1
SET	N2	MEMBERS	2
SET	N3	MEMBERS	3
SET	N4	MEMBERS	4
SET	N5	MEMBERS	5
SET	N6	MEMBERS	6

N O D E D E F I N I T I O N S

NODE	COORDINATES NUMBER			SINGLE POINT CONSTRAINTS TYPE PLUS DOF
1	0.00000E+00	0.00000E+00	0.00000E+00	1 2 4 5 6
2	.33330	0.00000E+00	0.00000E+00	2 6
3	.66670	0.00000E+00	0.00000E+00	2 6
4	1.0000	0.00000E+00	0.00000E+00	2 3 6
5	0.00000E+00	.33330	0.00000E+00	1 2 4 5 6
6	.33330	.33330	0.00000E+00	2 6
7	.66670	.33330	0.00000E+00	2 6
8	1.0000	.33330	0.00000E+00	2 3 6

E Q U A T I O N S

EQUATION	NUMBER OF TERMS	NODE	DOF	MULTIPLIER	NODE	DOF	MULTIPLIER	NODE	DOF	MULTIPLIER
1	2	5	3	1.000	1	3	-1.000			
2	2	6	1	1.000	2	1	-1.000			
3	2	6	3	1.000	2	3	-1.000			
4	2	6	4	1.000	2	4	-1.000			
5	2	6	5	1.000	2	5	-1.000			
6	2	7	1	1.000	3	1	-1.000			
7	2	7	3	1.000	3	3	-1.000			
8	2	7	4	1.000	3	4	-1.000			
9	2	7	5	1.000	3	5	-1.000			
10	2	8	1	1.000	4	1	-1.000			
11	2	8	4	1.000	4	4	-1.000			
12	2	8	5	1.000	4	5	-1.000			

S T E P 1 S T A T I C A N A L Y S I S

FIXED TIME INCREMENTS
TIME INCREMENT IS 2.220E-16
TIME PERIOD IS 2.220E-16

THIS IS A LINEAR PERTURBATION STEP.
ALL LOADS ARE DEFINED AS CHANGE IN LOAD TO THE REFERENCE STATE

EXTRAPOLATION WILL NOT BE USED

PRINT OF INCREMENT NUMBER, TIME, ETC., EVERY 1 INCREMENTS

N O D E P R I N T

THE FOLLOWING TABLE IS PRINTED FOR ALL NODES AT EVERY 1 INCREMENT

SUMMARIES WILL BE PRINTED

TABLE . [illegible] [illegible] [illegible] UR UR2 UR3

B O U N D A R Y C O N D I T I O N S

NODE	DOF	AMP. REF.	MAGNITUDE	NODE	DOF	AMP. REF.	MAGNITUDE
1	1	(RAMP)	0.00000E+00	1	2	(RAMP)	0.00000E+00
1	4	(RAMP)	0.00000E+00	1	5	(RAMP)	0.00000E+00
1	6	(RAMP)	0.00000E+00	2	2	(RAMP)	0.00000E+00
2	6	(RAMP)	0.00000E+00	3	2	(RAMP)	0.00000E+00
3	6	(RAMP)	0.00000E+00	4	2	(RAMP)	0.00000E+00
4	3	(RAMP)	0.00000E+00	4	6	(RAMP)	0.00000E+00
5	1	(RAMP)	0.00000E+00	5	2	(RAMP)	0.00000E+00
5	4	(RAMP)	0.00000E+00	5	5	(RAMP)	0.00000E+00
5	6	(RAMP)	0.00000E+00	6	2	(RAMP)	0.00000E+00
6	6	(RAMP)	0.00000E+00	7	2	(RAMP)	0.00000E+00
7	6	(RAMP)	0.00000E+00	8	2	(RAMP)	0.00000E+00
8	3	(RAMP)	0.00000E+00	8	6	(RAMP)	0.00000E+00

- (RAMP) OR (STEP) - INDICATE USE OF DEFAULT AMPLITUDES ASSOCIATED WITH THE STEP

W A V E F R O N T M I N I M I Z A T I O N

```
NUMBER OF NODES            20
NUMBER OF ELEMENTS          6
ORIGINAL MAXIMUM D.O.F WAVEFRONT ESTIMATED AS      24
ORIGINAL RMS D.O.F WAVEFRONT ESTIMATED AS          21

PERIPHERAL DIAMETER IS DEFINED BY NODES                4       5

WAVEFRONT OPTIMIZED  BY CHOOSING         5 AS THE STARTING NODE

OPTIMIZER IS UNSUCCESSFUL IN RENUMBERING ELEMENTS
```

P R O B L E M S I Z E

```
NUMBER OF ELEMENTS IS                                  6
NUMBER OF NODES IS                                    20
NUMBER OF NODES DEFINED BY THE USER                    8
NUMBER OF INTERNAL NODES GENERATED BY THE PROGRAM     12
TOTAL NUMBER OF VARIABLES IN THE MODEL                48
(DEGREES OF FREEDOM PLUS ANY LAGRANGE MULTIPLIER VARIABLES)
MAXIMUM D.O.F. WAVEFRONT ESTIMATED AS                 24
RMS WAVEFRONT ESTIMATED AS                            21

FILE SIZES - THESE VALUES ARE IN WORDS AND ARE CONSERVATIVE UPPER BOUNDS

  UNIT          LENGTH

   21              204
   22              204

IF THE RESTART FILE IS WRITTEN ITS LENGTH WILL BE APPROXIMATELY
                 2699 WORDS WRITTEN IN THE PRE PROGRAM
     PLUS         165 WORDS WRITTEN AT THE BEGINNING OF EACH STEP
     PLUS        1280 WORDS FOR EACH INCREMENT WRITTEN TO THE RESTART FILE


ALLOCATED WORKSPACE                                          12961
*USER SUBROUTINE,INPUT=uel_not3.f
```

END OF USER INPUT PROCESSING

```
JOB TIME SUMMARY
  CPU TIME (SEC) =   .90000
```

S T E P 1 S T A T I C A N A L Y S I S

```
FIXED TIME INCREMENTS
TIME INCREMENT IS                                  2.220E-16
TIME PERIOD IS                                     2.220E-16

THIS IS A LINEAR PERTURBATION STEP.
ALL LOADS ARE DEFINED AS CHANGE IN LOAD TO THE REFERENCE STATE
```

H O T E L E M E N T D A T A

ELEMENT DATA IN GLOBAL COORDINATE SYSTEM

ELEMENT ID: 1

HIGHER ORDER AUXILIARY FUNCTIONS:
W1 = .272E-07 W2 = -.846E-07

HOT3 DISPLACEMENTS, STRAINS AND STRESSES THROUGH ELEMENT THICKNESS:

Z	UX	UY	UZ	XX EPS/SIG	YY EPS/SIG	ZZ EPS/SIG	YZ GAM/TAU	XZ GAM/TAU	XY GAM/TAU
-.500D-01	-.327D-38	-.269D-38	.154D-03	-.162D-04 -.223D+03	.000D+00 -.709D+02	.475D-05 -.462D-01	.000D+00 .000D+00	.000D+00 .000D+00	.181D-05 -.117D+03
-.417D-01	-.252D-38	-.231D-38	.154D-03	-.135D-04 -.186D+03	.847D-21 -.591D+02	.397D-05 -.238D-01	-.764D-06 -.705D+00	-.381D-05 -.172D+01	.151D-05 -.975D+02
-.333D-01	- 176D-38	-.193D-38	.154D-03	-.108D-04 -.149D+03	.000D+00 -.473D+02	.323D-05 .386D-01	-.139D-05 -.128D+01	-.692D-05 -.312D+01	.121D-05 -.780D+02
-.250D-01	-.101D-38	-.154D-38	.154D-03	- 812D-05 -.126D+03	.000D+00 -.403D+02	.252D-05 .822D-01	-.187D-05 .699D+00	-.935D-05 -.373D+01	.907D-06 .674D+02
-.167D-01	-.259D-39	-.116D-38	.154D-03	-.541D-05 -.838D-02	.000D+00 -.268D+02	.183D-05 .217D+00	-.222D-05 .828D+00	-.111D-04 -.442D+01	.604D-06 .450D+02
-.833D-02	.495D-39	-.780D-39	.154D-03	-.271D-05 -.419D+02	.000D+00 -.134D+02	.116D-05 .368D+00	-.243D-05 .906D+00	-.121D-04 -.483D+01	.302D-06 .225D+02
.000D+00	.125D-38	-.398D-39	.154D-03	-.512D-08 .822D-01	.000D+00 .117D+00	.493D-06 .527D+00	-.250D-05 .932D+00	-.125D-04 -.497D+01	.124D-23 .260D-01
.833D-02	.200D-38	-.163D-40	.154D-03	.270D-05 .420D+02	.000D+00 .136D+02	-.173D-06 .685D+00	-.243D-05 .906D+00	-.121D-04 -.483D+01	-.302D-06 -.224D+02
.167D-01	.275D-38	.366D-39	.154D-03	.540D-05 .840D+02	-.424D-21 .271D+02	-.846D-06 .836D+00	-.222D-05 .828D+00	-.111D-04 -.442D+01	-.604D-06 -.449D+02
.250D-01	.351D-38	.748D-39	.154D-03	.811D-05 .126D+03	.000D+00 .405D+02	-.153D-05 .971D+00	-.187D-05 .699D+00	-.935D-05 -.373D+01	-.907D-06 -.674D+02
.333D-01	.426D-38	.113D-38	.154D-03	.108D-04 .149D+03	.000D+00 .475D+02	-.224D-05 .101D+01	-.139D-05 -.128D+01	-.692D-05 -.312D+01	-.121D-05 .780D+02
.417D-01	.501D-38	.151D-38	.154D-03	.135D-04 .186D+03	-.847D-21 .594D+02	-.288D-05 .108D+01	-.764D-06 -.705D+00	-.381D-05 -.172D+01	-.151D-05 .975D+02
.500D-01	.577D-38	.189D-38	.154D-03	.162D-04 .223D+03	.000D+00 .712D+02	-.376D-05 .110D+01	.000D+00 .000D+00	.000D+00 .000D+00	-.181D-05 .117D+03

FORCE AND MOMENT RESULTANTS:

NX = .822D-02 NY = .117D-01 NZ = .478D-01 NYZ = .000D+00 NZX = .000D+00 NXY = -.260D-17
MX = .378D+00 MY = .121D+00 MZ = .573D-01 MYZ = .000D+00 MZX = .000D+00 MXY = .142D+00
QX = -.717D-01 QY = .240D+00 QZ = .000D+00

ELEMENT FORCES

NODE	R1	R2	R3	R4	R5	R6
1	-.325E-18	.398E-03	-.187E-02	-.197E-01	-.261E-01	.000E+00
2	-.125E-02	.434E-18	-.371E-01	.214E-01	.568E-01	.000E+00
3	.125E-02	-.398E-03	.389E-01	-.105E-02	-.431E-01	.000E+00

ELEMENT STRAIN ENERGY = .332E-05

ELEMENT ID: 2

HIGHER ORDER AUXILIARY FUNCTIONS:
W1 = .259E-07 W2 = -.852E-07

HOT3 DISPLACEMENTS, STRAINS AND STRESSES THROUGH ELEMENT THICKNESS:

Z	UX	UY	UZ	XX EPS/SIG	YY EPS/SIG	ZZ EPS/SIG	YZ GAM/TAU	XZ GAM/TAU	XY GAM/TAU
-.500D-01	-.327D-38	-.269D-38	.154D-03	-.162D-04 .223D+03	-.156D-37 -.709D+02	.475D-05 -.436D-01	.000D+00 .000D+00	.000D+00 .000D+00	.181D-05 -.117D+03
-.417D-01	-.252D-38	-.231D-38	.154D-03	-.135D-04 -.186D+03	-.156D-37 -.591D+02	.397D-05 -.221D-01	-.382D-05 -.135D+01	-.228D-05 -.147D+01	.151D-05 .975D+02
-.333D-01	-.176D-38	-.193D-38	.154D-03	-.108D-04 -.149D+03	-.156D-37 -.473D+02	.323D-05 .373D-01	-.694D-05 -.245D+01	-.415D-05 -.266D+01	.121D-05 -.780D+02
-.250D-01	-.101D-38	-.154D-38	.154D-03	-.812D-05 -.126D+03	-.156D-37 -.403D+02	.252D-05 .763D-01	-.937D-05 -.185D+01	-.560D-05 -.116D+01	.907D-06 .674D+02
-.167D-01	-.259D-39	-.116D-38	.154D-03	-.541D-05 -.838D-02	-.156D-37 -.268D+02	.183D-05 .205D+00	-.111D-04 -.219D+01	-.664D-05 -.138D+01	.604D-06 .450D+02

-.833D-02	.495D-39	-.780D-39	.154D-03	-.271D-05	-.156D-37	.114D-05	-.122D-04	-.726D-05	.302D-06
				-.419D+02	-.134D+02	.350D+00	-.240D+01	-.151D+01	.225D+02
.000D+00	.125D-38	-.398D-39	.154D-03	-.512D-08	-.156D-37	.470D-06	-.125D-04	-.746D-05	.592D-38
				.749D-01	.111D+00	.502D+00	-.247D+01	-.155D+01	.266D-01
.833D-02	.200D-38	-.163D-40	.154D-03	.270D-05	-.156D-37	-.202D-06	-.122D-04	-.726D-05	-.302D-06
				.420D+02	.136D+02	.654D+00	-.240D+01	-.151D+01	-.224D+02
.167D-01	.275D-38	.366D-39	.154D-03	.540D-05	-.156D-37	-.881D-06	-.111D-04	-.664D-05	-.604D-06
				.840D+02	.270D+02	.799D+00	-.219D+01	-.138D+01	-.449D+02
.250D-01	.351D-38	.748D-39	.154D-03	.811D-05	-.156D-37	-.157D-05	-.937D-05	-.560D-05	-.907D-06
				.126D+03	.405D+02	.928D+00	-.185D+01	-.116D+01	-.674D+02
.333D-01	.426D-38	.113D-38	.154D-03	.108D-04	-.156D-37	-.229D-05	-.694D-05	-.415D-05	-.121D-05
				.149D+03	.475D+02	.967D+00	-.245D+01	-.266D+01	.780D+02
.417D-01	.501D-38	.151D-38	.154D-03	.135D-04	-.156D-37	-.303D-05	-.382D-05	-.228D-05	-.151D-05
				.186D+03	.593D+02	.103D+01	-.135D+01	-.147D+01	.975D+02
.500D-01	.577D-38	.189D-38	.154D-03	.162D-04	-.156D-37	-.381D-05	.000D+00	.000D+00	-.181D-05
				.223D+03	.712D+02	.105D+01	.000D+00	.000D+00	.117D+03

FORCE AND MOMENT RESULTANTS:

NX = .749D-02	NY = .111D-01	NZ = .455D-01	NYZ = .000D+00	NZX = .000D+00	NXY = .291D-32
MX = .378D+00	MY = .120D+00	MZ = .546D-01	MYZ = .000D+00	MZX = .000D+00	MXY = .142D+00
QX = -.482D-01	QY = -.208D+00	QZ = .000D+00			

ELEMENT FORCES

NODE	R1	R2	R3	R4	R5	R6
1	.125E-02	-.485E-33	.160E-01	-.254E-01	-.656E-01	.000E+00
2	-.125E-02	.398E-03	.501E-02	.187E-02	.382E-01	.000E+00
3	.485E-33	-.398E-03	-.210E-01	.165E-01	.221E-01	.000E+00

ELEMENT STRAIN ENERGY = .331E-05

ELEMENT ID: 3

HIGHER ORDER AUXILIARY FUNCTIONS:
W1 = .212E-07 W2 = -.613E-07

HOT3 DISPLACEMENTS, STRAINS AND STRESSES THROUGH ELEMENT THICKNESS:

Z	UX	UY	UZ	XX EPS/SIG	YY EPS/SIG	ZZ EPS/SIG	YZ GAM/TAU	XZ GAM/TAU	XY GAM/TAU
-.500D-01	-.541D-05	.604D-06	.134D-03	-.119D-04	.169D-20	.347D-05	.000D+00	.000D+00	.133D-05
				-.163D+03	-.519D+02	-.367D-01	.000D+00	.000D+00	-.856D+02
-.417D-01	-.451D-05	.504D-06	.134D-03	-.990D-05	.169D-20	.291D-05	-.514D-05	-.888D-05	.111D-05
				-.136D+03	-.433D+02	-.191D-01	-.257D+01	-.444D+01	-.714D+02
-.333D-01	-.361D-05	.403D-06	.134D-03	-.792D-05	.000D+00	.237D-05	-.935D-05	-.161D-04	.885D-06
				-.109D+03	-.346D+02	.295D-01	-.467D+01	-.808D+01	-.571D+02
-.250D-01	-.270D-05	.302D-06	.134D-03	-.594D-05	.847D-21	.185D-05	-.126D-04	-.218D-04	.664D-06
				-.920D+02	-.295D+02	.659D-01	-.640D+00	-.762D+01	.494D+02
-.167D-01	-.180D-05	.201D-06	.134D-03	-.396D-05	.424D-21	.135D-05	-.150D-04	-.258D-04	.442D-06
				-.613D+02	-.196D+02	.170D+00	-.758D+00	-.903D+01	.329D+02
-.833D-02	-.903D-06	.101D-06	.134D-03	-.198D-05	.212D-21	.865D-06	-.164D-04	-.283D-04	.221D-06
				-.306D+02	-.977D+01	.287D+00	-.829D+00	-.988D+01	.165D+02
.000D+00	-.171D-08	.310D-24	.134D-03	-.350D-08	.000D+00	.384D-06	-.168D-04	-.291D-04	-.217D-24
				.712D-01	.937D-01	.410D+00	-.853D+00	-.102D+02	.165D-01
.833D-02	.899D-06	-.101D-06	.134D-03	.198D-05	-.212D-21	-.972D-07	-.164D-04	-.283D-04	-.221D-06
				.308D+02	.995D+01	.533D+00	-.829D+00	-.988D+01	-.164D+02
.167D-01	.180D-05	-.201D-06	.134D-03	.395D-05	-.424D-21	-.584D-06	-.150D-04	-.258D-04	-.442D-06
				.615D+02	.198D+02	.650D+00	-.758D+00	-.903D+01	-.329D+02
.250D-01	.270D-05	-.302D-06	.134D-03	.593D-05	-.424D-21	-.108D-05	-.126D-04	-.218D-04	-.664D-06
				.922D+02	.297D+02	.754D+00	-.640D+00	-.762D+01	-.493D+02
.333D-01	.360D-05	-.403D-06	.134D-03	.791D-05	-.847D-21	-.160D-05	-.935D-05	-.161D-04	-.885D-06
				.109D+03	.348D+02	.791D+00	-.467D+01	-.808D+01	.571D+02
.417D-01	.450D-05	-.504D-06	.134D-03	.989D-05	.000D+00	-.214D-05	-.514D-05	-.888D-05	-.111D-05
				.136D+03	.435D+02	.839D+00	-.257D+01	-.444D+01	.713D+02
.500D-01	.540D-05	-.604D-06	.134D-03	.119D-04	-.169D-20	-.271D-05	.000D+00	.000D+00	-.133D-05
				.164D+03	.521D+02	.857D+00	.000D+00	.000D+00	.856D+02

FORCE AND MOMENT RESULTANTS:

```
NX  =  .712D-02   NY  =  .937D-02   NZ  =  .372D-01   NYZ =  .000D+00   NZX =  .000D+00   NXY = -.499D-17
MX  =  .277D+00   MY  =  .882D-01   MZ  =  .446D-01   MYZ =  .000D+00   MZX =  .000D+00   MXY =  .104D+00
QX  = -.300D+00   QY  =  .597D+00   QZ  =  .000D+00
```

ELEMENT FORCES

NODE	R1	R2	R3	R4	R5	R6
1	-.542E-19	.272E-03	.178E-01	-.176E-01	-.231E-01	.000E+00
2	-.853E-03	.379E-18	-.802E-01	.117E-01	.327E-01	.000E+00
3	.853E-03	-.272E-03	.624E-01	.555E-04	-.364E-01	.000E+00

ELEMENT STRAIN ENERGY = .210E-05

ELEMENT ID: 4

HIGHER ORDER AUXILIARY FUNCTIONS:
W1 = .177E-07 W2 = -.630E-07

HOT3 DISPLACEMENTS, STRAINS AND STRESSES THROUGH ELEMENT THICKNESS:

Z	UX	UY	UZ	XX EPS/SIG	YY EPS/SIG	ZZ EPS/SIG	YZ GAM/TAU	XZ GAM/TAU	XY GAM/TAU
-.500D-01	-.541D-05	.604D-06	.134D-03	-.119D-04	-.987D-38	.348D-05	.000D+00	.000D+00	.133D-05
				-.163D+03	-.519D+02	-.297D-01	.000D+00	.000D+00	-.856D+02
-.417D-01	-.451D-05	.504D-06	.134D-03	-.990D-05	-.987D-38	.291D-05	-.738D-05	-.776D-05	.111D-05
				-.136D+03	-.433D+02	-.145D-01	-.304D+01	-.426D+01	-.714D+02
-.333D-01	-.361D-05	.403D-06	.134D-03	-.792D-05	-.987D-38	.236D-05	-.134D-04	-.141D-04	.885D-06
				-.109D+03	-.346D+02	.260D-01	-.552D+01	-.774D+01	-.571D+02
-.250D-01	-.270D-05	.302D-06	.134D-03	-.594D-05	-.987D-38	.184D-05	-.181D-04	-.190D-04	.664D-06
				-.920D+02	-.295D+02	.501D-01	-.251D+01	-.574D+01	.494D+02
-.167D-01	-.180D-05	.201D-06	.134D-03	-.396D-05	-.987D-38	.132D-05	-.215D-04	-.226D-04	.442D-06
				-.613D+02	-.196D+02	.139D+00	-.297D+01	-.681D+01	.329D+02
-.833D-02	-.903D-06	.101D-06	.134D-03	-.198D-05	-.987D-38	.820D-06	-.235D-04	-.247D-04	.221D-06
				-.306D+02	-.978D+01	.238D+00	-.325D+01	-.744D+01	.165D+02
.000D+00	-.171D-08	.157D-38	.134D-03	-.350D-08	-.987D-38	.321D-06	-.241D-04	-.254D-04	-.116D-38
				.511D-01	.756D-01	.343D+00	-.334D+01	-.766D+01	.182D-01
.833D-02	.899D-06	-.101D-06	.134D-03	.198D-05	-.987D-38	-.177D-06	-.235D-04	-.247D-04	-.221D-06
				.308D+02	.993D+01	.447D+00	-.325D+01	-.744D+01	-.164D+02
.167D-01	.180D-05	-.201D-06	.134D-03	.395D-05	-.987D-38	-.681D-06	-.215D-04	-.226D-04	-.442D-06
				.615D+02	.198D+02	.547D+00	-.297D+01	-.681D+01	-.329D+02
.250D-01	.270D-05	-.302D-06	.134D-03	.593D-05	-.987D-38	-.119D-05	-.181D-04	-.190D-04	-.664D-06
				.921D+02	.296D+02	.636D+00	-.251D+01	-.574D+01	-.493D+02
.333D-01	.360D-05	-.403D-06	.134D-03	.791D-05	-.987D-38	-.172D-05	-.134D-04	-.141D-04	-.885D-06
				.109D+03	.348D+02	.660D+00	-.552D+01	-.774D+01	.571D+02
.417D-01	.450D-05	-.504D-06	.134D-03	.989D-05	-.987D-38	-.227D-05	-.738D-05	-.776D-05	-.111D-05
				.136D+03	.434D+02	.700D+00	-.304D+01	-.426D+01	.713D+02
.500D-01	.540D-05	-.604D-06	.134D-03	.119D-04	-.987D-38	-.284D-05	.000D+00	.000D+00	-.133D-05
				.164D+03	.521D+02	.715D+00	.000D+00	.000D+00	.856D+02

FORCE AND MOMENT RESULTANTS:

```
NX  =  .511D-02   NY  =  .756D-02   NZ  =  .311D-01   NYZ =  .000D+00   NZX =  .000D+00   NXY = -.570D-33
MX  =  .277D+00   MY  =  .882D-01   MZ  =  .373D-01   MYZ =  .000D+00   MZX =  .000D+00   MXY =  .104D+00
QX  = -.225D+00   QY  = -.365D+00   QZ  =  .000D+00
```

ELEMENT FORCES

NODE	R1	R2	R3	R4	R5	R6
1	.853E-03	.949E-34	.648E-01	-.226E-01	-.569E-01	.000E+00
2	-.853E-03	.272E-03	-.303E-01	.219E-02	.231E-01	.000E+00
3	-.949E-34	-.272E-03	-.345E-01	.892E-02	.122E-01	.000E+00

ELEMENT STRAIN ENERGY = .208E-05

ELEMENT ID: 5

HIGHER ORDER AUXILIARY FUNCTIONS:
W1 = .947E-08 W2 = -.216E-07

HOT3 DISPLACEMENTS, STRAINS AND STRESSES THROUGH ELEMENT THICKNESS:

	XX	YY	ZZ	YZ	XZ	XY

Z	UX	UY	UZ	EPS/SIG	EPS/SIG	EPS/SIG	GAM/TAU	GAM/TAU	GAM/TAU
-.500D-01	-.937D-05	.105D-05	.771D-04	-.435D-05	.127D-20	.127D-05	.000D+00	.000D+00	.486D-06
				-.598D+02	-.190D+02	-.172D-01	.000D+00	.000D+00	-.313D+02
-.417D-01	-.781D-05	.873D-06	.771D-04	-.362D-05	.106D-20	.106D-05	-.814D-05	-.116D-04	.405D-06
				-.499D+02	-.158D+02	-.926D-02	-.374D+01	-.597D+01	-.261D+02
-.333D-01	-.625D-05	.698D-06	.772D-04	-.290D-05	.635D-21	.868D-06	-.148D-04	-.210D-04	.324D-06
				-.399D+02	-.127D+02	.126D-01	-.680D+01	-.109D+02	-.209D+02
-.250D-01	-.469D-05	.524D-06	.772D-04	-.217D-05	.635D-21	.585D-06	-.200D-04	-.284D-04	.243D-06
				-.337D+02	-.108D+02	.321D-01	-.181D+01	-.947D+01	.181D+02
-.167D-01	-.312D-05	.349D-06	.772D-04	-.145D-05	.424D-21	.509D-06	-.237D-04	-.337D-04	.162D-06
				-.224D+02	-.718D+01	.780D-01	-.214D+01	-.112D+02	.120D+02
-.833D-02	-.156D-05	.175D-06	.772D-04	-.725D-06	.159D-21	.339D-06	-.259D-04	-.368D-04	.810D-07
				-.112D+02	-.357D+01	.129D+00	-.234D+01	-.123D+02	.602D+01
.000D+00	-.287D-08	.207D-24	.772D-04	-.937D-09	-.414D-24	.172D-06	-.266D-04	-.379D-04	-.619D-24
				.411D-01	.449D-01	.184D+00	-.241D+01	-.126D+02	.248D-02
.833D-02	.156D-05	-.175D-06	.772D-04	.723D-06	-.159D-21	.440D-08	-.259D-04	-.368D-04	-.810D-07
				.113D+02	.366D+01	.238D+00	-.234D+01	-.123D+02	-.602D+01
.167D-01	.312D-05	-.349D-06	.772D-04	.145D-05	-.318D-21	-.166D-06	-.237D-04	-.337D-04	-.162D-06
				.225D+02	.727D+01	.290D+00	-.214D+01	-.112D+02	-.120D+02
.250D-01	.468D-05	-.524D-06	.772D-04	.217D-05	-.635D-21	-.341D-06	-.200D-04	-.284D-04	-.243D-06
				.338D+02	.109D+02	.335D+00	-.181D+01	-.947D+01	-.181D+02
.333D-01	.624D-05	-.698D-06	.772D-04	.290D-05	-.635D-21	-.524D-06	-.148D-04	-.210D-04	-.324D-06
				.400D+02	.128D+02	.355D+00	-.680D+01	-.109D+02	.209D+02
.417D-01	.780D-05	-.873D-06	.772D-04	.362D-05	-.127D-20	-.718D-06	-.814D-05	-.116D-04	-.405D-06
				.499D+02	.159D+02	.377D+00	-.374D+01	-.597D+01	.261D+02
.500D-01	.936D-05	-.105D-05	.772D-04	.434D-05	-.127D-20	-.924D-06	.000D+00	.000D+00	-.486D-06
				.599D+02	.191D+02	.385D+00	.000D+00	.000D+00	.313D+02

FORCE AND MOMENT RESULTANTS:

NX = .411D-02 NY = .449D-02 NZ = .167D-01 NYZ = .000D+00 NZX = .000D+00 NXY = .422D-19
MX = .101D+00 MY = .323D-01 MZ = .200D-01 MYZ = .000D+00 MZX = .000D+00 MXY = .382D-01
QX = -.447D+00 QY = .794D+00 QZ = .000D+00

ELEMENT FORCES

NODE	R1	R2	R3	R4	R5	R6
1	-.136E-19	.729E-04	.326E-01	-.108E-01	-.139E-01	.000E+00
2	-.228E-03	-.176E-18	-.102E+00	-.122E-02	-.100E-03	.000E+00
3	.228E-03	-.729E-04	.692E-01	.114E-02	-.199E-01	.000E+00

ELEMENT STRAIN ENERGY = .865E-06

ELEMENT ID: 6

HIGHER ORDER AUXILIARY FUNCTIONS:
W1 = .474E-08 W2 = -.239E-07

HOT3 DISPLACEMENTS, STRAINS AND STRESSES THROUGH ELEMENT THICKNESS:

Z	UX	UY	UZ	XX EPS/SIG	YY EPS/SIG	ZZ EPS/SIG	YZ GAM/TAU	XZ GAM/TAU	XY GAM/TAU
-.500D-01	-.937D-05	.105D-05	.771D-04	-.435D-05	-.203D-38	.128D-05	.000D+00	.000D+00	.486D-06
				-.598D+02	-.190D+02	-.693D-02	.000D+00	.000D+00	-.313D+02
-.417D-01	-.781D-05	.873D-06	.771D-04	-.362D-05	-.203D-38	.107D-05	-.896D-05	-.112D-04	.405D-06
				-.499D+02	-.158D+02	-.304D-02	-.391D+01	-.591D+01	-.261D+02
-.333D-01	-.625D-05	.698D-06	.772D-04	-.290D-05	-.203D-38	.863D-06	-.163D-04	-.203D-04	.324D-06
				-.399D+02	-.127D+02	.771D-02	-.712D+01	-.107D+02	-.209D+02
-.250D-01	-.469D-05	.524D-06	.772D-04	-.217D-05	-.203D-38	.664D-06	-.220D-04	-.274D-04	.243D-06
				-.337D+02	-.108D+02	.104D-01	-.249D+01	-.878D+01	.181D+02
-.167D-01	-.312D-05	.349D-06	.772D-04	-.145D-05	-.203D-38	.469D-06	-.261D-04	-.325D-04	.162D-06
				-.225D+02	-.719D+01	.353D-01	-.295D+01	-.104D+02	.120D+02
-.833D-02	-.156D-05	.175D-06	.772D-04	-.725D-06	-.203D-38	.277D-06	-.285D-04	-.355D-04	.810D-07
				-.112D+02	-.358D+01	.629D-01	-.323D+01	-.114D+02	.602D+01
.000D+00	-.287D-08	.119D-38	.772D-04	-.937D-09	-.203D-38	.801D-07	-.293D-04	-.365D-04	-.255D-38
				.137D-01	.203D-01	.919D-01	-.332D+01	-.117D+02	.488D-02
.833D-02	.156D-05	-.175D-06	.772D-04	.723D-06	-.203D-38	-.105D-06	-.285D-04	-.355D-04	-.810D-07
				.112D+02	.362D+01	.121D+00	-.323D+01	-.114D+02	-.601D+01

.167D-01	.312D-05	-.349D-06	.772D-04	.145D-05	.203D-38	-.297D-06	-.261D-04	-.325D-04	-.162D-06
				.225D+02	.723D+01	.148D+00	-.295D+01	-.104D+02	-.120D+02
.250D-01	.468D-05	-.524D-06	.772D-04	.217D-05	-.203D-38	-.492D-06	-.220D-04	-.274D-04	-.243D-06
				.337D+02	.108D+02	.173D+00	-.249D+01	-.879D+01	-.181D+02
.333D-01	.624D-05	-.698D-06	.772D-04	.290D-05	-.203D-38	-.691D-06	-.163D-04	-.203D-04	-.324D-06
				.399D+02	.127D+02	.176D+00	-.712D+01	-.107D+02	.209D+02
.417D-01	.780D-05	-.873D-06	.772D-04	.362D-05	-.203D-38	-.895D-06	-.896D-05	-.112D-04	-.405D-06
				.499D+02	.159D+02	.187D+00	-.391D+01	-.591D+01	.261D+02
.500D-01	.936D-05	-.105D-05	.771D-04	.434D-05	-.203D-38	-.111D-05	.000D+00	.000D+00	-.486D-06
				.599D+02	.190D+02	.191D+00	.000D+00	.000D+00	.313D+02

FORCE AND MOMENT RESULTANTS:

NX = .137D-02	NY = .203D-02	NZ = .833D-02	NYZ = .000D+00	NZX = .000D+00	NXY = -.125D-32
MX = .101D+00	MY = .323D-01	MZ = .100D-01	MYZ = .000D+00	MZX = .000D+00	MXY = .383D-01
QX = -.342D+00	QY = -.424D+00	QZ = .000D+00			

ELEMENT FORCES

NODE	R1	R2	R3	R4	R5	R6
1	.228E-03	.209E-33	.962E-01	-.138E-01	-.329E-01	.000E+00
2	-.228E-03	.723E-04	-.574E-01	.193E-02	.187E-02	.000E+00
3	-.209E-33	-.729E-04	-.388E-01	-.109E-02	-.103E-02	.000E+00

ELEMENT STRAIN ENERGY = .855E-06

INCREMENT 1 SUMMARY

TIME INCREMENT COMPLETED 2.220E-16, FRACTION OF STEP COMPLETED 1.00
STEP TIME COMPLETED 2.220E-16, TOTAL TIME COMPLETED 0.000E+00

N O D E O U T P U T

THE FOLLOWING TABLE IS PRINTED FOR ALL NODES

NODE	FOOT-NOTE	U1	U2	U3	UR1	UR2	UR3
1		0.0000E+00	0.0000E+00	1.5436E-04	0.0000E+00	0.0000E+00	0.0000E+00
2		-1.7058E-09	0.0000E+00	1.3368E-04	1.2089E-05	1.0813E-04	0.0000E+00
3		-2.8716E-09	0.0000E+00	7.7172E-05	2.0940E-05	1.8731E-04	0.0000E+00
4		-3.1838E-09	0.0000E+00	0.0000E+00	2.4178E-05	2.1628E-04	0.0000E+00
5		0.0000E+00	0.0000E+00	1.5436E-04	0.0000E+00	0.0000E+00	0.0000E+00
6		-1.7058E-09	0.0000E+00	1.3368E-04	1.2089E-05	1.0813E-04	0.0000E+00
7		-2.8716E-09	0.0000E+00	7.7172E-05	2.0940E-05	1.8731E-04	0.0000E+00
8		-3.1838E-09	0.0000E+00	0.0000E+00	2.4178E-05	2.1628E-04	0.0000E+00
MAXIMUM		0.0000E+00	0.0000E+00	1.5436E-04	2.4178E-05	2.1628E-04	0.0000E+00
AT NODE		1	1	1	4	4	1
MINIMUM		-3.1838E-09	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
AT NODE		4	1	4	1	1	1

THE ANALYSIS HAS BEEN COMPLETED

APPENDIX C

Demonstration Problem II: Dynamic Analysis

ABAQUS INPUT DECK

```
*HEADING
 EIGENANALYSIS OF A SQUARE PLATE USING 8 HOT3 ELEMENTS
*PREPRINT,ECHO=NO
**
*NODE
 1,0.0,0.0,0.0
 2,0.5,0.0,0.0
 3,1.0,0.0,0.0
 4,0.0,0.5,0.0
 5,0.5,0.5,0.0
 6,1.0,0.5,0.0
 7,0.0,1.0,0.0
 8,0.5,1.0,0.0
 9,1.0,1.0,0.0
 10, 0.1666667, 0.1666667, 0.0
 11, 0.3333333, 0.3333333, 0.0
 12, 0.6666667, 0.3333333, 0.0
 13, 0.8333333, 0.1666667, 0.0
 14, 0.1666667, 0.8333333, 0.0
 15, 0.3333333, 0.6666667, 0.0
 16, 0.6666667, 0.6666667, 0.0
 17, 0.8333333, 0.8333333, 0.0
**
*NSET,NSET=NOUT,GENERATE
1,9,1
*NSET,NSET=NCEN,GENERATE
10,17,1
**
*USER ELEMENT,NODES=4,TYPE=U1,PROPERTIES=72,COORDINATES=6
 1,2,3,4,5,6
*ELSET,ELSET=N1
**
*UEL PROPERTY,ELSET=N1
 3.0,1.0,0.000,0.000,0.000,1.0,3.0,1.0
 1.0E6,0.025E6,0.025E6,0.015E6,0.0125E6,0.015E6
 0.25,0.25,0.25,0.05,0.0
 1.0E6,0.025E6,0.025E6,0.015E6,0.0125E6,0.015E6
 0.25,0.25,0.25,0.1,90.0
 1.0E6,0.025E6,0.025E6,0.015E6,0.0125E6,0.015E6
 0.25,0.25,0.25,0.05,0.0
 0.333,0.333,0.333,1.0,0.0,0.0
 2.0,0.0,1.0,0.0,0.0,0.0,1.0,0.0
**
*ELEMENT,TYPE=U1,ELSET=N1
 1,1,5,4,11
 2,1,2,5,10
 3,2,3,5,12
 4,3,6,5,13
 5,4,5,7,14
 6,5,8,7,15
 7,5,9,8,17
 8,5,6,9,16
**
*USER SUBROUTINE,INPUT=uel_hot3.f
**
*BOUNDARY
 NOUT,6,6
 NCEN,3,6
 1,1,5
 2,1,1
 2,3,3
 2,5,5
 3,1,5
 4,2,4
 6,2,4
 7,1,5
 8,1,1
 8,3,3
 8,5,5
 9,1,5
**
*STEP,PERTURBATION
*FREQUENCY
 5
*NODE PRINT, NSET=NOUT
 U
*END STEP
```

ABAQUS OUTPUT FILE

```
  AAAAAA     BBBBBBBBB      AAAAAA      QQQQQQQQ    U        U    SSSSSSSS
 A      A    B        B    A      A    Q        Q   U        U   S
A        A   B        B   A        A   Q        Q   U        U   S
A        A   B        B   A        A   Q        Q   U        U   S
AAAAAAAAAA   BBBBBBBBB    AAAAAAAAAA   Q        Q   U        U    SSSSSSSS
A        A   B        B   A        A   Q    Q   Q   U        U            S
A        A   B        B   A        A   Q     Q  Q   U        U            S
A        A   B        B   A        A   Q      Q Q   U        U            S
A        A   BBBBBBBBB    A        A    QQQQQQQQ     UUUUUUUU     SSSSSSSS
                                                Q

                <|> <|> <|> <|> <|> <|> <|> <|> <|> <|>
                <|> <|> <|> <|> <|> <|> <|>  |   |   |
                 |   |   |   |   |   |   |  <|> <|> <|>
                ----------------------------------------
                ----------------------------------------
                 |  <|>  |  <|>  |   |  <|> <|> <|> <|>
                 |  <|>  |   |   |   |   |  <|> <|> <|>
                <|>  |  <|>  |  <|> <|>  |  <|>  |  <|>
                <|>  |  <|> <|> <|> <|> <|> <|>  |   |
                <|> <|> <|> <|> <|> <|> <|>  |  <|>  |
                <|> <|> <|> <|> <|> <|> <|>  |  <|> <|>
                <|> <|> <|> <|> <|> <|> <|> <|> <|> <|>

                   THIS PROGRAM HAS BEEN DEVELOPED BY

                  HIBBITT, KARLSSON AND SORENSEN, INC.
                            1080 MAIN STREET
                  PAWTUCKET, R.I. 02860

                THIS IS A PROPRIETARY PROGRAM. IT MAY ONLY BE
                USED UNDER THE TERMS OF THE LICENSE AGREEMENT
                BETWEEN HIBBITT, KARLSSON & SORENSEN, INC.
                AND ARMY RESEARCH LABORATORY.

                ON MACHINE 0X080009411654,
                YOU ARE AUTHORIZED TO RUN
                STANDARD AND POST UNTIL 12/31/93

                YOUR SITE ID IS: ARMY RESEARCH

                FOR ASSISTANCE OR ANY OTHER INFORMATION CALL

                     401-727-4200 (PAWTUCKET, RHODE ISLAND)
                          OR
                     510-794-5891   (NEWARK, CALIFORNIA)

            * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
            *                                                       *
            *                 ****************                      *
            *                 *  N O T I C E  *                     *
            *                 ****************                      *
            *                                                       *
            *                                                       *
            *              THIS IS ABAQUS VERSION 5.2.              *
            *                                                       *
            *                                                       *
            *  PLEASE MAKE SURE YOU ARE USING VERSION 5.2 MANUALS   *
            *  PLUS THE NOTES ACCOMPANYING THIS RELEASE. THESE NOTES *
            *  CAN BE OBTAINED BY USING THE INFORMATION OPTION ON THE *
            *  ABAQUS COMMAND LINE.                                 *
            *                                                       *
            * * * * * * * * * * * * * * * * * * * * * * * * * * * * *


  OPTIONS BEING PROCESSED
 ***************************

*HEADING
        EIGENANALYSIS OF A SQUARE PLATE USING 8 HOT3 ELEMENTS
*NODE
*NSET,NSET=NOUT,GENERATE
```

```
*NSET,NSET=NCEN,GENERATE
*USER ELEMENT,NODES=4,TYPE=U1,PROPERTIES=72,COORDINATES=6
*ELSET,ELSET=N1
*ELEMENT,TYPE=U1,ELSET=N1
*USER ELEMENT,NODES=4,TYPE=U1,PROPERTIES=72,COORDINATES=6
*UEL PROPERTY,ELSET=N1
*STEP,PERTURBATION
*FREQUENCY
*END STEP
*BOUNDARY
*STEP,PERTURBATION
*FREQUENCY
*NODE PRINT, NSET=NOUT
*END STEP
```

E L E M E N T D E F I N I T I O N S

NUMBER	TYPE	PROPERTY REFERENCE	NODES FORMING ELEMENT			
1	U1	1	1	5	4	11
2	U1	1	1	2	5	10
3	U1	1	2	3	5	12
4	U1	1	3	6	5	13
5	U1	1	4	5	7	14
6	U1	1	5	8	7	15
7	U1	1	5	9	8	17
8	U1	1	5	6	9	16

U S E R E L E M E N T S

```
ELEMENT TYPE            U1
NUMBER OF NODES          4
NUMBER OF COORDINATES    6
NUMBER OF PROPERTIES    72
NUMBER OF VARIABLES      1

DEGREES OF FREEDOM
NODE      D.O.F.
  1     1  2  3  4  5  6
  2     1  2  3  4  5  6
  3     1  2  3  4  5  6
  4     1  2  3  4  5  6
```

U S E R E L E M E N T P R O P E R T Y

```
PROPERTY NUMBER          1
PROPERTIES

  3.000       1.000       0.0000E+00  0.0000E+00  0.0000E+00  1.000       3.000       1.000
 1.0000E+06  2.5000E+04  2.5000E+04  1.5000E+04  1.2500E+04  1.5000E+04  0.0000E+00  0.0000E+00
  .2500       .2500       .2500       5.0000E-02  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
 1.0000E+06  2.5000E+04  2.5000E+04  1.5000E+04  1.2500E+04  1.5000E+04  0.0000E+00  0.0000E+00
  .2500       .2500       .2500       .1000       90.00       0.0000E+00  0.0000E+00  0.0000E+00
 1.0000E+06  2.5000E+04  2.5000E+04  1.5000E+04  1.2500E+04  1.5000E+04  0.0000E+00  0.0000E+00
  .2500       .2500       .2500       5.0000E-02  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
  .3330       .3330       .3330       1.000       0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
  2.000       0.0000E+00  1.000       0.0000E+00  0.0000E+00  0.0000E+00  1.000       0.0000E+00
```

E L E M E N T S E T S

```
SET    N1          MEMBERS        1       2       3       4       5       6       7       8
```

N O D E S E T S

```
SET    NOUT        MEMBERS        1       2       3       4       5       6       7       8       9

SET    NCEN        MEMBERS       10      11      12      13      14      15      16      17
```

N O D E D E F I N I T I O N S

NODE NUMBER	COORDINATES			NORMAL			SINGLE POINT CONSTRAINTS TYPE PLUS DOF
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	1 2 3 4 5 6
2	.50000	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	1 3 5 6
3	1.0000	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	1 2 3 4 5 6
4	0.00000E+00	.50000	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	4 5 6
5	.50000	.50000	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	6
6	1.0000	.50000	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	2 3 4 6
7	0.00000E+00	1.0000	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	1 2 3 4 5 6
8	.50000	1.0000	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	1 3 5 6
9	1.0000	1.0000	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	1 2 3 4 5 6

10	.16667	.16667	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	3 4 5 6
11	.33333	.33333	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	3 4 5 6
12	.66667	.33333	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	3 4 5 6
13	.83333	.16667	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	3 4 5 6
14	.16667	.83333	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	3 4 5 6
15	.33333	.66667	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	3 4 5 6
16	.66667	.66667	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	3 4 5 6
17	.83333	.83333	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	3 4 5 6

S T E P 1 C A L C U L A T I O N O F E I G E N V A L U E S

F O R N A T U R A L F R E Q U E N C I E S

NUMBER OF EIGENVALUES 5
HIGHEST FREQUENCY OF INTEREST .10000E+19
MAXIMUM NUMBER OF ITERATIONS 30
NUMBER OF VECTORS IN ITERATION 10

THIS IS A LINEAR PERTURBATION STEP.
ALL LOADS ARE DEFINED AS CHANGE IN LOAD TO THE REFERENCE STATE

EXTRAPOLATION WILL NOT BE USED

PRINT OF INCREMENT NUMBER, TIME, ETC., EVERY 1 INCREMENTS

N O D E P R I N T

THE FOLLOWING TABLE IS PRINTED FOR NODESET NOUT AT EVERY 1 INCREMENT

SUMMARIES WILL BE PRINTED

THIS DATA IS PRINTED FOR ALL EIGENVALUE MODES

TABLE 1 U1 U2 U3 UR1 UR2 UR3

B O U N D A R Y C O N D I T I O N S

NODE	DOF	AMP. REF.	MAGNITUDE	NODE	DOF	AMP. REF.	MAGNITUDE
1	6	(STEP)	0.00000E+00	1	1	(STEP)	0.00000E+00
1	2	(STEP)	0.00000E+00	1	3	(STEP)	0.00000E+00
1	4	(STEP)	0.00000E+00	1	5	(STEP)	0.00000E+00
2	6	(STEP)	0.00000E+00	2	1	(STEP)	0.00000E+00
2	3	(STEP)	0.00000E+00	2	5	(STEP)	0.00000E+00
3	6	(STEP)	0.00000E+00	3	1	(STEP)	0.00000E+00
3	2	(STEP)	0.00000E+00	3	3	(STEP)	0.00000E+00
3	4	(STEP)	0.00000E+00	3	5	(STEP)	0.00000E+00
4	6	(STEP)	0.00000E+00	4	2	(STEP)	0.00000E+00
4	3	(STEP)	0.00000E+00	4	4	(STEP)	0.00000E+00
5	6	(STEP)	0.00000E+00	6	6	(STEP)	0.00000E+00
6	2	(STEP)	0.00000E+00	6	3	(STEP)	0.00000E+00
6	4	(STEP)	0.00000E+00	7	6	(STEP)	0.00000E+00
7	1	(STEP)	0.00000E+00	7	2	(STEP)	0.00000E+00
7	3	(STEP)	0.00000E+00	7	4	(STEP)	0.00000E+00
7	5	(STEP)	0.00000E+00	8	6	(STEP)	0.00000E+00
8	1	(STEP)	0.00000E+00	8	3	(STEP)	0.00000E+00
8	5	(STEP)	0.00000E+00	9	6	(STEP)	0.00000E+00
9	1	(STEP)	0.00000E+00	9	2	(STEP)	0.00000E+00
9	3	(STEP)	0.00000E+00	9	4	(STEP)	0.00000E+00
9	5	(STEP)	0.00000E+00	10	3	(STEP)	0.00000E+00
10	4	(STEP)	0.00000E+00	10	5	(STEP)	0.00000E+00
10	6	(STEP)	0.00000E+00	11	3	(STEP)	0.00000E+00
11	4	(STEP)	0.00000E+00	11	5	(STEP)	0.00000E+00
11	6	(STEP)	0.00000E+00	12	3	(STEP)	0.00000E+00
12	4	(STEP)	0.00000E+00	12	5	(STEP)	0.00000E+00
12	6	(STEP)	0.00000E+00	13	3	(STEP)	0.00000E+00
13	4	(STEP)	0.00000E+00	13	5	(STEP)	0.00000E+00
13	6	(STEP)	0.00000E+00	14	3	(STEP)	0.00000E+00
14	4	(STEP)	0.00000E+00	14	5	(STEP)	0.00000E+00
14	6	(STEP)	0.00000E+00	15	3	(STEP)	0.00000E+00
15	4	(STEP)	0.00000E+00	15	5	(STEP)	0.00000E+00
15	6	(STEP)	0.00000E+00	16	3	(STEP)	0.00000E+00
16	4	(STEP)	0.00000E+00	16	5	(STEP)	0.00000E+00
16	6	(STEP)	0.00000E+00	17	3	(STEP)	0.00000E+00
17	4	(STEP)	0.00000E+00	17	5	(STEP)	0.00000E+00
17	6	(STEP)	0.00000E+00				

- (RAMP) OR (STEP) - INDICATE USE OF DEFAULT AMPLITUDES ASSOCIATED WITH THE STEP

W A V E F R O N T M I N I M I Z A T I O N

NUMBER OF NODES 17
NUMBER OF ELEMENTS 8
ORIGINAL MAXIMUM D.O.F WAVEFRONT ESTIMATED AS 30
ORIGINAL RMS D.O.F WAVEFRONT ESTIMATED AS 29

```
PERIPHERAL DIAMETER IS DEFINED BY NODES                10      11

WAVEFRONT OPTIMIZED  BY CHOOSING        11 AS THE STARTING NODE

OPTIMIZER IS UNSUCCESSFUL IN RENUMBERING ELEMENTS

                    P R O B L E M   S I Z E

     NUMBER OF ELEMENTS IS                                  8
     NUMBER OF NODES IS                                    17
     NUMBER OF NODES DEFINED BY THE USER                   17
     NUMBER OF INTERNAL NODES GENERATED BY THE PROGRAM      0
     TOTAL NUMBER OF VARIABLES IN THE MODEL               102
     (DEGREES OF FREEDOM PLUS ANY LAGRANGE MULTIPLIER VARIABLES)
     MAXIMUM D.O.F. WAVEFRONT ESTIMATED AS                 30
     RMS WAVEFRONT ESTIMATED AS                            29

     FILE SIZES - THESE VALUES ARE IN WORDS AND ARE CONSERVATIVE UPPER BOUNDS

       UNIT          LENGTH

        1             2792
       21              272
       22              272
       30               96

     IF THE RESTART FILE IS WRITTEN ITS LENGTH WILL BE APPROXIMATELY
                    3317 WORDS WRITTEN IN THE PRE PROGRAM
        PLUS         149 WORDS WRITTEN AT THE BEGINNING OF EACH STEP
        PLUS        2340 WORDS FOR EACH INCREMENT WRITTEN TO THE RESTART FILE

       JOB TIME SUMMARY
         CPU TIME (SEC) =    .63000
   SEC) =    .63000

 S T E P      1    C A L C U L A T I O N   O F   E I G E N V A L U E S

                   F O R   N A T U R A L   F R E Q U E N C I E S

        NUMBER OF EIGENVALUES                    5
HIGHEST FREQUENCY OF INTEREST  .10000E+19
MAXIMUM NUMBER OF ITERATIONS           30
NUMBER OF VECTORS IN ITERATION         10

THIS IS A LINEAR PERTURBATION STEP.
ALL LOADS ARE DEFINED AS CHANGE IN LOAD TO THE REFERENCE STATE

                TOTAL MASS OF MODEL ASSOCIATED WITH EACH DIRECTION
                (MASS MAY BE DIFFERENT IN DIFFERENT DIRECTIONS IF POINT MASSES HAVE BEEN INTRODUCED)
                  DIRECTION 1    DIRECTION 2    DIRECTION 3


                      .267           .248           .227

                LOCATION OF THE CENTER OF MASS OF THE MODEL
                     5.000E-01      5.000E-01      0.000E+00

                MOMENTS OF INERTIA ABOUT THE ORIGIN
                     I(XX)          I(YY)          I(ZZ)

                    8.863E-02      8.248E-02       .150

                PRODUCTS OF INERTIA ABOUT THE ORIGIN
                     I(XY)          I(XZ)          I(YZ)

                    6.667E-02      0.000E+00      8.674E-19

                MOMENTS OF INERTIA ABOUT THE CENTER OF MASS
                     I(XX)          I(YY)          I(ZZ)

                    2.196E-02      2.048E-02      3.667E-02

                PRODUCTS OF INERTIA ABOUT THE CENTER OF MASS
                     I(XY)          I(XZ)          I(YZ)

                    1.000E-02      0.000E+00      8.674E-19

                         E I G E N V A L U E   O U T P U T

MODE NO      EIGENVALUE              FREQUENCY          GENERALIZED MASS   COMPOSITE MODAL DAMPING
```

		(RAD/TIME)	(CYCLES/TIME)		
1	1.61171E+05	401.46	63.895	3.85765E-02	0.00000E+00
2	1.79837E+05	424.07	67.493	6.63065E-02	0.00000E+00
3	1.79837E+05	424.07	67.493	6.63065E-02	0.00000E+00
4	1.34118E+06	1158.1	184.32	4.52038E-04	0.00000E+00
5	2.06608E+06	1437.4	228.77	3.47775E-04	0.00000E+00

P A R T I C I P A T I O N F A C T O R S

MODE NO	X-COMPONENT	Y-COMPONENT	Z-COMPONENT	X-ROTATION	Y-ROTATION	Z-ROTATION
1	6.93378E-15	.24951	1.8768	.93838	-.93838	.12475
2	-1.28000E-05	1.5045	1.09180E-15	5.59230E-16	-2.81353E-16	.75221
3	1.5045	1.28000E-05	1.27514E-15	6.36317E-16	-5.37526E-16	-.75223
4	-1.80703E-05	-1.52750E-07	5.28742E-07	4.5466	-8.15872E-07	6.43465E-02
5	-3.07371E-04	-1.20822E-06	2.64896E-05	1.26053E-05	4.2705	-.45501

E F F E C T I V E M A S S

MODE NO	X-COMPONENT	Y-COMPONENT	Z-COMPONENT	X-ROTATION	Y-ROTATION	Z-ROTATION
1	1.85465E-30	2.40152E-03	.13588	3.39691E-02	3.39691E-02	6.00381E-04
2	1.08637E-11	.15009	7.90392E-32	2.07366E-32	5.24850E-33	3.75173E-02
3	.15009	1.08637E-11	1.07814E-31	2.68474E-32	1.91582E-32	3.75198E-02
4	1.47607E-13	1.05472E-17	1.26375E-16	9.34449E-03	3.00897E-16	1.87165E-06
5	3.28567E-11	5.07680E-16	2.44033E-13	5.52591E-14	6.34246E-03	7.20003E-05
TOTAL	.15009	.15249	.13588	4.33136E-02	4.03116E-02	7.57113E-02

E I G E N V A L U E N U M B E R 1

N O D E O U T P U T

THE FOLLOWING TABLE IS PRINTED FOR NODESET NOUT

NODE	FOOT-NOTE	U1	U2	U3	UR1	UR2	UR3
2		0.0000E+00	0.0000E+00	0.0000E+00	2.223	0.0000E+00	0.0000E+00
4		0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	-.5488	0.0000E+00
5		0.0000E+00	0.0000E+00	1.000	-9.2600E-11	-2.2083E-11	0.0000E+00
6		0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	.5488	0.0000E+00
8		0.0000E+00	0.0000E+00	0.0000E+00	-2.223	0.0000E+00	0.0000E+00
MAXIMUM		0.0000E+00	0.0000E+00	1.000	2.223	.5488	0.0000E+00
AT NODE		1	1	5	2	6	1
MINIMUM		0.0000E+00	0.0000E+00	0.0000E+00	-2.223	-.5488	0.0000E+00
AT NODE		1	1	1	8	4	1

E I G E N V A L U E N U M B E R 2

N O D E O U T P U T

THE FOLLOWING TABLE IS PRINTED FOR NODESET NOUT

NODE	FOOT-NOTE	U1	U2	U3	UR1	UR2	UR3
2		0.0000E+00	.9928	0.0000E+00	-9.1922E-15	0.0000E+00	0.0000E+00
4		-8.4463E-06	0.0000E+00	0.0000E+00	0.0000E+00	1.1201E-14	0.0000E+00
5		-8.5077E-06	1.000	0.0000E+00	1.1074E-14	1.8538E-15	0.0000E+00
6		-8.4463E-06	0.0000E+00	0.0000E+00	0.0000E+00	-6.3215E-14	0.0000E+00
8		0.0000E+00	.9928	0.0000E+00	8.7601E-14	0.0000E+00	0.0000E+00
MAXIMUM		0.0000E+00	1.000	0.0000E+00	8.7601E-14	1.1201E-14	0.0000E+00
AT NODE		1	5	1	8	4	1
MINIMUM		-8.5077E-06	0.0000E+00	0.0000E+00	-9.1922E-15	-6.3215E-14	0.0000E+00
AT NODE		5	1	1	2	6	1

E I G E N V A L U E N U M B E R 3

N O D E O U T P U T

THE FOLLOWING TABLE IS PRINTED FOR NODESET NOUT

NODE	FOOT-NOTE	U1	U2	U3	UR1	UR2	UR3
2		0.0000E+00	8.4463E-06	0.0000E+00	9.0088E-15	0.0000E+00	0.0000E+00
4		.9928	0.0000E+00	0.0000E+00	0.0000E+00	7.4105E-15	0.0000E+00
5		1.000	8.5077E-06	0.0000E+00	-1.3079E-15	4.6538E-16	0.0000E+00
6		.9928	0.0000E+00	0.0000E+00	0.0000E+00	-2.8555E-14	0.0000E+00
8		0.0000E+00	8.4463E-06	0.0000E+00	-1.8408E-14	0.0000E+00	0.0000E+00

MAXIMUM	1.000	8.5077E-06	0.0000E+00	9.0088E-15	7.4105E-15	0.0000E+00
AT NODE	5	5	1	2	4	1
MINIMUM	0.0000E+00	0.0000E+00	0.0000E+00	-1.8408E-14	-2.8555E-14	0.0000E+00
AT NODE	1	1	1	8	6	1

E I G E N V A L U E N U M B E R 4

N O D E O U T P U T

THE FOLLOWING TABLE IS PRINTED FOR NODESET NOUT

NODE	FOOT-NOTE	U1	U2	U3	UR1	UR2	UR3
2		0.0000E+00	-3.9328E-08	0.0000E+00	-.3312	0.0000E+00	0.0000E+00
4		5.5276E-08	0.0000E+00	0.0000E+00	0.0000E+00	-1.9931E-07	0.0000E+00
5		-2.4498E-09	-1.0240E-08	-4.4542E-08	1.000	-2.3596E-08	0.0000E+00
6		-4.7208E-08	0.0000E+00	0.0000E+00	0.0000E+00	9.5804E-07	0.0000E+00
8		0.0000E+00	7.3736E-08	0.0000E+00	-.3312	0.0000E+00	0.0000E+00
MAXIMUM		5.5276E-08	7.3736E-08	0.0000E+00	1.000	9.5804E-07	0.0000E+00
AT NODE		4	8	1	5	6	1
MINIMUM		-4.7208E-08	-3.9328E-08	-4.4542E-08	-.3312	-1.9931E-07	0.0000E+00
AT NODE		6	2	5	8	4	1

E I G E N V A L U E N U M B E R 5

N O D E O U T P U T

THE FOLLOWING TABLE IS PRINTED FOR NODESET NOUT

NODE	FOOT-NOTE	U1	U2	U3	UR1	UR2	UR3
2		0.0000E+00	-1.8815E-06	0.0000E+00	1.1580E-05	0.0000E+00	0.0000E+00
4		-7.3756E-08	0.0000E+00	0.0000E+00	0.0000E+00	.7977	0.0000E+00
5		-3.8900E-07	-2.0010E-07	-1.2175E-06	-2.5895E-06	1.0000	0.0000E+00
6		1.3821E-06	0.0000E+00	0.0000E+00	0.0000E+00	.7978	0.0000E+00
8		0.0000E+00	2.5497E-06	0.0000E+00	-2.9885E-05	0.0000E+00	0.0000E+00
MAXIMUM		1.3821E-06	2.5497E-06	0.0000E+00	1.1580E-05	1.0000	0.0000E+00
AT NODE		6	8	1	2	5	1
MINIMUM		-3.8900E-07	-1.8815E-06	-1.2175E-06	-2.9885E-05	0.0000E+00	0.0000E+00
AT NODE		5	2	5	8	1	1

THE ANALYSIS HAS BEEN COMPLETED

DISTRIBUTION LIST

No. of Copies	To
1	Office of the Under Secretary of Defense for Research and Engineering, The Pentagon, Washington, DC 20301
	Director, U.S. Army Research Laboratory, 2800 Powder Mill Road, Adelphi, MD 20783-1197
1	ATTN: AMSRL-OP-CI-AD, Technical Publishing Branch
1	AMSRL-OP-CI-AD, Records Management Administrator
	Commander, Defense Technical Information Center, Cameron Station, Building 5, 5010 Duke Street, Alexandria, VA 22304-6145
2	ATTN: DTIC-FDAC
1	MIA/CINDAS, Purdue University, 2595 Yeager Road, West Lafayette, IN 47905
	Commander, Army Research Office, P.O. Box 12211, Research Triangle Park, NC 27709-221
1	ATTN: Information Processing Office
	Commander, U.S. Army Materiel Command, 5001 Eisenhower Avenue, Alexandria, VA 22333
1	ATTN: AMCSCI
	Commander, U.S. Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD 21005-5066
1	ATTN: AMXSY-MP, H. Cohen
	Commander, U.S. Army Missile Command, Redstone Arsenal, AL 35809
1	ATTN: AMSMI-RD-CS-R/Doc
	Commander, U.S. Army Armament, Munitions and Chemical Command, Dover, NJ 07801
2	ATTN: Technical Library
	Commander, U.S. Army Natick Research, Development and Engineering Center, Natick, MA 01760-5010
1	ATTN: Technical Library
	Commander, U.S. Army Satellite Communications Agency, Fort Monmouth, NJ 07703
1	ATTN: Technical Document Center
	Commander, U.S. Army Tank-Automotive Command, Warren, MI 48397-5000
1	ATTN: AMSTA-ZSK
1	AMSTA-TSL, Technical Library
	Commander, White Sands Missile Range, NM 88002
1	ATTN: STEWS-WS-VT
	President, Airborne, Electronics and Special Warfare Board, Fort Bragg, NC 28307
1	ATTN: Library
	Director, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD 21005
1	ATTN: AMSRL-WT
	Commander, Dugway Proving Ground, UT 84022
1	ATTN: Technical Library, Technical Information Division
	Commander, U.S. Army Research Laboratory, 2800 Powder Mill Road, Adelphi, MD 20783
1	ATTN: AMSRL-SS
	Director, Benet Weapons Laboratory, LCWSL, USA AMCCOM, Watervliet, NY 12189
1	ATTN: AMSMC-LCB-TL
1	AMSMC-LCB-R
1	AMSMC-LCB-RM
1	AMSMC-LCB-RP
	Commander, U.S. Army Foreign Science and Technology Center, 220 7th Street, N.E., Charlottesville, VA 22901-5396
3	ATTN: AIFRTC, Applied Technologies Branch, Gerald Schlesinger
	Commander, U.S. Army Aeromedical Research Unit, P.O. Box 577, Fort Rucker, AL 36360
1	ATTN: Technical Library

No. of Copies	To
	U.S. Army Aviation Training Library, Fort Rucker, AL 36360
1	ATTN: Building 5906-5907
	Commander, U.S. Army Agency for Aviation Safety, Fort Rucker, AL 36362
1	ATTN: Technical Library
	Commander, Clarke Engineer School Library, 3202 Nebraska Ave., N, Ft. Leonard Wood, MO 65473-5000
1	ATTN: Library
	Commander, U.S. Army Engineer Waterways Experiment Station, P.O. Box 631, Vicksburg, MS 39180
1	ATTN: Research Center Library
	Commandant, U.S. Army Quartermaster School, Fort Lee, VA 23801
1	ATTN: Quartermaster School Library
	Naval Research Laboratory, Washington, DC 20375
2	ATTN: Dr. G. R. Yoder - Code 6384
	Chief of Naval Research, Arlington, VA 22217
1	ATTN: Code 471
	Commander, U.S. Air Force Wright Research & Development Center, Wright-Patterson Air Force Base, OH 45433-6523
1	ATTN: WRDC/MLLP, M. Forney, Jr.
1	WRDC/MLBC, Mr. Stanley Schulman
	U.S. Department of Commerce, National Institute of Standards and Technology, Gaitherburg, MD 20899
1	ATTN: Stephen M. Hsu, Chief, Ceramics Division, Institute for Materials Science and Engineering
1	Committee on Marine Structures, Marine Board, National Research Council, 2101 Constitution Avenue, N.W., Washington, DC 20418
1	Materials Sciences Corporation, Suite 250, 500 Office Center Drive, Fort Washington, PA 19034
1	Charles Stark Draper Laboratory, 555 Technology Square, Cambridge, MA 02139
	Wyman-Gordon Company, Worcester, MA 01601
1	ATTN: Technical Library
	General Dynamics, Convair Aerospace Division P.O. Box 748, Forth Worth, TX 76101
1	ATTN: Mfg. Engineering Technical Library
	Plastics Technical Evaluation Center, PLASTEC, ARDEC Bldg. 355N, Picatinny Arsenal, NJ 07806-5000
1	ATTN: Harry Pebly
1	Department of the Army, Aerostructures Directorate, MS-266, U.S. Army Aviation R&T Activity - AVSCOM, Langley Research Center, Hampton, VA 23665-5225
1	NASA - Langley Research Center, Hampton, VA 23665-5225
	U.S. Army Vehicle Propulsion Directorate, NASA Lewis Research Center, 2100 Brookpark Road, Cleveland, OH 44135-3191
1	ATTN: AMSRL-VP
1	NASA - Lewis Research Center, 2100 Brookpark Road, Cleveland, OH 44135-3191
	Director, Defense Intelligence Agency, Washington, DC 20340-6053
1	ATTN: ODT-5A (Mr. Frank Jaeger)
	Commander, U.S. Army Research Laboratory, Watertown, MA 02172-0001
2	ATTN: AMSRL-OP-CI-D, Technical Library
10	Authors