IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)
e-ISSN: 2278-1684,p-ISSN: 2320-334X, Volume 15, Issue 3 Ver. IV (May. - June. 2018), PP 85-94
www.iosrjournals.org

Development of Quadratic Triangular (T6) Elements 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)*
Corresponding author: I.E Umeonyiagu

Abstract: *This study developed Quadratic Triangular (T6) elements 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 which addresses the deficiencies of Constant Strain Triangle (CST) elements developed by Kansara (2004). 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. The results obtained from the analysis of the example problems were found to be very accurate when compared to those obtained from Kansara (2004), SAP 2000 and LISA. The difference in displacements and stresses computed by the two programs was very negligible.*
Keywords – *Finite Element Model, Object Oriented Approach, Java Programming Language, Quadratic Triangular Elements, Constant Strain Triangle.*

Date of Submission: 15-06-2018 Date of acceptance: 03-07-2018

I. Introduction

Since the evolution of the term finite element by Clough in 1951, there have been significant developments in finite element method. A large number of different finite elements have been developed (Kansara, 2004).

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.

Clough and Tocher (1965) developed the triangular plate bending element by dividing the main triangle into three subtriangles.

Green et al (1961) first developed the concept of using triangular flat shell elements to model arbitrary shaped shell structures.

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) advocated for a future research work that can handle most of the deficiencies inherent in his work.

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.

Bazeley *et al.* (1966) developed confirming and non-confirming plate bending elements. They developed a triangular plate bending element by using shape functions based on the area coordinates.

This work improved on the work of Kansara (2004) using the procedures prescribed by Nikishkov (2010). Six node Quadratic Triangular (T6) Elements also known as Linear Strain Triangle (LST) were developed and a finite element analysis program was also written 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.

II. Literature Review

2.1 *Quadratic Triangular (T6) Element*

T6 element is a higher-order triangular element, called the linear-strain triangle (LST). This element has six nodes and twelve displacement degrees of freedom. For a given number of nodes, a better representation of true stress and displacement is generally obtained using LST elements than is obtained using the same number of nodes a finer subdivision of CST elements.

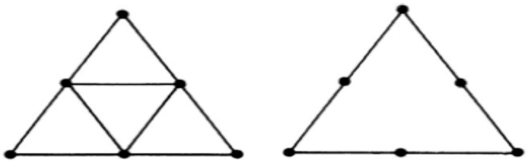

Figure 2.1: A single LST element gives better results than four CST elements.

The nodes of the LST element are numbered in a counterclockwise direction as shown in Figure 2.2.

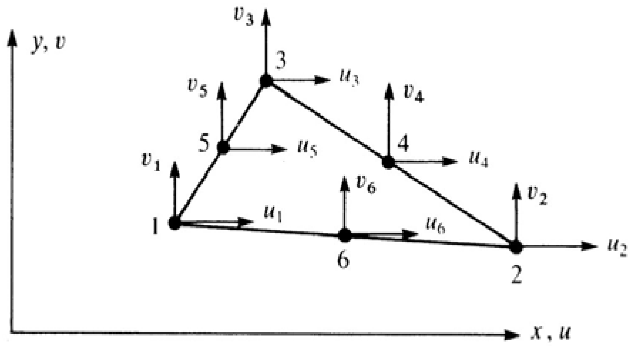

Fig. 2.2: Linear Strain Triangle (LST)

Each node has two degrees of freedom: displacements in the *x* and *y* directions. Let u_i and v_i represent the node *i* displacement components in the *x* and *y* directions, respectively.

The assumed displacement field can be defined by,

$$\left.\begin{aligned} u(x,y) &= a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2 \\ v(x,y) &= a_7 + a_8x + a_9y + a_{10}x^2 + a_{11}xy + a_{12}y^2 \end{aligned}\right\} \qquad (21)$$

Where *u*(*x*, *y*) is the displacement in the x direction, and *v* (*x*, *y*) is the displacement in the y direction.

The above equations can be written in matrix form as,

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} 1 & x & y & x^2 & xy & y^2 & \vdots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \vdots & 1 & x & y & x^2 & xy & y^2 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{11} \\ a_{12} \end{Bmatrix} \qquad (2.3)$$

Or

$$\{U(x,y)\} = [X]\{a\} \qquad (2.4)$$

The strains are found to be,

$$\left.\begin{aligned} \varepsilon_x &= b_2 + 2b_4x + b_5y \\ \varepsilon_y &= b_9 + b_4x + b_5y \\ \gamma_{xy} &= (b_3 + b_8) + (b_5 + 2b_{10})x + (2b_6 + b_{11})y \end{aligned}\right\} \qquad (2.5)$$

which are linear functions. Thus, we have the “linear strain triangle” (LST), which provides better results than the CST. In the natural coordinate system, the six shape functions for the LST element are,

$$\left.\begin{aligned} N_1 &= \xi(2\xi - 1) \\ N_2 &= \eta(2\eta - 1) \\ N_3 &= \zeta(2\zeta - 1) \\ N_4 &= 4\xi\eta \\ N_5 &= 4\eta\zeta \\ N_6 &= 4\zeta\xi \end{aligned}\right\} \qquad (2.6)$$

In which $\zeta = 1 - \xi - \eta$

Each of these six shape functions represents a quadratic form on the element as shown in figure 2.3.

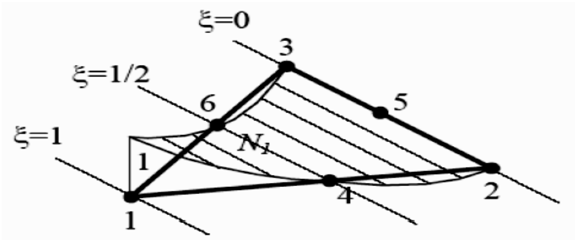

Fig. 2.3: Linear Strain Triangle shape functions

Displacements can be written as,

$$u = \sum_{i=1}^{6} N_i u_i \tag{2.4}$$

$$v = \sum_{i=1}^{6} N_i v_i \tag{2.4}$$

III. 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 postprocessing (results calculation and visualization). Rfea finite element system is organized into eight class packages. 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.

3.1.2 Package model

This package contain finite element model and loading classes

3.1.3 Package util

Utility classes which include the FePrintWriter, FeScanner, GaussRule, UTIL

3.1.4 Package elem

This package contains classes such ElementT6., ShapeT6.

3.1.5 Package material

This package contains the constitutive relations for materials.

3.1.6 Package solver

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

3.1.7 Package gener

The classes of this package will be used for generation of mesh

3.1.8 Package visual

Contains classes visualization of models and results. 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 Quadratic Triangular (T6) Elements*

3.2.1 Formulation of Shape functions

Consider a straight-sided triangular element shown in fig 3.1:

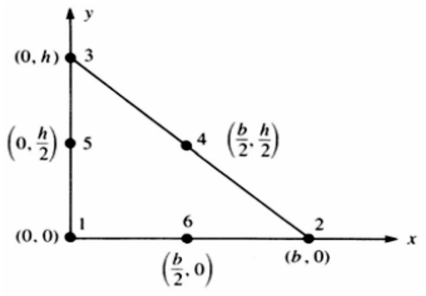

Fig 3.1

The triangle has a base dimension of *b* and a height *h*, with midside nodes. The general form of the displacement expressions in terms of the interpolation functions is given as

$$\begin{Bmatrix} u \\ v \end{Bmatrix} \begin{bmatrix} N_1 & 0 & \vdots & N_2 & 0 & \vdots & N_3 & 0 & \vdots & N_4 & 0 & \vdots & N_5 & 0 & \vdots & N_6 & 0 \\ 0 & N_1 & \vdots & 0 & N_2 & \vdots & 0 & N_3 & \vdots & 0 & N_4 & \vdots & 0 & N_5 & \vdots & 0 & N_6 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ \vdots \\ u_6 \\ u_6 \end{Bmatrix} \tag{3.1}$$

Where the interpolation functions are

$$\left.\begin{aligned} N_1 &= 1 - \frac{3x}{b} - \frac{3y}{h} + \frac{2x^2}{b^2} + \frac{4xy}{bh} + \frac{2y^2}{h^2} \\ N_2 &= -\frac{x}{b} + \frac{2x^2}{b^2} \\ N_3 &= -\frac{y}{h} + \frac{2y^2}{h^2} \\ N_4 &= \frac{4xy}{bh} \\ N_5 &= \frac{4y}{h} - \frac{4xy}{bh} - \frac{4y^2}{h^2} \\ N_6 &= \frac{4x}{b} - \frac{4xy}{bh} - \frac{4x^2}{b^2} \end{aligned}\right\} \tag{3.2}$$

The shape function N_1 is shown below:

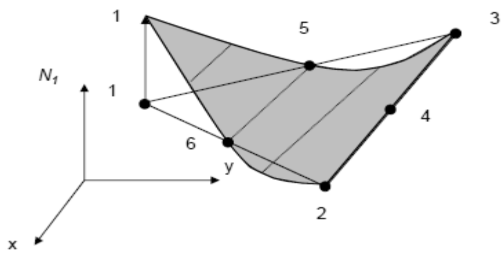

Fig 3.2: Interpolation function N_1

The second type of interpolation function is valid for functions N_4, N_5, and N_6. The function N_5 is shown below:

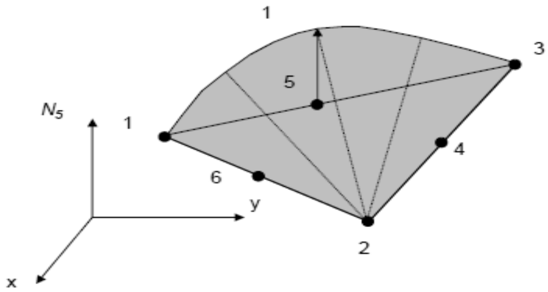

Fig 3.3: Interpolation function N_5

3.2.2 Calculation of strain-displacement matrix

The element strain is given as:

$$\{\varepsilon\} = [B]\{d\} \tag{3.3}$$

Where the $[B]$ matrix is:

$$[B] = \frac{1}{2A}\begin{bmatrix} \beta_1 & 0 & \beta_2 & 0 & \beta_3 & 0 & \beta_4 & 0 & \beta_5 & 0 & \beta_6 & 0 \\ 0 & \gamma_1 \vdots & 0 & \gamma_2 \vdots & 0 & \gamma_3 \vdots & 0 & \gamma_4 \vdots & 0 & \gamma_5 \vdots & 0 & \gamma_6 \\ \gamma_1 & \beta_1 & \gamma_2 & \beta_2 & \gamma_3 & \beta_3 & \gamma_4 & \beta_4 & \gamma_5 & \beta_5 & \gamma_6 & \beta_6 \end{bmatrix}$$

$$\left.\begin{aligned} \beta_i &= 2A\left(\frac{\partial N_i}{\partial x}\right) \\ \gamma_i &= 2A\left(\frac{\partial N_i}{\partial y}\right) \end{aligned}\right\} \tag{3.4}$$

Therefore since $\beta_i = 2A\left(\frac{\partial N_i}{\partial x}\right)$

$$A = \frac{bh}{2}$$

$$\left.\begin{gathered} \beta_1 = -3h + \frac{4hx}{b} + 4y \\ \beta_2 = -h + \frac{4hx}{b} \\ \beta_3 = 0 \\ \beta_4 = 4y \\ \beta_5 = -4y \\ \beta_6 = 4h - \frac{8hx}{b} - 4y \end{gathered}\right\} \tag{3.5}$$

Therefore since $\gamma_i = 2A\left(\frac{\partial N_i}{\partial y}\right)$

$$A = \frac{bh}{2}$$

$$\left.\begin{gathered} \gamma_1 = -3b + \frac{4by}{h} + 4x \\ \gamma_2 = 0 \\ \gamma_3 = -b + \frac{4by}{h} \\ \gamma_4 = 4x \\ \gamma_5 = 4b - 4x - \frac{8by}{h} \\ \gamma_6 = -4x \end{gathered}\right\} \tag{3.6}$$

3.2.3 Calculating Element Stiffness Matrix

The stiffness matrix for a constant thickness element can be obtained by substituting the β's and the γ's into the $[B]$ and then substituting $[B]$ into the following expression and evaluating the integral numerically.

$$[k] = \int_v [B]^T[D][B]dv \tag{3.7}$$

3.2.4 Class ElementT6

Class ElementT6 is created to extend class Element and to implement methods for computing element matrices and vectors. The following arrays are declared:

an – shape functions; dnxy – derivatives of shape functions with respect to global coordinates *x*, *y* (first index is related to node number, second index to *x* and *y*);bmat – displacement differentiation matrix, emat – elasticity matrix; ept – vector of thermal strains.

`GaussRule` objects for integration of the element stiffness matrix, thermal vector, surface load, and equivalent stress vector are created

Constructor `ElementT6(){}` calls the constructor of parent class `Element` and passes to it the element name `lst`, the number of element nodes (6) and the number of points for storing stresses and strains. Method `stiffnessMatrixLST() performs the` computation of the element stiffness matrix according to Equation (3.7) and the stiffness matrix `kmat` is set to zero.

Integer variable `ld` represents a length of the strain or stress vector. For plane problems `ld` is equal to 3. Extraction of `Material` object `mat` is done from the hash table `materials` using the material name. Numerical integration of the element stiffness matrix is performed in a loop with a parameter `ip` denoting integration point number. Integration is performed inside a single loop. Other important methods and their description are given in table 3.2

Table 3.2 Methods in the ElementDkQ8 class

Method	*Description*
`setBmatrixT6 ()`	Performs computation of a displacement differentiation matrix `bmat` for specified local coordinates `xi` and `et` and returns the determinant of the Jacobian matrix.
`DeriveT6 ()`	Computes the derivatives of shape functions `dnxy` with respect to global coordinates *x*, *y*
`ElasticityMatrixT6 ()`	Sets the elasticity matrix `emat`.
`equivFaceLoadT6 ()`	Computing a nodal equivalent of surface load
`RearrangeT6 ()`	Used to put load information in order, corresponding to local element numbering
`shapeDerivFaceT6 ()`	Provides one-dimensional shape functions `an` and derivatives of shape functions `xin` with respect to local coordinate ξ changing along the considered element side.
`equivStressVectorT6 ()`	Computation of a nodal force vector, which is equivalent to element stress field.
`extrapolateToNodesT6 ()`	Stress extrapolation from integration points to nodes
`getElemFacesT6 ()`	returns local numbers for six element sides specified in array `faceInd`
`getStrainsAtIntPointT6 ()`	returns strains at the requested integration point `ip`.

3.2.5 Class ShapeT6

Class `ShapeT6` is placed in package `elem`. It is created to calculate the shape functions for T6 elements. Element nodes are numbered in an anticlockwise direction starting from any corner node.

Method `shapeT6()` 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.

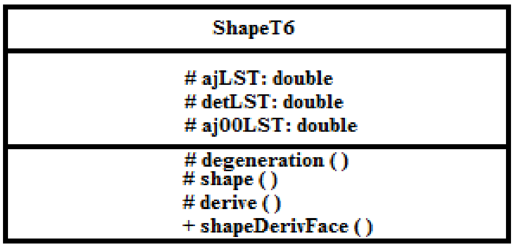

Figure 3.4: UML diagram for ShapeT6 with its attributes and operations

Derivatives of shape functions with respect to global coordinates *x*, *y* are obtained by multiplication of the Jacobian matrix and derivatives with respect to local coordinates ξ , η

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

IV. Numericalanalysis

4.1 Verification of Six Node Quadratic Triangular (T6) Element.

4.1.1 Finite element Analysis of a cantilever beam using eight-six node quadratic triangular (T6) elements

This test example consists of a cantilever beam of length 48 in., depth 12 in. and thickness of 1 in. The beam is modeled using 8 six node triangular (LST) elements. Vertical loads of 20 kips are applied at the free end of the cantilever (nodes 9 and 27). Fig. 4.1 shows the finite element model of the cantilever beam.

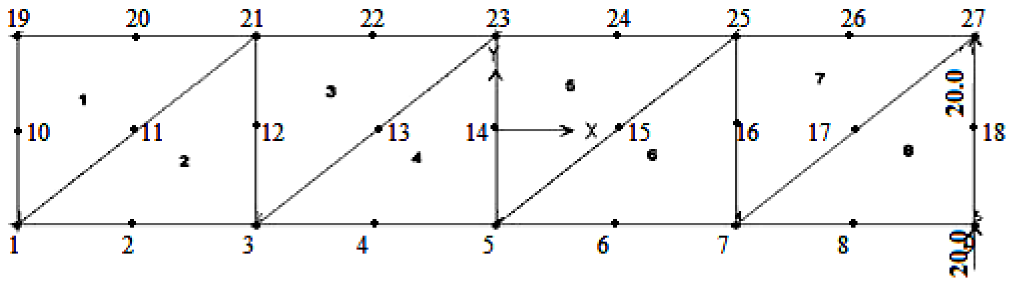

Fig. 4.1 FE Model for Test Example 1 – Cantilever Beam.

Geometric Data:
Length L = 48.0 in.
Depth h = 12 in.
Thickness t = 1.0 in.

Material Properties:
Modulus of elasticity E = 30000 ksi.
Poisson's ratio n = 0.25

Boundary Conditions:
Restraints are provided in the x and y directions at the left end of the cantilever (nodes 1 and 19).

Loading:
A concentrated load of 20 kips is applied to nodes 9 and 27.

Comparison of Results:

The results from the analysis obtained from the LST elements using the program and from CST elements Kansara (2004) are shown in Table 4.1. The results shown are the displacements at nodes 27 and 23, and stresses at node 19. From Table 4.1, it is seen that the results given by the program are higher which points to the fact that the larger the number of degrees of freedom for a given type of triangular element, the closer the solution converges to the exact one.

Table 4.1 Displacements and Stresses for Test Example 1

Location		**CST Element Kansara (2004)**	**T6 Element Programe Rfea**	**Difference**
Node - 27	**UX**	-0.014150	-0.039153	0.025003
	UY	0.090347	0.111347	0.021000
Node - 23	**UX**	-0.010825	-0.050825	0.040000
	UY	0.030403	0.066603	0.003620
Node- 19	**S12**	-17.128727	-21.33855	4.209823
	S22	-4.282182	-5.538562	1.256380
	S12	9.537940	11.571534	2.033594

4.1.2 Finite element Analysis of a cantilever beam using thirty two-six node quadratic triangular elements

The second test example consists of the same cantilever beam of Example 1. The length of the beam is 48 in., depth is 12 in. and the thickness is 1 in. The finite element model was previously with 32 three node triangular (CST) elements but now consist of 32 three node (T6) elements. Vertical loads of 40 kips is applied at the free end of the cantilever beam. The finite element model of the cantilever beam as shown in Fig. 4.2.

DOI: 10.9790/1684-1503048594

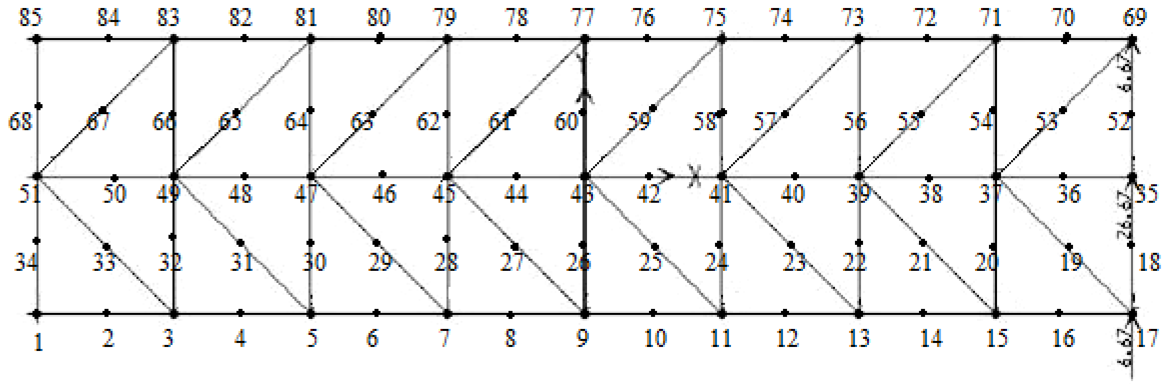

Fig. 7.2 FE Model for a thirty two-six node quadratic triangular elements

Geometric Data:
Length L = 48.0 in.
Depth h = 12 in.
Thickness t = 1.0 in.

Material Properties:
Modulus of elasticity E = 30000 ksi.
Poisson's ratio n = 0.25

Boundary Conditions:
Restraints are provided in the x and y directions at the left end of the cantilever.

Loading:
A concentrated load P_1 = 6.67 kips is applied at node 69.
A concentrated load P_2 = 26.67 kips is applied at node 35.
A concentrated load P_3 = 6.67 kips is applied at node 17.

Comparisons of Results:

Table 4.2 represents the results obtained for the second test example from the developed program and SAP 2000. The displacements at nodes 69 and 77 and stresses at node 85 are shown. The results indicate higher values from T6 elements showing fast convergence to the exact solution.

Table 4.1 Displacements and Stresses for Test Example 1

Location		**CST Element Kansara (2004)**	**T6 Element Programe Rfea**	**Difference**
Node - 69	**UX**	-0.034263	-0.063416	0.029153
	UY	0.194412	0.216732	0.022320
Node - 77	**UX**	-0.025599	-0.0643041	0.0387051
	UY	0.062956	0.102456	0.003950
Node- 85	**S12**	-41.493731	-47.90255	6.408820
	S22	-10.373433	-12.131518	1.758085
	S12	11.840936	14.010997	2.170061

4.1.3 Finite element Analysis of a Semi Circular Hole

The third verification example is a plate with semi circular hole of radius 3 in. at the center. The plate is fixed at the top. A downward vertical load of 0.67 kips/in is applied at the free edge. The length of the plate is 16 in., the width is 6 in. and the thickness is 0.45 in. The plate was previously modeled using 110 three node triangular (CST) elements, now modified to 110 six node triangular (T6) elements, The FE model is shown in Fig 4.3.

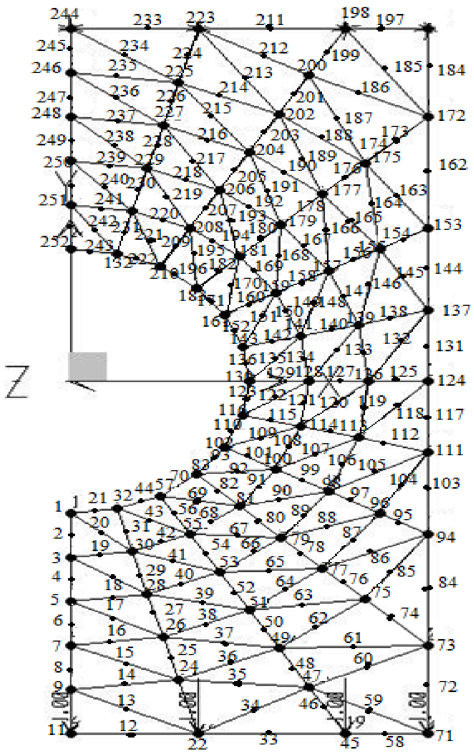

Fig. 7.3 FE Model for Test Example 3 – Plate with Semi Circular Hole.

Geometric Data:
Width L = 16.0 in.
Width b = 6.0 in.
Thickness t = 0.45 in.
Material Properties:
Modulus of elasticity E = 30000 ksi.
Poisson's ratio n = 0.3
Boundary Conditions:
The plate is restrained at the top in both the x and y directions.
Loading:
Concentrated loads of 1.0 kips are applied at each node at the bottom of the plate. i.e. nodes 11, 22, 45, and 71.
Comparison of Results:
The displacements at nodes 1 and 11 and stresses at nodes 116 and 130 are given in Table 4.3. As can be seen from the table, results obtained from the developed program converges close to the exact solution. Similar results are obtained for the other nodes.

Table 4.1 Displacements and Stresses for Test Example 1

Location		CST Element Kansara (2004)	T6 Element Programe Rfea	Difference
Node - 1	**UX**	0.001545	0.005155	0.003610
	UY	-0.003132	-0.005732	0.002600
Node - 11	**UX**	0.004213	0.008113	0.003900
	UY	-0.003355	-0.006975	0.003620
Node- 116	**S11**	1.250061	2.359682	1.109621
	S22	7.050394	8.235704	1.185310
	S12	0.480698	0.614292	0.133594
Node- 130	**S11**	1.156083	2.012463	0.856380
	S22	7.547362	8.757185	1.209823

V. Conclusions

This research presented the development of Six Node Quadratic Triangular (T6) elements 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 results obtained from the analysis of the example problems were found to converge faster to the exact solution when compared to those obtained from Kansara(2004). In general, both the T6 and CST analyses yield sufficient results for most plane stress/strain problems provided a sufficient number of elements are used. The T6 model is preferred over the CST model for plane stress applications when a relatively small number of nodes is used.

References

[1]. 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.

[2]. McNeal R. H., (1978) A Simple Quadrilateral Shell Element, Computers and Structures, Vol. 8, pp. 175-183.

[3]. Clough R. W. and Tocher J. L.,(1965) Finite Element Stiffness Matrices for Analysis of Plate Bending, *Proc. Conference on Matrix Methods in Structural Mechanic*, WPAFB, Ohio, pp. 515-545.

[4]. Green B. E., Strome D. R., and Weikel R. C.,(1961) Application of the stiffness method to the analysis of shell structures, *Procedures on Aviation Conference, American Society of Mechanical Engineers*, Los Angeles, March.

[5]. Nikishkov, G.P., (2010) Programming Finite Elements in Java™, Springer-Verlag London Limited.

[6]. Bazeley, G. P., Cheung Y. K., Irons B. M. and Zienkiewicz, O. C., (1966) Triangular Elements in Plate Bending, Confirming and Non – Confirming Solutions, *Proc. 1st Conference on Matrix Methods in Structural Mechanics*, pp. 547-576, Wright Patterson AF Base, Ohio.

I.E Umeonyiagu "Development of Quadratic Triangular (T6) Elements in Java." IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE) , vol. 15, no. 3, 2018, pp. 85-94.