Introduction

Our goal in this assignment is to code linear and quadratic *isoparametric* triagular finite elements.In finite element analysis, both the geometry and the unknown solution field are discretized using suitable interpolating functions. The term *parametric* indicates that we will develop shape functions for a standard parametric domain. Using suitable Jacobian matrix we will map these shape functions to the actual domain of integration. The term *iso* indicates that we will use the same shape functions for both the geometrical domain and the unknown solution field. To develop a numerical scheme for isoparametric triangular elements requires triangular shape functions and quadrature scheme for integration on triangular domains. These are the topics of the succeeding sections.

Triangular Shape Functions

We plan to develop shape functions for a linear 3-node triangular element and quadratic 6-node triangular element. The standard parametric triangular domains along with the nodes and their parametric coordinates for these elements are shown in figures 1 and 2. The next two subsections give the equations for shape functions and their derivatives for the two cases.

Linear 3-node Triangular Shape Functions and Their Derivatives

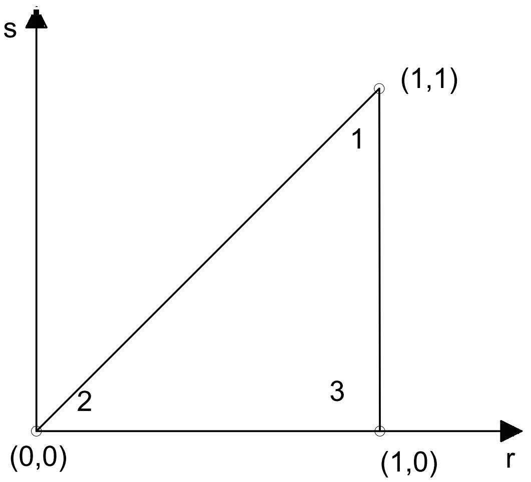


Figure 1: Nodal positions and numbering for linear parametric triangular domain

For three nodes we need three shape functions that satisfy the Kronecker-Delta property. Barycentric co-ordinates for triangles naturally satisfy this requirement. Hence the shape functions in terms of the barycentric co-ordinates are

$$\hat{N}_i = \lambda_i$$

For the node numbering shown in the standard (r, s) parametric domain of Figure 1, the barycentric coordinates, and thus the shape functions, are given by

$$\lambda_1 = s \qquad \lambda_2 = 1 - r \qquad \lambda_3 = r - s \tag{1}$$

The derivatives are

$$
\begin{aligned}
\hat{N}_{1,r} &= 0 & \hat{N}_{1,s} &= 1 \\
\hat{N}_{2,r} &= -1 & \hat{N}_{2,s} &= 0 \\
\hat{N}_{3,r} &= 1 & \hat{N}_{3,s} &= -1
\end{aligned}
$$

Quadratic 6-node Triangular Shape Functions and Their Derivatives

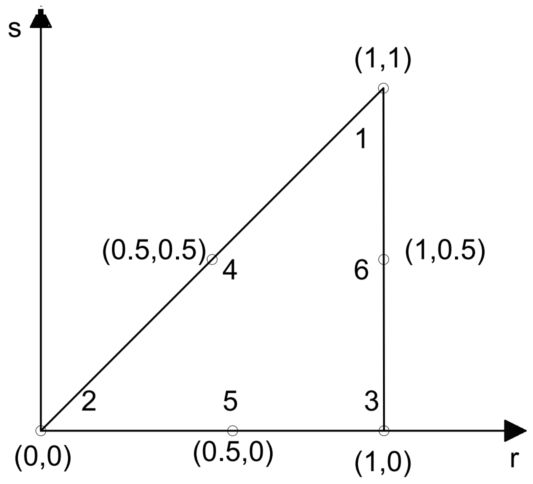


Figure 2: Node numbering and locations for quadratic parametric triangular domain

The node numbering is shown in Figure 2. As the node numbers 1, 2 and 3 are same as for the linear 3-node case, we can use the same barycentric coordinates as in Equation 1. The six shape functions in terms of the barycentric coordinates are

$$
\begin{aligned}
\hat{N}_1 &= \lambda_1(2\lambda_1 - 1) = -s + 2s^2 \\
\hat{N}_2 &= \lambda_2(2\lambda_2 - 1) = 2r^2 - 3r + 1 \\
\hat{N}_3 &= \lambda_3(2\lambda_3 - 1) = 2r^2 - r - 4rs + s + 2s^2 \\
\hat{N}_4 &= 4\lambda_1\lambda_2 \qquad\quad = -4rs + 4s \\
\hat{N}_5 &= 4\lambda_2\lambda_3 \qquad\quad = -4(r^2 - r - rs + s) \\
\hat{N}_6 &= 4\lambda_3\lambda_1 \qquad\quad = 4rs - 4s^2
\end{aligned}
$$

The derivatives are

$$
\begin{aligned}
\hat{N}_{1,r} &= 0 & \hat{N}_{1,s} &= 4s - 1 \\
\hat{N}_{2,r} &= 4r - 3 & \hat{N}_{2,s} &= 0 \\
\hat{N}_{3,r} &= 4r - 4s - 1 & \hat{N}_{3,s} &= 1 - 4r + 4s \\
\hat{N}_{4,r} &= -4s & \hat{N}_{4,s} &= 4(1 - r) \\
\hat{N}_{5,r} &= 4(1 - 2r + s) & \hat{N}_{5,s} &= 4(r - 1) \\
\hat{N}_{6,r} &= 4s & \hat{N}_{6,s} &= 4(r - 2s)
\end{aligned}
$$

Table 1: Sampling points and weights for 1-point and 3-point Gaussian quadrature

n_G	$\hat{w}_i$	(r_i, s_i)
1	$w_1 = 1$	$(2/3, 1/3)$
3	$w_1 = (1/3)$ $w_2 = (1/3)$ $w_3 = (1/3)$	$(1/2, 1/2)$ $(1, 1/2)$ $(1/2, 0)$

Gaussian Quadrature on Triangular Domain

The equations for elastic response of isoparametric triangular elements have integrals evaluated over a triangular domain of integration. We will use Gaussian Quadrature rules for numerical integration. For the standard parametric domain of Figure 1 the general form of Gaussian quadrature is

$$\int_0^1 \int_0^r f(r,s)\,\mathrm{d}s\mathrm{d}r = \sum_{i=1}^{n_G} \hat{w}_i f(r_i, s_i)\hat{\Omega}$$

where n_G is the number of sampling points (r_i, s_i) and $\hat{w}_i$ is the corresponding weight normalized by the area of the domain $\hat{\Omega} = (1/2)$. Table 1 gives the values of n_G, $\hat{w}_i$ and (r_i, s_i) for the 1-point and 3-point quadrature.

Isoparametric Triangular Elements

Using either of the shape functions and one of the quadrature rules discussed in previous sections we can write the equations for strain energy, internal force and stiffness modulus of isoparametric triangular elements. Let $\mathbf{X}$ and $\mathbf{x}$ represent the reference and deformed configurations of the element. Also, we will denote the parametric domain as θ^α where $\theta^1 = r$ and $\theta^2 = s$. Let n be the number of nodes in the element. In all the following equations $i, J, k, L, \alpha, \beta = 1, 2$ and $a = 1, 2 \ldots n$. The Jacobian matrix for transformation from parametric domain to reference configuration is then given as

$$\mathbb{J}_{I\alpha} = \frac{\partial X_I}{\partial \theta^\alpha}$$

The derivative of the shape functions with respect to $\mathbf{X}$ are

$$\begin{aligned} N_{a,J} &= \frac{\partial N_a}{\partial X_{aJ}} \\ &= \frac{\partial \hat{N}_a}{\partial \theta_\beta}\frac{\partial \theta_\beta}{\partial X_J} \\ &= \hat{N}_{a,\beta}\mathbb{J}^{-1}_{\beta J} \end{aligned}$$

The deformation gradient can now be written as

$$F_{iJ} = x_{ia}N_{a,J}$$

It is important that the sequence of vertices of the triangular domain $\mathbf{X}$ and $\mathbf{x}$ must be same as formed by the node numbering of the standard domain in Figures 1 and 2. If this requirement is not met, the determinant of $\mathbf{F}$ will become negative and give rise to imaginary Piola-Kirchhoff stress tensor. Also, the triangles should not have very high asspect ratio as that can cause determinant of $\mathbf{F}$ approach 0 and cause problems in calculating $\mathbf{F}^{-1}$.

Once we have a valid deformation gradient we can calculate the strain energy density w, the Piola-Kirchhoff stress tensor P_{iJ} and the consistent two-dimensional stiffness tensor C_{iJkL} using the plane stress neo-Hookean constitutive relationship formulated in HW 1. We need the following integral equations to find strain energy W, the internal force f_{ia}^{int} and the stiffness modulus K_{iakb} for the element Ω^e

$$\begin{aligned} W &= \int_{\Omega^e} w \, \mathrm{d}\Omega^e \\ &= \sum_{q=1}^{n_G} w \Big|_{\theta_q} \hat{w}_q |\mathbb{J}| \hat{\Omega} \end{aligned}$$

$$\begin{aligned} f_{ia}^{\text{int}} &= \int_{\Omega^e} P_{iJ} N_{a,J} \, \mathrm{d}\Omega^e \\ &= \sum_{q=1}^{n_G} \left(P_{iJ} \hat{N}_{a,\beta} \mathbb{J}^{-1}_{\beta J} \right) \Big|_{\theta_q} \hat{w}_q |\mathbb{J}| \hat{\Omega} \end{aligned}$$

$$\begin{aligned} K_{iakb} &= \int_{\Omega^e} C_{iJkL} N_{a,J} N_{b,L} \, \mathrm{d}\Omega^e \\ &= \sum_{q=1}^{n_G} \left(C_{iJkL} \hat{N}_{a,\beta} \mathbb{J}^{-1}_{\beta J} \hat{N}_{b,\alpha} \mathbb{J}^{-1}_{\alpha L} \right) \Big|_{\theta_q} \hat{w}_q |\mathbb{J}| \hat{\Omega} \end{aligned}$$

Verification Tests for Shape Functions, Quadrature and Elements

To verify the correctness of our computer implementations, we will check for some known properties that the shape functions, quadrature and elements should satisfy.

Verification of Shape Functions

- Partition of unity: At any point (r_1, s_1) in the standard domain, the shape functions should have

$$\sum_{a=1}^{n} \hat{N}_a(r_1, s_1) = 1$$

  We generated 1000 random points in the domain and checked for this property for the linear 3-node and quadratic 6-node triangular shape functions. It was satisfied with a numberical tolerance of 1×10^{-15} in all cases.

- Partition of nullity: The shape function derivatives should satisfy

$$\sum_{a=1}^{n} \hat{N}_{a,\alpha}(r,s) = 0 \qquad \alpha = 1,2$$

at all points in the standard domain. For 1000 randomly generated points in the domain this property was found to be satisfied with a numerical tolerance of 1×10^{-15} for both the 3-node linear and 6-node quadratic shape functions.

- Consistency: The shape function derivatives calculated by our code should tally with finite difference approximations of the derivatives within a suitable numerical tolerance. To get finite difference approximation of derivative with respect to one of the coordiantes r and s we perturb that co-ordinate with a suitably small value while holding the other coordinate constant. For example, using central finite differences we can write

$$\hat{N}_{a,r} \approx \frac{\hat{N}_a(r+h,s) - \hat{N}_a(r-h,s)}{2h}$$

where h is the perturbation. The value of h was chosen to be of the order 10^{-6}. For values of h smaller than this, large rounding errors make the approximation to have errors larger than $\mathcal{O}(h)$. For both the 3-node linear and 6-node quadratic shape functions the derivatives calculated by our code tallied with the corresponding central finite difference approximations to an acceptable tolerance of 10^{-6}.

- C^0-completeness: For a finite element formulation to converge to a solution, one of the requirements is that the shape functions should be able to exactly reproduce a linear field. Irrespective of the actual functional form of the solution field over the full domain, if it is discretized over sufficiently small elements then the local form of the solution over the element can always be reasonably approximated using linear functions. This is the rationale behind C^0-completeness requirement. To test it we will use a linear polynomial in (r,s), that is, $p(r,s) = ar + bs + c$ where a, b and c are randomly chosen co-efficients. We will evaluate p at the nodes of the standard domain to get $p(r_a, s_a)$. For a randomly chosen point (r_1, s_1) in the domain, C^0-completeness requires

$$p(r_1,s_1) = \sum_{a=1}^{n} \hat{N}_a(r_a,s_a) p(r_a,s_a)$$

For both the 3-node linear and 6-node quadratic shape functions we found this to be true with an error of the order 10^{-14}. The error is small enough to consider this verification test as passed.

Verification of Quadrature Implementation

A Gaussian quadrature rule that uses n points should exactly reproduce integration of a polynomial of degree less than or equal to $2n-1$. We have implemented 1-point and 3-point quadrature rules. But our shape functions are only linear or quadratic. So we will check, if our 1-point quadrature code can reproduce integration of a linear polynomial in (r,s) and if our 3-point quadrature code can reproduce integration of a quadratic polynomial in (r,s) exactly over the standard domain of Figure 1.

- For a linear polynomial,

$$I_1 = \int_0^1 \int_0^r (ar + bs + c)\,\mathrm{d}s\mathrm{d}r$$
$$= \frac{a}{3} + \frac{b}{6} + \frac{c}{2}$$

  For 1-point scheme the sampling point is $(r_1, s_1) = ((2/3), (1/3))$ and $\hat{w}_1 = 1$. Area of the standard domain is $\hat{\Omega} = (1/2)$. We want to check if our code returns

$$(ar_1 + bs_1 + c)\hat{w}_1\hat{\Omega} = I_1$$

  We found this to satisfied for each of 1000 random choices for a, b and c with a tolerance of the order 10^{-15}.

- For a quadratic polynomial,

$$I_3 = \int_0^1 \int_0^r (ar^2 + bs^2 + crs + dr + es + f)\,\mathrm{d}s\mathrm{d}r$$
$$= \frac{1}{24}(6a + 2b + 3c + 8d + 4e + 12f)$$

  For 3-point scheme the sampling points and weights are given in Table 1. Area of the standard domain is $\hat{\Omega} = (1/2)$. We want to check if our code returns

$$\sum_{q=1}^{3}(ar_q^2 + bs_q^2 + cr_qs_q + dr_q + es_q + f)\hat{w}_q\hat{\Omega} = I_3$$

  We found this to be satisfied for each of 1000 random choices for a, b, c, d, e and f with a tolerance of the order 10^{-15}.

Verification Tests for Isoparametric Triangular Elements

Consistency

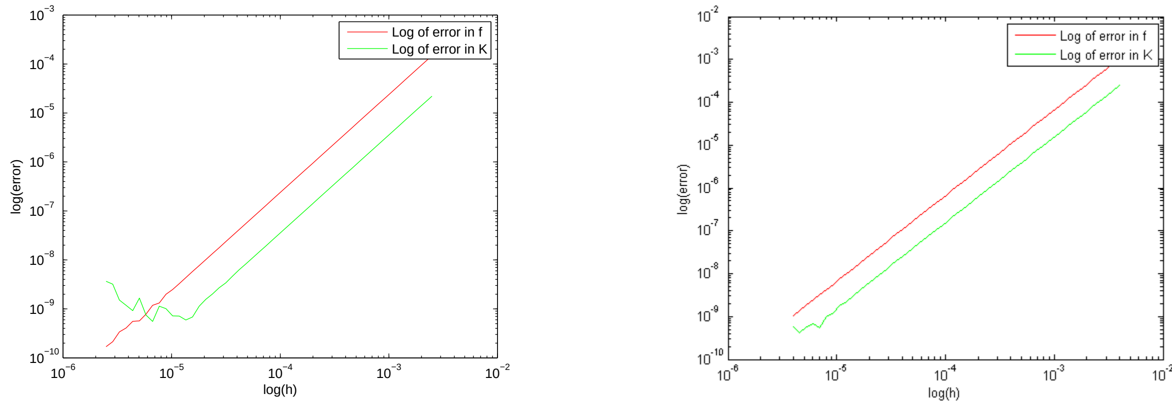
We have implemented the calculation of strain energy, internal force and stiffness modulus for the isoparametric triangular elements. These quantities are interrelated as

$$f_{ia}^{\text{int}} = \frac{\partial W}{\partial x_{ia}} \qquad \text{and}$$
$$K_{iakb} = \frac{\partial f_{ia}^{\text{int}}}{\partial x_{kb}}$$

We can use these two equations to check our implementation for bugs. If there are no bugs, the finite difference approximation of derivatives of W and f_{ia}^{int} with respect to x_{ia} and x_{kb} should match f_{ia}^{int} and K_{iakb}, obtained directly by our code, respectively to acceptable numerical tolerance. The central finite difference approximation of the above equations are given as

$$f_{ia}^{\text{int}} \approx \frac{W(x_{ia} + h) - W(x_{ia} - h)}{2h}$$
$$K_{iakb} \approx \frac{f_{ia}^{\text{int}}(x_{kb} + h) - f_{ia}^{\text{int}}(x_{kb} - h)}{2h}$$

We chose 50 values for perturbation $h \in (10^{-6}, 10^{-3})$. The error in the numerical derivatives scaled quadratically with h as can be seen from Figure 3. For the smaller values of h the effect of large rounding errors can be noticed in Figure 3a.



(a) Linear 3-node triangular element (b) Quadratic 6-node triangular element

Figure 3: Log-log plot of error versus h shows straight lines with slope 2 indicating error of $\mathcal{O}(h^2)$ as expected for a central finite difference numerical differentiation. Smaller vlues of h introduce large rounding errors as seen in Figure 3a

Rank of Stiffness Matrix

The fourth order stiffness modulus K_{iakb} can be *unrolled* into a square 2-D matrix of size $ia \times kb$. For linear 3-node triangular element this will give a 6×6 matrix and for the quadratic 6-node triangular element we get a 12×12 matrix. A $n \times n$ matrix can have maximum of n non-zero eigenvalues. Then the *rank* of that matrix is said to be n. Each zero eigenvalue reduces the rank by 1 which makes the matrix *rank deficient*. We calculated the rank of the stiffness matrix for the case of zero deformation i.e. $\mathbf{X} = \mathbf{x}$ and for finite deformation for different combinations of shape function and quadrature order choices. The results are presented in Table 2. A discrete finite-element model must give unique solution in the same way as the corresponding continuous mathematical model. In a elasticity problem, rigid body motions do not result in any internal forces and thus are called zero-energy modes. Zero-energy modes make the solution degenerate i.e. non-unique. Each zero eigen-value of the stiffness matrix should represents a rigid-body mode and we should not have any extra zero-energy modes in our finite element implementation than as physically expected. For zero deformation case, there are three degrees of freedom — two translations and one rotation. Hence, we expect the stiffness matrix to be rank-deficient by 3. For finite deformation case there are only two degrees of freedom — the translations. Hence, we expect the matrix to be rank deficient by 2. The results presented in Table 2 are consistent with our expectations for all but the two cases when we use 1-point Gaussian quadrature rule with a quadratic shape function. A 1-point Gaussian quadrature rule can exactly reproduce an integration of a polynomial of degree less than or equal to 1. For exactly reproducing integration of quadratic shape function we will need at least 3-point quadrature. By using lesser number of sampling points we are introducing extra non-physical zero body modes.

Table 2: Rank of stiffness matrix for various combinations of shape functions and order of Gaussian quadrature

Deformation	Shape Function	Quadrature	Rank
Zero	Linear	1-point	3
	Quadratic	1-point	3
	Quadratic	3-point	9
Finite	Linear	1-point	4
	Quadratic	1-point	4
	Quadratic	3-point	10

Source Code Listing

The source code files for this assignment are listed below.

1. `aspectRation.m`:This function takes coordinates of vertices of a triangle as input and returns its aspect ratio as an output. We use it to ensure that we do not generate skewed triangles which can cause problems due determinant of F approaching zero.

2. `checkOutwardNormal.m`: This function takes coordinates of vertices of a triangle as an input and checks if the sequence of the vertices is consistent with the node numbering of the standard parametric triangle. This is used to ensure that we don't get deformation gradients with negative determinant.

3. `findStiffnessRank.m`: This function accepts a four dimensional array as an input and unrolls it into a 2D matrix and returns the number of non-zero eigenvalues of the unrolled matrix as output.

4. `neoHookean.m` This is the 3D neo-Hookean constitutive model implementation from HW 1.

5. `planeStressNH.m` This is the plane-stress implementation of the 3D neo-Hookean consitutive model as developed for HW 1.

6. `Stiffness_rank.m` This is a script file that calculates and prints a table for the rank of stiffness matrix for different combination of shape function and quadrature rule choices.

7. `T3Lin.m` This function takes (r, s) coordinates as input and returns the shape function and its derivatives evaluated for a linear 3-node triangular element at that point.

8. `T6quad.m` This function takes (r, s) coordinates as input and returns the shape function and its derivatives evaluated for a quadratic 6-node triangular element at that point.

9. `T3Lin_Verification.m` This script runs the various tests for the linear 3-node shape functions.

10. `T6quad_Verification.m` This script runs the various tests for the quadratic 6-node shape functions.

11. `T3LinEle.m` This function takes X, x, quadrature order and the material properties μ and λ as input and returns the strain energy, internal force and stiffness modulus for a linear 3-node triangular finite element.

12. `T6QuadEle.m` This function takes X, x, quadrature order and the material properties μ and λ as input and returns the strain energy, internal force and stiffness modulus for a quadratic 6-node triangular finite element.

13. `T3LinEle_Verification.m`: This script runs and prints the results of various tests for the linear 3-node triangular element.

14. `T6QuadEle_Verification.m`: This script runs and prints the results of various tests for the quadratic 6-node triangular element.

15. `TriGaussQuad.m`: This function takes the order of Gaussian quadrature as input and returns the corresponding sampling points and weights as output for a standard triangular domain.

16. `TriGaussQuad_Verification.m`: This script runs verification test for the Gaussian quadrature implementation and prints its results.