University of Cape Town

Elastic Mesh Deformation for Internal Flows with Moving Boundaries

Author: **Alaa Mohammed Ali**
Supervisor: Prof. Arnaud G. Malan
Co-Supervisors: Dr. Tomas Lundquist,
Dr. Niran Ilangakoon

A thesis submitted to the University of Cape Town for the degree of Master of Science in Engineering

Department of Mechanical Engineering
Computational Fluid Dynamics Research Group

June, 2023

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This thesis/dissertation has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Abstract

This project aims to study elastic-based mesh deformation for internal flow problems with obstructions. As such a tank with a flexible internal beam serves as the tested case employed in this work. As such a computer code was developed and the elastic deformation was implemented via the Finite Element Galerkin method with two methods to control the element's distortion near the moving boundaries, Jacobian-based stiffening modification and the distance criterion method. Verification of the scheme was performed by conducting the same test as described in a journal and comparing the obtained results. The project focused on mesh deformation resulting from prescribing the displacements to the inner boundaries of the beam. The results were validated by assessing mesh quality using aspect ratio as a metric, for the 2D unstructured triangular mesh, the element mesh quality showed improvement when the Jacobian-based stiffening method was applied, which involved excluding the determinant of the Jacobian from the stiffness matrix formulation. Similarly, for the 3D unstructured mesh, the aspect ratio improved from elements collapsing to a value close to 1. In both cases, optimal mesh quality was achieved using the distance criterion method for 2D and 3D structured meshes. Specifically, for the 2D structured mesh, the aspect ratio demonstrated improvement, while for the 3D structured mesh, a further enhancement was also observed.

Acknowledgment

The following people have contributed to constructing and reaching the finish line of this project, the majority have shared their knowledge and time generously. Some have been the only source of confidence I sometimes lacked. For all, I am sincerely grateful.

My supervisor for his patience, vast knowledge, and guidance, and my co-supervisors Dr. Tomas and Dr. Niran for the experience they shared in a humble manner, which made my route a lot easier to walk.

The team of InCFD, Dr. Bevan, YusufAli, Prince, Michael, Dr.Leon, and Roy. The fragments of information I learned from all of you have assembled in many parts of this thesis in skeleton and design.

For my family and friends for endless support and confidence in me.

This work is dedicated to my family, and the resilient Sudanese women. Your excellence is enough to inspire the whole world.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

α	Aspect ratio
RMS	Root mean square

Greek Symbols

$\boldsymbol{\sigma}$	Caushy stress
ϵ	Strain
η	Coordinate of natural space
γ	Shear strain
Ω	Volume of Element
ρ	Density
τ	Stress vector
ξ	Coordinate of natural space
v	Poisson's ratio

Mathematical Operators

$(\bullet)^{-1}$	Inverse of $\bullet$
∇	Gradient Operator
$\partial\bullet$	Partial Derivative of $\bullet$
T	Transpose of matrix

Roman Symbols

s	Surface area
$\boldsymbol{f}$	The accelerations caused by body forces

C Material Property

K Stiffness matrix

L Lagrangian shape functions

u Displacements vector

Chapter 1

Introduction

Mesh deformation is crucial for fluid-structure interaction modeling, it plays a lead role in projects with moving boundaries when re-meshing the geometry after each time step would be expensive [1, 2]. In numerous research domains like ballistics, aeronautics, biomedical engineering, and transportation, simulations involve various challenges such as handling unsteady conditions, moving meshes, and coupling between fluid-structure interactions or structural interactions [3] especially in CFD simulations [4], for example shape optimization incorporates mesh deformation at each iteration when there is a moving boundary [5] In this project an internal flow geometry is considered which consists of a tank containing a flexible beam. The beam is subject to prescribed motion to mimic FSI behavior resulting in both rotation and "shear" type distortion to the mesh if using mesh deformation [6]. Mesh deformation is achieved in this work via a linear elastic methodology with added element stiffening to reduce deterioration in element quality. This chapter includes a literature review, problem statement, and thesis outline.

1.1 Literature Review

The main mesh deformation methods in CFD are the elasticity-based method first proposed by Tezduyar [7], spring analogy method by Batina [8] and interpolation-based method by Hardy [9]. Persson and Strang used solving for equilibrium in a truss structure to deform a mesh generated by the Delaunay triangulation algorithm, using node movement and distance function, this method generated elements with high quality, implementing topology updates such as edge flips in 2004 [10] which was also used by Wang and Persson in 2013 [11] and 2015 cite as a mesh deformation method, using a spring-based analogy for nodes movement, Persson and Peraire used a non-linear elasticity analogy for mesh deformation of a straight mesh to a curved mesh using a neo-Hookean constitutive model for the stress tensor where the mesh is modeled as a linear solid, the system is discretized and solved for the unknowns using the standard finite element method, used again by Froehle and Persson to model. The above methods can be classified into two basic classes: physical analogy and interpolation Jacobian-based stiffening modification and the distance

criterion method [2]. Both approaches are briefly discussed below.

1-Mesh Deformation using Interpolation Analogy

In the interpolation-based approaches, an interpolation function is used as a mesh deformation scheme by prescribing the displacements of the boundary and interpolating the deformation to the rest of mesh [12]. To deform the mesh, displacements are calculated for each vertex based on its initial and target positions. The mesh nodes' positions are updated using an interpolation function of boundary nodes movement to compute the displacements for each node, the interpolation process is frequently carried out in multiple iterations, such as in dynamic simulations involving continuous deformation of boundaries at each time step [13]. These methods do not utilize grid connectivity information, in general, they are more computationally efficient but suffer from a lack of robustness, especially for large meshes, which can range from hundreds of thousands to millions of elements [2]. These methods are:

1-Transfinite interpolation.

2-Algebraic damping method.

3-Inverse distance weighting method.

4-Radial Basis Functions Interpolation.

2-Mesh Deformation using Physical Analogy

The physical analogy approach uses spring or elasticity governing equations and numerical methods to solve for displacements. Two main approaches are used in this analogy and they are:

1-Spring analogy: In the mesh, each edge is modeled as a spring with stiffness determined by the reciprocal of its length. These springs simulate the connections between vertices in the mesh, which can be any edges or a subset of edges. The methods used for spring analogy are linear spring analogy, torsional spring analogy, semi-torsional spring analogy, and ball vertex spring analogy [14]. These methods require grid connectivity information which results in more storage requirements [2].

2-Elasticity-Based methods: Three elasticity-based mesh deformation methods are commonly used: a)Linear elasticity-based method. It was first introduced by Tezduyar [7] where the linear elasticity equation is used with a modification in the discretizing scheme by dropping the Jacobian from the system equation that was later named Jacobian-based stiffening. The linear elasticity equation used is:

$$\boldsymbol{\nabla}.\boldsymbol{\sigma} = \rho \boldsymbol{f}$$

where $\boldsymbol{\sigma}$ is the stress and $\boldsymbol{f}$ is the accelerations caused by body forces and ρ is the density. b)Nonlinear elasticity-based method based on size criterion. c)Nonlinear elasticity-based method based on the distance criterion [15]. and the distance criterion method [15] The linear elasticity-based method was used in this

project because it utilizes grid connectivity information and hence allows for limiting deterioration in element quality due to mesh motion. It is also simpler and more efficient than the non-linear versions [7, 15].

1.2 Problem Statement

The problem solved in this thesis describes the motion of mesh elements occurring due to displacements of the domain boundary. The test case used in this work will be a tank with an internal flexible beam, with the beam deflecting due to prescribed motion. The material used to model the mesh is isotropic homogeneous and uses geometrically linear elastic formulation. In addition, a quasi-static approach will be employed i.e. steady state and neglecting body forces under the following assumptions [16]:

1- The behavior of the material is linear.

2- Dynamic effects are neglected, i.e. small acceleration.

3- No gaps or overlaps occur during the deformation of the solid. As the solid deforms, it does not crack or undergo any interpenetration of material.

Linear elasticity theory is generally valid for small deformations. If the amplitude of the mesh deformation is small (i.e., the deformation is within the linear range), linear elasticity can provide reasonably accurate results. The boundary motion was divided into small deformation values the maintain mesh quality. The stiffness matrix was subsequently updated thus mimicking a non-linear method similar to the work of Bogaers [17].

1.3 Thesis Outline

This thesis is made of 5 chapters, a description of the following four chapters is below:

• **Chapter Two: The Finite Element Method Formulation** Contains governing equations of 2D and 3D linear elasticity equation and discretization using the finite element Galerkin method and validation of implementation.

• **Chapter Three: Verification and Validation** This chapter focuses on the verification and validation of the method and code utilized in this study. It involves verifying the accuracy and reliability of the method by comparing it against previous tests conducted using a similar approach.

• **Chapter Four: Mesh Deformation In a Tank** Validation of the finite element scheme using a test case of mesh in a fuel tank with a beam in motion.

• **Chapter Five: Conclusions and Recommendations** Contains a brief description of observations and recommended future research's focus.

Chapter 2

The Finite Element Formulation

The mathematical scheme used to model mesh deformation is the Finite Element scheme. In this chapter, the governing equations and discretization necessary for the scheme are reported in a two-dimensional domain.

2.1 Continuous Form of 2D Finite Element Formulation

The Governing equation used is the momentum conservation equation for a solid body undergoing linear elastic motion while neglecting body forces and inertial forces is [18]:

$$\frac{\partial \sigma_{ij}}{\partial x_j} = 0$$

where σ_{ij} is the stress tensor. The first subscript refers to the positive direction of the normal to the surface. The second subscript refers to the positive direction of a component of the stress vector acting on the surface. In a 2D cartesian coordinates system, the result in 2D is:

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} = 0 \tag{2.1}$$

$$\frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} = 0 \tag{2.2}$$

Next, introducing the constitutive relation between stress and strain which is described by the generalized Hooke's law:

$$\sigma_{ij} = \mathrm{C}_{ijkl}\epsilon_{kl}$$

where C_{ijkl} is the fourth-order elasticity tensor, and ϵ_{kl} is the symmetric part of the Lagrangian strain tensor for linear materials. Writing the constitutive equation above in engineering notation yields:

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \frac{E}{(1+v)(1-2v)} \begin{bmatrix} 1-v & v & 0 \\ v & 1-v & 0 \\ 0 & 0 & \frac{1}{2}-v \end{bmatrix} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{bmatrix} = \boldsymbol{G\epsilon} \tag{2.3}$$

where ϵ and γ are the normal and shear strains respectively, note that In finite elements, the relationship $\gamma_{xy} = 2\epsilon_{xy}$ is incorporated to maintain the consistency of the definition of elastic energy density. E is Young's modulus of elasticity and v is Poisson's ratio. Also, note the adoption of Caushy's assumption of the stress tensor being symmetric which means $\sigma_{xy} = \sigma_{yx}$. Finally, introducing the relationship between the strain and displacement field:

$$\epsilon_{ij} = \frac{1}{2}[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}] \tag{2.4}$$

Here $\boldsymbol{u}$ is defined as a vector that extends from a material point P in the undeformed configuration to the same material point P in the deformed configuration. In 2D $\boldsymbol{u}$ has 2 components:

$$\boldsymbol{u}(x,y) = \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix}$$

2.2 Discrete Form

FEM is a numerical method that uses an approximate solution to solve partial differential equations that describe a certain physical system. It is used here to solve for displacements in the problem described. Defining the finite element Galerkin method -weak form-, being an approximate solution that applies the principle of weighted residuals. For the two Governing Equations (2.1) and (2.2) at a node i which is located in element e [18]:

$$\int_{\Omega_e} R_x L_i d\Omega_e = 0$$

$$\int_{\Omega_e} R_y L_i d\Omega_e = 0$$

where R_x and R_y are the residuals in x and y respectively and L_i is the Lagrangian shape function used as a weighting function - discussed later in this document -. Further Ω_e is the volume of the element. Equations (2.1) and (2.2) become:

$$\int_{\Omega_e} \left[\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} \right] L_i d\Omega_e = 0 \tag{2.5}$$

$$\int_{\Omega_e} \left[\frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} \right] L_i d\Omega_e = 0 \tag{2.6}$$

The integration by parts and Gauss Theorem applied to Equations (2.5) and (2.6) yield:

$$\int_{\Omega_e} \left[\sigma_{xx} \frac{\partial L_i}{\partial x} + \sigma_{xy} \frac{\partial L_i}{\partial y} \right] d\Omega_e = \oint_{d\Omega_e} \left[\sigma_{xx} n_x L_i + \sigma_{xy} n_y L_i \right] ds \tag{2.7}$$

$$\int_{\Omega_e} \left[\sigma_{yx} \frac{\partial L_i}{\partial x} + \sigma_{yy} \frac{\partial L_i}{\partial y} \right] d\Omega_e = \oint_{d\Omega_e} \left[\sigma_{yx} n_x L_i + \sigma_{yy} n_y L_i \right] ds \tag{2.8}$$

where n_x and n_y are the x and y components of the outward pointing element surface unit vector respectively.

The system equations for one node can be written in a matrix for each element as:

$$\begin{bmatrix} \sigma_{xx} \frac{\partial L_1}{\partial x} + \sigma_{xy} \frac{\partial L_1}{\partial y} \\ \sigma_{yx} \frac{\partial L_1}{\partial x} + \sigma_{yy} \frac{\partial L_1}{\partial y} \\ \sigma_{xx} \frac{\partial L_2}{\partial x} + \sigma_{xy} \frac{\partial L_2}{\partial y} \\ \sigma_{yx} \frac{\partial L_2}{\partial x} + \sigma_{yy} \frac{\partial L_2}{\partial y} \\ \vdots \\ \sigma_{xx} \frac{\partial L_n}{\partial x} + \sigma_{xy} \frac{\partial L_n}{\partial y} \\ \sigma_{yx} \frac{\partial L_n}{\partial x} + \sigma_{yy} \frac{\partial L_n}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial L_1}{\partial x} & 0 & \frac{\partial L_1}{\partial y} \\ 0 & \frac{\partial L_1}{\partial y} & \frac{\partial L_1}{\partial x} \\ \frac{\partial L_2}{\partial x} & 0 & \frac{\partial L_2}{\partial y} \\ 0 & \frac{\partial L_2}{\partial y} & \frac{\partial L_2}{\partial x} \\ \vdots & \vdots & \vdots \\ \frac{\partial L_n}{\partial x} & 0 & \frac{\partial L_n}{\partial y} \\ 0 & \frac{\partial L_n}{\partial y} & \frac{\partial L_n}{\partial x} \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = (\nabla \boldsymbol{L})^T \boldsymbol{\sigma} \tag{2.9}$$

where n is the number of nodes in the element, ∇ is a gradient in engineering notation and not a general form of the gradient.

Further reducing the RHS of equations (2.7) and (2.8) can be done using this relation:

$$\tau_i = \sigma_{ji} n_j \tag{2.10}$$

This means that the components of stress acting on any point are a linear combination of the components of the stress vectors acting on the coordinate planes at that point. In other words:

$$\tau_x = \sigma_{xx} n_x + \sigma_{xy} n_y \tag{2.11}$$

$$\tau_y = \sigma_{yx} n_x + \sigma_{yy} n_y \tag{2.12}$$

where τ_x and τ_y are the boundary fluxes . Then equations (2.7) and (2.8) become:

$$\int_{\Omega_e} \left[\sigma_{xx} \frac{\partial L_i}{\partial x} + \sigma_{xy} \frac{\partial L_i}{\partial y} \right] d\Omega_e = \int_{d\Omega_e} \tau_x L_i ds \tag{2.13}$$

$$\int_{\Omega_e} \left[\sigma_{yx} \frac{\partial L_i}{\partial x} + \sigma_{yy} \frac{\partial L_i}{\partial y} \right] d\Omega_e = \int_{d\Omega_e} \tau_y L_i ds \tag{2.14}$$

The RHS of equations (2.13) and (2.14) can be written in matrix form as follows:

$$\begin{bmatrix} \tau_x L_1 \\ \tau_y L_1 \\ \tau_x L_2 \\ \tau_y L_2 \\ . \\ . \\ \tau_x L_n \\ \tau_y L_n \end{bmatrix} = \begin{bmatrix} L_1 & 0 \\ 0 & L_1 \\ L_2 & 0 \\ 0 & L_2 \\ . & \\ . & \\ . & \\ L_n & 0 \\ 0 & L_n \end{bmatrix} \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix} = \boldsymbol{L}^T \boldsymbol{\tau} \tag{2.15}$$

using equations (2.9) and (2.15), equations (2.13) and (2.14) can be written as [18]:

$$\int_{\Omega_e} (\nabla \boldsymbol{L})^T \boldsymbol{\sigma} \, d\Omega_e = \oint_{d\Omega_e} \boldsymbol{L}^T \boldsymbol{\tau} \, ds \tag{2.16}$$

2.2.1 Trial Solution

The FEM obtains an approximate solution by using the classical trial-solution procedure. The trial solution for the two displacement components u and v is given by a linear combination of the basis functions L_j:

$$u(x, y) = u_j L_j(x, y)$$

$$v(x, y) = v_j L_j(x, y)$$

Substituting the constitutive equations introduces the displacement vector, i.e. the trial solution of displacements needs to be substituted into the strain vector:

$$\epsilon_{xx} = \frac{\partial u_1}{\partial x} = u_j \frac{\partial L_j}{\partial x}$$

$$\epsilon_{yy} = \frac{\partial u_2}{\partial y} = v_j \frac{\partial L_j}{\partial y}$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = u_j \frac{\partial L_j}{\partial y} + v_j \frac{\partial L_j}{\partial x}$$

which can be written in matrix-vector form for a single element:

$$\begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial L_1}{\partial x} & 0 & \frac{\partial L_2}{\partial x} & 0 & ... & \frac{\partial L_n}{\partial x} & 0 \\ 0 & \frac{\partial L_1}{\partial y} & 0 & \frac{\partial L_2}{\partial y} & ... & 0 & \frac{\partial L_n}{\partial y} \\ \frac{\partial L_1}{\partial y} & \frac{\partial L_1}{\partial x} & \frac{\partial L_2}{\partial y} & \frac{\partial L_2}{\partial x} & ... & \frac{\partial L_n}{\partial y} & \frac{\partial L_n}{\partial x} \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ . \\ . \\ u_n \\ v_n \end{bmatrix} \tag{2.17}$$

In vector notation, the relation above can be written as :

$$\boldsymbol{\epsilon} = (\nabla \boldsymbol{L})\boldsymbol{u} \tag{2.18}$$

Equation (2.18) can be substituted in Equation (2.3), the latter then becomes :

$$\boldsymbol{\sigma} = \boldsymbol{G}(\nabla \boldsymbol{L})\boldsymbol{u} \tag{2.19}$$

Now the system equation (2.16) can be written as:

$$\int_{\Omega_e} (\nabla \boldsymbol{L})^T \boldsymbol{G}(\nabla \boldsymbol{L})\boldsymbol{u} d\Omega_e = \int_{d\Omega_e} \boldsymbol{L}^T \boldsymbol{\tau} ds \tag{2.20}$$

where $\nabla \boldsymbol{L}$, $\boldsymbol{L}$ *and* $\boldsymbol{u}$ are defined in Cartesian coordinates. The RHS is the boundary stress applied on the surface of the element.

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{F} \tag{2.21}$$

The matrix $\boldsymbol{K}$ is called the stiffness matrix and is symmetric while $\boldsymbol{F}$ is the load vector where:

$$\boldsymbol{K} = \int_{\Omega_e} (\nabla \boldsymbol{L})^T \boldsymbol{G}(\nabla \boldsymbol{L}) d\Omega_e \tag{2.22}$$

$$\boldsymbol{F} = \int_{d\Omega_e} \boldsymbol{L}^T \boldsymbol{\tau} ds \tag{2.23}$$

to solve for $\boldsymbol{u}$ on a single element:

$$\boldsymbol{u} = \boldsymbol{K}^{-1}\boldsymbol{F} \tag{2.24}$$

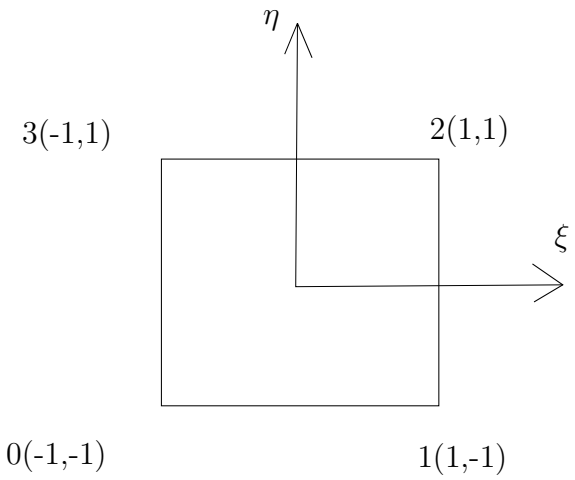


Figure 2.1: Quadrilateral Parent Element

2.2.2 Quadrilateral Element

Shape functions:

Next, the definition of the shape functions is required to compute the numerical solution to the system of equations. Consider a linear quadrilateral element, develop the specific shape function of the parent element then map it to your real element as follows: a) developing shape function for a bi-unit square, the parent element must be a perfect square in the (ξ, η) coordinate system (as per Figure 2.1) with corners positioned at (-1,-1), (1,-1), (1,1) and (-1,1).

The shape functions for the four nodes of the above element are as follows [18]:

$$L_0(\xi,\eta) = \frac{1}{4}(1-\xi)(1-\eta)$$

$$L_1(\xi,\eta) = \frac{1}{4}(1+\xi)(1-\eta)$$

$$L_2(\xi,\eta) = \frac{1}{4}(1+\xi)(1+\eta)$$

$$L_3(\xi,\eta) = \frac{1}{4}(1-\xi)(1+\eta)$$

which can also be written as:

$$\boldsymbol{L} = \sum_n \psi_i(\xi)_n \psi_j(\eta)_n$$

where

$$\psi_i(\xi)_0 = \frac{(\xi - \xi_0)}{2}$$

$$\psi_j(\eta)_0 = \frac{(\eta - \eta_0)}{2}$$

Next all the derivatives of the shape functions to use in the stiffness matrix are listed below:

$$\frac{\partial L_0}{\partial \xi} = -\frac{1}{4}(1-\eta) \qquad \frac{\partial L_0}{\partial \eta} = -\frac{1}{4}(1-\xi)$$
$$\frac{\partial L_1}{\partial \xi} = \frac{1}{4}(1-\eta) \qquad \frac{\partial L_1}{\partial \eta} = -\frac{1}{4}(1+\xi)$$
$$\frac{\partial L_2}{\partial \xi} = \frac{1}{4}(1+\eta) \qquad \frac{\partial L_2}{\partial \eta} = \frac{1}{4}(1+\xi)$$
$$\frac{\partial L_3}{\partial \xi} = -\frac{1}{4}(1+\eta) \qquad \frac{\partial L_3}{\partial \eta} = \frac{1}{4}(1-\xi)$$

or

$$\boldsymbol{L}_{\prime\xi} = \sum \psi_{i\prime\xi}\psi_j$$

where $\boldsymbol{L}_{\prime\xi}$ and $\psi_{i\prime\xi}$ mean the derivative of $\boldsymbol{L}_\xi$ and ψ_i with respect to ξ

Isoparametric Element

As mentioned before the shape functions above are in the natural coordinates system and the real element is in the Cartesian system. Therefore mapping is needed between them. Also to use numerical integration to obtain the required matrices a conversion to the natural coordinate system is needed. For this the isoparametric mapping was employed, given by the following formula:

$$x(\xi,\eta) = x_k L_k(\xi,\eta)$$

$$y(\xi,\eta) = y_k L_k(\xi,\eta)$$

where x_k is the x coordinate of the node in the Cartesian frame. To find $\nabla \boldsymbol{L}$ in (ξ,η) the following formulas are used:

$$\frac{\partial L_i}{\partial \xi} = \frac{\partial L_i}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial L_i}{\partial y}\frac{\partial y}{\partial \xi}$$
$$\frac{\partial L_i}{\partial \eta} = \frac{\partial L_i}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial L_i}{\partial y}\frac{\partial y}{\partial \eta}$$

$$\begin{bmatrix} \dfrac{\partial L_i}{\partial \xi} \\ \dfrac{\partial L_i}{\partial \eta} \end{bmatrix} = \boldsymbol{J} \begin{bmatrix} \dfrac{\partial L_i}{\partial x} \\ \dfrac{\partial L_i}{\partial y} \end{bmatrix}$$

where $\boldsymbol{J}$ is the Jacobian matrix given by

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} x_k \frac{\partial L_k}{\partial \xi} & y_k \frac{\partial L_k}{\partial \xi} \\ x_k \frac{\partial L_k}{\partial \eta} & y_k \frac{\partial L_k}{\partial \eta} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{\partial L_0}{\partial \xi} & \frac{\partial L_1}{\partial \xi} & \frac{\partial L_2}{\partial \xi} & \frac{\partial L_3}{\partial \xi} \\ \frac{\partial L_0}{\partial \eta} & \frac{\partial L_1}{\partial \eta} & \frac{\partial L_2}{\partial \eta} & \frac{\partial L_3}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix}$$

The Jacobian matrix can be used to map the derivatives of shape functions from the real element to the isoparametric element as follows:

$$\begin{bmatrix} \frac{\partial L_i}{\partial x} \\ \frac{\partial L_i}{\partial y} \end{bmatrix} = \boldsymbol{J}^{-1} \begin{bmatrix} \frac{\partial L_i}{\partial \xi} \\ \frac{\partial L_i}{\partial \eta} \end{bmatrix} \tag{2.25}$$

i.e:

$$\frac{\partial L_i}{\partial x} = J_{11}^{-1} \frac{\partial L_i}{\partial \xi} + J_{12}^{-1} \frac{\partial L_i}{\partial \eta}$$

where J_{ij}^{-1} denotes the i,j component of the inverse of $\boldsymbol{J}$.

Numerical Integration

In order to compute the integration Gauss quadrature is routinely applied to the above in the (ξ,η) frame. Using the mapping described in the previous subsection, all the expressions are in terms of (ξ, η) except $d\Omega$. This is resolved in 2D using the determinant of $\boldsymbol{J}$ as follows:

$$d\Omega_e = |\boldsymbol{J}| d\xi d\eta$$

where $|\boldsymbol{J}|$ is the determinant of the jacobian matrix $\boldsymbol{J}$. This results in:

$$\boldsymbol{K} = \int_{-1}^{1} \int_{-1}^{1} (\nabla \boldsymbol{L})^T(\xi, \eta) \boldsymbol{G} \nabla \boldsymbol{L}(\xi, \eta) |\boldsymbol{J}(\xi, \eta)| d\xi d\eta$$

Gauss Quadrature in 2D is obtained by [18]:

$$\int_{-1}^{1} \int_{-1}^{1} I(\xi, \eta) d\xi d\eta = \sum_{k=1}^{n} \sum_{l=1}^{n} w_k w_l I(\xi_k, \eta_l)$$

where w_{nk}, w_{nl} are Gauss weights and ξ_{nk}, η_{nl} are Gauss points

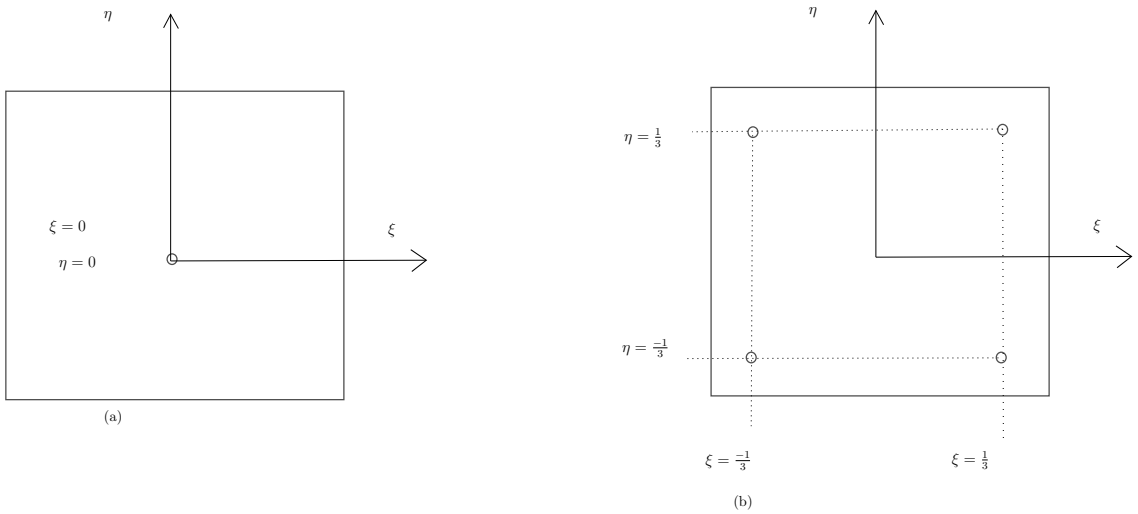


Figure 2.2: Gauss Points

Two different numbers of integration points in the Gauss quadrature were considered, as shown in the Figure above. The discrete stiffness matrix becomes

$$\boldsymbol{K} = \sum_{k=1}^{n}\sum_{l=1}^{n} w_k w_l (\nabla \boldsymbol{L})^T \boldsymbol{G} \nabla \boldsymbol{L} |\boldsymbol{J}| \Big|_{(\xi_k, \eta_l)} \tag{2.26}$$

Now returning to the surface integrals on the RHS:

$$\boldsymbol{F}(x, y) = \int_{d\Omega_e} \boldsymbol{L}^T \boldsymbol{\tau} ds \tag{2.27}$$

In (ξ, η) equation (2.27) becomes:

$$\boldsymbol{F}(\xi, \eta) = \int_{-1}^{1} \boldsymbol{L}^T \boldsymbol{\tau} J_\Gamma ds' \tag{2.28}$$

$\boldsymbol{\tau}$ is a vector of constants which are the stresses on the boundary.

$$\boldsymbol{F}(\xi, \eta) = \boldsymbol{\tau} \int_{-1}^{1} \boldsymbol{L}^T J_\Gamma ds' \tag{2.29}$$

where ds' is the surface in the (ξ, η) coordinate system. J_Γ is the Jacobian of the surface.

$$J_\Gamma = \frac{L_r}{L_p} \tag{2.30}$$

where L_r is the length of the side where the load is applied and L_p is the length of the corresponding

side of the parent element. For the domain (-1,1) the length of the side of the parent element is = 2 $\boldsymbol{L}$ is equal to the shape functions in 1D so we can call it $\boldsymbol{L}_\Gamma$. This allows the use of one of the following shape functions for each node:

$$L_0 = \frac{1-\xi}{2} \tag{2.31}$$

$$L_1 = \frac{1+\xi}{2} \tag{2.32}$$

$\boldsymbol{F}$ can be evaluated using Gauss Quadrature for 1D

$$\int_{-1}^{1} I(\xi)d\xi = \sum_{l=1}^{n} w_{nl} I(\xi_{nl}) \tag{2.33}$$

Equation (2.29) thus becomes:

$$\boldsymbol{F} = \boldsymbol{\tau} J_\Gamma \sum_{l=1}^{n} w_{nl} \boldsymbol{L}_\Gamma^T\big|_{nl} \tag{2.34}$$

Summary

The system equations that need to be solved for a single element are:

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{F}$$

$$\boldsymbol{u} = \boldsymbol{K}^{-1}\boldsymbol{F}$$

$$\boldsymbol{K} = \sum_{k=1}^{n}\sum_{l=1}^{n} w_k w_l (\nabla \boldsymbol{L})^T \boldsymbol{G} \nabla \boldsymbol{L} |\boldsymbol{J}| \Big|_{(\xi_k,\eta_l)}$$

$$\boldsymbol{F} = \boldsymbol{\tau} J_\Gamma \sum_{l=1}^{n} w_l \boldsymbol{L}_\Gamma^T\big|_{nl}$$

2.2.3 Linear Triangular Elements (CST)

In the case of the linear triangular element, the strain matrix for triangles is found to be made of constants only. Hence it is known as CST (constant strain triangle), and despite it being less accurate than the quadrilateral element, it is still found useful for complex geometries [16]. The solution steps are the same as a quadrilateral element, with the only difference being in the different shape functions employed.

Shape Functions of CST

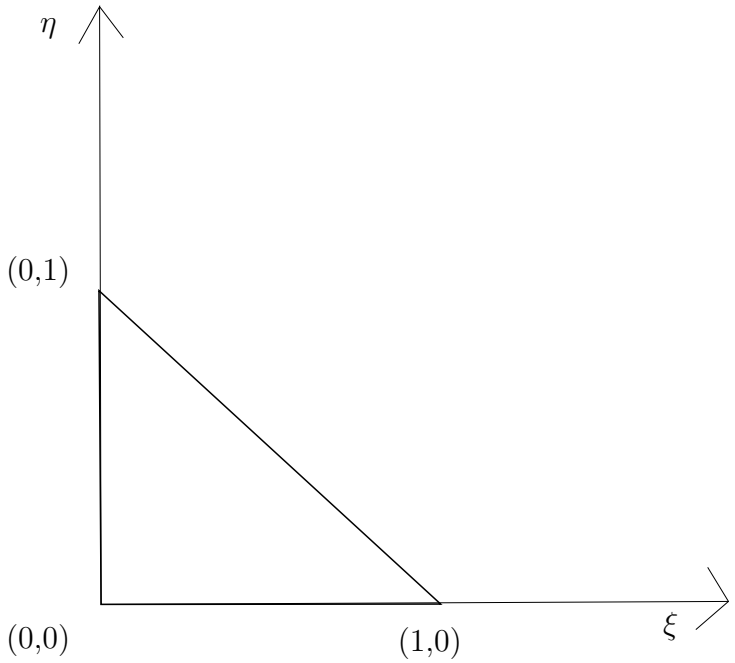


Figure 2.3: Parent Linear Triangle

The definition of the shape functions is required to compute the numerical solution to the system of equations The set of Lagrange shape functions for a triangular parent element in (ξ, η) domain is as follows[19]:

$$L_0 = 1 - \xi - \eta$$

$$L_1 = \xi$$

$$L_2 = \eta$$

The system equations to be solved are:

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{F} \tag{2.35}$$

$$\boldsymbol{u} = \boldsymbol{K}^{-1}\boldsymbol{F} \tag{2.36}$$

$$\boldsymbol{K} = \int_0^1 \int_0^{1-\xi} (\nabla \boldsymbol{L})^T \boldsymbol{G} \nabla \boldsymbol{L} |\boldsymbol{J}| \, d\xi \, d\eta \tag{2.37}$$

$$\boldsymbol{F} = \boldsymbol{\tau} \int_0^1 \boldsymbol{L}^T J_\Gamma \, ds' \tag{2.38}$$

It is evident that equation 2.37 provides that the integrand function is constant. For a linear triangle element, there is no need to introduce an isoparametric element if neglecting the inertial forces. However, in this work, it was chosen to allow the expansion of the implementation of the scheme to different triangle elements of higher order than first order.

Integration of System Equations

To compute the stiffness matrix equation (2.37), $\nabla \boldsymbol{L}$ is needed, derivatives of the shape functions equations:

$$\frac{\partial L_0}{\partial \xi} = -1 \qquad \frac{\partial L_0}{\partial \eta} = -1$$
$$\frac{\partial L_1}{\partial \xi} = 1 \qquad \frac{\partial L_1}{\partial \eta} = 0$$
$$\frac{\partial L_2}{\partial \xi} = 0 \qquad \frac{\partial L_2}{\partial \eta} = 1$$

This makes $\nabla \boldsymbol{L}$ a matrix of constants, while

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} x_k \frac{\partial L_k}{\partial \xi} & y_k \frac{\partial L_k}{\partial \xi} \\ x_k \frac{\partial L_k}{\partial \eta} & y_k \frac{\partial L_k}{\partial \eta} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{\partial L_0}{\partial \xi} & \frac{\partial L_1}{\partial \xi} & \frac{\partial L_2}{\partial \xi} \\ \frac{\partial L_0}{\partial \eta} & \frac{\partial L_1}{\partial \eta} & \frac{\partial L_2}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix}$$

Recall that $\boldsymbol{G}$ is a material property matrix, which is also a matrix of constants, this means the value of the integration of equation (2.37) is:

$$\boldsymbol{K} = (\nabla \boldsymbol{L})^T \boldsymbol{G} \nabla \boldsymbol{L} |\boldsymbol{J}| A$$

where A is the area of the parent element shown in Figure 2.3, Isoparametric mapping is again employed:

$$x(\xi, \eta) = x_k L_k(\xi, \eta)$$

$$y(\xi, \eta) = y_k L_k(\xi, \eta)$$

$$\boldsymbol{F} = \boldsymbol{\tau} \int_0^1 \boldsymbol{L}^T J_\Gamma ds' \tag{2.39}$$

To use Gauss 1D integration it is needed to write the equation in the (-1,1) domain. This changes equation (2.39) into:

$$\boldsymbol{F} = \frac{1}{2} \boldsymbol{\tau} J_\Gamma \int_{-1}^1 \boldsymbol{L}_\Gamma^T ds'$$

$$\boldsymbol{F} = \frac{1}{2} \boldsymbol{\tau} J_\Gamma \sum_{l=1}^{n} w_{nl} \boldsymbol{L}_\Gamma^T \big|_{nl}$$

2.2.4 Assembly

A standard assembly method was used where the global stiffness matrix was assembled from the contribution of element stiffness matrices in a full squared global matrix [20]. The method uses two inputs to perform the assembly process the two inputs are:

1-Elements connectivity matrix: A matrix that holds the information of the mesh, each row contains the global indices for the nodes of one element.

2- Coordinates of the nodes matrix: A matrix that holds the information of the nodes' coordinates, each row contains the x and y coordinates of the node, and the rows are arranged in the global position of each node.

The method is split into three steps:

1- Calculating the stiffness matrix for each element.

2- Looping over all the elements in the mesh.

3-Finding the indices of the coefficients of each node from its local element using the formula:
$[row][column] = [Ni * mvar + vk][Nj * mvar + vk]$ where
Ni and Nj are the global row and column index respectively, $mvar$ is the number of variables which is equal to 2 in the 2D cases, v_k is equal to 0 for the equations in the x direction and equal to 1 for the equations in the y direction.

4-Adding the local stiffness matrix made of the coefficients to their positions in The following pseudocode shows how the element matrices are combined and assembled directly into the complete square global matrix.

---

n = number of degrees of freedom per element
N = total number of degrees of freedom in the domain
mE = number of elements
$C[mE, n]$ = connectivity matrix
$k[n, n]$ = local stiffness matrix

$K[N, N]$ = global stiffness matrix
for $i = 1, N$
for $j = 1, N$
$K[i, j] = 0$
end for
end for
for $e = 1, mE$
generate k
for $i = 1, n$
for $j = 1, n$
$K[C[e, i], C[e, j]] = K[C[e, i], C[e, j]] + k[i, j]$
end for
end for
end for

2.3 Benchmark Test

A computer code was written in C++ in which the aforementioned numerical scheme was implemented. The sections to follow are to prove the correctness of this via modeling a range of benchmark problems.

2.3.1 Uni-Axial Tension

Testing the mathematical scheme considering a 2D body in uni-axial tension, modeled as a concentrated force at some nodes defining the edge where the uni-axial tension is applied, as shown in Figure (2.4) below:

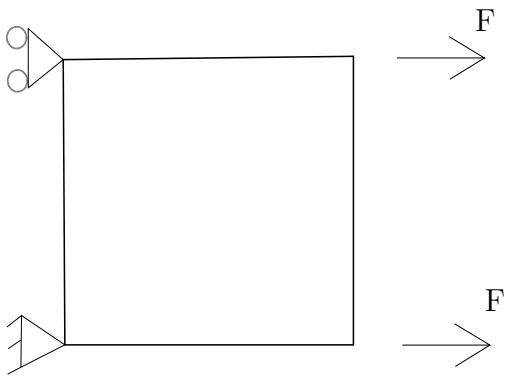


Figure 2.4: Uni-Axial Tension

The material has a Young's modulus of E= 210 GPa, Poission's ratio =0.3, and a length and width of 1 mm. The analytical solution was computed using:

$$\Delta L = \epsilon_{xx} L$$

where ΔL represents the change in length of the material after tension was applied.

$$\Delta d = -\epsilon_{yy} d$$

where Δd represents the verticle change in width of the material after tension was applied.

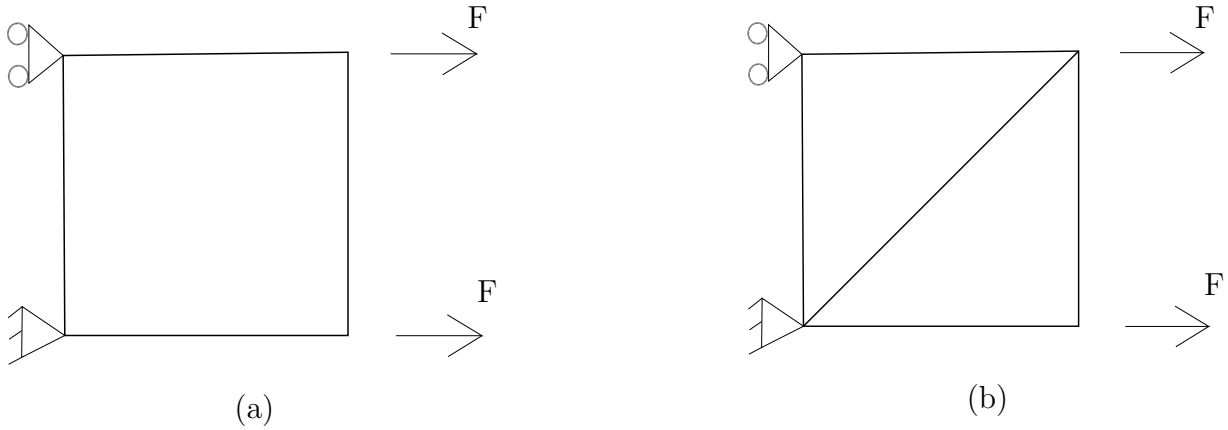


Figure 2.5: Domain under Uni-axial tension (a)Quadrilaterals (b) Triangles

For the mesh above the numerical solution results are as follows:

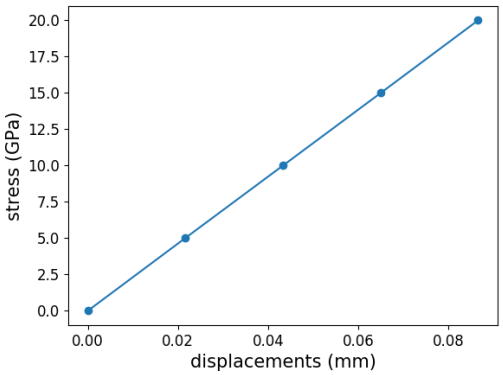


Figure 2.6: Uni-axial stress plot

The plot above shows a linear relation between the displacement and the stress applied perpendicular to the surface. The plots represent the results computed for both of the meshes shown in Figure 2.5.

2.3.2 Pure Bending

The next test case considered was that of a beam in pure bending. The beam was clamped at one end and subjected to a moment at the free end. The material properties used were Young's modulus $E = 210$ GPa and Poisson's ratio $\nu = 0.3$. The length L = 2 m, height h = 0.1 m and a F= 0.1 MN.m, the plane strain assumption were used.

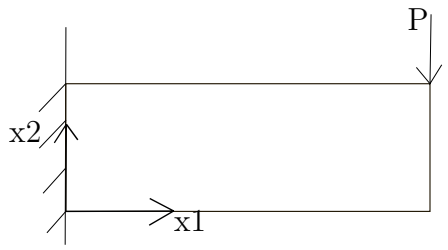


Figure 2.7: Tip load applied to a beam

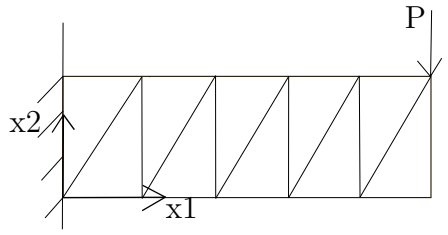


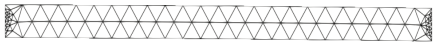
Figure 2.8: Beam under tip load with mesh of triangles shown

The two Figures above show the beam when the load is applied to it, before and after a mesh of triangles is introduced to the beam. The Figure below shows different triangular meshes for the test described above and the error plot in this Figure is explained below.

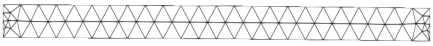
(a)

(b)

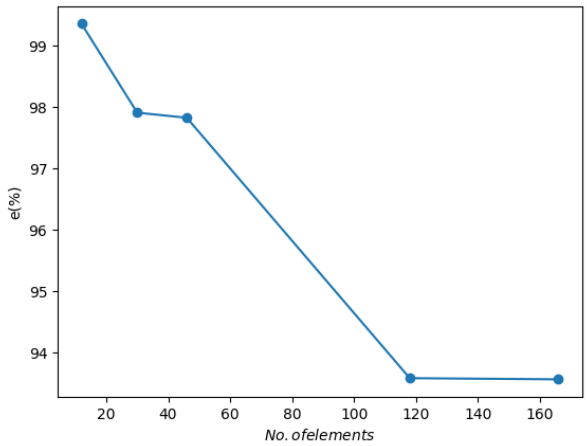
(c)



(d)



(e)



(f)

Figure 2.9: Different triangle meshes and their error plot

The error in the bending case shown in the Figure above is high compared to the uni-axial tension case, due to the elements used being the linear elements which produces shear locking in bending cases [21]. The analytical solution [22]

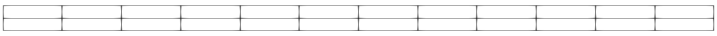
$$v_{ana} = -\frac{Mx_2^2}{2EI}$$

where v_{ana} is the analytical solution. Percent error was used to verify the method in the case of bending.

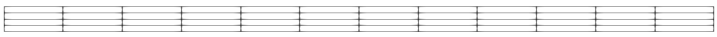
$$e = \frac{|v_{ana} - v_n|}{v_{ana}} * 100$$

where e is percent error, v_n is the displacements in the y direction of the nodes at the tip in a mesh of nx1 elements. The classic locking expected from linear shape functions is shown in Figure(2.9). The Figure below shows different quadrilateral elements' meshes for the same pure bending test described above and the error plot in this Figure is explained below.

(a)



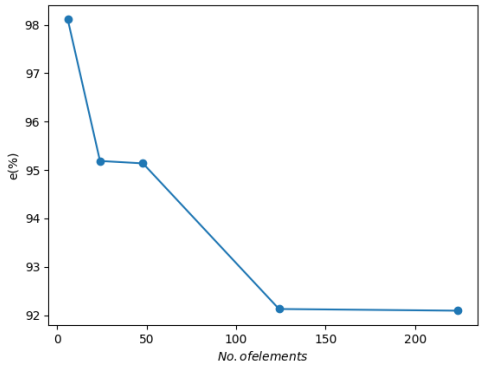
(b)



(c)

(d)

(e)



(f)

Figure 2.10: Different quadrilateral meshes

Similar to the triangular meshes the error in the bending case shown in the Figure above is high compared to the uni-axial tension case, but lower than the triangular meshes under bending due to quadrilateral elements showing less stiffness than triangular elements for this test case.

2.4 Continuous Form of 3D Finite Element Formulation

The solution techniques can be applied similarly in both 2D and 3D FEM. The dimensionality of the governing equations is increased to 3D in this section and a similar discretization method is applied. To balance of forces equations in x, y, and z directions in 3D the governing equation becomes:

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} = 0 \tag{2.40}$$

$$\frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} = 0 \tag{2.41}$$

$$\frac{\partial \sigma_{zx}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} = 0 \tag{2.42}$$

Writing the constitutive equation again assuming Hooke's law yields:

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{zx} \\ \sigma_{yz} \end{bmatrix} = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{xy} \\ \gamma_{zx} \\ \gamma_{yz} \end{bmatrix} = \boldsymbol{G\epsilon}$$

where $\lambda = \dfrac{v}{(1+v)(1-2v)}E$ and $\mu = \dfrac{1}{2(1+v)}E$ ϵ and γ are the normal and shear strains respectively. E is Young's modulus of elasticity and v is poisson's ratio. Finally introducing the relationship between the strain and displacement field:

$$\epsilon_{ij} = \frac{1}{2}[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}] \tag{2.43}$$

In 3D $\boldsymbol{u}$ has 3 components:

$$\boldsymbol{u}(x,y,z) = \begin{bmatrix} u(x,y,z) \\ v(x,y,z) \\ w(x,y,z) \end{bmatrix}$$

2.5 Discrete Form

Recall the discretization method explained in equations (2.5-2.14) After integration by parts, the governing equation for an element results in weak form:

$$\int_{\Omega_e} (\nabla \boldsymbol{L})^T \boldsymbol{\sigma}\, d\Omega_e = \oint_{d\Omega_e} \boldsymbol{L}^T \boldsymbol{\sigma}\, \boldsymbol{n}\, ds \tag{2.44}$$

Make $\nabla \boldsymbol{L} = \boldsymbol{B}$ where $\boldsymbol{B}$ is made of submatrices as follows:

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{B}_2 & \boldsymbol{B}_3 & . & . & . & \boldsymbol{B}_i \end{bmatrix} \tag{2.45}$$

where i is the number of nodes in the element

$$\boldsymbol{B}_i = \begin{bmatrix} \frac{\partial L_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial L_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial L_i}{\partial z} \\ \frac{\partial L_i}{\partial y} & \frac{\partial L_i}{\partial x} & 0 \\ \frac{\partial L_i}{\partial z} & 0 & \frac{\partial L_i}{\partial x} \\ 0 & \frac{\partial L_i}{\partial z} & \frac{\partial L_i}{\partial y} \end{bmatrix} \tag{2.46}$$

The RHS of equation (3.5) can be further reduced by using this relation:

$$\tau_i = \sigma_{ji} n_j \tag{2.47}$$

This means the components of stress acting on any point are a linear combination of the components of the stress vectors acting on the coordinate planes at that point. This gives us:

$$\tau_x = \sigma_{xx} n_x + \sigma_{xy} n_y + \sigma_{xz} n_z$$

$$\tau_y = \sigma_{yx} n_x + \sigma_{yy} n_y + \sigma_{yz} n_z$$

$$\tau_z = \sigma_{zx} n_x + \sigma_{zy} n_y + \sigma_{zz} n_z$$

Here τ_x, τ_y and τ_z are the boundary fluxes. Recall equation (2.44) and subsection 2.2.1, the FEM obtains an approximate solution by using the classical trial-solution procedure, the trial solution for the three

displacement components u and v :

$$u(x,y,z) = u_j L_j(x,y,z)$$

$$v(x,y,z) = v_j L_j(x,y,z)$$

$$w(x,y,z) = w_j L_j(x,y,z)$$

Substituting the constitutive equations to introduce the displacement vector, first substituting the trial solution of displacement into the strain vector:

$$\epsilon_{xx} = \frac{\partial u}{\partial x} = u_j \frac{\partial L_j}{\partial x}$$

$$\epsilon_{yy} = \frac{\partial v}{\partial y} = v_j \frac{\partial L_j}{\partial y}$$

$$\epsilon_{zz} = \frac{\partial w}{\partial z} = w_j \frac{\partial L_j}{\partial z}$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = u_j \frac{\partial L_j}{\partial y} + v_j \frac{\partial L_j}{\partial x}$$

$$\gamma_{yz} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} = v_j \frac{\partial L_j}{\partial z} + w_j \frac{\partial L_j}{\partial y}$$

$$\gamma_{xz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} = u_j \frac{\partial L_j}{\partial z} + w_j \frac{\partial L_j}{\partial x}$$

As previously, this leads to:

$$\int_{\Omega_e} (\nabla \boldsymbol{L})^T \boldsymbol{G} (\nabla \boldsymbol{L}) \boldsymbol{u} d\Omega_e = \int_{d\Omega_e} \boldsymbol{L}^T \boldsymbol{\tau} ds \tag{2.48}$$

or:

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{F} \tag{2.49}$$

2.5.1 Hexahedron Element

A commonly used 3D element is a hexahedron for its accuracy compared to Tetrahedral elements [23].

Shape functions:

The shape functions for the eight nodes of the above element are as follows:

$$L_i(\xi,\eta,\zeta) = \frac{1}{8}(1+\xi\xi_i)(1+\eta\eta_i)(1+\zeta\zeta_i) \tag{2.50}$$

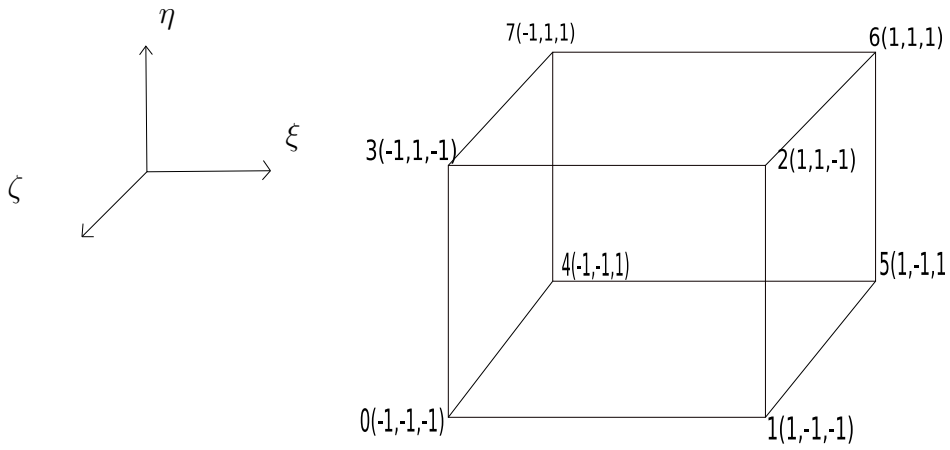


Figure 2.11: Parent Hexahedron

where i is the index of the node. This can also be written as:

$$\boldsymbol{L} = \sum_n \psi_i(\xi)_n \psi_j(\eta)_n \psi_k(z)_n \tag{2.51}$$

where

$$\psi_i(\xi)_0 = \frac{(\xi - \xi_0)}{2} \tag{2.52}$$

$$\psi_j(\eta)_0 = \frac{(\eta - \eta_0)}{2} \tag{2.53}$$

$$\psi_k(z)_0 = \frac{(z - z_0)}{2} \tag{2.54}$$

$$\boldsymbol{L}_{\prime\xi} = \sum \psi_{i\prime\xi} \psi_j \psi_k \tag{2.55}$$

where $\boldsymbol{L}_{\prime\xi}$ and $\psi_{i\prime\xi}$ mean the derivative of $\boldsymbol{L}_\xi$ and ψ_i with respect to ξ

$$\boldsymbol{L}_{\prime\eta} = \sum \psi_i \psi_{j\prime\eta} \psi_k \tag{2.56}$$

Applying again isoparametric mappings yields:

$$\begin{bmatrix} \frac{\partial L_i}{\partial x} \\ \frac{\partial L_i}{\partial y} \\ \frac{\partial L_i}{\partial z} \end{bmatrix} = \boldsymbol{J}^{-1} \begin{bmatrix} \frac{\partial L_i}{\partial \xi} \\ \frac{\partial L_i}{\partial \eta} \\ \frac{\partial L_i}{\partial \zeta} \end{bmatrix} \tag{2.57}$$

i.e:

$$\frac{\partial L_i}{\partial x} = J_{00}^{-1}\frac{\partial L_i}{\partial \xi} + J_{01}^{-1}\frac{\partial L_i}{\partial \eta} + J_{02}^{-1}\frac{\partial L_i}{\partial \zeta} \tag{2.58}$$

$$\frac{\partial L_i}{\partial y} = J_{10}^{-1}\frac{\partial L_i}{\partial \xi} + J_{11}^{-1}\frac{\partial L_i}{\partial \eta} + J_{12}^{-1}\frac{\partial L_i}{\partial \zeta} \tag{2.59}$$

$$\frac{\partial L_i}{\partial z} = J_{20}^{-1}\frac{\partial L_i}{\partial \xi} + J_{21}^{-1}\frac{\partial L_i}{\partial \eta} + J_{22}^{-1}\frac{\partial L_i}{\partial \zeta} \tag{2.60}$$

Numerical Integration

To again employ Gauss quadrature efficiently, all expressions are to be written in terms of (ξ, η, ζ). Hence the volume integrant is written as:

$$d\Omega_e = |\boldsymbol{J}| d\xi d\eta d\zeta \tag{2.61}$$

from which the stiffness matrix is computed as:

$$\boldsymbol{K} = \sum_{k=1}^{n}\sum_{l=1}^{n}\sum_{m=1}^{n} w_k w_l w_m (\nabla \boldsymbol{L})^T \boldsymbol{G} \nabla \boldsymbol{L} |\boldsymbol{J}| \Big|_{(\xi_k, \eta_l, \zeta_m)} \tag{2.62}$$

The RHS, in the x-y-z coordinate system:

$$\boldsymbol{F}(x, y, z) = \int_{d\Omega_e} \boldsymbol{L}^T \boldsymbol{\tau} ds \tag{2.63}$$

becomes:

$$\boldsymbol{F}(\xi, \eta, \zeta) = \oint \boldsymbol{L}^T \boldsymbol{\tau} J_\Gamma ds'$$

where ds' is the surface area in ξ, η, ζ coordinate system. J_Γ is the Jacobian of the surface and is a matrix made of constants which changes the above equation into:

$$\boldsymbol{F} = J_\Gamma \boldsymbol{\tau} \oint \boldsymbol{L}^T ds'$$

Also, $\boldsymbol{L}^T$ leads to the load vector being evaluated at each node at the surface where a stress is applied as follows:

$$\boldsymbol{F} = J_\Gamma \boldsymbol{\tau} A$$

where A is the area of the parent element surface.

2.5.2 Linear Tetrahedral Elements

The linear Tetrahedral element like its 2D equivalent gives a constant strain matrix since its shape functions -shown in the next subsection- are all linear functions.

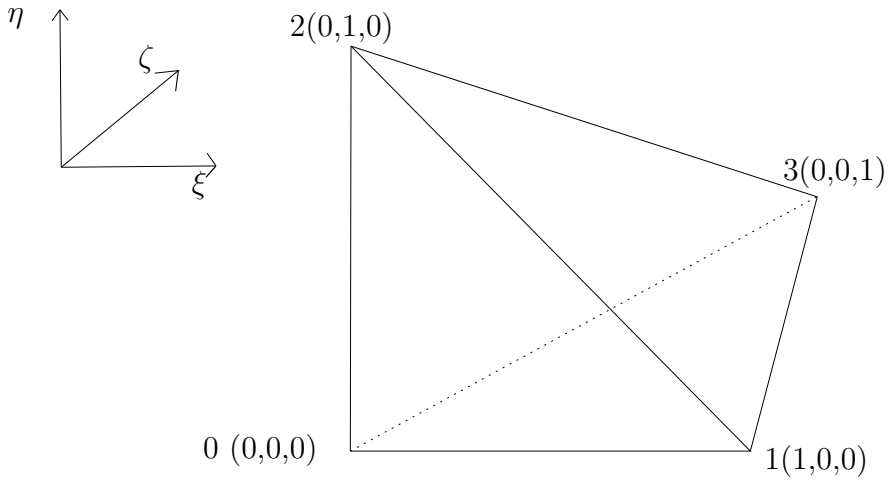


Figure 2.12: Parent Linear Tetrahedral

Expressing the displacement at any point in the element using the displacements at the nodes and shape functions.

$$\boldsymbol{U}^h(x,y) = \boldsymbol{L}(x,y)\boldsymbol{d}_e \tag{2.64}$$

where the superscript h indicates that the displacement is approximated, and $\boldsymbol{d}_e$ is a vector of the nodal displacements arranged in the order of

$$\boldsymbol{d}_e = \begin{bmatrix} u_0 \\ v_0 \\ u_1 \\ v_1 \\ u_2 \\ v_2 \end{bmatrix} \tag{2.65}$$

and the matrix of shape functions N is arranged as

$$\boldsymbol{L} = \begin{bmatrix} L_0 & 0 & L_1 & 0 & L_2 & 0 \\ 0 & L_0 & 0 & L_1 & 0 & L_2 \end{bmatrix} \tag{2.66}$$

in which $L_i (i = 1, 2, 3)$ are three shape functions corresponding to the three nodes of the triangular element

Shape Functions of Linear Tetrahedral Elements

The following are the Lagrangian shape functions of the tetrahedral element [19], shown on Figure (3.2):

$$L_0 = 1 - \xi - \eta - \zeta$$

$$L_1 = \xi$$

$$L_2 = \eta$$

$$L_3 = \zeta$$

The system equations that need to be solved are:

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{F} \tag{2.67}$$

$$\boldsymbol{u} = \boldsymbol{K}^{-1}\boldsymbol{F} \tag{2.68}$$

$$\boldsymbol{K} = \int_0^1 \int_0^1 \int_0^1 (\nabla \boldsymbol{L})^T \boldsymbol{G} \nabla \boldsymbol{L} |\boldsymbol{J}| d\xi d\eta d\zeta \tag{2.69}$$

$$\boldsymbol{F} = \oint \boldsymbol{L}^T \boldsymbol{\tau} J_\Gamma ds' \tag{2.70}$$

2.6 Benchmark Tests

Uni-Axial Tension

Using the same test as in the 2D case where the material has a Young's modulus of E= 210 GPa, Poission's ratio =0.3, and a length, width, and height of 1 mm. The structured mesh consisted of a single cell and the stress varies from 0 to 20 GPa.

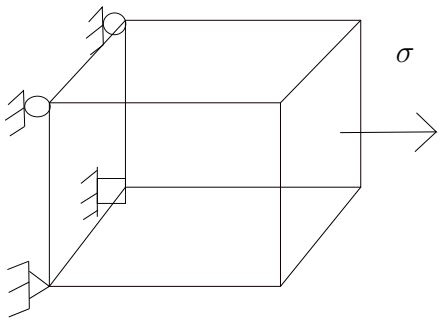


Figure 2.13: 3D domain under uni-axial tension

Results are found to be equal to the analytical solution. The analytical solution was computed using:

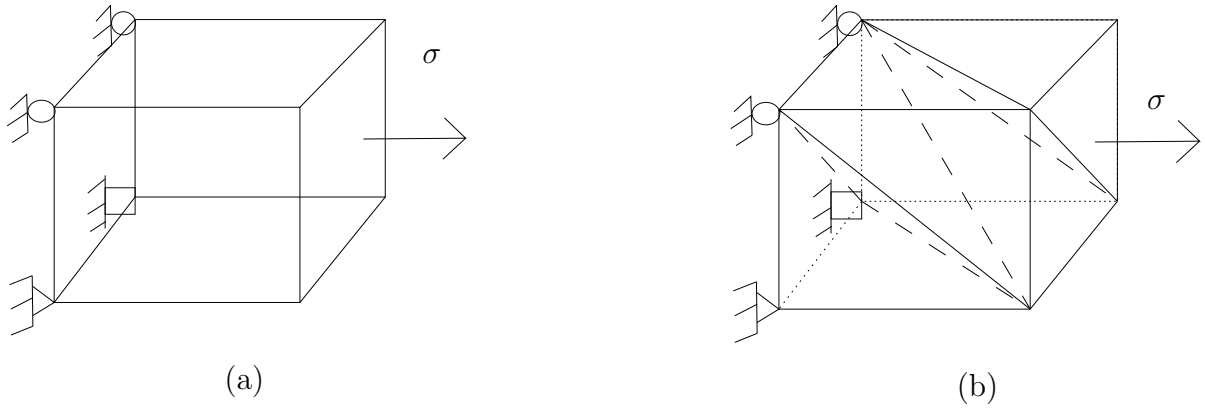
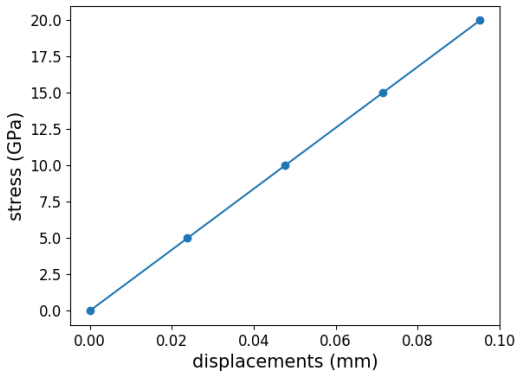


Figure 2.14: A mesh consisting of (a)one hexahedron (b) six tetrahedrals



$\boldsymbol{\sigma} = 10$ GPa, $\boldsymbol{E} = 210$ GPa

$$\Delta L = \epsilon_{xx} L = 0.047619$$

where ΔL represents the change in length of the material in the direction tension was applied.

$$\Delta d = -\epsilon_{yy} d = -0.0142857$$

where Δd represents the changes in the length of the material in the lateral direction to where the tension was applied.

Similar to the 2D test case, a 3D domain under uni-axial tension was examined. The test revealed a linear relationship between stress and displacement, with the displacements closely matching the analytical solution.

Pure Bending

Similar to the benchmark tests for the 2D scheme, pure bending was tested for a 3D Beam. The beam was clamped at one end and subjected to a moment at the free end. The material properties used were

a Young's modulus $E = 210$ GPa and Poisson's ratio $\nu = 0.3$. The length L = 2 m, height h = 0.1 m, thickness t=0.1 m and a F= 0.1 MN.m.

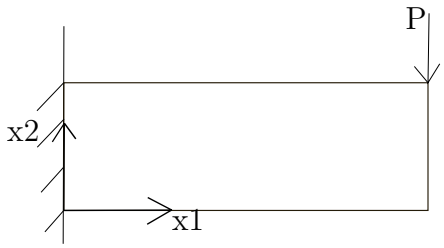
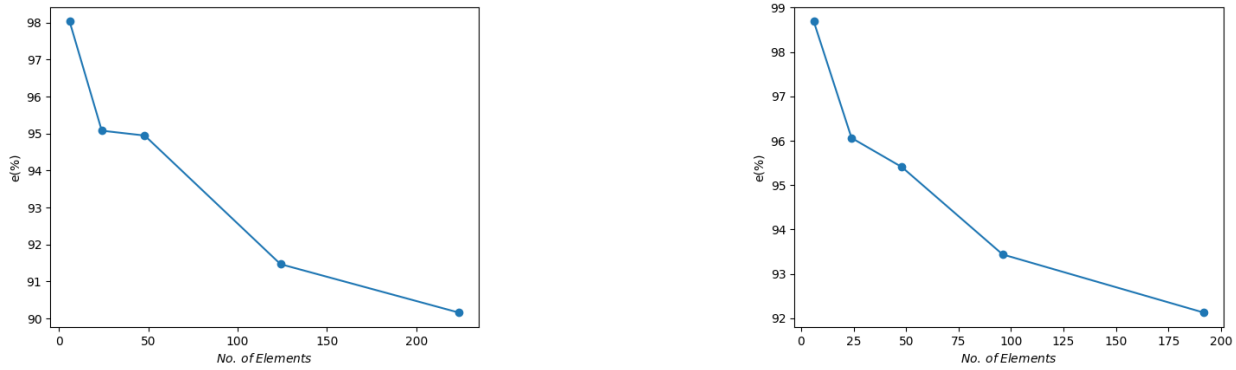


Figure 2.15: Tip load applied to a beam

The bending case served as a benchmark for evaluating the performance of the 3D elements, similar to the 2D case. The error plots for various numbers and types of elements are presented below. The plot below illustrates the error computed in determining the displacement resulting from a bending case applied both in tetrahedral meshes and hexahedral meshes.



(a) (b)

Figure 2.16: Error against number of elements for bending test case

Similar to the 2D Benchmark test case of pure bending, the error found in the bending case is high compared to the uni-axial tension case, due to the elements used being linear elements which produces shear locking in bending cases especially for full integration quadrilateral and hexahedral elements. The error decreases when increasing the number of elements for both tetrahedral and hexahedral meshes.

2.7 Modification to the FEM scheme

Recall the general formula for obtaining the stiffness of an element:

$$\boldsymbol{K} = (\nabla \boldsymbol{L})^T \boldsymbol{G} \nabla \boldsymbol{L} |\boldsymbol{J}| A$$

The formula above has been modified to improve the quality of the mesh after undergoing deformation as discussed in the introduction of this thesis. Two methods were used, they are the Jacobian-based stiffening method and the distance method. 1-The Jacobian-based stiffening method is a technique used to control the quality of a mesh by adjusting the stiffness of smaller elements relative to larger ones. This adjustment is achieved by modifying the determinant of the Jacobian, denoted as $|\boldsymbol{J}|$. By introducing a stiffening factor, the control over mesh quality is extended, allowing for more flexibility and improved results [17]. To implement the Jacobian stiffening method, the following formula is used:

$$|\boldsymbol{J}|_{modified} = |\boldsymbol{J}|_{stiffening\ factor}$$

In this formula, $|\boldsymbol{J}|_{\text{modified}}$ represents the modified determinant of the Jacobian, obtained by multiplying the original determinant $|\boldsymbol{J}|$ by the stiffening factor. By adjusting the stiffening factor, you can influence the stiffness of the smaller elements relative to the larger ones, thereby enhancing the quality and behavior of the mesh. The exact value of the stiffening factor depends on the specific requirements and characteristics of the mesh and should be carefully chosen to achieve the desired results.

$$\boldsymbol{K} = \int_{-1}^{1}\int_{-1}^{1} (\nabla \boldsymbol{L})^T \boldsymbol{G} \nabla \boldsymbol{L} |\boldsymbol{J}| \left(\frac{J^0}{|\boldsymbol{J}|}\right)^{\chi} d\xi d\eta \tag{2.71}$$

where $\boldsymbol{K}$ is the stiffness matrix and J^0 is an arbitrary scaling parameter that is inserted into the formulation to make the alteration dimensionally consistent [24] and χ is the stiffening factor. 2-Another method is the distance method. A technique used to control mesh quality and modify material properties is found in equation 2.3. It involves defining a distance function that assigns values based on proximity to specific features. By using this function, material properties can be adjusted based on distance, allowing for smooth transitions and accurate representation of complex geometries [15]:

$$E \propto \frac{1}{d}$$

where E is Young's modulus and d is the distance between the element and the moving boundary.

2.7.1 Solving a Linear System

Iterative solvers are used to efficiently and accurately solve the linear system of equations. These solvers iteratively improve the solution by updating it based on the residual [25]. Popular iterative solvers include Conjugate Gradient (CG) for symmetric systems and Generalized Minimal Residual (GMRES) for nonsymmetric systems. Iterative solvers handle large-scale problems, avoid storing dense matrices, and can exploit problem-specific structures. They provide an efficient and flexible approach to solving linear

FEM systems. A bi-conjugate gradient stabilized solver for a sparse stiffness matrix:

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{F}$$

Chapter 3

Verification Of The FEM Scheme:

The objective of this chapter is to validate the method and its implementation by comparing it to a previous study that involved rotation and translation of inner boundaries. Three other tests used for verification and validation involve examining a full rotation case as well as vertical and diagonal translations to explore various potential failure scenarios.

3.0.1 Prescribed Displacements

The Dirichlet boundary condition designates a fixed value at the boundary [26], used to prescribed displacements. By making the row relevant to the degree of freedom to be equal to zero except for the diagonal entry, and putting the value of the displacement prescribed in the load vector as follows:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ K_{10} & . & . & K_{13} \\ . & . & . & . \\ K_{30} & . & . & K_{33} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} a \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.1}$$

where a is the value prescribed for the degree of freedom 0 whose displacement is $u_0 = a$.

3.0.2 Application of Modification methods

First, the Jacobian-based stiffening. It renders the smaller elements stiffer than the larger ones by altering the determinant of the Jacobian $|\boldsymbol{J}|$. A modification of adding a stiffening factor widens the control of mesh quality [17]. Recall the formula for the Jacobian stiffening method:

$$\boldsymbol{K} = \int_{-1}^{1}\int_{-1}^{1} (\nabla \boldsymbol{L})^T \boldsymbol{G} \nabla \boldsymbol{L} |\boldsymbol{J}| \left(\frac{J^0}{|\boldsymbol{J}|}\right)^{\chi} d\xi d\eta \tag{3.2}$$

where $\boldsymbol{K}$ is the stiffness matrix and J^0 is an arbitrary scaling parameter that is inserted into the formulation to make the alteration dimensionally consistent [24] and χ is the stiffening factor.

Test Problem

The purpose of performing this test is to verify the method and its implementation, below is a description of the test problem and the mesh metric used for verification of the accuracy of the scheme. A simple test problem involving translation and rotation. The domain consists of a 200x200 units square domain with an inner boundary being a rectangle. The inner rectangle is translated by y-direction the initial mesh shown in Figure 3.1(b) was generated using Pointwise the mesh consists of triangle elements. The Figure below shows the initial meshes and the mesh after deformation for two implementations, they both use the same stiffening method. For this test stiffening factor $\chi = 1.0$ was chosen [17]. Plotted is the value of mesh metric in greyscale. The mesh metric used in this plot is further discussed below

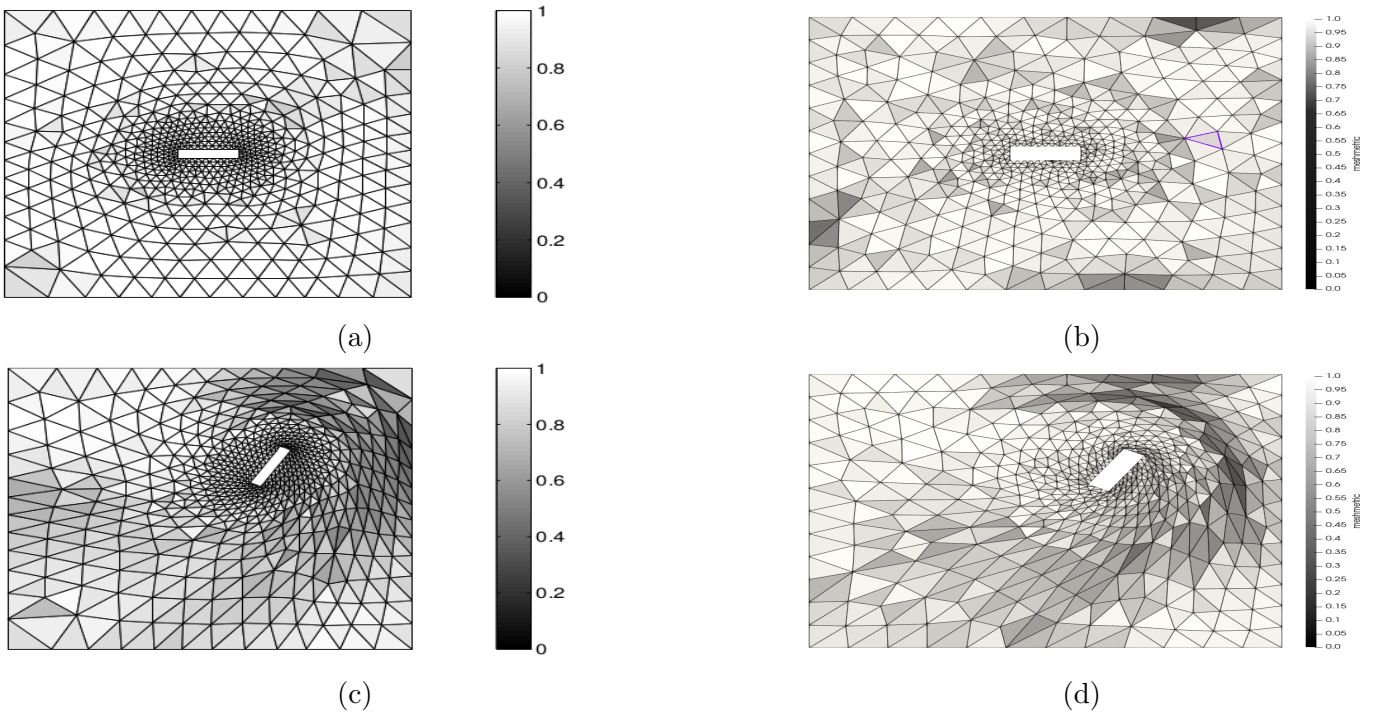


Figure 3.1: (a)Shows the initial mesh Boager used created using NETGEN, (b)the initial mesh created using Pointwise,(c)Boager's mesh after movement, and (d)our mesh verified after movement

Mesh quality metric f_{ss} is given by the following formula [17]:

$$f_{\text{ss}} = \sqrt{f_{\text{size}}}\, f_{\text{shape}} \tag{3.3}$$

where f_{size} and f_{shape} are metrics associated with the size and shape of the element respectively. Below is a breakdown of formulas used to obtain f_{ss} .

1- f_{ss} ranging from $(0-1)$, 1 being a perfect element and 0 a degenerate element. f_{shape} ranges also between $(0-1)$, $f_{\text{shape}} = 1.0$ for an equilateral triangle. To obtain the value of f_{shape} the following formulas are used:

$$f_{\text{shape}} = \frac{\sqrt{3}\alpha}{\lambda_{11} + \lambda_{22} - \lambda_{12}}. \tag{3.4}$$

$$\alpha = \det(\boldsymbol{A}) \tag{3.5}$$

A Jacobian matrix for the element a triangular element, around node k is

$$\boldsymbol{A} = \begin{bmatrix} x_{k+1} - x_k & x_{k+2} - x_k \\ y_{k+1} - y_k & y_{k+2} - y_k \end{bmatrix}. \tag{3.6}$$

where k is the vertices of the triangle. and λ_{ij} is component of the shape metric $\boldsymbol{\lambda}$ defined as

$$\boldsymbol{\lambda} = \boldsymbol{A}^T \boldsymbol{A} \tag{3.7}$$

2- f_{size} The size metric is defined as

$$f_{\text{size}} = \frac{2\tau}{1 + \tau^2} \tag{3.8}$$

where

$$\tau = \frac{w_d}{w_o} \tag{3.9}$$

w_d and w_o are the volumes of element after and before deformation respectively. The range of f_{size} is $0 \leq f_{\text{size}} \leq 1$, where 1 is obtained if the mesh before and after deformation had the same size.

Results:

To compare the two results for the sake of verifying the method used in this thesis. A metric for the deviation of the two solutions was plotted. The formula of the deviation was:

$$Error = \frac{|f_{\text{ss0}} - f_{\text{ss1}}|}{f_{\text{ss0}}} * 100 \tag{3.10}$$

where f_{ss0} is the metric of mesh produced by Boager's and f_{ss1} is the metric of mesh produced by the method described in this thesis.

The Figure below shows the deviation of the mesh metric for different numbers of elements for the same test described above, the plot shows convergence of the mesh metric values for finer meshes.

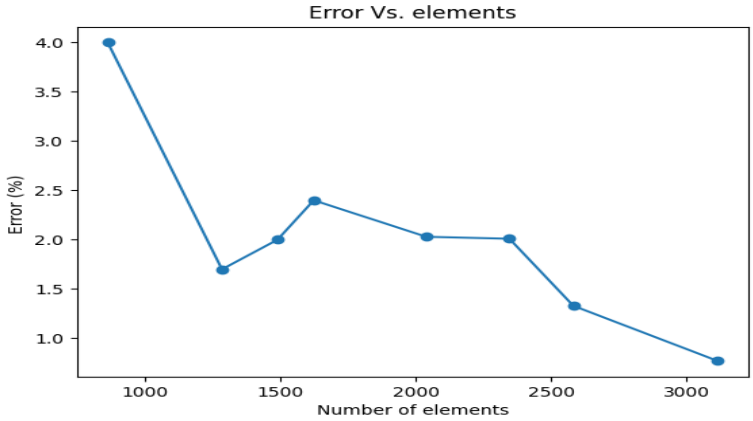


Figure 3.2: A plot of the error Vs. the number of elements used in the mesh

Full Rotation:

A simple test problem involving rotation only was conducted to examine the conditions that produce failure of the scheme. The domain consists of a 200x200 units square domain with an inner boundary being a rectangle. The inner rectangle rotated by 0° -360°.

The mesh shown in Figure 3.3(a) was generated using Pointwise the mesh consists of triangle elements. The Figure below shows the initial meshes and the mesh after deformation. For this test stiffening factor $\chi = 1.0$ was chosen [17]. The mesh metric used in this plot is f_{ss} found in equation (3.3) above.

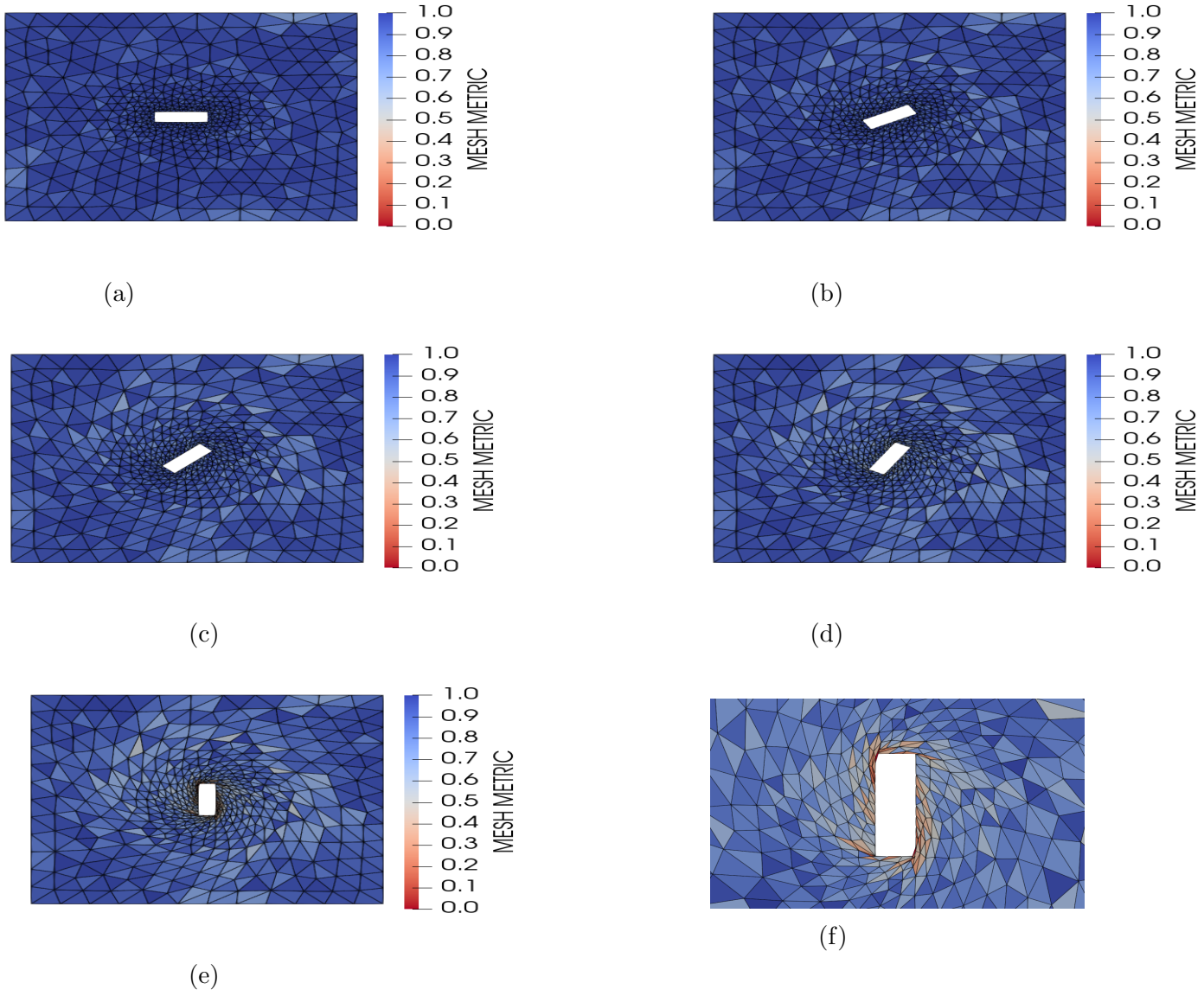


Figure 3.3: Representation of different angles of rotation (a) Shows the initial mesh (b-e) shows the mesh when rotated by 30°-90° (f) shows a close snapshot of the mesh around the rectangle when rotated by 90°

The mesh deformation results produced overlapping elements at rotation by 90° shown in the Figure above. 2-Same rotation described above was applied to a mesh consisting of quadrilateral elements, the stiffening method was the Jacobian method stiffening factor $\chi = 1.0$, and the distance method was tested for this mesh and produced better results than the Jacobian-stiffening. In this method the stiffness of the element is controlled by altering the magnitude of Young's modulus depending on the distance between the element and the boundary, the relationship between Young's modulus and the distance criterion can be defined as [15]:

$$E \propto \frac{1}{d}$$

where E is Young's modulus and d is the distance between the element and the moving boundary.

$$E = \frac{E_{max}}{d^2}$$

where E_{max} is a contsant chosen to be $= 250$. For the structured mesh the numerical integration for quadrilateral elements differs from the one for triangle elements, recall numerical integration from Chapter 2: Gauss Quadrature in 2D is written:

$$\int_{-1}^{1}\int_{-1}^{1} I(\xi,\eta)d\xi d\eta = \sum_{k=1}^{n}\sum_{l=1}^{n} w_k w_l I(\xi_k,\eta_l)$$

where w_{nk}, w_{nl} are Gauss weights and ξ_{nk}, η_{nl} are Gauss points

The discrete stiffness matrix becomes

$$\boldsymbol{K} = \sum_{k=1}^{n}\sum_{l=1}^{n} w_k w_l (\nabla \boldsymbol{L})^T \boldsymbol{G} \nabla \boldsymbol{L} |\boldsymbol{J}| \Big|_{(\xi_k,\eta_l)} \tag{3.11}$$

1- One gauss point was used where: $w_k w_l = 4.0$ and $(\xi_k, \eta_l) = (0.0, 0.0)$ shown in Figure 2.2(a)

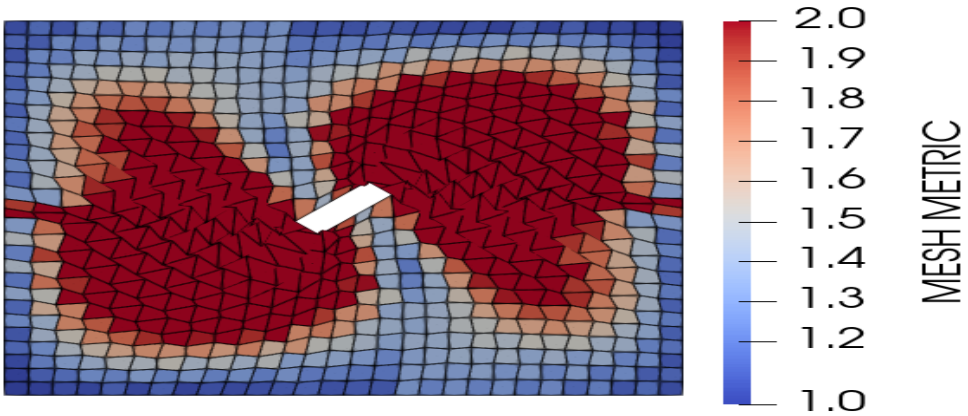


Figure 3.4

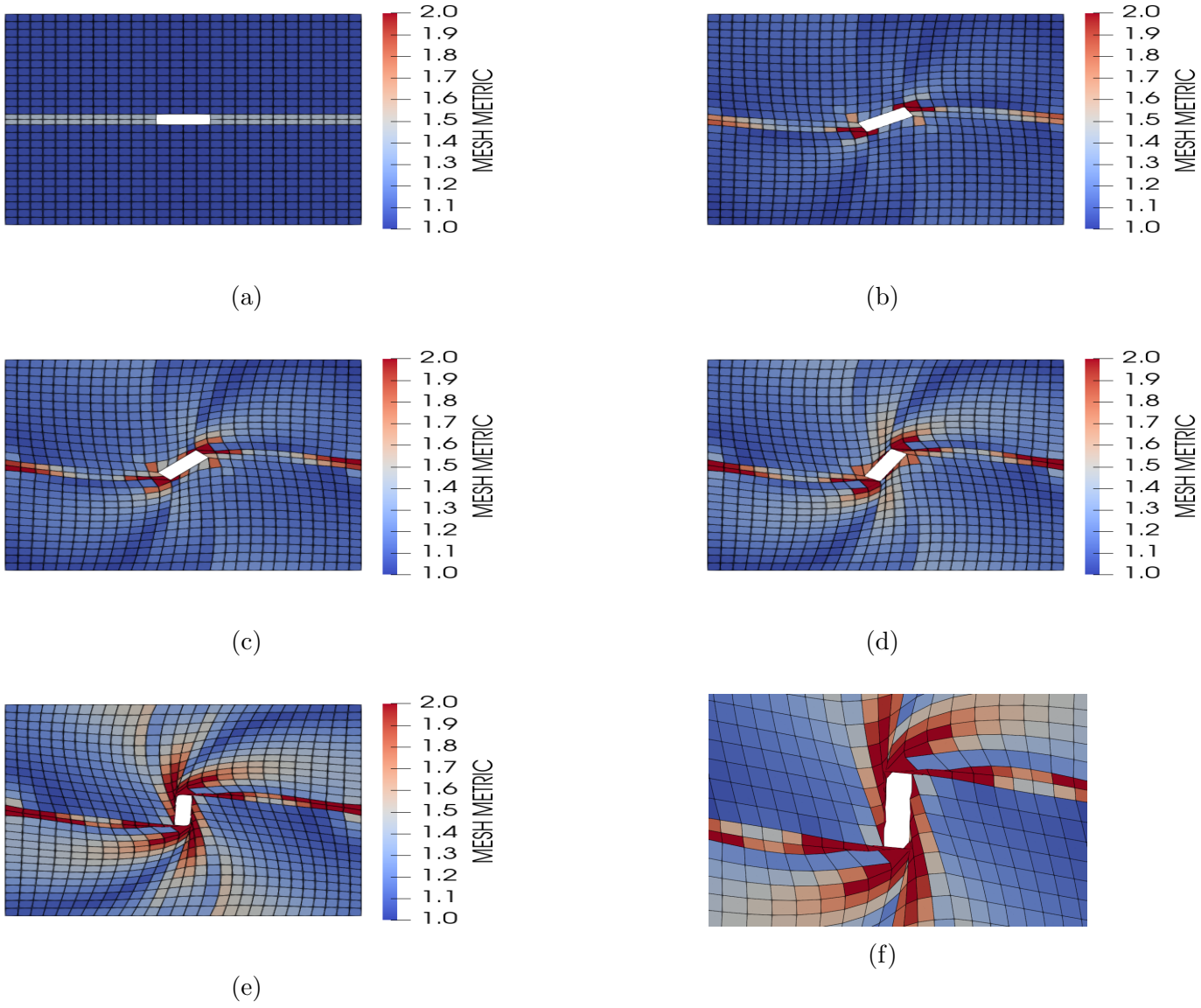
The mesh deformation results produced overlapping elements at rotation by 45°,

2- Four Gauss points of integration were used below where: $w_k w_l = 1.0$ and the Gauss points shown in Figure 2.2(b)

The Mesh quality metric plotted is the aspect ratio for a quadrilateral element found by the following formula:

$$\alpha = \frac{L_{max}}{L_{min}}$$

where α is the aspect ratio of an element, $L_{max}\ and\ L_{min}$ are the maximum and minimum lengths of the side of a quadrilateral respectively. that produced the results shown below



(a) (b) (c) (d) (e) (f)

Figure 3.5: Representation of different angles of rotation (a) Shows the initial mesh (b-e) shows the mesh when rotated by 30°-90° (f) shows a close snapshot of the mesh around the rectangle when rotated by 90°

The mesh deformation results produced overlapping elements at rotation by 90° similar to the triangular mesh.

Translation:

Another test problem involving translation only was incorporated to verify the robustness of the scheme. The tests involved prescribing displacements of the inner boundary both in vertical and diagonal directions for meshes consisting of triangles and quadrilaterals. The domain consists of a 200x200 units square domain with an inner boundary being a rectangle. The inner rectangle underwent translation in the vertical direction only. The Figure below shows a triangular mesh where the inner boundary undergoes a translation in the upward vertical direction by 5 units in each step.

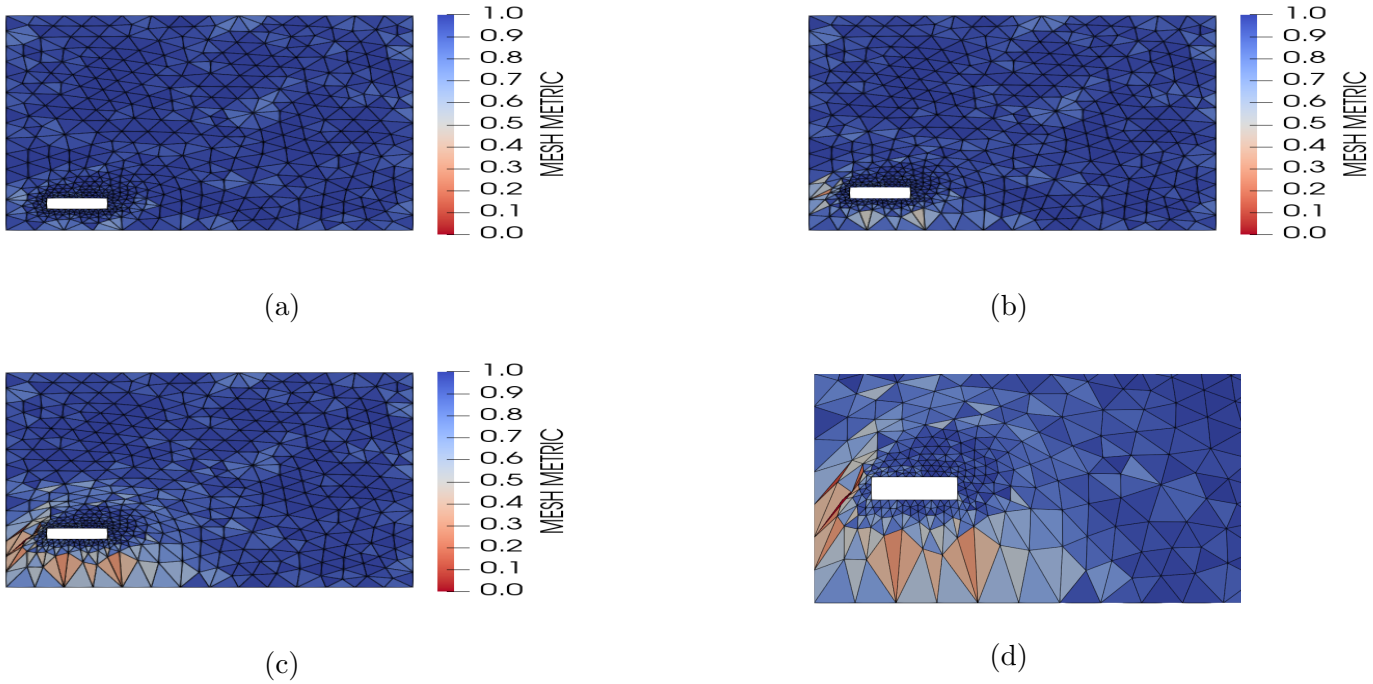


Figure 3.6: Representation of different magnitudes of translation (a) Shows the initial mesh (b and c) show the mesh after translation by 10 and 25 units respectively (d) shows a close snapshot of the mesh translated by 25 units

The mesh deformation results produced overlapping elements after 25 units of vertical translation as shown in the Figure above.

The Figure below shows a mesh consisting of quadrilateral elements where the inner boundary undergoes a translation in the upward vertical direction by 5 units in each step. The distance method explained in Chapter 2 is used and produced the results shown in the Figure below:

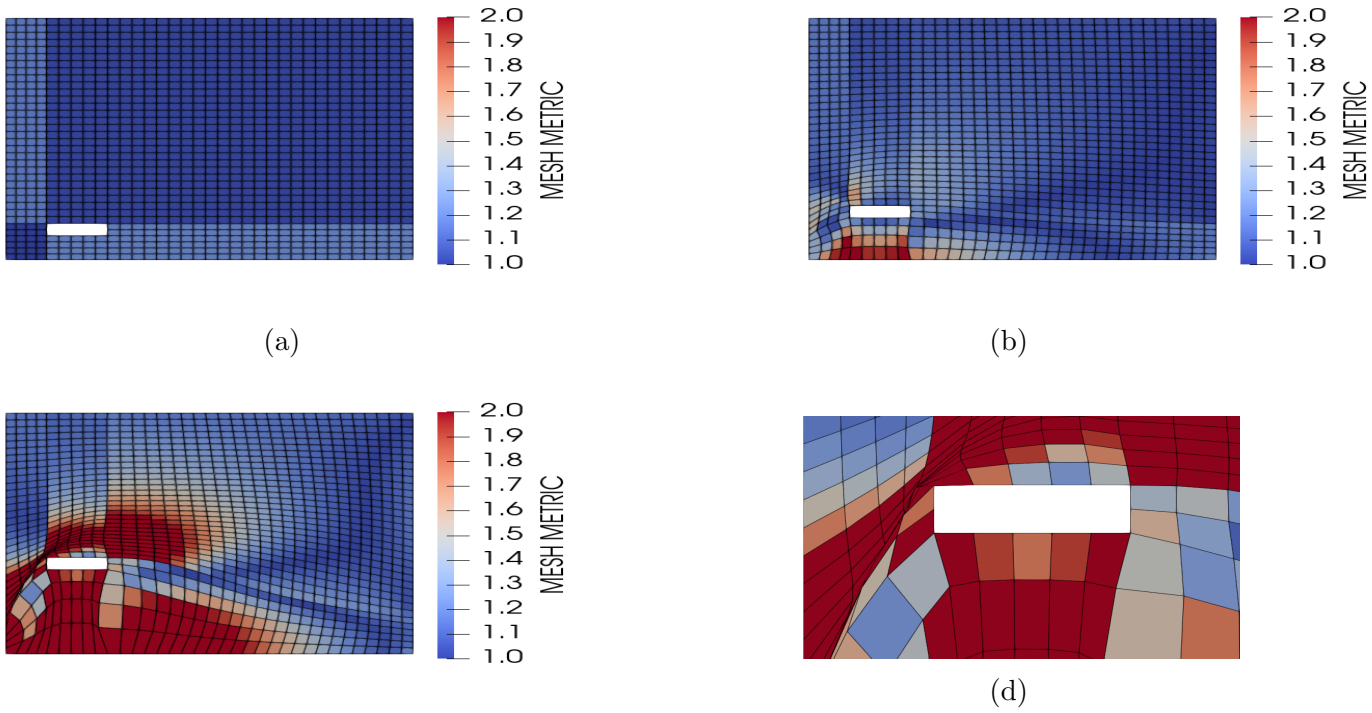


Figure 3.7: Representation of different magnitudes of translation (a) Shows the initial mesh (b-e) shows the mesh after translation by units (d) shows a close snapshot of the mesh translated by 50 units

The mesh deformation results produced overlapping elements after 50 units of translation vertically.

The Figure below shows a triangular mesh where the inner boundary undergoes a translation in both vertical and horizontal directions equally by 5 units in each step.

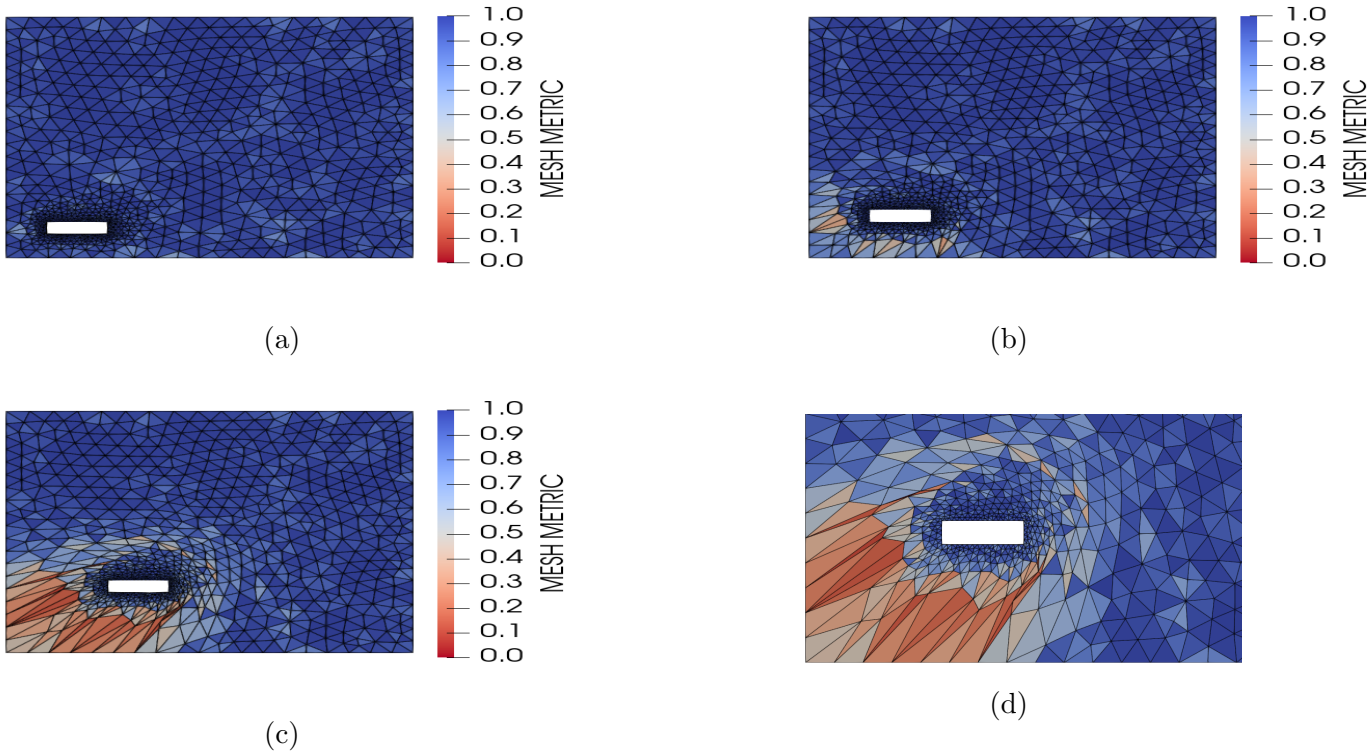


Figure 3.8: Representation of different magnitudes of translation (a) Shows the initial mesh (b-d) shows the mesh after translation (e) shows a close snapshot of the mesh translated by 30 units

The mesh deformation results produced overlapping elements after 30 units of translation.

The Figure below shows a mesh consisting of quadrilaterals where the inner boundary undergoes a translation in both vertical and horizontal directions equally by 5

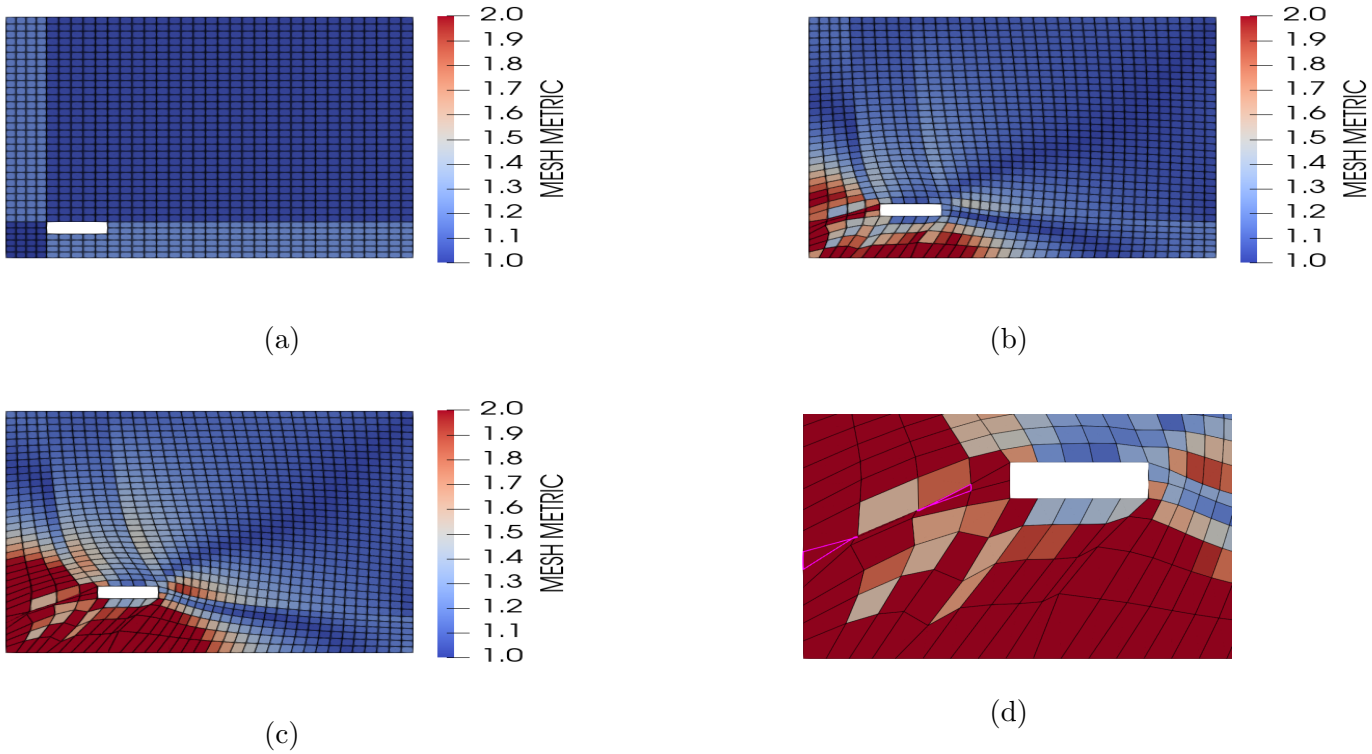


Figure 3.9: Representation of different magnitudes of translation (a) Shows the initial mesh (b-c) shows the mesh after translation (d) shows a close snapshot of the mesh translated by 25 units

The mesh deformation results produced overlapping elements after 25 units of translation.

Chapter 4

Mesh Deformation in a Tank

This project aimed to study the use of the elastic mesh deformation method by considering the case of a fluid domain with a flexible solid beam. For this purpose, the beam is caused to move via a large prescribed motion as detailed in Section 4.1. This chapter contains the formulas used to prescribe this movement as well as how the distortion of elements was controlled in the interest of mesh quality, by using different optimizing techniques to the finite element method. The resulting mesh deformations and element qualities achieved are finally presented in both 2D and 3D on structures as well as unstructured meshes. Results of mesh deformation are presented.

4.1 Displacements of Beam

The horizontal displacement of the beam tip (see Figure 4.1) is described as follows:

$$x_{tip} = 0.075 * sin\theta \tag{4.1}$$

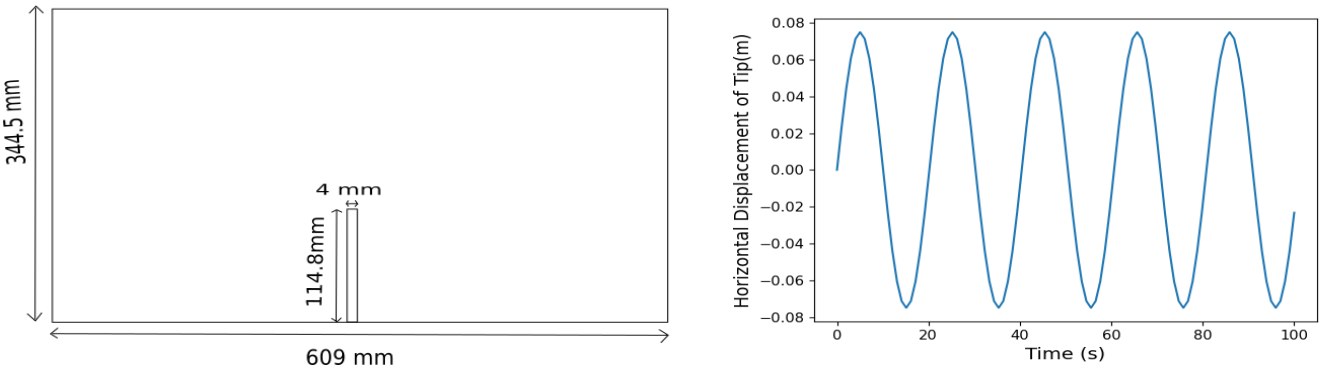
Here x_{tip} is the horizontal prescribed displacement at the tip of the beam, and θ varies at the tip to produce different modes of beam movement. The following quadratic equation was used to calculate the deformation at each node of the beam

$$x = cy^2 \tag{4.2}$$

where

$$c = \frac{x_{tip}}{h} \tag{4.3}$$

and h is the height of the beam.



(a) (b)

Figure 4.1: (a) Dimensions of tank and beam, (b) Horizontal displacements of tip of beam

To implement the scheme a method for prescribing displacements was needed.

4.2 Mesh Quality

In this project, two methods were employed to maintain mesh quality following deformation: the element size method and the distance criterion method. These approaches were discussed and validated in chapters 2 and 3. Both methods essentially seek to stiffen up elements as a function of their proximity to the moving boundary [15]. The mesh quality is utilized to showcase the element quality in a deformed mesh, highlighting the elements with poor quality in red.

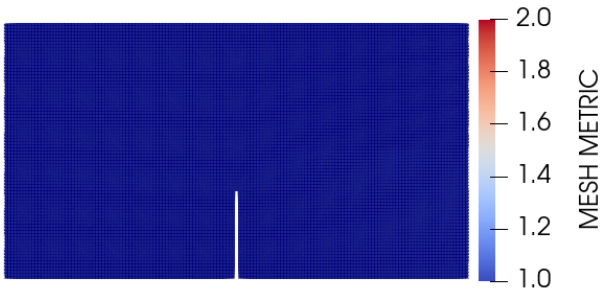
4.2.1 Structured Meshes

Recall the distance method, where the magnitude of Young's modulus depends on the distance between the element and boundary, the relationship between Young's modulus and the distance criterion can be defined as [15]:

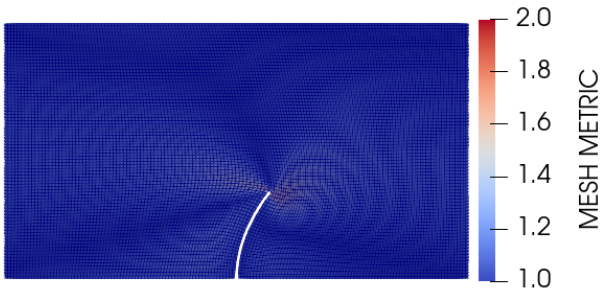
$$E \propto \frac{1}{d}$$

where E is Young's modulus and d is the distance between the element and the moving boundary. In this project, this method is used on the structured mesh in 2D and 3D below:

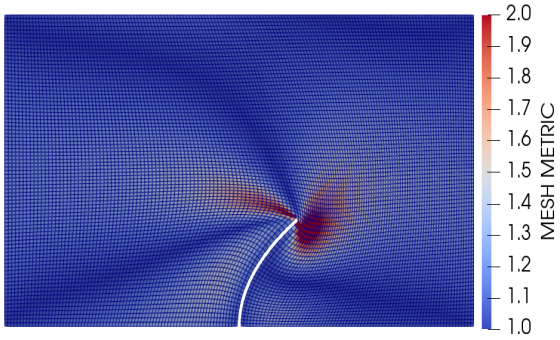
A 2D mesh composed of quadrilateral elements, has been used to model the deformation of elements around the beam using the displacements shown in Figure (4.1). Figure (4.2) shows the mesh after the prescribed displacement was applied:



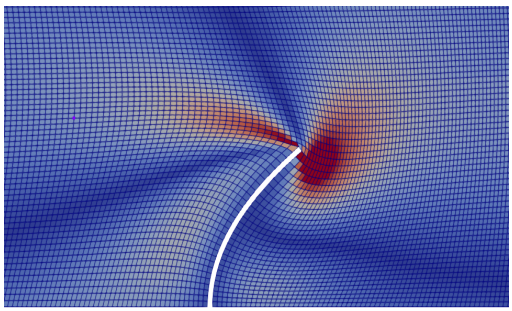
a



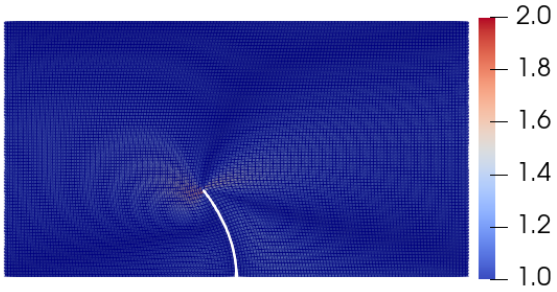
b



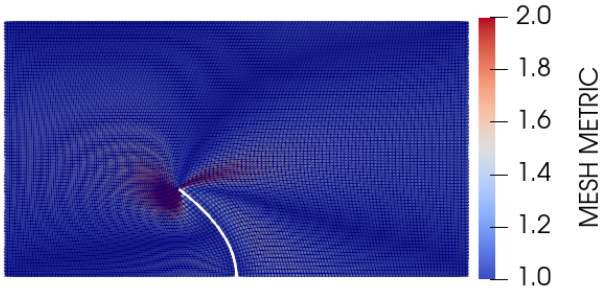
c



d



e



f

Figure 4.2: 2D Structured Mesh under deformation using the distance criterion method

where the plot in Figure 4.2(a) is a close-up of the mesh under maximum deformation of the inner boundary. Aspect ratio for a quadrilateral element:

$$\alpha = \frac{L_{max}}{L_{min}}$$

where α is the aspect ratio of an element, $L_{max}\, and\, L_{min}$ are the maximum and minimum lengths of the side of a quadrilateral respectively.α was found to be equal 1.207 at maximum deformation.

For the mesh above, α was found to be equal to 1.2. No elements were overlapping in this mesh deformation case, which is expected after applying modifications to the FEM scheme.

Next, a 3D mesh composed of hexahedral elements was used to model the deformation of elements. Figure (4.3) shows the mesh after maximum deformation.

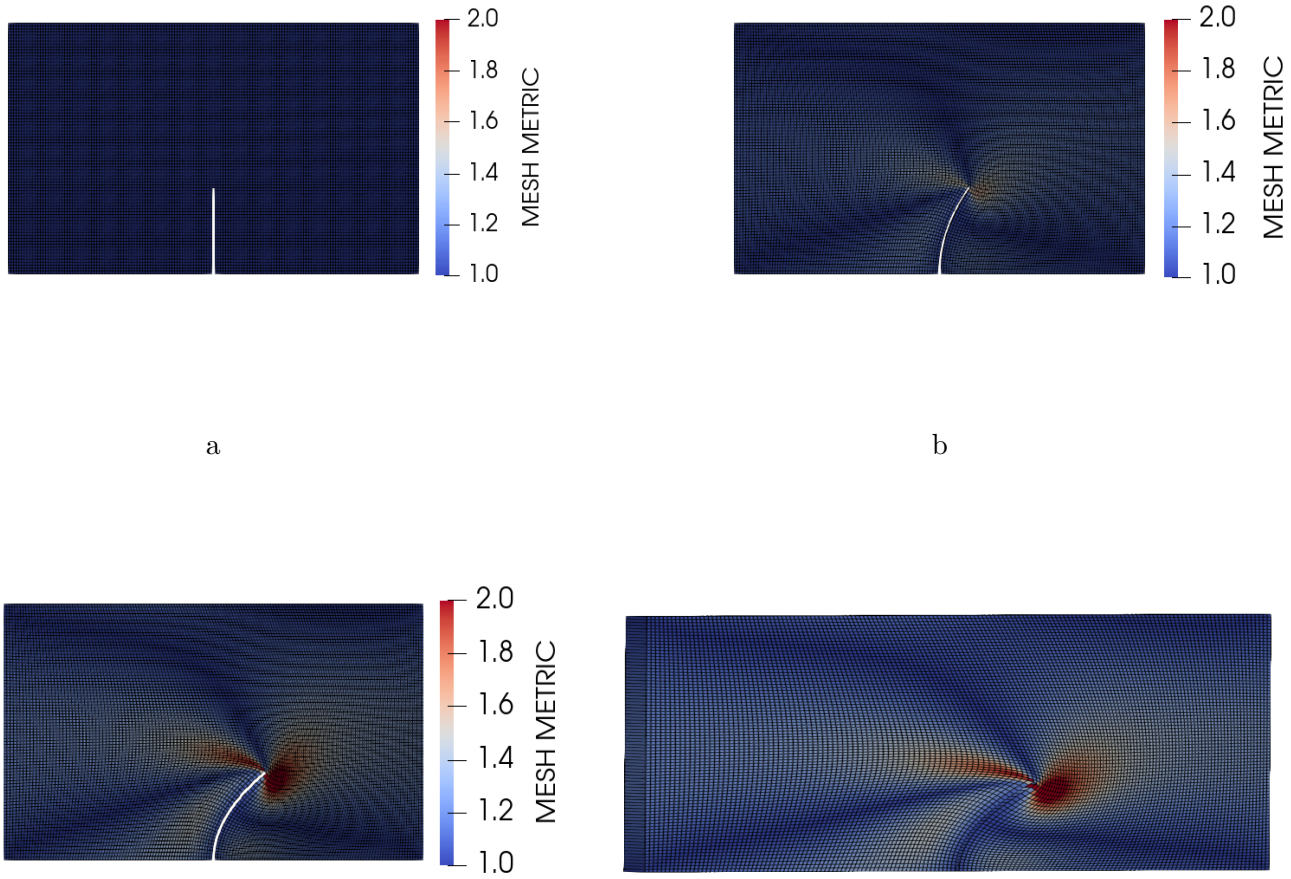


Figure 4.3: 3D Structured Mesh Under deformation using distance criterion method

For the mesh above, α was found to be equal to 1.21. The 3D mesh exhibited a mesh metric slightly higher than that of the 2D structured mesh, yielding an approximate similarity in the results. That is due to the difference in accuracy between the two dimensionalities.

4.2.2 Unstructured Meshes

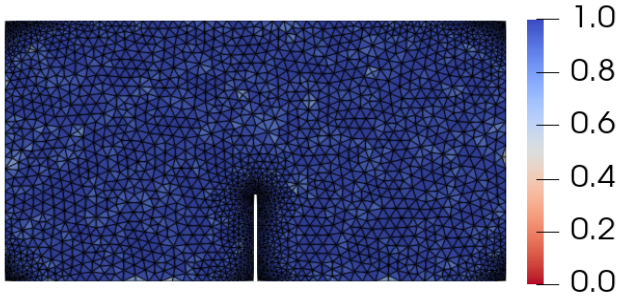
Recall the Jacobian-based stiffening. Using the Jacobian-based approach results in making smaller elements more rigid compared to the larger ones. This is achieved by altering the determinant of the Jacobian $|\boldsymbol{J}|$. In [24] the following formula was used to calculate the determinant of the Jacobian in the stiffness matrix:

$$\boldsymbol{K} = \int_{-1}^{1}\int_{-1}^{1} (\nabla \boldsymbol{L})^T \boldsymbol{G} \nabla \boldsymbol{L} |\boldsymbol{J}| \left(\frac{J^0}{|\boldsymbol{J}|}\right)^{\chi} d\xi d\eta \tag{4.4}$$

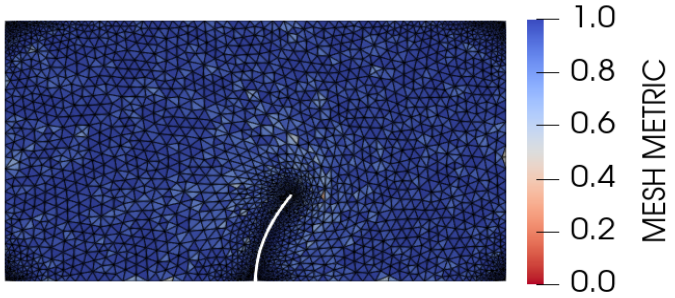
where J^0 "is an arbitrary scaling parameter which is inserted into the formulation to make the alteration dimensionally consistent" [24]. In this project, it was chosen to be equal to 1.0 since in this case, it's a constant.

To determine the optimum value for χ in equation (4.4); since it varies depending on the test case, the aspect ratio was used to measure Mesh Quality after deformation, for the following types of elements:

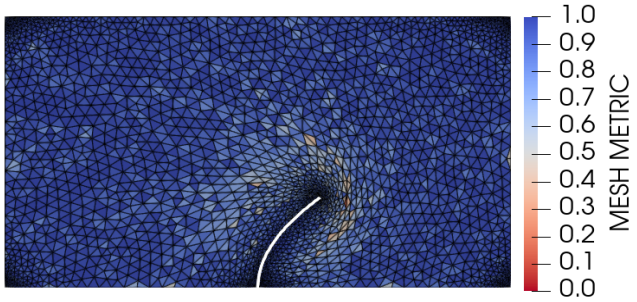
A mesh composed of triangle elements was used to model the 2D mesh deformation around the beam. Figure (4.4) shows the mesh after maximum deformation.



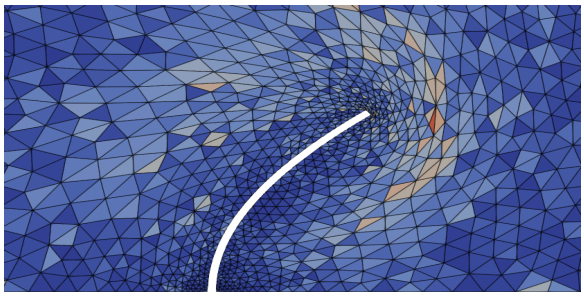
a



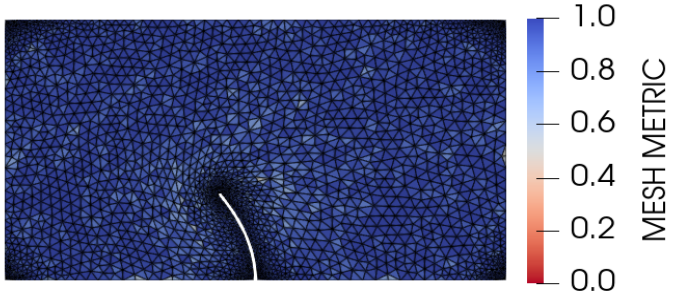
b



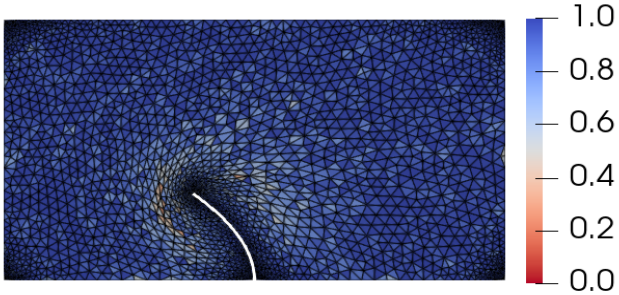
c



b



b



b

Figure 4.4: 2D Unstructured Mesh Under deformation using Element size method

For the mesh above composed of triangle elements, the following formula of mesh metric was used to determine the optimal method of Jacobian-based stiffening. The mesh quality of the whole mesh was calculated using root mean square

$$RMS = \sqrt{\frac{\sum_{n=0}^{n}(f_{ss})^2}{n}} \tag{4.5}$$

where RMS is the root mean square of f_{ss} the mesh metric described in Chapter 3, calculated after the prescribed deformation was applied. As shown in Fig. 4.5, the optimal value is $\chi = 1.0$.

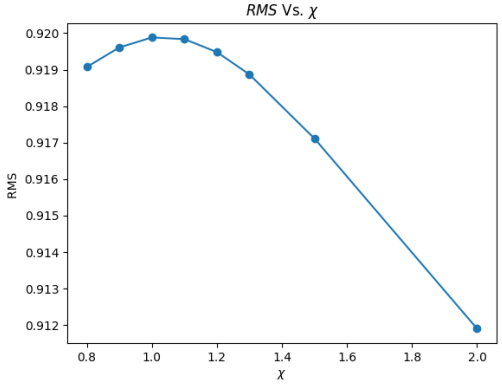


Figure 4.5: RMS Vs. χ for triangles

Notice that the optimal value of $\chi = 1.0$ was found to be equal to the value used in chapter 3 by [17].

In the end, a mesh composed of tetrahedral elements was used to model the deformation of elements around the beam using the displacements shown in Figure (4.1) as a prescribed displacement to the beam. Figure (4.6) shows the result of the 3D linear tetrahedral element.

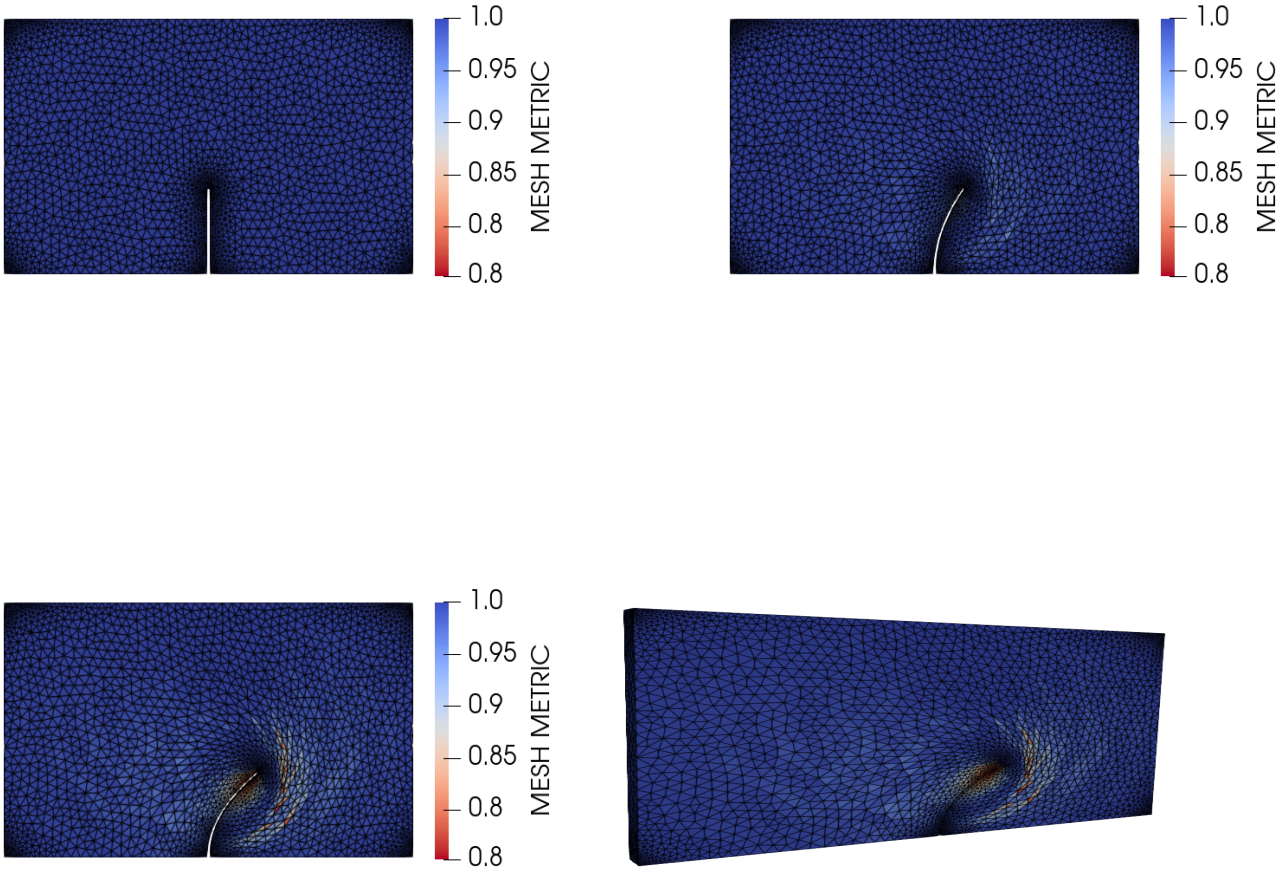


Figure 4.6: 3D Structured Mesh Under deformation using distance criterion method

For the mesh above composed of 36 801 elements, the following formula of aspect ratio was used to determine the optimal value of χ for this case. As shown in Figure 4.7, the optimal aspect ratio is achieved for $\chi = 1.1$.

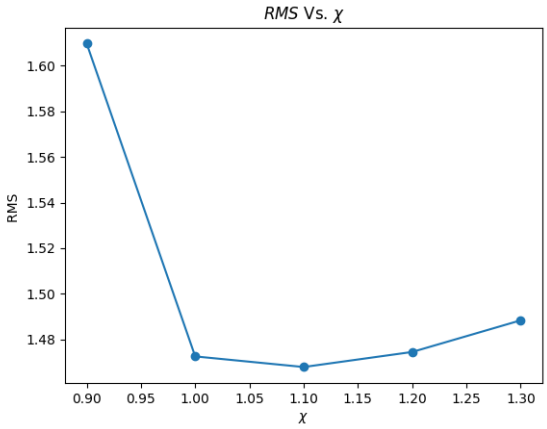


Figure 4.7: RMS Vs. χ for tetrahedrals

Unlike the 2D case, the 3D unstructured mesh showed optimal mesh quality at $\chi = 1.1$. Similar to the case of structured mesh the 3D unstructured mesh behaves differently under the same test case, due to the accuracy of 3D elements being higher.

Chapter 5

Conclusions and Recommendations

Conclusion

This project aimed to study the use of a linear elastic mesh deformation method for internal flow problems. As such the geometry selected is that of a tank with an internal flexible beam. This required implementing a finite element Galerkin method in a computer program to model the deformation of the mesh as a result of the motion of the beam.

The first part of this project involved implementing the mathematical scheme for problems in 2D and 3D domains, using both structured and unstructured elements. This was followed by testing the accuracy of the program by running small-scale tests and comparing them to the analytical solution. Verification of the scheme was performed by conducting the same test as described in a journal and comparing the obtained results.

The second part consisted of setting up the mesh motion problem by prescribing displacements to the boundaries of the beam in the tank, in 2D and 3D for both structured and unstructured meshes.

Two methods to control mesh quality were used, one was the distance criterion method for structured meshes and the other was Jacobian-based stiffening. For unstructured meshes, for both methods element aspect ratio was used as a metric to validate the element quality achieved. For 2D unstructured an overall element quality equal to 0.9199 was achieved with $\chi = 1$. For 3D Unstructured an optimal element aspect ratio (element quality) of 1.468 was obtained using the Jacobian-based stiffening method with $\chi = 1.1$. The mesh quality for the 2D structured mesh was 1.207 and for 3D structured was 1.1, for both 2D and 3D structured meshes the distance criterion method was used. Overall no mesh contained elements overlapping for different deformations.

Recommendations

The tests proved that the optimal mesh control method was different for different types of mesh elements and also for different numbers of dimensions for the problem, in the future it's suggested to:

- Further study the effect of changing the type of elements on the mesh quality.
- Investigate the use of additional mesh quality metrics for structured meshes e.g. non-orthogonality.
- Implement non-linear elasticity equations and high-order elements.

Appendices

Example of Assembly

A demonstration of the assembly method is shown below for a mesh of two triangular elements.

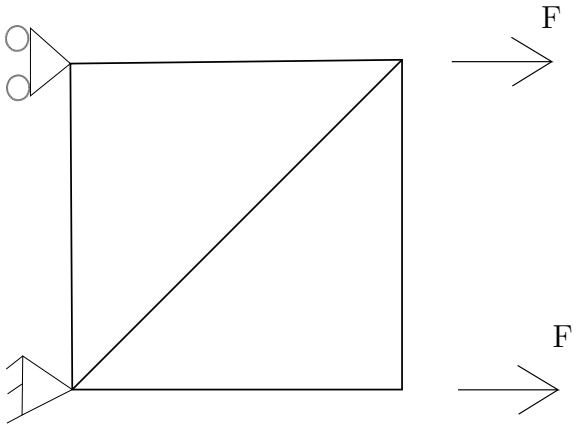


Figure 1: Multi-Triangles test case

$$ECM = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 2 & 3 \end{bmatrix}, CONM = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$$

where ECM stands for element connectivity matrix and CONM stands for coordinates of noes matrix.

$$\boldsymbol{K}^i = \begin{bmatrix} K^i_{(0,0)} & K^i_{(0,1)} & K^i_{(0,2)} & K^i_{(0,3)} & K^i_{(0,4)} & K^i_{(0,5)} \\ K^i_{(1,0)} & K^i_{(1,1)} & K^i_{(1,2)} & K^i_{(1,3)} & K^i_{(1,4)} & K^i_{(1,5)} \\ K^i_{(2,0)} & K^i_{(2,1)} & K^i_{(2,2)} & K^i_{(2,3)} & K^i_{(2,4)} & K^i_{(2,5)} \\ K^i_{(3,0)} & K^i_{(3,1)} & K^i_{(3,2)} & K^i_{(3,3)} & K^i_{(3,4)} & K^i_{(3,5)} \\ K^i_{(4,0)} & K^i_{(4,1)} & K^i_{(4,2)} & K^i_{(4,3)} & K^i_{(4,4)} & K^i_{(4,5)} \\ K^i_{(5,0)} & K^i_{(5,1)} & K^i_{(5,2)} & K^i_{(5,3)} & K^i_{(5,4)} & K^i_{(,5)} \end{bmatrix}$$

where i refers to the element index, in this case i=0,1, the Global stiffness matrix is given by

$$\boldsymbol{K} = \begin{bmatrix} K^2_{(0,0)} + K^1_{(0,0)} & K^2_{(0,1)} + K^1_{(0,1)} & K^2_{(0,2)} & K^2_{(0,3)} & K^2_{(0,4)} + K^1_{(0,2)} & K^2_{(0,5)} + K^1_{(0,3)} & K^1_{(0,4)} & K^1_{(0,5)} \\ K^2_{(1,0)} + K^1_{(1,0)} & K^2_{(1,1)} + K^1_{(1,1)} & K^2_{(1,2)} & K^2_{(1,3)} & K^2_{(1,4)} + K^1_{(1,2)} & K^2_{(1,5)} + K^1_{(1,3)} & K^1_{(1,4)} & K^1_{(1,5)} \\ K^2_{(2,0)} & K^2_{(2,1)} & K^2_{(2,2)} & K^2_{(2,3)} & K^2_{(2,4)} & K^2_{(2,5)} & 0 & 0 \\ K^2_{(3,0)} & K^2_{(3,1)} & K^2_{(3,2)} & K^2_{(3,3)} & K^2_{(3,4)} & K^2_{(3,5)} & 0 & 0 \\ K^2_{(4,0)} + K^1_{(4,0)} & K^2_{(4,1)} + K^1_{(4,1)} & K^2_{(4,2)} & K^2_{(4,3)} & K^2_{(4,4)} + K^1_{(4,2)} & K^2_{(4,5)} + K^1_{(4,3)} & K^1_{(4,4)} & K^1_{(4,5)} \\ K^2_{(5,0)} + K^1_{(5,0)} & K^2_{(5,1)} + K^1_{(5,1)} & K^2_{(5,2)} & K^2_{(5,3)} & K^2_{(5,4)} + K^1_{(5,2)} & K^2_{(5,5)} + K^1_{(5,3)} & K^5_{(0,4)} & K^5_{(0,5)} \\ K^1_{(4,0)} & K^1_{(4,1)} & 0 & 0 & K^1_{(4,2)} & K^1_{(4,3)} & K^1_{(4,4)} & K^1_{(4,5)} \\ K^1_{(5,0)} & K^1_{(5,1)} & 0 & 0 & K^1_{(5,2)} & K^1_{(5,3)} & K^1_{(5,4)} & K^1_{(5,5)} \end{bmatrix}$$

Shape Functions of a Hexahedron

$$L_0 = \frac{1}{8}(1-\xi)(1-\eta)(1-\zeta)$$

$$L_1 = \frac{1}{8}(1+\xi)(1-\eta)(1-\zeta)$$

$$L_2 = \frac{1}{8}(1-\xi)(1+\eta)(1-\zeta)$$

$$L_3 = \frac{1}{8}(1+\xi)(1+\eta)(1-\zeta)$$

$$L_4 = \frac{1}{8}(1-\xi)(1-\eta)(1+\zeta)$$

$$L_5 = \frac{1}{8}(1+\xi)(1-\eta)(1+\zeta)$$

$$L_6 = \frac{1}{8}(1-\xi)(1+\eta)(1+\zeta)$$

$$L_7 = \frac{1}{8}(1+\xi)(1+\eta)(1+\zeta)$$

Derivatives of shape functions of a hexahedron

Shape Functions of a Tetrahedral

$$L_0 = 1 - \xi - \eta - \zeta$$

$$L_1 = \xi$$

$$L_2 = \eta$$

$$L_3 = \zeta$$

Derivative to ξ	Derivative to η	Derivative to ζ
$L_{0'\xi} = \frac{-1}{8}(1-\eta)(1-\zeta)$	$L_{0'\eta} = \frac{-1}{8}(1-\xi)(1-\zeta)$	$L_{0'\zeta} = \frac{-1}{8}(1-\xi)(1-\eta)$
$L_{1'\xi} = \frac{1}{8}(1-\eta)(1-\zeta)$	$L_{1'\eta} = \frac{-1}{8}(1+\xi)(1-\zeta)$	$L_{1'\zeta} = \frac{-1}{8}(1+\xi)(1-\eta)$
$L_{2'\xi} = \frac{-1}{8}(1+\eta)(1-\zeta)$	$L_{2'\eta} = \frac{1}{8}(1-\xi)(1-\zeta)$	$L_{2'\zeta} = \frac{-1}{8}(1-\xi)(1+\eta)$
$L_{3'\xi} = \frac{1}{8}(1+\eta)(1-\zeta)$	$L_{3'\eta} = \frac{1}{8}(1+\xi)(1-\zeta)$	$L_{3'\zeta} = \frac{-1}{8}(1+\xi)(1+\eta)$
$L_{4'\xi} = \frac{-1}{8}(1-\eta)(1+\zeta)$	$L_{4'\eta} = \frac{-1}{8}(1-\xi)(1+\zeta)$	$L_{4'\zeta} = \frac{1}{8}(1-\xi)(1-\eta)$
$L_{5'\xi} = \frac{1}{8}(1-\eta)(1+\zeta)$	$L_{5'\eta} = \frac{-1}{8}(1+\xi)(1+\zeta)$	$L_{5'\zeta} = \frac{1}{8}(1+\xi)(1-\eta)$
$L_{6'\xi} = \frac{-1}{8}(1+\eta)(1+\zeta)$	$L_{6'\eta} = \frac{1}{8}(1-\xi)(1+\zeta)$	$L_{6'\zeta} = \frac{1}{8}(1-\xi)(1+\eta)$
$L_{7'\xi} = \frac{1}{8}(1+\eta)(1+\zeta)$	$L_{7'\eta} = \frac{1}{8}(1+\xi)(1+\zeta)$	$L_{7'\zeta} = \frac{1}{8}(1+\xi)(1+\eta)$

Table 1: Derivatives of Shape Functions

Derivatives of Shape Functions of a Tetrahedral

Derivative to ξ	Derivative to η	Derivative to ζ
$L_{0'\xi} = -1$	$L_{0'\eta} = -1$	$L_{0'\zeta} = -1$
$L_{1'\xi} = 1$	$L_{1'\eta} = 0$	$L_{1'\zeta} = 0$
$L_{2'\xi} = 0$	$L_{2'\eta} = 1$	$L_{2'\zeta} = 0$
$L_{3'\xi} = 0$	$L_{3'\eta} = 0$	$L_{3'\zeta} = 1$

Table 2: Derivatives of Shape Functions of a Tetrahedral

Bibliography

[1] Richard P Dwight. Robust mesh deformation using the linear elasticity equations. In *Computational fluid dynamics 2006*. Springer, 2009.

[2] MM Selim, RP Koomullil, et al. Mesh deformation approaches–a survey. *Journal of Physical Mathematics*, (2), 2016.

[3] Nicolas Barral, Edward Luke, and Frédéric Alauzet. Two mesh deformation methods coupled with a changing-connectivity moving mesh method for cfd applications. *Procedia Engineering*, 82, 2014.

[4] Shuvam Sen, Guillaume De Nayer, and Michael Breuer. A fast and robust hybrid method for block-structured mesh deformation with emphasis on fsi-les applications. *International Journal for Numerical Methods in Engineering*, (3).

[5] Gang WANG, Xin CHEN, and Zhikan LIU. Mesh deformation on 3d complex configurations using multistep radial basis functions interpolation. *Chinese Journal of Aeronautics*, (4), 2018.

[6] Matteo Lombardi, Nicola Parolini, Alfio Quarteroni, and Gianluigi Rozza. Numerical simulation of sailing boats: dynamics, fsi, and shape optimization. Springer, 2012.

[7] Tayfun E Tezduyar, M Behr, S Mittal, and AA Johnson. Computation of unsteady incompressible flows with the stabilized finite element methods: Space-time formulations, iterative strategies and massively parallel implementations. *ASME Pressure Vessels Piping DIV PUBL PVP., ASME, NEW YORK, NY(USA), 1992,*, 1992.

[8] JT Batina. Unsteady euler airfoil solutions using unstructured dynamic meshes. *AIAA Journal*, (8), 1990.

[9] RL Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of geophysical research*, (8), 1971.

[10] Per-Olof Persson and Gilbert Strang. A simple mesh generator in matlab. *Siam Review - SIAM REV*, 46, 2004.

[11] Luming Wang and Per-Olof Persson. A discontinuous galerkin method for the navier-stokes equations on deforming domains using unstructured moving space-time meshes. 2013.

[12] Yosheph Yang, Serkan Özgen, and Haedong Kim. Improvement in the spring analogy mesh deformation method through the cell-center concept. *Aerospace Science and Technology*, 2021.

[13] M Cordero-Gracia, M Gómez, J Ponsin, and E Valero. An interpolation tool for aerodynamic mesh deformation problems based on octree decomposition. *Aerospace Science and Technology*, (1), 2012.

[14] Yang, Y. and Özgen, S. Comparison of various spring analogy related mesh deformation techniques in two-dimensional airfoil design optimization, 2017.

[15] Bin Zhang, Zhiwei Feng, Tao Yang, Boting Xu, and Xiaojian Sun. Integrated improvement of the elasticity-based mesh deformation method based on robust parameters and mesh quality. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, (8), 2019.

[16] Fish Jacob and Belytschko Ted. *A first course in finite elements*. Number 1. Wiley, 2007.

[17] Alfred Edward Jules Bogaers, Schalk Kok, and AG Malan. Highly efficient optimization mesh movement method based on proper orthogonal decomposition. *International Journal for Numerical Methods in Engineering*, (8), 2011.

[18] David S Burnett. *Finite element analysis: from concepts to applications*. Prentice Hall, 1987.

[19] Klaus-Jürgen Bathe. *Finite element procedures*. Klaus-Jurgen Bathe, 2006.

[20] David C O'Neal. Optimization of finite element codes. *Journal of scientific computing*, 1990.

[21] Xiaoshan Lin, YX Zhang, and Prabin Pathak. *Nonlinear finite element analysis of composite and reinforced concrete beams*. Woodhead Publishing, 2019.

[22] Timothy A Philpot. Mechanics of materials: an integrated learning system. 2008.

[23] Steven E Benzley, Ernest Perry, Karl Merkley, Brett Clark, and Greg Sjaardama. A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elasto-plastic analysis. 1995.

[24] Keith Stein and Tayfun Tezduyar. Advanced mesh update techniques for problems involving large displacements. In *Proceedings of the Fifth World Congress on Computational Mechanics, On-line publication: http://wccm. tuwien. ac. at/, Paper-ID*, 2002.

[25] Travis B Fillmore, Varun Gupta, and Carlos Armando Duarte. Preconditioned conjugate gradient solvers for the generalized finite element method. *Meshfree Methods for Partial Differential Equations IX 9*, 2019.

[26] King-Hay Yang. *Basic finite element method as applied to injury biomechanics*. Academic Press, 2017.