International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Development of Flat Shell Elements with Drilling Degrees of Freedom in Java

I.E Umeonyiagu[1], N.P Ogbonna[2]

[1]Department of Civil Engineering, Chukwuemeka Odumegwu Ojukwu University, Uli. Nigeria.

[2]Department of Civil Engineering, Chukwuemeka Odumegwu Ojukwu University, Uli. Nigeria.

Abstract: ***This study developed flat shell elements with drilling degree of freedom using an object oriented approach and the Java programming language. A review of the work of Kansara (2004) and Nikishkov (2010) was conducted to develop a finite element program using Java programming language which addresses most of deficiencies inherent in the work of Kansara (2004). Input to the program is done through a text file and the loads that can be applied to the structure include concentrated loads and uniformly distributed surface loads. Various load combinations can also be used. The program computes displacements and stresses at each node of the finite element model. Several test examples were analyzed using the program and results were compared with those obtained from Kansara(2004), the commercial finite element analysis program SAP 2000 and LISA respectively. Results were compared at the points of maximum displacements and stresses. The average stress was taken in to consideration to calculate stresses at specific point. The results obtained from the analysis of the example problems were found to be very accurate when compared to those obtained from the Kansara (2004), SAP 2000 and LISA. The difference in displacements computed by the two programs was less than 1, which is very negligible. The difference in stresses was also quite close. Results from the bilinear degenerated shell element with drilling degree of freedom was in agreement with those obtain from LISA and SAP 2000 but differed considerably with those of Kansara (2004).***

Keywords: Finite element model, Java programming language, Drilling degree of freedom, bilinear degenerated shell element.

List of Symbols: $\theta_x, \theta_y, and\ \theta_z$ are rotations about the *x, y* and *z* respectively,N_i are Shape functions , $\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} are$ Strain-displacement relationships for the four node, *G* is the shear modulus, $\boldsymbol{V_{3i}}$ is called node director, *L* is Length, *H is* Depth, *t is* Thickness , E is Modulus of elasticity $n$is Poisson's ratio, $\sigma_x, \sigma_y, \sigma_x, \tau_z, \tau_{yz}, \tau_{zx}$ are the state of stress at any point $\epsilon_x, \epsilon_y, \epsilon_z, \gamma_{xy}, \gamma_{yz}, \gamma_{zx}$ are the components of the strain $\alpha\ is$ Coefficient of thermal expansion T *is* Change in temperature , ψ is the non-zero residual due to the approximate representation of a function inside a finite element.

1. Introduction

McNeal (1978) developed the quadrilateral shell element QUAD4, by considering two inplane displacements that represent membrane properties and one out-of-plane displacement and two rotations, which represents the bending properties. McNeal (1978) included modifications in terms of a reduced order integration scheme for shear terms. He also included curvature and transverse shear flexibility to deal with the deficiency in the bending strain energy. The simplest method adopted to remove the rotational singularity is to add a fictitious rotational stiffness. However, Yang et al. (2000) suggested that, although the method solves the problem of singularity it creates a convergence problem that sometimes leads to poor results. Recent developments include using membrane elements with rotational degrees of freedom to develop an efficient flat shell element (Kansara, 2004). Several methods have been suggested by various authors for removing the singularity in the stiffness matrix based on variational principles such as those formulated by Gruttmann et al (1992). A degenerated shell element with drilling degrees of freedom was developed recently by Djermane *et al.* (2006) for application in linear and nonlinear analysis of thin shell structures for isotropic or anisotropic materials with using the assumed natural strains technique to alleviate locking phenomenon. Djermane et al. (2007) also extended the formulation by using the same techniques to study the dynamic responses of thick and thin nonlinear shells. Kanok–Kanukulchai (1979) devised utilization of the penalty approach to remove the rotational singularity caused by the addition of a fictitious rotational stiffness. He introduced a constraint equation which "links" the drilling rotations in the fiber coordinate system to the in-plane twisting mode of the mid-surface.

Adam et al (2013) developed a bilinear degenerated four nodes shell elements with drilling degrees of freedom. They examined the element performance with respect to sensitivity to the value of penalty parameter and to evaluate the suitable value. Forte et al (1990) in one of the first publications on the object oriented approach to the finite element development, presented essential finite element classes such as elements, nodes, displacement and force boundary conditions, vectors and matrices. Several authors described detailed finite element architecture using the object oriented approach. Nikiskov (2010) developed procedures for programming finite element in Java. He was able to demonstrate on how to solve finite element problems for solid mechanics as well as Heat transfer problems.

Kansara (2004) developed membrane, plate and flat shell element in Java Programming language. He created a finite element analysis program using Java Programming Language to check the accuracy of the developed elements. Kansara (2004) while testing the output of his Java code with a commercial program SAP 2000 noted that for the test examples of a cantilever I-beam and a folded plate structure, the stresses from the program differed considerably from those obtained from SAP 2000. This was due to the fact that his Java program lacked the ability to solve flat shell problems with rotational degrees of freedom (also known as "drilling degree of freedom"). He advocated for a future

research work that can handle most of the deficiencies inherent in his work. These were the inability to developing flat shell elements in which the membrane elements have rotational (drilling) degrees of freedom, the use of a band storage scheme for storing the structure stiffness matrix and band solvers in the program to solve large finite element analysis problems, absence of a graphical user interface in the program and, extending the application of the program to include other elements such as truss, and frame elements.

This work improved on the work of Kansara (2004) using the procedures prescribed by Nikishkov (2010). Flat shell elements developed are the bilinear degenerated four nodes shell element with drilling degree of Freedom (Adam et al, 2013). A finite element analysis program was also developed with Java programming language to check the accuracy of the developed elements. Several test structures will be analyzed using the Java program and the results compared with those from other standard test validation examples.

2. Literature Review

2.1 Bilinear Degenerated Four Nodes Shell Element

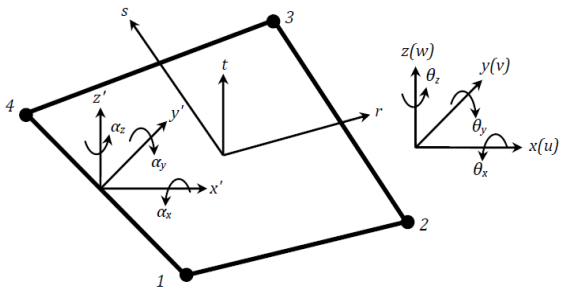

Figure 2.1: Four nodes shell element (Adam et al, 2013).

The mid-surface shown in Figure 2.1 is defined by natural coordinates (*r*, *s*, *t*). The displacements *u*, *v* and *w* are the displacements in global Cartesian coordinates *x*, *y* and *z* respectively. $\theta_x, \theta_y, and\ \theta_z$ are rotations about the *x*, *y* and *z* respectively. The rotations $\alpha_x, \alpha_y\ and\ \alpha_z$ are about local coordinates *x'*, *y'*, and *z'* respectively.
The shape functions to describe the mid-surface in terms of natural coordinates are:

$$N_i(r,s) = \frac{1}{4}(1 + r_i r)(1 + S_i S) \tag{2.1}$$

The thickness at each node h_i is computed in the direction normal to the mid-surface.
$\boldsymbol{V_{3i}}$ is called node director and defined by:

$$V_{3i} = \begin{Bmatrix} x_{itop} - x_{ibottom} \\ y_{itop} - y_{ibottom} \\ z_{itop} - z_{ibottom} \end{Bmatrix} \tag{2.2}$$

The coordinates of any point in the element can be derived from the 8-nodes solid element to 4-nodes element as:

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \sum_{i=1}^{4} \frac{1}{2}(1+t)N_i \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix}_{top} + \sum_{i=1}^{4} \frac{1}{2}(1-t)N_i \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix}_{bottom} \tag{2.3}$$

Since $\frac{1}{2}(1+t)$ is the part of shape function in 8-nodes solid element in direction of the thickness x_i, y_i and z_i are the global coordinates of the midpoint *i*
The displacement variation in the element can be expressed as:

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \sum_{i=1}^{4} N_i \left(\begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} + \begin{Bmatrix} u_i^* \\ v_i^* \\ w_i^* \end{Bmatrix} \right) \tag{2.4}$$

Where $u_i, v_i\ and\ w_i$ are the displacements at mid point *i* along global direction, and $u*_i, v*_i\ and\ w*_i$ are the relative nodal displacements along global direction produced by rotation of the normal at node *i* and can be expressed in terms of rotations $\theta_{xi}, \theta_{yi}, and\ \theta_{zi}$ at each node *i* about global axes. The stress-strain relation can be stated after imposing $\sigma_{z'} = 0$ as:

$$\begin{Bmatrix} \sigma_{x'} \\ \sigma_{y'} \\ \tau_{x'y'} \\ \tau_{x'z'} \\ \tau_{y'z'} \end{Bmatrix} = \frac{E'}{1-v^2} \begin{bmatrix} 1 & v & 0 & 0 & 0 \\ v & 1 & 0 & 0 & 0 \\ 0 & 0 & A & 0 & 0 \\ 0 & 0 & 0 & D & 0 \\ 0 & 0 & 0 & 0 & D \end{bmatrix} \begin{Bmatrix} \varepsilon_{x'} \\ \varepsilon_{y'} \\ \gamma_{x'y'} \\ \gamma_{x'z'} \\ \gamma_{y'z'} \end{Bmatrix} \tag{2.5}$$

$$A = \frac{(1-v)}{2} \quad and \quad D = \mu A \tag{2.6}$$

where $\mu = 5/6$ is a factor that accounts for the thickness-direction variation of transverse shear strain, E' is the modulus of elasticity and v is Poisson's ratio. The constitutive matrix in Equation (2.6) is split into $\boldsymbol{C_m}$ and $\boldsymbol{C_s}$ as follows:

$$\left.\begin{aligned} C_m &= \frac{E}{1-v^2} \begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & \frac{1-v}{2} \end{bmatrix} \\ C_s &= \frac{E'\mu}{2(1+v)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}\right\} \tag{2.7}$$

In a degenerated shell, the rotation of the normal and the mid-surface displacement field are independent. The idea then is to derive an additional constraint between the torsional rotation of the normal, α_z, and the rotation of the mid-surface,

$$\frac{1}{2}\left(\frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'}\right) \tag{2.8}$$

The derivation of the torsional rotation of the normal from that of the mid-surface is assumed to have governing strain energy (Djermane, 2007) and (Kanok-Nukulchai ,1979) as:

$$\left.\begin{aligned} U_t = \alpha_t G \int_A \int \left[\alpha_z - \frac{1}{2}\left(\frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'}\right)\right]^2_{(r,s,0)} dA \\ = u^T k_t u \end{aligned}\right\} \tag{2.9}$$

Where α_t is the torsional constant and it is problem dependent, G is the shear modulus, $\frac{\partial v'}{\partial x'}$ and $\frac{\partial u'}{\partial y'}$ can be calculated from Eqn. (2.5). The matrix B_3 can be written after extracting the vector of nodal displacements u from the relation $\left[\alpha_z - \frac{1}{2}\left(\frac{\partial v'}{\partial x'} - \frac{\partial u'}{\partial y'}\right)\right]$, then the torsional stiffness matrix is given by:

$$k_t = \alpha_t G \iint_{-1-1}^{11} (B_3^T B_3) dA \qquad (2.10)$$

α_t is a penalty parameter and must be determined to insure good convergence (Adam et al, 2013).

3. Materials and Method

3.1 Creation of Robot Finite Element Analysis Program (Rfea)

The required input to Rfea is in the form of a text file and the results from the program saved in an output file in text format. A simple GUI was used during visualization of finite element models and results. Rfea performs three tasks of the finite element analysis: preprocessing (finite element model generation), processing (problem solution); and *post* processing (results calculation and visualization). Rfea finite element system is organized into eight class packages. Using a Unified Modelling Language (UML) diagram, the various packages and their individual classes are presented as follows:

3.1.1 Package fea
This package shall contain the main classes which include FE, JFEM, JMGEN and JVIS.

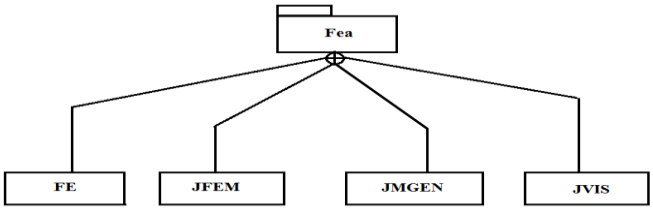

Figure 3.1: Package fea showing its member classes

3.1.2 Package model
This package shall contain finite element model and loading classes which are illustrated on the UML diagram in figure 3.2.

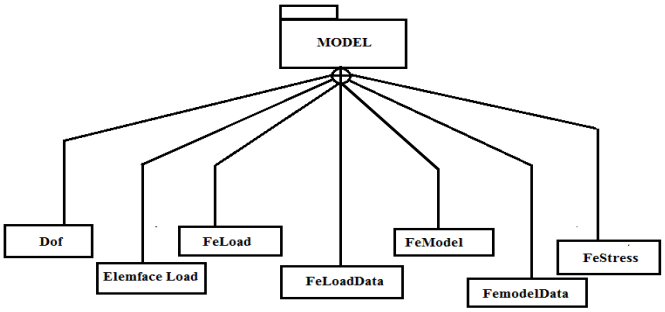

Figure 3.2: Package Model showing its member classes

3.1.3 Package util
Utility classes which include the FePrintWriter, FeScanner, GaussRule, UTIL

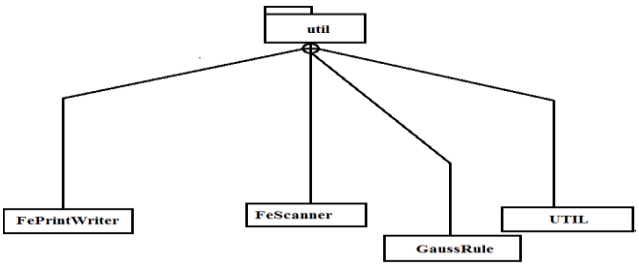

Figure 3.3: Package util with its member classes

3.1.4 Package elem
This package contains classes such ElementShellD, ShapeShellD.

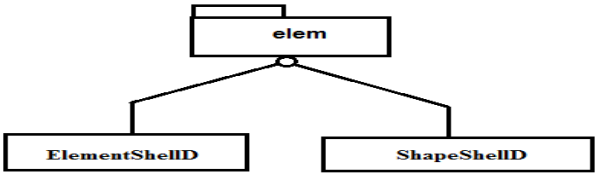

Figure 3.4: Package elem with its member classes

3.1.5 Package material
This package contains the constitutive relations for materials.

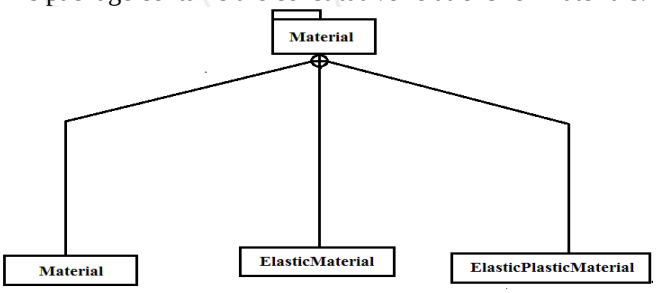

Figure 3.5: Package material with its member classes

3.1.6 Package solver
Classes for the Assembly and solution of global finite element equation systems:

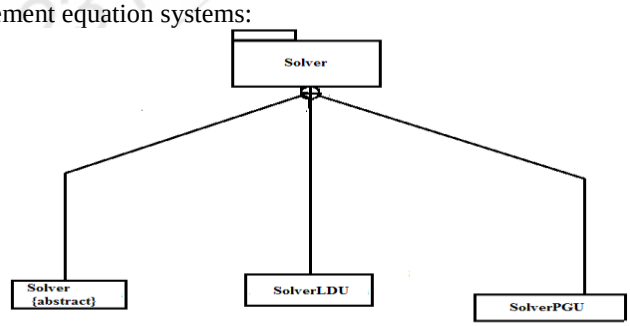

Figure 3.6: Package solver with its member classes

3.1.7 Package gener
The classes of this package will be used for generation of mesh and are represented with a UML diagram shown in figure 3.7.

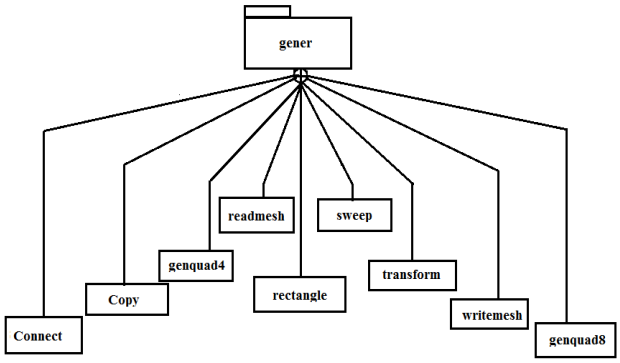

Figure 3.7: Package gener with its member classes

3.1.8 Package visual

Contains classes visualization of models and results as shown in figure 3.7

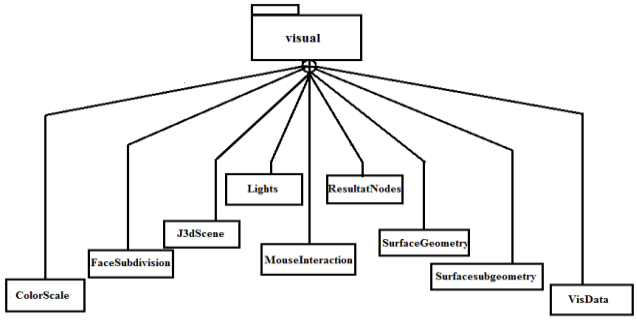

Figure 3.8: Package visual with its member classes

Detailed description of some Rfea classes imported from Nikishkov (2010), were presented in the last sections of chapter two of this thesis work. The new Java Programming classes developed to modify Kansara (2004) work using Nikishkov's approach will be explained in the successive sections of this thesis work

3.2 Implementation of Bilinear Degenerated Four Nodes Elements with Drilling Degree of Freedom

3.2.1 Class ElementShell4

Class ElementShell4 is created to extend class Element and to implement methods for computing element matrices and vectors.

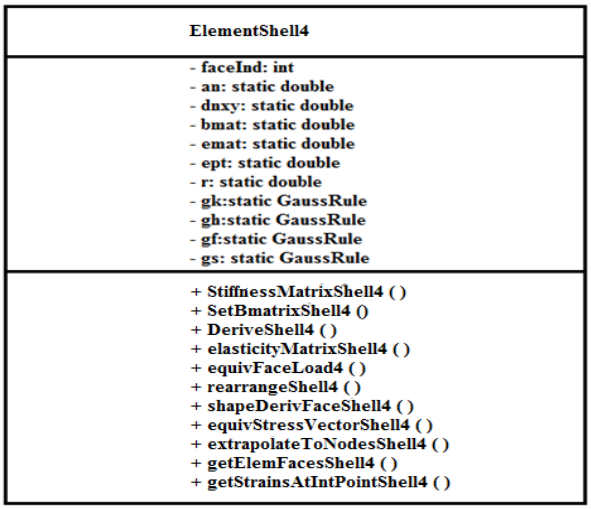

Figure 3.1: ElementShell4 with its attributes and operations

Constructor ElementShell4()calls the constructor of parent class Element and passes to it the element name shell4, the number of element nodes (4) and the number of points for storing stresses and strains. Method stiffnessMatrixBshell4() performs the computation of the element stiffness matrix according to Equation (2.110)

Other important methods and their description are given in table 3.5

Table 3.5: Methods in the ElementShell4 class

Method	Description
setBmatrixShell4()	Performs computation of a displacement differentiation matrix bmat for specified local coordinates xi and et and returns the determinant of the Jacobian matrix.
Derive Shell4()	Computes the derivatives of shape functions dnxy with respect to global coordinates *x, y*
elasticity Matrix Shell4()	Sets the elasticity matrix emat.
Equiv Face Load Shell4()	Computing a nodal equivalent of surface load
rearrange Shell4()	Used to put load information in order, corresponding to local element numbering
Shape Deriv Face Shell4()	Provides one-dimensional shape functions an and derivatives of shape functions xin with respect to local coordinate ξ changing along the considered element side.
equivStress Vector Shell4()	Computation of a nodal force vector, which is equivalent to element stress field.
Extrapolate To NodesShell4()	Stress extrapolation from integration points to nodes
getElem FacesShell4 ()	returns local numbers for four element sides specified in array faceInd

3.2 Class ShapeShell4

Class ShapeShell4 is created to calculate the shape functions for 4 node quadratic shell elements with drilling degrees of freedom. Element nodes are numbered in an anticlockwise direction starting from any corner node.

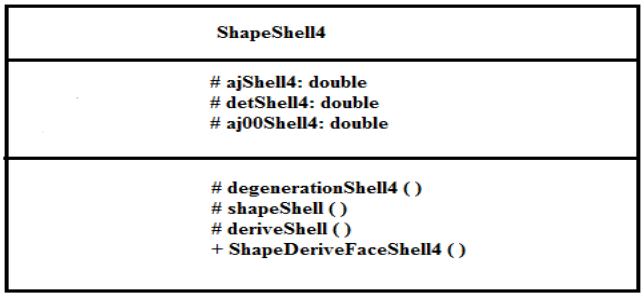

Figure 3.19: ShapeShell4 with its attributes and operations

Method shapeShell4 () computes element shape functions an for specified local coordinates xi (ξ) and et (η). Connectivity numbers ind are used as information on the existence of midside nodes.

Derivatives of shape functions with respect to global coordinates dnxy are given by method deriv. The method parameters are:
xi, et – local coordinates ξ and η, ind – element connectivities, xy – array of nodal coordinates.

Method shapeDerivFaceShell4 () calculates three shape functions an and their derivatives dndxi with respect to the local coordinate ξ.

4. Numerical Analysis

4.1 Verification of Bilinear Degenerated Shell Element with Drilling Degree of Freedom

4.1.1 Analysis of a cantilever channel section with fifty Six –four node quadrilateral shell elements

The finite element model is generated using 56 four node quadrilateral shell elements. Figure 4.1 shows the finite element discretization for the structure.

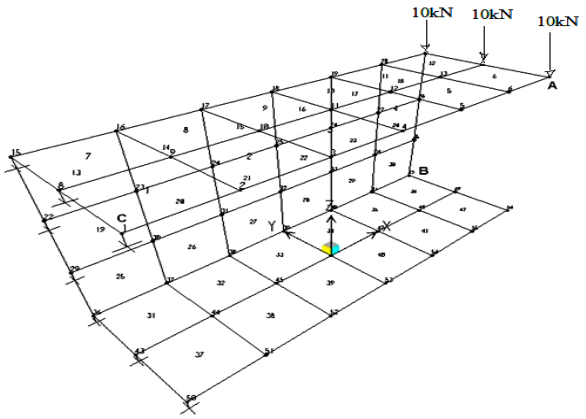

Figure 4.1: FE Model for a Cantilever Channel Section (Kansara 2004)

Geometric Data: Length = 0.6.m, Flange width = 0. 2m, Height of the section = 0.3.m., Thickness t = 0.1m
Material Properties: Modulus of elasticity E = 3600 kN/m^2, Poisson's ratio n = 0.2
Boundary Conditions: Restraints in all six directions are provided at the left end of the cantilever.
Loading: A concentrated loads of 10 kN each is applied to the nodes at the top of the free end of the cantilever (nodes 7, 14, and 21) as shown in Figure 4.1
Comparison of Results:
Table 4.1 shows the comparison of displacements and stresses for the verification example. The displacements at nodes 11 and 14 are compared with those obtained from SAP 2000. The difference in the displacements is less than 5%.

Table 4.1: Displacements and Stresses for a cantilever channel section with fifty Six –four node quadrilateral shell elements

Location		*Program Rfea*	*SAP 2000*	*Difference (%)*
POINT A (NODE 14)	**UX**	0.0155091	0.014522	0.0246775
	UY	-0.067292	-0.077952	0.26650
	UZ	-0.2385	-0.223337	-0.379075
POINT B (NODE 11)	**UX**	0.009755	0.010005	-0.00625
	UY	-0.02876	-0.025712	-0.0762
	UZ	-0.069415	-0.070682	0.031675
POINT C (NODE 1)	**S11**	59.353877	59.538842	-4.624125
	S22	11.997768	12.050775	-1.325175
POINT D (NODE 7)	**S12**	-26.876871	-26.862638	-0.355825

4.2.2 Finite Element Analysis of a Cantilever I – Beam with forty eight - 48 four node quadrilateral shell elements

This example consists of a cantilever I – beam. It tests the behavior of the element when the inplane bending stresses are severe.. The finite element model of this example problem contains 48 four node quadrilateral shell elements and is shown in Figure 4.2.

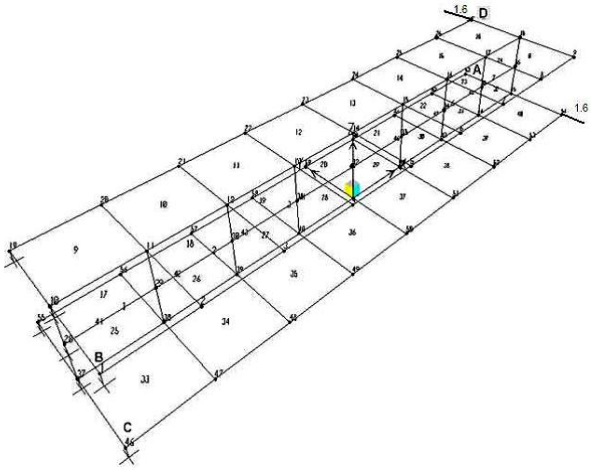

Figure 4.2: FE Model for a Cantilever I – Beam (Kansara 2004)

Geometric Data:
Length L = 0.4m, Width = 0.1m, Height h = 0.05m, Thickness t = 0.025.

Material Properties: Modulus of elasticity E = 10000 N/mm^2, Poisson's ratio η = 0.3

Boundary Conditions: One end of the cantilever is fixed.

Loading: A concentrated load of 1.6 kN is applied at the top and bottom of the flange in opposite directions as shown in Figure 4.2

Table 4.2: Displacements and Stresses for a Cantilever I – Beam with forty eight - 48 four node quadrilateral shell elements

Location		Program Rfea	SAP 2000	Difference (%)
POINT A (NODE 63)	UX	-0.02712	-0.027162	0.00105
	UY	0.149943	0.151049	-0.02765
	UZ	0.243467	0.255308	-0.296025
POINT E (NODE 9)	UX	-0.02512	-0.027162	0.05105
	UY	-0.139963	-0.151049	0.27715
	UZ	-0.209469	-0.255308	1.145975
POINT B (NODE 1)	S11	-12.38834	-12.256818	-3.28805
POINT C (NODE 46)	S22	4.640982	4.683511	-1.063225
POINT D (NODE 27)	S12	-1.593552	-1.683754	2.25505

Comparison of Results: The displacements at nodes 9 and 63 and stresses at nodes 1, 46, and 27 are obtained from the developed program Rfea and SAP 2000 are shown in Table 4.2. The displacements are compared at nodes 63 and node 9, which are the nodes opposite to the nodes where the loads are applied. When a torque is applied to a cantilever I – beam it was expected that the displacements in y and z should be in the opposite directions and of same value. The tabulated results agree with the expected results thus verifying the

accuracy of the assembly of structure stiffness matrix and the equation solver used in this program.

5. Conclusion

This research presented the development of bilinear degenerated shell element with drilling degree of freedom using the object oriented programming concept in Java as an alternative to the traditional procedural programming approach. A finite element analysis program was developed to verify the accuracy of the results.

Some test example problems were analyzed using the developed program. The results from these analyses were compared with those obtained from Kansara (2004), the commercial finite element analysis program SAP 2000 and LISA respectively in order to verify the accuracy of the developed program. The results obtained from the analysis of the example problems were found to be very accurate when compared to those obtained from SAP 2000

References

[1] Adam, F. M. and Mohamed, A. E. (2013), Finite Element Analysis of Shell structures, LAP LAMBERT Academic Publishing.
[2] B. Irons and S. Ahmad,(1980) Techniques of Finite Elements. Ellis Horwood, Chichester, UK.
[3] Djermane, M., Chelghoum, A., Amieur, B. and Labbaci, B. (2006), Linear and Nonlinear Thin Shell Analysis Using A Mixed Finite Element with Drilling Degrees of Freedom,International Journal of Applied Engineering Research, Volume 1 Number 2 pp. 217-236
[4] Fathelrahman. M. Adam, Abdelrahman. E. Mohamed, A. E. Hassaballa (2013) Degenerated Four Nodes Shell Element with Drilling Degree of Freedom,, IOSR Journal of Engineering (IOSRJEN).
[5] Gallagher R. H., (1975) Finite Element Analysis Fundamentals, Prentice-Hall.
[6] G.P. Nikishkov, (2010) Programming Finite Elements in Java™, Springer-Verlag London Limited.
[7] Kansara, K. (2004), Development of Membrane, Plate and Flat Shell Elements in Java, M.sc. Thesis, Faculty of the Virginia Polytechnic Institute & State University, 2004.
[8] McNeal R. H., (1978) A Simple Quadrilateral Shell Element, Computers and Structures, Vol. 8, pp. 175-183.

Author Profile

Umeonyiagu Ikechukwu received his B.Eng and M.Eng degrees from University of PortHarcourt and Nnamdi Azikiwe University respectively and phD from Nnamdi Azikiwe University. He is currently the Dean Faculty of Engineering Chukwuemeka Odumuegwu Ojukwu University (COOU), Uli.

Ogbonna Nnamdi received HND and PGD in Civil Engineering from Federal Polytechnic Nekede, Owerri and Bayero University, Kano. He is a Java programmer and is currently an M.Eng student of Chukwuemeka Odumuegwu Ojukwu University,(COOU).