CHAPTER - 6

COMPUTER PROGRAM DEVELOPMENT AND VALIDITY
DEVELOPMENT OF COMPUTER PROGRAM

6.0 GENERAL

A finite element program which provides the stress analysis of composite structures is presented.

The present software (finite element program) is based on a displacement solution. The various solution techniques are active column solution or the skyline method, Cholesky factorization, static condensation, substructures, and frontal solution. The present software uses a front solver technique by Irons and Ahmad to reduce the amount of nodes considered at the same time. This technique implies that during the solving process, the program looks for nodes which will not be involved in other elements.

The equations pertaining to such nodes are then extracted from the global stiffness matrix and stored for later back-substitution. The stiffness matrices are evaluated by Gauss-Quadrature integration technique. The most active part of the equations are kept in computer memory allowing for a flexible limit on the wavefront's size. The bulk of the required storage space is kept on a hard disk.

The F.E. program was designed to handle stress analysis of composite structures while working with plane stress, plane strain or plate bending elements. Stresses are evaluated at the gauss points of the elements (Gauss-Quadrature locations).

The FORTRAN-77 language was chosen over BASIC, C, Pascal for its clarity and structure. FORTRAN version 5.0 was the selected compiler for its execution, speed, unlimited code size and spectrum of available functions.

6.1 FLOW-CHART FOR COMPUTER PROGRAM

The flow diagram is a technique of systematically representing the computational process such as operations and decisions involved in the process and of the order in which these must be executed. The more complete the diagram, the easier the job of actually writing the program.
Hence a flow-chart is a means of communication among people who use computers.

The detailed flow-chart of the developed computer program is presented.
The computer program COMP.FOR has about 817 lines and 12 subroutines and is given in Annexure 6.1.

The function of each subroutine is described below

6.1.1 SUBROUTINE INPUT

Subroutine INPUT inputs the control data, geometric data and boundary conditions.

In addition, it calls subroutine NODEXY whose function is to generate the co-ordinates of mid-side nodes which lie on a straight line connecting adjacent corner nodes.

It also calls subroutine GAUSSQ whose function is to generate the sampling point positions and weighting factors according to order of integration rule specified through NGAUS in the control data.

6.1.2 SUBROUTINE NODEXY

Subroutine NODEXY calculates the co-ordinates of mid-side nodes which lie on a straight line connecting two adjacent corner nodes.

6.1.3 SUBROUTINE GAUSSQ

Subroutine GAUSSQ sets up the sampling point positions and weighting factors according to the order of integration rule specified through NGAUS in the control data.

6.1.4 SUBROUTINE STIFPC

Subroutine STIFPC calculates stiffness matrix for each element.
6.1.5 SUBROUTINE MODPC

Subroutine MODPC calculates the components of the matrix of elastic rigidities $D$ from the element material properties.

6.1.6 SUBROUTINE SFR2

Subroutine SFR2 calculates the shape functions and their derivatives at any sampling point usually the gauss point within an element.

6.1.7 SUBROUTINE JACOB2

Subroutine JACOB2 calculates the co-ordinates of the Gauss points, the Jacobian matrix, the inverse of Jacobian matrix and the cartesian shape function derivatives.

6.1.8 SUBROUTINE BMATPC

Subroutine BMATPC calculates the strain matrix $B$ at any point within the element using the shape functions and their cartesian derivatives.

6.1.9 SUBROUTINE DBE

Subroutine DBE multiplies matrix $D$ by matrix $B$.

6.1.10 SUBROUTINE LOADPC

Subroutine LOADPC calculates the element consistent nodal forces.

6.1.11 SUBROUTINE FRONT

Subroutine FRONT calculates nodal displacements, nodal rotations and reactions at restrained nodes.
6.1.12 SUBROUTINE STREPC

Subroutine STREPC calculates the stresses at the Gauss points after the displacements have been evaluated.

6.2 MANUAL FOR COMPUTER PROGRAM

Descriptive manual is provided for the program. Herein it is described how to label the laminated composite plate and how to prepare data file line by line. With the help of manual, one can write data files without error and execute them to get accurate results. Those who do not even know finite element method and computer programming, can analyse laminated composite plate with the help of this manual.

6.2.1 LABELING THE ELEMENTS

Once the structure to be analysed has been discretised into number of finite elements, the structural geometry must be defined numerically.

Number the elements in the order in which it is intended to process them. An attempt being necessary to optimize this order so as to minimize frontwidth.

6.2.2 NUMBERING NODES

The nodal points can be numbered in a random manner.

6.2.3 PREPARATION OF DATA FILE

The data file for a problem consists of following data cards:

6.2.3.1 DATA CARD 1

This data card will include total number of nodes, total number of elements, total number of restrained boundary points, number of nodes per element, number of degrees of freedom per
node, order of integration formula for numerical integration, number of coordinate dimensions, number of independent generalized stress components, thickness of laminated composite plate in the form NPOIN, NELEM, NVFIX, NNODE, NDOFN, NGAUS, NDIME, NSTRE, TH.

The variables are:

- **NPOIN**: Total number of nodes (nodal points)
- **NELEM**: Total number of elements
- **NVFIX**: Total number of restrained boundary points where one or more degrees of freedom are restrained
- **NNODE**: Number of nodes per element
- **NDOFN**: Number of degrees of freedom per node
- **NGAUS**: Order of integration formula for numerical integration
- **NDIME**: Number of coordinate dimensions
- **NSTRE**: Number of independent generalized stress components
- **TH**: Thickness of laminated composite plate

### 6.2.3.2 DATA CARD 2

This data card will include element number and corresponding node numbers in the form NUMEL, (LNODS(NUMEL,INODE), INODE = 1, NNODE). Total number of lines in this data card will equal to number of elements. The specification of the nodal connection numbers must follow a systematic pattern. An anti-clockwise sequence is adhered to, beginning from any corner node.

The variables are:

- **NUMEL**: Element Number
- **LNODS (NUMEL, 1)**: 1st nodal connection number
LNODS (NUMEL, 2): 2nd nodal connection number

LNODS (NUMEL, 8): 8th nodal connection number

6.2.3.3 DATA CARD 3

This data card will include nodal point number, co-ordinates of node in the form IPOIN, (COORD(IPOIN, IDIME), IDIME=1, NDIME)

Total number of lines in this data card will equal to node numbers whose co-ordinates are to be input in the program.

The co-ordinates of only corner nodes are given as input data when sides of element are represented by straight line. In this case, mid-side nodal co-ordinates are automatically generated. Hence the total number of lines will equal the corner nodes. The co-ordinates of the highest numbered node must be input, regardless of whether it is a mid-side node or not.

The variables are:

IPOIN : Nodal point number
COORD (IPOIN, 1) : X- coordinate of node
COORD (IPOIN, 2) : Y- coordinate of node

6.2.3.4 DATA CARD 4

This data card will include restrained node number, conditions of restraints and prescribed values of nodal displacements and rotations in the form
NOFIX(IVFIX),(IFPRE(IVFIX,IDOFN),IDOFN=1,NDOFN),(PRESC(IVFIX,IDOFN), IDOFN=1, NDOFN).

Total number of lines in this card will equal to number of restrained nodes.

The variables are:
NOFIX (IVFIX) : Restained Node number
IFPRE (IVFIX, 1) : Condition of restraint on nodal displacement, u
 0  No displacement restraint
 1  Nodal displacement restrained
IFPRE (IVFIX, 2) : Condition of restraint on nodal displacement, v
 0  No displacement restraint
 1  Nodal displacement restrained
IFPRE (IVFIX, 3) : Condition of restraint on nodal displacement, w
 0  No displacement restraint
 1  Nodal displacement restrained
IFPRE (IVFIX, 4) : Condition of restraint on nodal rotation, θx
 0  No rotation restraint
 1  Nodal rotation restrained
IFPRE (IVFIX, 5) : Condition of restraint on nodal rotation, θy
 0  No rotation restraint
 1  Nodal rotation restrained
PRES (IVFIX, 1) : Prescribed value of nodal displacement, u
PRES (IVFIX, 2) : Prescribed value of nodal displacement, v
PRES (IVFIX, 3) : Prescribed value of nodal displacement, w
PRES (IVFIX, 4) : Prescribed Value of nodal rotation, θx
PRES (IVFIX, 5) : Prescribed value of nodal rotation, θy

6.2.3.5 DATA CARD 5

This data card is divided into three parts. 1st part will include element number, number of layers in the element in the form N, N-LAYER (IELEM) . 2nd part will include Z - coordinates of layers in the element w.r.t. mid-plane in the form (H(k), k=1, LTMP) (Ref. Fig. 6.1) and 3rd part
will include layer number, various material properties in the form N, EL (IL), ET (IL), VLT (IL) VTL (IL), GLT (IL), THETA (IL), G13 (IL), G23 (IL).

The angle THETA i.e. $\theta$ is positive when angle of the L-T axes measured from X-Y axes is in the counter clockwise direction (Ref. Fig. 6.2).

This data card will be repeated number of times equal to number of elements.

The variables in 1st part are:

- N : Element number
- NLAYER (IELEM) : Number of layers in element
Fig 6.1 Z-CO-ORDINATES OF LAYERS W.R.T. MID-PLANE

Fig 6.2 ORIENTATION OF PRINCIPAL MATERIAL AXES WITH THE REFERENCE CO-ORDINATE AXES
The variables in 2nd part are:

- **H (k)**: Z - coordinates of layers of element w.r.t. mid-plane
  
  \( k=1, \) LTMP and LTMP = NLAYER+1

The variables in the 3rd part are:

- **N**: Layer number
- **EL(IL)**: Young's modulus of Elasticity in longitudinal direction L
- **ET(IL)**: Young's modulus of Elasticity in transverse direction T
- **VLT(IL)**: Poisson's ratio with load in longitudinal direction L
- **VTL(IL)**: Poisson's ratio with load in transverse direction T
- **GLT(IL)**: Inplane shear modulus of layer
- **THETA(IL)**: Orientation of principal material axes with the reference coordinate axes (Ref. Fig. 6.2)
- **G13(IL)**: Transverse shear modulus
- **G23(IL)**: Transverse shear modulus

### 6.2.3.6 DATA CARD 6

This data card will include applied nodal load control parameter in the form IPLOD.

The variable is:

- **IPLOD**: Applied nodal load control parameter
  
  - 0: No applied nodal loads to be input
  - 1: Applied nodal loads to be input

### 6.2.3.7 DATA CARD 7

This data card will include node number, nodal load components and nodal couples in the form LODPT, (POINT (IDOFN), IDOFN=1, NDOFN)
The number of lines in this card will equal to number of nodes whose applied load values are to be input. The last line should be that for the highest numbered node whether it is loaded or not.

The variables are:

- **LODPT** : Node number
- **POINT (1)** : Load component in the x-direction
- **POINT (2)** : Load component in the y-direction
- **POINT (3)** : Load component in the z-direction
- **POINT (4)** : Nodal couple in the xz plane
- **POINT (5)** : Nodal couple in the yz plane

6.2.3.8 DATA CARD 8

This data card will include value of uniformly distributed load on an element in the form **UDLOD**.

Total number of lines in this data card will equal to number of elements.

The variables is:

- **UDLOD** : Value of uniformly distributed load on an element
VALIDITY OF COMPUTER PROGRAM

6.3 TEST PROBLEMS

Whenever a computer program is developed, it is customary to check its validity. For testing validity of the program, some problems available in the literature or problems whose solution can be obtained by classical method are executed by the program. The results obtained by executing the program are compared with those available in the literature or calculated by classical method. A good comparison establishes the validity of the program.

To test the validity of the program, nature of the problems selected from the literature should be such that it addresses the special features of program developed. Therefore in the present studies, a simple problem of cantilever beam made up of isotropic material and some problems of laminated plates presented by Pandya and Kant in the literature have been executed. A good comparison of the results establishes validity of the program developed.

6.3.1 TEST PROBLEM 1

Analyse the cantilever beam shown in Fig.6.3

Area of X-section of beam = 0.3 x 0.6 = 0.18 m^2
M.O.I. of beam = 5.4 x 10^3 m^4
E = 8 x 10^7 N/m^2
v = 0.25

i) Vertical reactions are - 2264.53 N, 14110.7 N and -2267.26 N

Sum of all vertical reactions = -2264.53 + 14110.7 - 2267.26
= 9578.915 N

Vertical loading on the beam = 8000 x 0.3 x 4.
= 9600 N

Hence it satisfies the condition \( \sum V = 0 \)
Fig. 6.3 CANTILEVER BEAM
ii) Moments obtained at fixed end are 3694.52 N-m, 11748 N-m and 3695.34 N-m

Sum of obtained moments at fixed end = 3694.52 + 11748 + 3695.34
= 19137.86 N-m

Calculated moment at fixed end = 8000 x 0.3 x 4²/2
= 19200 N-m

Hence it satisfies the condition $\Sigma M = 0$

iii) Horizontal reactions obtained in X and Y directions are 0 and 0

Sum of horizontal reactions = 0

Horizontal loading on beam = 0

Hence it satisfies the condition $\Sigma H = 0$

iv) Computed deflection at free end = $\frac{(180.351 + 180.344 + 180.344)}{3}$
= 180.346 mm

Calculated deflection at free end = 182.78 mm

\[ a) \text{ due to bending} = \frac{wL^4}{8EI} = \frac{8000 \times 0.3 \times 4^4}{8 \times 8 \times 10^7 \times 5.4 \times 10^3} = 0.17778 \text{ m} = 177.78 \text{ mm} \]

\[ b) \text{ due to shearing} = \frac{3pl^2}{2(2AG)} = \frac{3 \times 8000 \times 0.3 \times 4^2 \times 2(1+.25)}{2 \times 2 \times 0.3 \times 0.6 \times 8 \times 10^7} = 0.005 \text{ m} = 5 \text{ mm} \]

v) Computed slope at free end = $\frac{wL^3}{6EI}$

Calculated slope at free end = $\frac{0.0587849 + 0.0588612 + 0.0587847}{3}$
= 0.0588102
The values of deflection and slope at free end of the cantilever beam computed from program and calculated by classical method are presented in table 6.1.

Table 6.1 Deflection and Slope at free end of cantilever beam under uniform loading

<table>
<thead>
<tr>
<th></th>
<th>DEFLECTION</th>
<th>SLOPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computed value</td>
<td>180.346 mm</td>
<td>0.0588102</td>
</tr>
<tr>
<td>Value calculated by classical method</td>
<td>182.78 mm</td>
<td>0.0592592</td>
</tr>
<tr>
<td>Percentage Difference</td>
<td>- 1.35</td>
<td>- 0.76</td>
</tr>
</tbody>
</table>

Note: Percentage difference is w. r. t computed values.

Conclusions

Since the results obtained satisfy the conditions of statical equilibrium, hence the validity of the program is established.

Since percentage difference between computed values and values calculated by classical method for deflection and slope is 1.35 and 0.76 which is very less.

Hence the validity of computer program is established.

6.3.2 TEST PROBLEMS FROM LITERATURE

To further substantiate the suitability of the program, the problems reported in the literature were studied and it was decided to solve a few problems reported by Pandya and Kant. The material properties etc. were taken as given in their paper.

The material properties considered for test problems 2 to 5 are:

$$\frac{E_1}{E_3} = \frac{E_2}{E_3} = 25, \quad \frac{G_{12}}{G_{13}} = \frac{G_{23}}{E_3} = 0.5, \quad \frac{G_{23}}{E_3} = 0.2$$
\[ v_{12} = v_{23} = v_{31} = 0.25 \]

The deflections represented are nondimensionalized with the help of following multiplying factor for uniform pressure \( p_0 \):

\[ m_1 = \frac{100 \ h^3 \ E_2}{p_0 \ a^4} \]

Multiplying factor for central point load is obtained by replacing \( p_0 \ a^2 \) by \( P \).

\( E_2 = E_3 = 10^6 \ \text{N/m}^2 \). The uniform pressure \( p_0 \) and central point load are taken 10000 N/m² and 10000 N respectively.

Simply supported and clamped boundary conditions often occur in many practical situations.

Similarly uniform pressure and central point load are commonly occurring loading cases.

Numerical results involving these boundary conditions and loading cases are not available in the literature.

With a view to provide numerical results for future references, Kant and present investigator considered a square laminated cross-ply plate \((0^\circ/90^\circ/0^\circ)\) made up of three equally thick layers involving these boundary conditions and loading cases.

### 6.3.2.1 TEST PROBLEM 2

A clamped three layer cross-ply \((0^\circ/90^\circ/0^\circ)\) square laminate under uniform pressure is considered for comparison of maximum deflection.

The finite element mesh is shown in Fig. 6.4

\( a \) is side of square laminate

\( h \) is thickness of laminate
Fig. 6.4 FINITE ELEMENT MESH FOR TEST PROBLEM 2 AND TEST PROBLEM 3
Case I:

\[ a = 1.2 \text{ m}, \quad h = 0.3 \text{ m} \]

\[ m_1 = \frac{100 \times (0.3)^3 \times 10^6}{10000 \times (1.2)^4} = \frac{10^5}{768} \]

The data file PC20.DAT and output file PC20.RES for case I obtained by executing the program is given in appendix 6.1 as specimen file.

The nondimensionalized deflection at centre by present theory:

\[ \frac{0.0151845 \times 10^5}{768} = 0.019771484 \]

(ii) by Kant^9 = 1.8891

Case II:

\[ a = 3 \text{ m}, \quad h = 0.3 \text{ m} \]

\[ m_1 = \frac{100 \times (0.3)^3 \times 10^6}{10000 \times (3)^4} = \frac{10}{3}, \quad w_0 \times m_1 = \frac{0.144294 \times 10}{3} = 0.48098 \]

The nondimensionalized deflection at centre by present = 0.48098

The nondimensionalized deflection at centre by Kant^9 = 0.5247
Case III

\[
a = 30 \text{ m}, \ h = 0.3 \text{ m}
\]

\[
m_1 = \frac{100 \times (3)^3 \times 10^6}{10000 \times (30)^4} = \frac{10^8 \times 3^3 \times 1}{10^4 \times 10^3 \times 3^4 \times 10^4} = \frac{1}{3000}
\]

The nondimensionalized deflection at centre by Kant\(^9\) = 0.1421

The nondimensionalized deflection at centre by present = \[
\frac{343.465 \times 1}{3000} = 0.1144883
\]

The comparison of nondimensionalized deflections at centre is represented in table 6.2 for three cases.

**Table 6.2 Central deflections in a clamped three layer cross-ply \((0º/90º/0º)\) square laminate under uniform pressure**

<table>
<thead>
<tr>
<th>a/h →</th>
<th>4</th>
<th>10</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kant(^9)</td>
<td>1.8891</td>
<td>0.5247</td>
<td>0.1421</td>
</tr>
<tr>
<td>Mindlin Theory</td>
<td>1.9203</td>
<td>0.4829</td>
<td>0.1388</td>
</tr>
<tr>
<td>Present</td>
<td>1.97715</td>
<td>0.48098</td>
<td>0.1145</td>
</tr>
<tr>
<td>Percentage Difference w.r.t Kant(^9)</td>
<td>+ 4.45</td>
<td>- 9.09</td>
<td>- 24.1</td>
</tr>
<tr>
<td>Percentage Difference w.r.t Mindlin Theory</td>
<td>+ 2.88</td>
<td>- 0.4</td>
<td>- 21.22</td>
</tr>
</tbody>
</table>

Note: Percentage difference is w. r. t present results
Conclusions

Effect of plate side to thickness ratio \( \frac{a}{h} \) on central deflection for a clamped square laminate under uniform pressure has been studied.

For thick \( \left( \frac{a}{h} = 4 \right) \) plates, the present theory gives 4.45 percent higher value in comparison to Kant\(^2\) theory and 2.88 percent higher value in comparison to Mindlin theory.

For moderately thick \( \left( \frac{a}{h} = 10 \right) \) plates, the present theory gives 0.4 and 9.09 percent lower values in comparison to Mindlin and Kant theories.

For thin \( \left( \frac{a}{h} = 100 \right) \) plates, the present theory gives 21.22 and 24.1 percent lower values in comparison to Mindlin and Kant theories.

For thick and moderately thick plates, present formulation gives nearly same results as given by Mindlin theory.

6.3.2.2 TEST PROBLEM 3

A clamped three layer cross-ply \((0^\circ/90^\circ/0^\circ)\) square laminate under central concentrated load is considered for comparison of maximum deflection.

The finite element mesh is shown in Fig. 6.4

The central point load is taken = 10000 N

The deflections are represented nondimensionalized with the help of following multiplying factor:

\[
m_i = \frac{100 h^3 E_2}{P a^2}
\]

Case I:

\[
\frac{a}{h} = 4
\]

\[a = 1.2 \text{ m}, \ h = 0.3 \text{ m}
\]

\[
m_1 = \frac{100 \times (0.3)^3 \times 10^6}{10000 \times (1.2)^2} = \frac{3000}{16}
\]

The nondimensionalized deflection at centre = \(w_0 \times m_1\)
Case II:

\[ a = 10, \quad a = 3 \text{m}, \quad h = 0.3 \text{ m} \]

\[
100 \times (0.3)^3 \times 10^6
\]

\[
m_1 = \frac{1}{10000 \times (3)^2} = 30
\]

The nondimensionalized deflection at centre = \( w_0 \times m_1 \)

\[
= 0.103592 \times 30 = 3.10776
\]

Case III:

\[ a = 100 \]

\[ a = 30 \text{ m}, \quad h = 0.3 \text{ m} \]

\[
100 \times (0.3)^3 \times 10^6
\]

\[
m_1 = \frac{3}{10000 \times (30)^2} = 10
\]

The nondimensionalized deflection at centre = \( w_0 \times m_1 \)

\[
= 2.02394 \times \frac{3}{10} = 0.607182
\]

The comparison of nondimensionalized deflections at centre is represented in table 6.3 for three cases.
Table 6.3 Central deflections in a clamped three layer cross-ply (0°/90°/0°) square laminate under central point load

<table>
<thead>
<tr>
<th>Source</th>
<th>a/h</th>
<th>4</th>
<th>10</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kant⁹</td>
<td>4</td>
<td>19.5659</td>
<td>4.0740</td>
<td>0.8478</td>
</tr>
<tr>
<td>Mindlin Theory</td>
<td>10</td>
<td>14.1203</td>
<td>3.1840</td>
<td>0.8186</td>
</tr>
<tr>
<td>Present</td>
<td>100</td>
<td>14.2958</td>
<td>3.1078</td>
<td>0.6072</td>
</tr>
<tr>
<td>Percentage Difference w.r.t Kant⁹</td>
<td>4</td>
<td>-36.86</td>
<td>-31.09</td>
<td>-39.62</td>
</tr>
<tr>
<td>Percentage Difference w.r.t Mindlin Theory</td>
<td>10</td>
<td>+1.23</td>
<td>-2.45</td>
<td>-34.82</td>
</tr>
</tbody>
</table>

Note: Percentage difference is w.r.t present results.

Conclusions

Effect of plate side to thickness ratio (a/h) on central deflection for a clamped square laminate under central point load has been studied.

For thick (a/h = 4) plates, the present theory gives 1.23 percent higher value in comparison to Mindlin theory and 36.86 percent lower value in comparison to Kant⁹ theory.

For moderately thick (a/h = 10) plates, the present theory gives 2.45 percent and 31.09 percent lower values in comparison to Mindlin and Kant theories respectively.

For thin (a/h = 100) plates, the present theory gives 34.82 and 39.62 percent lower values in comparison to Mindlin and Kant theories respectively.

Thus for thick plates and for moderately thick plates, present formulation gives nearly same results as given by Mindlin theory.
6.3.2.3 TEST PROBLEM 4

A simply supported three layer cross-ply (0°/90°/0°) square laminate under uniform pressure is considered.

The finite element mesh is shown in Fig. 6.5

Case I:
\[ \frac{a}{h} = 4 \]

Nondimensionalized deflection at centre by present
\[ \frac{10^3}{768} = 0.0187348 \times \frac{10^3}{768} = 2.439427 \]

Nondimensionalized deflection at centre by Kant\(^9\) = 2.8765

Case II:
\[ \frac{a}{h} = 10 \]

Nondimensionalized deflection at centre
\[ \frac{10}{3} = 0.230196 \times \frac{10}{3} = 0.76732 \]
Fig. 6.5 FINITE ELEMENT MESH FOR TEST PROBLEM 4 AND TEST PROBLEM 5
Case III: \( a = 100 \)

The nondimensionalized deflection at centre

\[
\frac{l}{3000} = 0.133889 \times 10^4 \times \frac{3000}{3000} = 0.4463
\]

The comparison of nondimensionalized deflections at centre is represented in table 6.4 for three cases.
Table 6.4: Central deflections in a simply supported three layer cross-ply (0°/90°/0°) square laminate under uniform pressure.

<table>
<thead>
<tr>
<th>a/h</th>
<th>Source</th>
<th>4</th>
<th>10</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Present</td>
<td>2.4394</td>
<td>0.7673</td>
<td>0.4463</td>
</tr>
<tr>
<td></td>
<td>Kant⁹</td>
<td>2.8765</td>
<td>1.0968</td>
<td>0.6713</td>
</tr>
<tr>
<td></td>
<td>Mindlin Theory</td>
<td>2.6559</td>
<td>1.0211</td>
<td>0.6701</td>
</tr>
<tr>
<td></td>
<td>Higher-order</td>
<td>2.9091</td>
<td>1.0900</td>
<td>0.6705</td>
</tr>
<tr>
<td></td>
<td>shear deformation theory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>First -order</td>
<td>2.6597</td>
<td>1.0220</td>
<td>0.6697</td>
</tr>
<tr>
<td></td>
<td>shear deformation theory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Percentage Difference w.r.t Kant⁹</td>
<td>- 17.92</td>
<td>- 42.94</td>
<td>- 50.41</td>
</tr>
<tr>
<td></td>
<td>Percentage Difference w.r.t Mindlin Theory</td>
<td>- 8.88</td>
<td>- 33.08</td>
<td>- 50.15</td>
</tr>
<tr>
<td></td>
<td>Percentage Difference w.r.t HSDT</td>
<td>- 19.25</td>
<td>- 42.06</td>
<td>- 50.24</td>
</tr>
<tr>
<td></td>
<td>Percentage Difference w.r.t FSDT</td>
<td>- 9.03</td>
<td>- 33.19</td>
<td>- 50.06</td>
</tr>
</tbody>
</table>

Note: Percentage difference is w.r.t present results.
Conclusions

Effect of plate side to thickness ratio \( \frac{a}{h} \) on central deflection for a simply supported square laminate under uniform pressure has been studied.

For thick \( \frac{a}{h} = 4 \), moderately thick \( \frac{a}{h} = 10 \) and thin \( \frac{a}{h} = 100 \) plates, present formulation provides lower values of deflection in comparison to other theories.

6.3.2.4 TEST PROBLEM 5

A simply supported three layer cross-ply \((0\degree/90\degree/0\degree)\) square laminate is considered under central point load.

The finite element mesh is shown in Fig. 6.5

Case I:

\[
\frac{a}{h} = 4
\]

Nondimensionalized deflection at centre

\[
\frac{3000}{16} = 0.0814851 \times 15.2785
\]

Case II:

\[
\frac{a}{h} = 10
\]

Nondimensionalized deflection at centre

\[
0.123217 \times 30 = 3.6965
\]

Case III:

\[
\frac{a}{h} = 100
\]
The comparison of nondimensionalized deflections at centre is represented in table 6.5 for three cases.

**Table 6.5 Central deflections in a simply supported three layer cross-ply (0°/90°/0°) square laminate under Central point load**

<table>
<thead>
<tr>
<th>Source</th>
<th>4</th>
<th>10</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kant</td>
<td>21.7089</td>
<td>5.3434</td>
<td>2.1593</td>
</tr>
<tr>
<td>Mindlin Theory</td>
<td>15.6905</td>
<td>4.3989</td>
<td>2.1177</td>
</tr>
<tr>
<td>Present</td>
<td>15.2785</td>
<td>3.6965</td>
<td>1.4146</td>
</tr>
<tr>
<td>Percentage Difference w.r.t Kant</td>
<td>-42.09</td>
<td>-44.55</td>
<td>-52.64</td>
</tr>
<tr>
<td>Percentage Difference w.r.t Mindlin Theory</td>
<td>-2.697</td>
<td>-19.0</td>
<td>-49.70</td>
</tr>
</tbody>
</table>

Note: Percentage difference is w.r.t present results

**Conclusions**

Effect of plate side to thickness ratio (a/h) on central deflection for a simply supported square laminate under central point load has been studied.

For thick (a/h = 4) plates, present theory gives nearly same results as given by Mindlin theory where-as present theory differs by 42.09 percent in comparison to Kant theory.
The results given by present formulation differs for moderately thick and thin plates in comparison to Mindlin and Kant theories.

6.4 CONCLUSIONS

The flow-chart and manual of computer program has been presented.

The validity of computer program has been established by analysing cantilever beam. With a view to provide numerical results for future references, a square laminated cross-ply plate made up of three equally thick layers involving simply supported and clamped boundary conditions and uniform pressure and central point load cases has been analysed and effect of plate side to thickness ratio on central deflection has been studied.

For thick \( \frac{a}{h} = 4 \) plates, results obtained by present formulation match with Mindlin theory and differs with Kant\(^9\) theory.

For moderately thick and thin plates, present formulation provides different results in comparison to Mindlin and Kant\(^9\) theories.
C PROGRAM FOR ANALYSIS OF LAMINATED COMPOSITE PLATES
COMMON/CONTROL/NPOIN, NELEM, NNODE, NDOFN, NDIME, NSTRE, TH,
1 NGAUS, NVFIX, NEVAB
COMMON/LGDATA/COORD(833,2), PRESC(66,5), ASDIS(4165),
1 ELOAD(256,40), NOPIX(66), IFPRE(66,5), LNO(256,8)
COMMON/WORK/ELCOD(2,8), SHAPE(8), DERIV(2,8), DMATX(8,8),
1 CAR(2,8), DBMATX(8,40), SMATX(8,40,4),
1 POSTG(2), WEGP(2), GPCOD(2,4), ESTIF(40,40)
OPEN(1,FILE='COM161.DAT', STATUS='OLD')
OPEN(2,FILE='R2.RES', STATUS='NEW')
OPEN(3,FILE='COM161.RES', STATUS='NEW')
OPEN(4,FILE='R4.RES', STATUS='NEW')
C CALL THE SUBROUTINE WHICH READS MOST OF
C THE PROBLEM DATA
C
CALL INPUT
C***** NEXT CREATE THE ELEMENT STIFFNESS FILE
CALL STIFPC
C****** COMPUTE LOADS, AFTER READING THE RELEVANT
C EXTRA DATA
CALL LOADPC
C****** MERGE AND SOLVE THE RESULTING EQUATIONS
C BY THE FRONTAL SOLVER
CALL FRONT
C****** COMPUTE THE STRESSES IN ALL THE ELEMENTS
CALL STREPC
STOP
END
SUBROUTINE INPUT
COMMON/CONTROL/NPOIN, NELEM, NNODE, NDOFN, NDIME, NSTRE, TH,
1 NGAUS, NVFIX, NEVAB
COMMON/LGDATA/COORD(833,2), PRESC(66,5), ASDIS(4165),
1 ELOAD(256,40), NOPIX(66), IFPRE(66,5), LNO(256,8)
COMMON/WORK/ELCOD(2,8), SHAPE(8), DERIV(2,8), DMATX(8,8),
1 CAR(2,8), DBMATX(8,40), SMATX(8,40,4),
1 POSTG(2), WEGP(2), GPCOD(2,4), ESTIF(40,40)
****** READ THE FIRST DATA CARD
C
READ(1,*)NPOIN, NELEM, NNODE, NDOFN, NDIME, NSTRE, TH
NEVAB=NDDOF*NNODE
WRITE(3,905)NPOIN, NELEM, NVFIX, NNODE, NDOFN,
1 NGAUS, NDIME, NSTRE, TH
905 FORMAT(/8H NPOIN =, I4, 4X, 8H NELEM =, I4, 4X, 8H NVFIX =, I4,
1 4X, 8H NDOFN =, I4, 4X, 8H NGAUS =, I4, 4X, 8H NDIME =, I4, 4X,
1 8H NSTRE =, I4, 4X, 8H NEVAB =, I4, 4X, 12H THICKNESS =, F10.4)
C***** READ THE ELEMENT NODAL CONNECTION NUMBERS
WRITE(3,910)
910 FORMAT(/9H ELEMENT, 10X, 12H NODAL NUMBERS)
DO 10 IELEM=1, NELEM
READ(1,*)NUMEL, (LNO(IELEM, INODE), INODE=1, NNODE)
10 WRITE(3,915)NUMEL, (LNO(IELEM, INODE), INODE=1, NNODE)
915 FORMAT(1X, I5, 6X, 8I5)
C**** ZERO ALL THE NODAL COORDINATES, PRIOR
C TO READING SOME OF THEM
DO 20 IPOINT=1, NPOIN
DO 20 IDIME=1, NDIME
20 COORD(IPOINT,IDIME)=0.0
C**** READ SOME NODAL COORDINATES, FINISHING WITH THE
LAST NODE OF ALL
WRITE(3,920)
920 FORMAT(//24H NODAL POINT COORDINATES)
WRITE(3,925)
925 FORMAT(6H NODE,9X,1HX,9X,1HY)
30 READ(1,*) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
IF(IPOIN .NE. NPOIN) GO TO 30
C**** INTERPOLATE COORDINATES OF MID-SIDE NODES
CALL NODEXY
DO 50 IPOIN=1,NPOIN
50 WRITE(3,935) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
935 FORMAT(1X,15,2F10.3)
C**** READ THE FIXED VALUES
WRITE(3,940)
940 FORMAT(//17H RESTRAINED NODES)
WRITE(3,945)
945 FORMAT(6H NODE,2X,4HCODE,12X,12HFIXED VALUES)
DO 80 IVFIX=1,NVFIX
READ(1,*) NOFIX(IVFIX),(IFPRE(IVFIX, IDOFN),IDOFN=1,NDOFN),(PRESC(IVFIX, 
1 IDOFN),IDOFN=1,NDOFN)
80 WRITE(3,955) NOFIX(IVFIX),(IFPRE(IVFIX, 
1 IDOFN),IDOFN=1,NDOFN),(PRESC(IVFIX, 
1 IDOFN),IDOFN=1,NDOFN)
955 FORMAT(1X,14,2X,5I1,5F10.6)
C*** SET UP GAUSSIAN INTEGRATION CONSTANTS
CALL GAUSSQ
RETURN
END

SUBROUTINE NODEXY
COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,TH,
1 NGAUS,NVFIX,NEVAB
COMMON/LGDATA/COORD(833,2),PRESC(66,5),ASDIS(4165),
1 ELOAD(256,40),NOFIX(66),(IFPRE(IVFIX, 
1 IDOFN),IDOFN=1,NDOFN),(PRESC(IVFIX, 
1 IDOFN),IDOFN=1,NDOFN)
80 WRITE(3,955) NOFIX(IVFIX),(IFPRE(IVFIX, 
1 IDOFN),IDOFN=1,NDOFN),(PRESC(IVFIX, 
1 IDOFN),IDOFN=1,NDOFN)
955 FORMAT(1X,14,2X,5I1,5F10.6)
C*** LOOP OVER EACH ELEMENT
DO 30 IELEM=1,NELEM
C*** LOOP OVER EACH ELEMENT EDGE
DO 20 INODE=1,NNODE,2
COMPUTE THE NODE NUMBER OF THE FIRST NODE
NODST=LNODS(IELEM,INODE)
IGASH=INODE+2
IF(IGASH .GT. NNODE) IGASH= 1
C*** COMPUTE THE NODE NUMBER OF THE LAST NODE
NODFN= LNODS(IELEM,IGASH)
MIDPT=INODE+1
C*** COMPUTE THE NODE NUMBER OF THE INTERMEDIATE NODE
NODMD=LNODS(IELEM,MIDPT)
TOTAL=ABS(COORD(NODMD,1)) + ABS(COORD(NODMD,2))
C*** IF THE COORDINATES OF THE INTERMEDIATE NODE ARE BOTH ZERO INTERPOLATE BY A STRAIGHT LINE
IF(TOTAL .GT. 0.0) GO TO 20
KOUNT = 1
10 COORD(NODMD, KOUNT) = (COORD(NODST, KOUNT) + COORD(NODFN, KOUNT)) / 2.
KOUNT=KOUNT+1
SUBROUTINE GAUSSQ
COMMON/CONTRO/NPOIN, NELEM, NNODE, NDOFN, NDIME, NSTRE, TH,
1 NGAUS, NVFIX, NEVAB
COMMON/LGDATA/COORD(833, 2), PRESC(66, 5), ASDIS(4165),
1 ELOAD(256, 40), NOFIX(66), IFPRE(66, 5), LNODS(256, 8)
COMMON/WORK/ELCOD(2, 8), SHAPE(8), DERIV(2, 8), DMATX(8, 8),
1 CARTD(2, 8), DMBAT(8, 40), BMATX(8, 40), SMATX(8, 40, 4),
1 POSGP(2), WEIGP(2), GPCOD(2, 4), ESTIF(40, 40)
IF(NGAUS .GT. 2) GO TO 10
POSGP(1) = 0.577350269189626
WEIGP(1) = 1.0
GO TO 20
10 POSGP(1) = -0.774596669241483
POSGP(2) = 0.0
WEIGP(1) = 0.555555555555556
WEIGP(2) = 0.888888888888889
20 KGAUS = NGAUS/2
DO 30 IGASH = 1, KGAUS
JGASH = NGAUS + 1 - IGASH
POSGP(JGASH) = -POSGP(IGASH)
WEIGP(JGASH) = WEIGP(IGASH)
30 CONTINUE
RETURN
END

SUBROUTINE STIFPC
COMMON/CONTRO/NPOIN, NELEM, NNODE, NDOFN, NDIME, NSTRE, TH,
1 NGAUS, NVFIX, NEVAB
COMMON/LGDATA/COORD(833, 2), PRESC(66, 5), ASDIS(4165),
1 ELOAD(256, 40), NOFIX(66), IFPRE(66, 5), LNODS(256, 8)
COMMON/WORK/ELCOD(2, 8), SHAPE(8), DERIV(2, 8), DMATX(8, 8),
1 CARTD(2, 8), DMBAT(8, 40), BMATX(8, 40), SMATX(8, 40, 4),
1 POSGP(2), WEIGP(2), GPCOD(2, 4), ESTIF(40, 40)
CALCULATES ELEMENT STIFFNESS MATRIX
WRITE(3, 441)
FORMAT(/23H NO. OF LAYERS , ,2X,13HZ-COORDINATES,
1 2X,5HW.R.T,2X,9HMID-PLANE/8H AND,2X,8HMATERIAL,2X,
1 10HPROPERTIES,2X,2HOF,2X,6HLAYERS,2X,2HIN,2X,3HTHE,2X,
1 3HTHE,2X,8HELEMENTS)
LOOP OVER EACH ELEMENT
DO 70 IELEM = 1, NELEM
EVALUATE THE COORDINATES OF THE
C ELEMENT NODAL POINTS
DO 10 INODE = 1, NNODE
LNODE = LNODS(IELEM, INODE)
DO 10 IDIME = 1, NDIME
ELCOD(IDIME, INODE) = COORD(LNODE, IDIME)
10 CONTINUE
CALCULATES ELEMENT STIFFNESS MATRIX
DO 20 IEVAB = 1, NEVAB
DO 20 JEVAB = 1, NEVAB
88
ESTIF(IEVAB, JEVAB) = 0.0

20 CONTINUE
CALL MODPC(IELEM)
KGASP = 0

C*** ENTER LOOPS FOR NUMERICAL INTEGRATION
DO 50 IGAUS = 1, NGAUS
EXISP = POSGP(IGAUS)
DO 50 JGAUS = 1, NGAUS
ETASP = POSGP(JGAUS)
KGASP = KGASP + 1

C*** EVALUATE THE SHAPE FUNCTIONS,
C ELEMENTAL AREA, ETC.
CALL SF2R(EXISP, ETASP)
CALL JACOB2 (IELEM, DJACB, KGASP)
DAREA = DJACB * WEIGP(IGAUS) * WEIGP(JGAUS)

C*** EVALUATE THE B AND DB MATRICES
CALL BMATPC
CALL DBE

C*** CALCULATE THE ELEMENT STIFFNESS
DO 30 IEVAB = 1, NEVAB
DO 30 JEVAB = IEVAB, NEVAB
DO 30 ISTRE = 1, NSTRE
ESTIF(IEVAB, JEVAB) = ESTIF(IEVAB, JEVAB) +
1 BMATX(ISTRE, IEVAB) * DBMAT(ISTRE, JEVAB) * DAREA
30 CONTINUE

C*** STORE THE COMPONENTS OF THE DB MATRIX
FOR THE ELEMENT
DO 40 ISTRE = 1, NSTRE
DO 40 IEVAB = 1, NEVAB
SMATX(ISTRE, IEVAB, KGASP) = DBMAT(ISTRE, IEVAB)
40 CONTINUE
50 CONTINUE

C*** CONSTRUCT THE LOWER TRIANGLE
OF THE STIFFNESS MATRIX
DO 60 IEVAB = 1, NEVAB
DO 60 JEVAB = 1, NEVAB
ESTIF(JEVAB, IEVAB) = ESTIF(IEVAB, JEVAB)
60 CONTINUE

C STORE STRESS MATRIX AND SAMPLING POINT
COORDINATES FOR EACH ELEMENT
WRITE (4, *) SMATX, GPCOD
70 CONTINUE
RETURN
END

SUBROUTINE MODPC(IELEM)

DIMENSION A(3, 3), B(3, 3), C(3, 3), D(3, 3), Q(3, 3), H(15)
DIMENSION E(256), ET(256), VLT(256), VTL(256), GLT(256), THETA(256)
DIMENSION NLayer(256), G13(256), G23(256), E(2, 2), F(2, 2)

COMMON/CONTRO/NPOIN, NELEM, NNODE, NDOFN, NDIME, NSTRE, TH,
1 NGAUS, NVFIX, NEVAB
COMMON/LDATA/COORD(833, 2), PRESC(66, 5), ASDIS(4165),
1 ELOAD(256, 40), NOPR(66), LNods(256, 8)
COMMON/WORK/ELCOD(2, 8), SHAPE(8), DERIV(2, 8), DMATX(8, 8),
1 CARTD(2, 8), DBMAT(8, 40), BMATX(8, 40), SMATX(8, 40, 4),
1 POSG(2), WIGP(2), GPCOD(2, 4), ESTIF(40, 40)

C*** CALCULATES MATRIX OF ELASTIC RIGIDITIES
READ (1, *) N, NLayer(IELEM)
WRITE (3, *) N, NLayer(IELEM)
LTMP = NLayer(IELEM) + 1

89
READ(1, *) (H(K), K=1, LTMP)
WRITE(3, *) (H(K), K=1, LTMP)
DO 111 IL=1, (LTMP-1)
READ(1, *) N, EL(IL), ET(IL), VLT(IL), VTL(IL), GLT(IL), THETA(IL),
1 G13(IL), G23(IL)
WRITE(3, 719) N, EL(IL), ET(IL), VLT(IL), VTL(IL), GLT(IL), THETA(IL),
1 G13(IL), G23(IL)
719 FORMAT(14, 2E12.6, 2F8.3, E10.4, F8.2, 2E10.4)
111 CONTINUE
DO 10 ISTR=1, NSTRE
DO 10 JSTRE=1, NSTRE
DMATX(ISTRE, JSTRE) = 0.0
10 CONTINUE
DO 14 I=1, 3
DO 14 J=1, 3
A(I, J) = 0.0
B(I, J) = 0.0
D(I, J) = 0.0
C(I, J) = 0.0
Q(I, J) = 0.0
14 CONTINUE
DO 15 I=1, 2
DO 15 J=1, 2
E(I, J) = 0.0
F(I, J) = 0.0
15 CONTINUE
DO 22 IL=1, NLayer(IELEM)
THETA(IL) = THETA(IL) * 3.14157/180.0
C(1, 1) = EL(IL) / (1.0 - VLT(IL) * VTL(IL))
C(2, 2) = ET(IL) / (1.0 - VLT(IL) * VTL(IL))
C(1, 2) = EL(IL) * VTL(IL) / (1.0 - VLT(IL) * VTL(IL))
C(3, 3) = GLT(IL)
Q(1, 1) = (C(1, 1) * COS(THETA(IL)) ** 4 + C(2, 2) * SIN(THETA(IL)) ** 4) + 2.0
1 * (C(1, 2) + 2.0*C(3, 3)) * (SIN(THETA(IL)) ** 2) * (COS(THETA(IL)) ** 2)
Q(1, 2) = (C(1, 1) + C(2, 2) - 4.0*C(3, 3)) * (SIN(THETA(IL)) ** 2)
1 * COS(THETA(IL)) ** 2
1 * C(1, 2) * (COS(THETA(IL)) ** 4 * SIN(THETA(IL)) ** 4)
Q(1, 3) = (C(1, 1) - C(1, 2) - 2.0*C(3, 3)) * (COS(THETA(IL)) ** 3)
1 * SIN(THETA(IL))
1 - (C(2, 2) - C(1, 2) - 2.0*C(3, 3)) * (SIN(THETA(IL)) ** 3) * COS(THETA(IL))
Q(2, 1) = Q(1, 2)
Q(2, 2) = C(1, 1) * SIN(THETA(IL)) ** 4 + C(2, 2) * COS(THETA(IL)) ** 4 + 2.0
1 * (C(1, 2) + 2.0*C(3, 3)) * (SIN(THETA(IL)) ** 2) * COS(THETA(IL)) ** 2
Q(2, 3) = (C(1, 1) - C(1, 2) - 2.0*C(3, 3)) * COS(THETA(IL))
1 * SIN(THETA(IL)) ** 3
1 - (C(2, 2) - C(1, 2) - 2.0*C(3, 3)) * SIN(THETA(IL)) * COS(THETA(IL)) ** 3
Q(3, 1) = Q(1, 3)
Q(3, 2) = Q(2, 3)
Q(3, 3) = (C(1, 1) + C(2, 2) - 2.0*C(1, 2) - 2.0*C(3, 3))
1 * (SIN(THETA(IL)) ** 2) * 2
1 * COS(THETA(IL)) ** 2 + C(3, 3) * (SIN(THETA(IL)) ** 4 + COS(THETA(IL)) ** 4)
E(1, 1) = G13(IL) * COS(THETA(IL)) ** 2 + G23(IL) * SIN(THETA(IL)) ** 2
E(1, 2) = G23(IL) - G13(IL) * COS(THETA(IL)) * SIN(THETA(IL))
E(2, 1) = E(1, 2)
E(2, 2) = G13(IL) * SIN(THETA(IL)) ** 2 + G23(IL) * COS(THETA(IL)) ** 2
DO 21 I=1, 2
DO 21 J=1, 2
F(I, J) = F(I, J) + 1.25 * (E(I, J) + (H(IL+1) - H(IL)) - 4.0 * (H(IL+1)**3 - H(IL)**3) / (3.0*TH*TH))
DO 21 I=1, 2
DO 21 J=1, 2
F(I, J) = F(I, J) + 1.25 * (E(I, J) + (H(IL+1) - H(IL)) - 4.0 * (H(IL+1)**3 - H(IL)**3) / (3.0*TH*TH))
CONTINUE
DO 22 I=1,3
DO 22 J=1,3
A(I,J)=A(I,J)+Q(I,J)*(H(IL+1)-H(IL))
B(I,J)=B(I,J)+Q(I,J)*(H(IL+1)**2-H(IL)**2)/2.0
D(I,J)=D(I,J)+Q(I,J)*(H(IL+1)**3-H(IL)**3)/3.0
END

CONTINUE
DO 42 I=1,3
DO 42 J=1,3
DMAXX(I,J)=A(I,J)
END

CONTINUE
DMAXX(1,4)=B(1,1)
DMAXX(1,5)=B(1,2)
DMAXX(1,6)=B(1,3)
DMAXX(2,4)=B(2,1)
DMAXX(2,5)=B(2,2)
DMAXX(2,6)=B(2,3)
DMAXX(3,4)=B(3,1)
DMAXX(3,5)=B(3,2)
DMAXX(3,6)=B(3,3)
DO 43 I=1,3
DO 43 J=4,6
DMAXX(J,I)=DMAXX(I,J)
END

RETURN

SUBROUTINE SFR2(S,T)
*** CALCULATES SHAPE FUNCTIONS AND THEIR DERIVATIVES FOR 2D ELEMENTS
C
COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOPN,NDIME,NSTRE,TH,
1 NGAUS,NVFIX,NEVAB
COMMON/LGDATA/COORD(833,2),PRESC(66,5),ASDIS(4165),
1 ELOAD(256,40),NOFIX(66),IFPRE(66,5),LNODS(256,8)
COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMAXX(8,8),
1 CARTD(2,8),DBMAT(8,40),BMATX(8,40),SMATX(8,40,4),
1 POSGP(2),WEIGP(2),GPCOD(2,4),BSTIF(40,40)
S2=S*2.0
T2=T*2.0
SS=S*S
TT=T*T
ST=S*T
SST=S*S*T
STT=S*T*T
ST2=S*T*T*2.0
*** SHAPE FUNCTIONS
SHAPE(1)=(-1.0+ST+SS+TT-SST-STT)/4.0

C***
SHAPE(2) = (1.0 - T - SS + SST) / 2.0
SHAPE(3) = (-1.0 - ST + SS + TT - SST + STT) / 4.0
SHAPE(4) = (1.0 + S - TT - STT) / 2.0
SHAPE(5) = (-1.0 + ST + SS + TT + SST + STT) / 4.0
SHAPE(6) = (1.0 - T - SS - SST) / 2.0
SHAPE(7) = (-1.0 - ST + SS + TT + SST - STT) / 4.0
SHAPE(8) = (1.0 - S - TT + STT) / 2.0

C* * * SHAPE FUNCTION DERIVATIVES
DERIV(1, 1) = (T + S^2 - ST^2 - TT) / 4.0
DERIV(1, 2) = -S + ST
DERIV(1, 3) = (-T + S^2 - ST^2 + TT) / 4.0
DERIV(1, 4) = (1.0 - TT) / 2.0
DERIV(1, 5) = (T + S^2 + ST^2 + TT) / 4.0
DERIV(1, 6) = -S - ST
DERIV(1, 7) = (-T + S^2 + ST^2 - TT) / 4.0
DERIV(1, 8) = (-1.0 + TT) / 2.0
DERIV(2, 1) = (S + T^2 - SS - ST^2) / 4.0
DERIV(2, 2) = (-1.0 + SS) / 2.0
DERIV(2, 3) = (-S + T^2 - SS + ST^2) / 4.0
DERIV(2, 4) = -T - ST
DERIV(2, 5) = (S + T^2 + SS + ST^2) / 4.0
DERIV(2, 6) = (1.0 - SS) / 2.0
DERIV(2, 7) = (-S + T^2 + SS - ST^2) / 4.0
DERIV(2, 8) = -T + ST
RETURN
END

SUBROUTINE JACOB2(IELEM, DJACB, KGASP)
CALCULATES COORDINATES OF GAUSS POINTS
AND THE JACOBIAN MATRIX AND ITS DETERMINANT
AND THE INVERSE FOR 2D ELEMENTS
DIMENSION XJACM(2,2), XJACI(2,2)
COMMON/CONTRO/NPOIN, NELEM, NNODE, NDOFN, NDIME, NSTRE, TH, 1 NGAUS, NVFIX, NEVAB
COMMON/LGDATA/COORD(833,2), PRESC(66,5), ASDIS(4165), 1 ELOAD(256,40), NOFIX(66), IFPRE(66,5), LNODS(256,8)
COMMON/WORK/ELCOD(2,8), SHAPE(8), DERIV(2,8), DMATX(8,8), 1 CARTD(2,8), DBMAT(8,40), BMATX(8,40), SMATX(8,40,4), 1 POSGP(2), WEIGP(2), GPCOD(2,4), ESTIF(40,40)
C CALUCALTE COORDINATES OF SAMPLING POINT
DO 10 IDIME=1, NDIME
GPCOD(IDIME,KGASP)=0.0
DO 10 INODE=1, NNODE
GPCOD(IDIME,KGASP)=GPCOD(IDIME,KGASP) + ELCOD(IDIME,INODE)* 1 SHAPE(INODE)
10 CONTINUE
C CREATE JACOBIAN MATRIX XJACM
DO 20 IDIME=1, NDIME
DO 20 JDIME=1, NDIME
XJACM(IDIME,JDIME)=0.0
DO 20 INODE=1, NNODE
XJACM(IDIME,JDIME)=XJACM(IDIME,JDIME) + DERIV(IDIME,INODE)*ELCOD 1 JDIME, INODE)
20 CONTINUE
C CALCULATE DETERMINE AND INVERSE OF JACOBIAN MATRIX
DJACB=XJACM(1,1)*XJACM(2,2)-XJACM(1,2)*XJACM(2,1)
IF(DJACB .GT. 0.0) GO TO 30
WRITE(6,900)IELEM
STOP
30 XJACI(1,1)=XJACM(2,2)/DJACB

92
XJACI(2,2)=XJACM(1,1)/DJACB  
XJACI(1,2)=XJACM(1,2)/DJACB  
XJACI(2,1)=XJACM(2,1)/DJACB  

C CALCULATE CARTESIAN DERIVATIVES
DO 40 IDIME=1,NDIME
DO 40 INODE=1,NNODE
CARTD(IDIME,INODE)=0.0
DO 40 JDIME=1,NDIME
CARTD(IDIME,INODE)=CARTD(IDIME,INODE)+XJACI(IDIME,JDIME)*DERIV
1 JDIME,INODE)
40 CONTINUE
900 FORMAT(//,24HPROGRAM HALTED IN JACOB2,,/11X,22H ZERO OR NEGATIVE
1 AREA,,/10X,16H ELEMENT NUMBER,15)
RETURN
END
SUBROUTINE BMATPC
C CALCULATE STRAIN MATRIX B
COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,TH,
1 NGAUS,NVFIX,NEVAB
COMMON/LGDATA/COORD(833,2),PRESC(66,5),ASDIS(4165),
1 ELOAD(256,40),NOFIX(66),INODS(256,8)
COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),BMATX(8,8),
1 CARTD(2,8),DBMAT(8,40),SMA(8,40,4),
1 POSGP(2),WEIGP(2),GPCOD(2,4),ESTIF(40,40)
DO 10 ISTRE=1,NSTRE
DO 10 IEVAB=1,NEVAB
BMATX(ISTRE,IEVAB)=0.0
10 CONTINUE
JGASH=0
DO 20 INODE=1,NNODE
MGASH=JGASH+1
NGASH=MGASH+1
BMATX(1,MGASH)=CARTD(1,INODE)
BMATX(2,NGASH)=CARTD(2,INODE)
BMATX(3,MGASH)=CARTD(2,INODE)
BMATX(3,NGASH)=CARTD(1,INODE)
IGASH=NGASH+1
BMATX(7,IGASH)=-CARTD(1,INODE)
BMATX(8,IGASH)=-CARTD(2,INODE)
IGASH=IGASH+1
JGASH=IGASH+1
BMATX(4,IGASH)=CARTD(1,INODE)
BMATX(5,JGASH)=CARTD(2,INODE)
BMATX(6,IGASH)=CARTD(2,INODE)
BMATX(7,IGASH)=SHAPE(INODE)
BMATX(8,JGASH)=SHAPE(INODE)
BMATX(6,JGASH)=CARTD(1,INODE)
20 CONTINUE
RETURN
END
SUBROUTINE DBE
C CALCULATE D*B
COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,TH,
1 NGAUS,NVFIX,NEVAB
COMMON/LGDATA/COORD(833,2),PRESC(66,5),ASDIS(4165),
93
SUBROUTINE LOADPC
C CALCULATE NODAL FORCES FOR PLATE ELEMENT
DIMENSION POINT(5)
COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,TH,
1 NGAUS,NVFIX,NEVAB
COMMON/LGDATA/COORD(833,2),PRESC(66,5),ASDIS(4165),
1 ELOAD(256,40),NOFIX(66),IFPRE(66,5),LNODS(256,8)
COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(8,8),
1 CARTO(2,8),DBMAT(8,40),BMATX(8,40),SMATX(8,40,4),
1 POSGP(2),WEIGP(2),GPCOD(2,4),ESTIF(40,40)
DO 10 IELEM=1,NELEM
DO 10 IEVAB=1,NEVAB
ELOAD(IELEM,IEVAB)=0.0
10 CONTINUE
READ DATA CONTROLLING LOADING TYPES TO BE INPUT
WRITE(3,442)
442 FORMAT(/25H INPUT  LOADING DATA)
READ(1,*)IPLOD
WRITE(3,*)IPLOD
READ NODAL POINT LOADS
IF(IPLOD.EQ.0)GO TO 60
20 READ(1,*)LODPT,(POINT(IDOFN),IDOFN=1,NDOFN)
WRITE(3,925)LODPT,(POINT(IDOFN),IDOFN=1,NDOFN)
925 FORMAT(15,5F10.2) °
ASSOCIATE THE NODAL POINT LOADS WITH AN ELEMENT
DO 30 IELEM=1,NELEM
DO 30 INODE=1,NNODE
NLOCA=LNODS(IELEM,INODE)
IF(LODPT .EQ. NLOCA)GO TO 40
30 CONTINUE
40 DO 50 IDOFN=1,NDOFN
NGASH=(INODE-1)*NDOFN+IDOFN
ELOAD(IELEM,NGASH)=POINT(IDOFN)
50 CONTINUE
IF(LODPT .NE. NPOIN) GO TO 20
60 CONTINUE
LOOP OVER EACH ELEMENT
DO 110 IELEM=1,NELEM
READ(1,*)UDLOD
WRITE(3,*)'UDLOD=',UDLOD
IF(UDLOD .EQ. 0.0) GO TO 110
EVALUATE THE COORDINATES OF THE ELEMENT NODAL POINTS
DO 70 INODE=1,NNODE
LNODE=LNODS(IELEM,INODE)
70 CONTINUE
C
SUBROUTINE FRONT
DIMENSION FIXED(4165), EQUAT(325), VECRV(325), GLOAD(325),
1 GSTIF(52975), IFFIX(4165), NACVA(325), LOCEL(40), NDEST(40)
COMMON/CONTRO/NPOIN, NELEM, NNODE, NDOFN, NDIME, NSTRE, TH,
1 NGAUS, NVFIX, NEVAB
COMMON/LGDATA/COORD(833,2), PRESC(66,5), ASDIS(4165),
1 ELOAD(256,40), NNOFIX(66), IFPRE(66,5), LNODS(256,8)
COMMON/WORK/ELCOD(2,8), SHAPE(8), DERIV(2,8), DMATX(8,40),
1 CARTD(2,8), DBMAT(8,40), BMATX(8,40), SMATX(8,40,4),
1 POSGP(2), WEIGP(2), GPCOD(2,4), ESTIF(40,40)
NFUNC(I,J)=(J*J-J)/2+I
MFRON=325
MSTIF=52975
C INTERPRET FIXITY DATA IN VECTOR FORM
NTOTV=NPOIN*NDOFN
DO 100 ITOTV=1, NTOTV
IFFIX(ITOTV)=0
100 FIXED(ITOTV)=0.0
DO 110 IVFIX=1, NVFIX
NLOCA=(NOFIX(IVFIX)-1)*NDOFN
DO 110 IDOFN=1, NDOFN
NGASH=NLOCA+IDOFN
IFFIX(NGASH)=IFPRE(IVFIX, IDOFN)
110 FIXED(NGASH)=PRESC(IVFIX, IDOFN)
C CHANGE THE SIGN OF THE LAST APPEARANCE OF EACH NODE
DO 140 IPOIN=1,NPOIN
KLAST=0
DO 130 IELEM=1,NELEM
DO 120 INODE=1,NNODE
IF(LNODS(IELEM,INODE).NE.IPOIN) GO TO 120
KLAST=IELEM
NLAST=INODE
120 CONTINUE
130 CONTINUE
IF(KLAST.NE.0) LNODS(KLAST,NLAST)=-IPOIN
140 CONTINUE

C START BY INITIALIZING EVERYTHING THAT MATTERS TO ZERO
DO 150 ISTIF=1,MSTIF
150 GSTIF(ISTIF)=0.0
DO 160 IFRON=1,MFRON
GLOAD(IFRON)=0.0
EQUAT(IFRON)=0.0
VECRV(IFRON)=0.0
160 CONTINUE
NACVA(IFRON)=0

C ENTER MAIN ELEMENT ASSEMBLY REDUCTION LOOP
NFRON=0
KELVA=0
DO 380 IELEM=1,NELEM
KEVAB=0
DO 170 INODE=1,NNODE
DO 170 IDOFN=1,NDOFN
NPOSI=(INODE-1)*NDOFN+IDOFN
LOCNO=LNODS(IELEM,INODE)
IF(LOCNO.GT.O) LOCEL(NPOSI)=(LOCNO-1)*NDOFN+IDOFN
IF(LOCNO.LT.0) LOCEL(NPOSI)=(LOCNO+1)*NDOFN-IDOFN
170 CONTINUE

C START BY LOOKING FOR EXISTING DESTINATIONS
DO 210 IEVAB=1,NEVAB
NIKNO=IABS(LOCEL(IEVAB))
KEXIS=0
DO 180 IFRON=1,NFRON
IF(NIKNO.NE.NACVA(IFRON)) GO TO 180
KEVAB=KEVAB+1
KEXIS=1
NDEST(KEVAB)=IFRON
180 CONTINUE
IF(KEXIS.NE.0) GO TO 210

C WE NOW SEEK NEW EMPTY PLACES FOR DESTINATION VECTOR
DO 190 IFRON=1,MFRON
IF(NACVA(IFRON).NE.0) GO TO 190
NACVA(IFRON)=NIKNO
KEVAB=KEVAB+1
NDEST(KEVAB)=IFRON
GO TO 200
190 CONTINUE

C THE NEW PLACES MAY DEMAND AN INCREASE IN CURRENT FRONTWIDTH
200 IF(NDEST(KEVAB).GT.NFRON) NFRON=NDEST(KEVAB)
210 CONTINUE

C ASSEMBLE ELEMENT LOADS
DO 240 IEVAB=1,NEVAB
IDEST=NDEST(IEVAB)
GLOAD(IDEST)=GLOAD(IDEST)+ELOAD(IELEM,IEVAB)

C ASSEMBLE THE ELEMENT STIFFNESSES BUT NOT IN RESOLUTION
DO 220 IEVAB=1,NEVAB
JDEST=NDEST(JEVAB)
NGASH=NFUNC(IDEST,JDEST)
NGISH=NFUNC(JDEST,IDEST)
IF(JDEST.GE.IDEST)GSTIF(NGASH)=GSTIF(NGASH)+ESTIF(IEVAB,JEVAB)
IF(JDEST.LT.IDEST)GSTIF(NGISH)=GSTIF(NGISH)+ESTIF(IEVAB,JEVAB)

CONTINUE

C RE-EXAMINE EACH ELEMENT NODE TO ENQUIRE WHICH CAN BE ELIMINATE!
DO 370 IEVAB=1,NEVAB
NIKNO=LOCEL(IEVAB)
IF(NIKNO.LE.0) GO TO 370

C FIND POSITIONS OF VARIABLES READY FOR ELIMINATIONS
DO 350 IFRON=1,NFRON
IF(NACVA(IFRON).NE.NIKNO) GO TO 350

C EXTRACT THE COEFFICIENTS OF THE NEW EQUATIONS FOR ELIMINATION
DO 250 JFRON=1,MFRON
IF(IFRON.LT.JFRON) NLOCA=NFUNC(IFRON,JFRON)
IF(IFRON.GE.JFRON) NLOCA=NFUNC(JFRON,IFRON)
EQUAT(JFRON)=GSTIF(NLOCA)
250 GSTIF(NLOCA)=0.0

C AND EXTRACT THE CORRESPONDING RIGHT HAND SIDES
EQRHS=GLOAD(IFRON)
GLOAD(IFRON)=0.0
KELVA=KELVA+1

C WRITE EQUATIONS TO DISC
WRITE(2,999)(EQUAT(KL),KL=1,325),EQRHS,IFRON,NIKNO

C DEAL WITH PIVOT
PIVOT=EQUAT(IFRON)
EQUAT(IFRON)=0.0

C ENQUIRE WHETHER PRESENT VARIABLE IS FREE OR PRESCRIBED
IF(IFFIX(NIKNO).EQ.0) GO TO 300

C DEAL WITH A PRESCRIBED DEFLECTION
DO 290 JFRON=1,NFRON
GLOAD(JFRON)=GLOAD(JFRON)-FIXED(NIKNO)*EQUAT(JFRON)
290 GO TO 340

C ELIMINATE A FREE VARIABLE -DEAL WITH RIGHT HAND SIDE FIRST
DO 330 JFRON=1,NFRON
GLOAD(JFRON)=GLOAD(JFRON)-EQUAT(JFRON)*EQRHS/PIVOT
IF(EQUAT(JFRON).EQ.0.0) GO TO 330
NLOCA=NFUNC(0,JFRON)
DO 310 LFRON=1,JFRON
NGASH=LFRON+NLOCA
310 GSTIF(NGASH)=GSTIF(NGASH)-EQUAT(JFRON)*EQUAT(LFRON)/PIVOT
330 CONTINUE

340 EQUAT(IFRON)=PIVOT

C RECORD THE NEW VACANT SPACE AND REDUCE FRONTWIDTH IF POSSIBLE
NACVA(IFRON)=0
GO TO 360

C COMPLETE THE ELEMENT LOOP IN THE FORWARD ELIMINATION
CONTINUE

350 CONTINUE

360 IP(NACVA(NFRON).NE.0) GO TO 370
NFRON=NFRON-1
IF(NFRON.GT.0) GO TO 360

370 CONTINUE

380 CONTINUE

C ENTER BACK-SUBSTITUTION PHASE, LOOP BACKWARDS THROUGH VARIABLES
DO 410 IELVA=1,KELVA
C READ A NEW EQUATION
BACKSPACE 2
C PREPARE TO BACK-SUBSTITUTION FROM THE CURRENT EQUATION
Pivot=EQUAT(IFRON)
IF(IFIX(NIKNO).EQ.1) VECRV(IFRON)=FIXED(NIKNO)
IF(IFIX(NIKNO).EQ.0) EQUAT(IFRON)=0.0
C BACK SUBSTITUTE IN THE CURRENT EQUATION
DO 400 JFRON=1,MFRON
400
C EQRHS=EQRHS-VECRV(JFRON)*EQUAT(JFRON)
PUT THE FINAL VALUES WHERE THEY BELONG
IF(IFIX(NIKNO).EQ.0)VECRV(IFRON)=EQRHS/PIVOT
IF(IFIX(NIKNO).EQ.1) FIXED(NIKNO)=-EQRHS
ASDIS(NIKNO)=VECRV(IFRON)
410 CONTINUE
900 FORMAT(I10,5E14.6)
WRITE(3,900)
920 FORMAT(I10,5E14.6)
WRITE(3,925)
940 FORMAT(I10,5E14.6)
WRITE(3,945)
520 CONTINUE
RETURN
END
C SUBROUTINE STREPC
CALCULATES STRESS RESULTANTS AT GAUSS POINTS FOR ALL ELEMENTS
DIMENSION ELDIS(5,8),STRSG(8)
COMMON/CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,TH,
1 NGAUS,NVFIX,NEVAB
COMMON/LGDATA/COORD(833,2),PRESC(66,5),ASDIS(4165),
1 ELOAD(256,40),NOFIX(66),IPREB(66,5),LNODS(256,8)
COMMON/WORK/ELCOD(2,8),SHAPE(8),DERIV(2,8),DMATX(8,40),
1 CARTD(2,8),DBMAT(8,40),BMATX(8,40),SMATX(8,40,4),
1 POSGP(2),WGRIP(2),GFCOD(2,4),ESTIF(40,40)
WRITE(3,900)
WRITE(3,905)
C  PREPARE FOR DISC READING
REWIND 4
C  LOOP OVER EACH ELEMENT
DO 40 IELEM=1,NELEM
C  READ THE STRESS MATRIX , SAMPLING POINT COORDINATES
C  FOR THE ELEMENT
READ(4,*)SMATX,GPCOD
WRITE(3,910)IELEM
C  IDENTIFY THE DISPLACEMENTS OF THE ELEMENT NODAL POINTS
DO 10 INODE=1,NNODE
   LNODE=LNODS(IELEM,INODE)
   NPOSN=(LNODE-1)*NDOFN
   DO 10 IDOFN=1,NDOFN
      NPOSN=NPOSN+1
      ELDIS(IDOFN,INODE)=ASDIS(NPOSN)
10 CONTINUE
KGASP=0
C  ENTER LOOPS OVER EACH SAMPLING POINT
DO 30 IGAUS=1,NGAUS
   DO 30 JGAUS=1,NGAUS
      KGASP=KGASP+1
      DO 20 ISTRE=1,NSTRE
      STRSG(ISTRE)=0.0
      KGASH=0
      C  COMPUTE THE STRESS RESULTANTS
      DO 20 INODE=1,NNODE
         DO 20 IDOFN=1,NDOFN
            KGASH=KGASH+1
            STRSG(ISTRE)=STRSG(ISTRE)+SMATX(ISTRE,KGASH,KGASP)*ELDIS(IDOFN,INODE)
20 CONTINUE
C  OUTPUT THE STRESS RESULTANTS
WRITE(3,915) KGASP, (GPCOD(IDIME,KGASP),IDIME=1,NDIME),
               (STRSG(ISTRE),ISTRE=1,NSTRE)
30 CONTINUE
40 CONTINUE
900 FORMAT(/,10X,8HSTRESSES,/)  
905 FORMAT(4HG,P.,2X,6HX-CRD,2X,6HY-CRD.,2X,2HMX,
               1 4X,2HNY,3X,3HNXY,5X,2HMX,
               1 7X,2HMY,9X,3HMYX,8X,3HQXZ,8X,3HQYZ)
910 FORMAT(/,5X,12HELEMENT NO.=-,I5)  
915 FORMAT(I3,2F8.4,3F5.1,5E11.4)
RETURN
END