Development of a quadratic structural solver on a 10 node tetrahedral element for FSI applications.

KAUSHIK KUMAR NAGARAJAN

CTFD, National Aerospace Laboratories, Bengaluru

Abstract :

The objective of this report is to present the development of an FEM based structural solver on a 10 node tetrahedron. In the previous report the limitation of the linear solver for bending problems of aero-elasticity was discussed and shown to be inaccurate. The numerical formulation for the calculation of the stiffness matrix is discussed. Test cases which measure the accuracy of the solver as compared to the linear solver is presented.

Keywords : FEM, 3d, Beam problems, Tetrahedral meshes, quadratic solver.

1 Introduction

Fluid Structure Interaction problems has been an active area of research and encompasses a wide range of applications such as turbomachinary, aeroelasticity, bio mechanics to name a few. The inputs of the interaction is a vital component in the design of aircraft wings, turbomachinary and compressor blades, high rise civil structures. Although a good number of structural solvers are available commercially they are not a viable when used for large scale problems. Also coupling this solvers with the in house solvers is difficult. Open source solvers are still in the developmental stages as far as FSI problems are concerned [2]. Hence there is a need to develop a structural solver which integrates with the fluid solver. A 3d solver on tetrahedral meshes was presented in the earlier report [7]. Although the tetrahedral elements was shown to work well for compression problems, the solver failed to give accurate results for problems involving beam bending. Also an error of around 33% with respect to the actual solution was reported. Mesh refinement which normally resolves this inaccuracy shows that the linear system tend to become very stiff with increasing mesh size. Also the solution does not improve considerably. This has also been widely reported by other FEM users who have reported large errors due to the tetrahedral elements. A detailed comparison of the various elements used in bending problems can be found in [6] where the linear elements are not recommended. Another issue with the linear elements is that the frequency predicted for the modal analysis are way off from the actual value. This necessitates the

development of a higher order solver. Tetrahedral meshes (also denoted by tet mesh from here onwards) are the most generic type which is available in almost all the grid generators. An advantage of a tet mesh is its ease when used for FSI problems as the surface meshes generated for the CFD solutions can be used as an surface interface for the structural solver. Unfortunately all the CFD grid generators which are employed in the CTFD division for example POINTWISE® just support linear elements in the current version. Generation of higher elements then is a challenge. A mesh generation tool which takes the tet mesh and generates the higher order mesh was found on [8]. The code was corrected for errors in discussion with the developer and is used in the current work. We present the mathematical details of the stiffness computation on the higher order mesh (also referred to as the tet10 mesh) and give the results for the beam bending problem which was earlier solved for the linear mesh. A very good agreement with the analytical solution is obtained.

2 3d problems and tet10 geometry

Although the governing equations are discussed in the previous report [7] we present them here again for the sake of continuity. The element stiffness matrix derivation follows closely the derivation given in [4]. The governing equation for the plane elasticity problems are nothing but the Newton's laws of motion given by

$$\rho_s \frac{\partial^2 u}{\partial t^2} = \nabla(D.S) + \rho_s f \; in \; \Omega \times [0, T] \tag{1}$$

where ρ_s the material density D represents the deformation gradient operator and S the Cauchy stress tensor. f represents the external force and u represents the material displacement [2]. The deformation gradient D is given by

$$D = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \tag{2}$$

The Cauchy stress tensor is given by σ_{ij} where the indical representation denotes the coordinates (x, y, z). The strain fields are related to the displacements by the relation $e = D\ u$. where the strain field e is given by $e_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})$. Here $u_{i,j}$ denotes the partial derivative of u_i with respect to j. The six independent strain components can be represented as

$$\epsilon = [e_{xx}, e_{yy}, e_{zz}, e_{xy}, e_{yz}, e_{zx}] \tag{3}$$

The stress fields are related to the strain field by the constitutive relation given as

$$\sigma = \boldsymbol{E}\epsilon \tag{4}$$

where $\boldsymbol{E}$ is the symmetric elasticity matrix representing the material property and is given by

$$\boldsymbol{E} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}-\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}-\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}-\nu \end{bmatrix} \tag{5}$$

Here E denotes the Young's modulus of the material and ν the Poisson ratio. Before deriving the FEM formulation of (1) it is worthwhile to remember the principle of virtual work [1] which states the balance of the internal work to the external work for the equilibrium of the body subjected to any compatible displacements δu so as to minimize the total strain energy. Stated mathematically the principle of virtual work can be written as

$$\int_{\Omega_e} \left(\epsilon\sigma + \rho_s \delta u \frac{\partial^2 u}{\partial t^2} \right) d\Omega - \int_{\Omega_e} \delta u \; f d\Omega - \int_{\Gamma_e} \delta u \; t d\Gamma_e = 0 \tag{6}$$

t represents the surface traction force and Γ_e denotes the boundary of the element e. To obtain the FEM formulation we expand the displacements u in a finite element basis

$$u = \sum_{i=0}^{N} u_i \psi_i \tag{7}$$

and substitute into the (6) and equating the variation δu to zero, to obtain the elemental FEM formulation as

$$M^e \frac{\partial^2 u}{\partial t^2} + K^e u = V^e + F^e \tag{8}$$

where M^e, K^e, V^e, F^e represent the element mass matrix, stiffness matrix, volume force vector and surface force vector respectively.

$$\begin{aligned} M^e &= \int_{\Omega_e} \psi^t \psi \; d\Omega \\ K^e &= \int_{\Omega_e} D^t(\psi) E D(\psi) \; d\Omega \\ V^e &= \int_{\Omega_e} \psi^t \; f d\Omega \\ F^e &= \int_{\Gamma_e} \psi^t f d\Gamma_e \end{aligned} \tag{9}$$

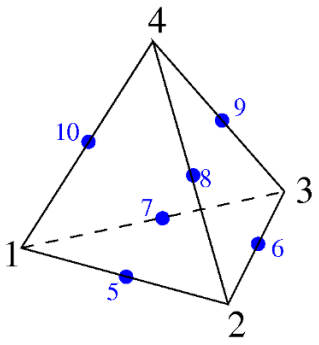

Figure 1: A schematic representation of a 4 node Tetrahedral element .

To construct the shape function of the tetrahedral element consider the representation shown in figure 1. let the natural coordinates of the tetrahedral be denoted by (x_i, y_i, z_i) $i = 1, 2, \ldots 10$ We denote the coordinate differences by

$$x_{ij} = x_i - y_j, \;\; y_{ij} = y_i - y_j \;\; z_{ij} = z_i - z_j, \;\; i, j = 1, ..4 \tag{10}$$

the volume of the tetrahedral V is then given by

$$\begin{aligned} 6V &= \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{vmatrix} \\ &= x_{21}(y_{23}z_{34} - y_{34}z_{23}) + x_{32}(y_{34}z_{12} - y_{12}z_{34}) + x_{43}(y_{12}z_{23} - y_{23}z_{12}) \end{aligned} \tag{11}$$

the position of any point inside a tetrahedron can be specified by the natural coordinates (x, y, z) or by the set of transformed tetrahedral coordinates

$$\xi_1, \xi_2, \xi_3, \xi_4 \tag{12}$$

where $\xi_i = \frac{V_{0i}}{V}$ and V_{0i} is the volume enclosed by the origin and the face opposite to vertex i. For a linear element the elemental shape function ψ_i for a linear tetrahedron is indeed ξ_i. The elemental shape function for the 10 node tetrahedron can be written in terms of the linear tetrahedral coordinates as

$$\text{Corner nodes:} \psi_i = \xi_i \left(2\xi_i - 1\right) \text{Mid-point nodes :} \psi_k = 4\xi_i\xi_j \tag{13}$$

where i, j denotes the corner nodes. We now consider a scalar function expressed in tetrahedral coordinates as $F(\xi_i, \xi_2, \xi_3, \xi_4)$. Before computing the stiffness matrix we express the partial

derivative of F with respect to the cartesian coordinates x_i i,e $\frac{\partial F}{\partial x_i}$. We know that the scalar function F over an element e expressed in terms of the elemental shape function ψ_i^e can be written as

$$F = F_i \, \psi_i^e \tag{14}$$

where F_i denotes the value of F at the i^{th} node. The partial derivative of F with respect to the cartesian x_i can be written as

$$\frac{\partial F}{\partial x_i} = F_k \frac{\partial \psi_k}{\partial \xi_j} \frac{\partial \xi_j}{\partial x_i} \tag{15}$$

The index of sums are $k = 1, 2, \ldots, 10$, $j = 1, \ldots 4$. The subscript e has been omitted for the sake of notational clarity. In matrix form we can write the above as

$$\left[F_k \frac{\partial \psi_k}{\partial \xi_1}, F_k \frac{\partial \psi_k}{\partial \xi_2}, F_k \frac{\partial \psi_k}{\partial \xi_3}, F_k \frac{\partial \psi_k}{\partial \xi_4} \right] \begin{bmatrix} \frac{\partial \xi_1}{\partial x} & \frac{\partial \xi_1}{\partial y} & \frac{\partial \xi_1}{\partial z} \\ \frac{\partial \xi_2}{\partial x} & \frac{\partial \xi_2}{\partial y} & \frac{\partial \xi_2}{\partial z} \\ \frac{\partial \xi_3}{\partial x} & \frac{\partial \xi_3}{\partial y} & \frac{\partial \xi_3}{\partial z} \\ \frac{\partial \xi_4}{\partial x} & \frac{\partial \xi_4}{\partial y} & \frac{\partial \xi_4}{\partial z} \end{bmatrix} = \left[\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z} \right] \tag{16}$$

We now set F to x, y and z and stack them along the row. Also noting that $\xi_1 + \xi_2 + \xi_3 + \xi_4 = 1$ and differentiating with respect to x, y and z and inserting in the first row to get

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ x_k \frac{\partial \psi_k}{\partial \xi_1} & x_k \frac{\partial \psi_k}{\partial \xi_2} & x_k \frac{\partial \psi_k}{\partial \xi_3} & x_k \frac{\partial \psi_k}{\partial \xi_4} \\ y_k \frac{\partial \psi_k}{\partial \xi_1} & y_k \frac{\partial \psi_k}{\partial \xi_2} & y_k \frac{\partial \psi_k}{\partial \xi_3} & y_k \frac{\partial \psi_k}{\partial \xi_4} \\ z_k \frac{\partial \psi_k}{\partial \xi_1} & z_k \frac{\partial \psi_k}{\partial \xi_2} & z_k \frac{\partial \psi_k}{\partial \xi_3} & z_k \frac{\partial \psi_k}{\partial \xi_4} \end{bmatrix} \begin{bmatrix} \frac{\partial \xi_1}{\partial x} & \frac{\partial \xi_1}{\partial y} & \frac{\partial \xi_1}{\partial z} \\ \frac{\partial \xi_2}{\partial x} & \frac{\partial \xi_2}{\partial y} & \frac{\partial \xi_2}{\partial z} \\ \frac{\partial \xi_3}{\partial x} & \frac{\partial \xi_3}{\partial y} & \frac{\partial \xi_3}{\partial z} \\ \frac{\partial \xi_4}{\partial x} & \frac{\partial \xi_4}{\partial y} & \frac{\partial \xi_4}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial 1}{\partial x} & \frac{\partial 1}{\partial y} & \frac{\partial 1}{\partial z} \\ \frac{\partial x}{\partial x} & \frac{\partial x}{\partial y} & \frac{\partial x}{\partial z} \\ \frac{\partial y}{\partial x} & \frac{\partial y}{\partial y} & \frac{\partial y}{\partial z} \\ \frac{\partial z}{\partial x} & \frac{\partial z}{\partial y} & \frac{\partial z}{\partial z} \end{bmatrix} \tag{17}$$

On noting the independence of coordinates x, y and z the above equation can be written in a compact form as

$$\mathbf{J} P = I_{aug} \tag{18}$$

where I_{aug} is the augmented unit matrix given by

$$I_{aug} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{19}$$

The matrix P which represents the derivative of the shape function is obtained by solving equation

(18) as $P = \mathbf{J}^{-1} I_{aug}$. which can be written in the matrix notation as

$$P = \begin{bmatrix} \frac{\partial \xi_1}{\partial x} & \frac{\partial \xi_1}{\partial y} & \frac{\partial \xi_1}{\partial z} \\ \frac{\partial \xi_2}{\partial x} & \frac{\partial \xi_2}{\partial y} & \frac{\partial \xi_2}{\partial z} \\ \frac{\partial \xi_3}{\partial x} & \frac{\partial \xi_3}{\partial y} & \frac{\partial \xi_3}{\partial z} \\ \frac{\partial \xi_4}{\partial x} & \frac{\partial \xi_4}{\partial y} & \frac{\partial \xi_4}{\partial z} \end{bmatrix} = \frac{1}{J} \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ a_4 & b_4 & c_4 \end{bmatrix} \tag{20}$$

where J is the determinant of the Jacobian matrix $\mathbf{J}$. The shape function derivatives are then given by $\frac{\partial \xi_i}{\partial x} = a_i/J$, $\frac{\partial \xi_i}{\partial y} = b_i/J$, $\frac{\partial \xi_i}{\partial z} = c_i/J$. The partial derivative of any function $F(x, y, z)$ is then given by

$$\frac{\partial F}{\partial x} = F_k \frac{\partial \psi_k}{\partial \xi_i} \frac{a_i}{J}, \ \frac{\partial F}{\partial y} = F_k \frac{\partial \psi_k}{\partial \xi_i} \frac{b_i}{J}, \ \frac{\partial F}{\partial z} = F_k \frac{\partial \psi_k}{\partial \xi_i} \frac{c_i}{J} \tag{21}$$

The range of the index above is $i = 1, 2, 3, 4$ and $k = 1, 2, \ldots, 10$. The element volume integral of the function is given by

$$\int_{\Omega_e} F(x, y, z) d\Omega_e = \int_{\Omega_e} F(x, y, z) dx dy dz = \int_{\Omega_e} F(\xi_1, \xi_2, \xi_3, \xi_4) \frac{1}{6} J d\xi_1 d\xi_2 d\xi_3 d\xi_4 \tag{22}$$

The shape function Cartesian derivatives are given by

$$\frac{\partial \psi_n}{\partial x} = (4\xi_n - 1)\frac{a_n}{J}, \ \frac{\partial \psi_n}{\partial y} = (4\xi_n - 1)\frac{b_n}{J}, \ \frac{\partial \psi_n}{\partial z} = (4\xi_n - 1)\frac{c_n}{J} \tag{23}$$

$$\frac{\partial \psi_m}{\partial x} = 4\frac{a_i \xi_j + a_j \xi_i}{J}, \ \frac{\partial \psi_m}{\partial y} = 4\frac{b_i \xi_j + b_j \xi_i}{J}, \ \frac{\partial \psi_m}{\partial z} = 4\frac{c_i \xi_j + c_j \xi_i}{J} \tag{24}$$

where n denotes the corner nodes. m denotes the mid-point nodes (5,6,7,8,9,10) joining the vertex pair (i, j). The element stiffness matrix can be now written for an element with constant elastic modulus as

$$K^e = \int_{\Omega_e} D^t(\psi) E D(\psi) \, d\Omega_e \tag{25}$$

Denoting the matrix integrand $F(\xi_1, \xi_2, \xi_3, \xi_4) = D^t(\psi) E D(\psi) \frac{1}{6} J$, a 4-point quadrature rule for the stiffness matrix gives

$$K^e = \sum_{i=1}^{4} w_k F(\xi_{1k}, \xi_{2k}, \xi_{3k}, \xi_{4k}) \approx \frac{1}{4} F(a, b, b, b) + \frac{1}{4} F(b, a, b, b) + \frac{1}{4} F(b, b, a, b) + \frac{1}{4} F(b, b, b, a) \tag{26}$$

where $a = (5 + 3\sqrt{5})/20, \ \ b = (5 - \sqrt{5})/20$ denote the Gaussian quadrature points for the tetrahedral volume. w_k represents the weights at the quadrature points and is equal to $1/4$ for a

4 noded quadrature. To verify the stiffness computation it is always useful to test on a particular element where the results are known. Also one more validation for a 10 node element is that six of the eigenvalue of the stiffness matrix has to be zero and the others positive. A standard tetrahedron with the coordinates given by $(2, 3, 4), (6, 3, 2), (2, 5, 1), (4, 3, 6)$ as given in the reference [4] was chosen. The eigenvalues plotted in figure 2 for the stiffness matrix K shows that exactly 6 of the eigenvalues are zero as in the reference.

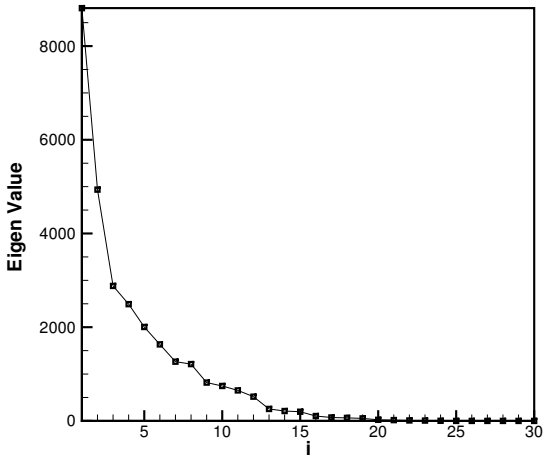

Figure 2: Eigenvalues of the stiffness matrix K

To compute the body forces due to any field variables it is necessary to distribute the field displacements to the nodes. if the body field such as gravity is given by

$$b = [b_x, b_y, b_z]^T \tag{27}$$

The body force vector is given in terms of the elemental shape functions as

$$f^e = \int_{\Omega_e} \psi^T b \, d\Omega_e \tag{28}$$

where ψ is the shape function matrix given by

$$\psi = \begin{bmatrix} \xi_1 & 0 & 0 & \cdots & \xi_{10} & 0 & 0 \\ 0 & \xi_1 & 0 & \cdots & 0 & \xi_{10} & 0 \\ 0 & 0 & \xi_1 & \cdots & 0 & 0 & \xi_{10} \end{bmatrix} \tag{29}$$

To calculate the surface integral is to calculate is basically a specific case of integrating a function over a surface defined in terms of two parametric coordinates. Lets first consider this general case. A fundamental relation from differential geometry gives the expression for the elemental surface measure as

$$d\mathbf{\Gamma_e} = \left\{ \begin{array}{c} \frac{\partial x}{\partial \xi_1} \\ \frac{\partial y}{\partial \xi_1} \\ \frac{\partial z}{\partial \xi_1} \end{array} \right\} \times \left\{ \begin{array}{c} \frac{\partial x}{\partial \xi_2} \\ \frac{\partial y}{\partial \xi_2} \\ \frac{\partial z}{\partial \xi_2} \end{array} \right\}$$

where ξ_1 and ξ_2 are the two parametric coordinates of the surface. An integral of a function, f over the surface would be written as

$$\int f d\Gamma_e = \int f(\xi_1, \xi_2) ||d\mathbf{\Gamma_e}||_2 d\xi_1 d\xi_2$$

where $||d\mathbf{\Gamma_e}||_2$ is just the magnitude of the vector.
Now lets consider the specific case of faces of the ten-node tetrahedron. In an isoparametric formulation, x, y, and z are written in terms of the shape functions, ψ_I, e.g.

$$x = \psi_I(\xi_1, \xi_2) x_I$$

where x_I are the values of x at the nodes of the triangular face. The expressions for ψ_e on a face can be obtained from the element shape functions by specializing them for a particular face, or equivalently, by using the shape functions for the six-node triangle. From these equations,one can easily evaluate the vector cross product above. The most general way to actually perform the integration is by using the appropriate Gauss-Legendre rule for the triangle [9].

3 Results

A C++ code has been developed to implement the method explained above. The EIGEN sparse library has been used to solve the global system. After having performed the traction case we then decided to test the code on an uniformly loaded cantilever beam. The tetrahedral mesh was generated using POINTWISE. The advantage of this grid generator is the seamless integration with the fluid solvers as a common grid generation tool. However the main disadvantage is that the current version only supports linear elements. A mesh generation code which takes the linear tet mesh and generates the higher order elements as in [8] was used. A total of around 9384 elements with around 2411 nodes was generated using the mesh generator. After higher order refinement a total of 7831 nodes was obtained. The number of elements in both the linear and quadratic grids remain the same. Only the mid nodes points are generated and numbered to have a consistent edge compatibility of the nodes. The Length of the beam $L = 20m$, breadth $B = 10m$ and depth $D = 1m$. The Young's modulus of the material is chosen as $E = 70KPa$ and Poisson ratio $\nu = \frac{1}{3}$. A uniform load of $F = -1750N/m$ is applied on the top surface. This test case is specially chosen as it represent a typical wing geometry on which the aerodynamic loads are given by a CFD solver which is used to compute t surface deflection and velocities which is fed-back

into the CFD solver.

$$\delta = \frac{fx^2}{24EI}\left(4Lx - 6L^2 - x^2\right) \tag{30}$$

A maximum deflection at the tip of the cantilever is given by

$$\delta_{max} = -\frac{FL^4}{8EI} = 6.1mm \tag{31}$$

The maximum deflection is found to be around $4.1mm$ using linear solver resulting in error of more than 33%. Also a typical of 15134 elements was used for the linear solver. This is typical a disadvantage of using linear tet meshes for bending problems which results in substantial error. Mesh refinement a possible alternative to obtain accurate solution results in very stiff matrices and poor convergence even after preconditioning, as in [6, 7]. This was found to be true where the ILU preconditioned of the EIGEN library failed to converge. Results for the tet10 element for the vertical displacement is shown in figure 3. A maximum deflection of $5.7mm$ was found at the tip. This corresponds to around 6% error in the accuracy.

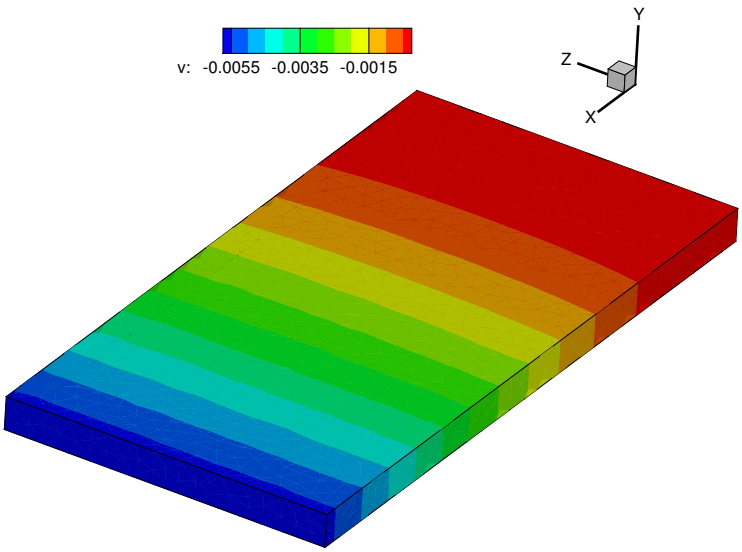

Figure 3: Deflection contours of the uniformly loaded cantilever beam

4 Conclusions

A FEM structural solver in 3d has been developed on tet10 elements and tested for a cantilever beam with uniformly distributed load. The numerical results shows a good match with the analytical solution. The higher order solver require lesser number of elements than the linear solver. Also there is no problems of convergence or ill conditioning of the linear solvers. Mesh convergence studies has been performed to measure the accuracy of the results. The solver can be integrated with any CFD solver for FSI problems. Currently dynamic analysis and modal analysis are being integrated. Also since scalability is an issue for large problems we are exploring the linear algebra library PETSc. We have developed a standalone structural solver which can be integrated with any CFD solver in the CTFD division so that a truly FSI solution to various aerodynamic problems.

Acknowledgments

The author would like to thank Dr Sintu Singha and C. K. Niranjanan for useful discussions during the course of this work. Also would like to acknowledge John Burkardt of Florida State University for maintaining a wonderful collection of code libraries for tet10 mesh generation. The author would also like to acknowledge the discussion which helped in correcting an ordering error in the code and provided more insights regarding the tet10 element generation.

References

[1] Bathe KJ., Finite Element Procedures, Prentice Hall Publications, 2015

[2] Ruben Sanchez (*et. al.), Towards a Fluid-Structure Interaction solver for problems with large deformations within the open source SU2 suite, AIAA-2016-0205*

[3] Reddy, J. N., Finite Element Method, TMH, 2005

[4] Carlos. Fellipa, Advanced FEM lecture notes, University of Colarado Boulder

[5] Sorna Rachel and William Weinlandt, 3d Finite Element Modelling: A comparison of common element types and patch test verification, Cornell University report, 2015.

[6] Erke Wang (et.al.), Back to Back elements- tetrahedra vs hexaheda, 9^{th} *Intl Ansys Conference, 2004.*

[7] Development of FEM based Structural Solver for FSI applications, PD/CTFD/2017/1002, 2017.

[8] John Burkardt, 10 Node Tet Mesh from 4 Node Tet Mesh, https://people.sc.fsu.edu/ jburkardt/

[9] Zienkiewicz O. C., The finite element method its basis and fundamentals, BH, 2013