SOLUTION TO THREE DIMENSIONAL INCOMPRESSIBLE NAVIER-STOKES EQUATIONS USING FINITE ELEMENT METHOD

by

SHRINIVAS G APTE

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2012

To my parents, Vidya and Ganesh, who always sacrificed their happiness to see me succeed in pursuit of my happ‘y’ness.

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Brian H Dennis for his inputs and guidance during my thesis work. His experience in the field of Computational Fluid Dynamics proved vital to overcome many situations when I thought my work has reached impasse. I am also thankful to him for teaching two valuable courses namely 'Computational Aerodynamics' and 'Finite Element Method in Heat Transfer and Fluid Flow' which gave me understanding of the subject and motivation to excel.

I would also like to thank all of those who directly, indirectly, knowingly or unknowingly contributed to this work.

December 7, 2012

ABSTRACT

SOLUTION TO THREE DIMENSIONAL INCOMPRESSIBLE NAVIER-STOKES EQUATIONS USING FINITE ELEMENT METHOD

SHRINIVAS G APTE, M.S.

The University of Texas at Arlington, 2012

Supervising Professor: Brian H Dennis

A primitive variable mixed order formulation of finite element method for solving three dimensional incompressible Navier-Stokes equations is presented. The method of weighted residuals is used for obtaining the approximate solutions of linear and nonlinear partial differential equations. The Physical domain is discretized by using unstructured tetrahedral elements. Unequal order interpolation functions are used for pressure & velocity variables while the temporal discretization is carried out by using an implicit time marching scheme based on finite differencing.

One of the major difficulties arising during the finite element solution of an incompressible Navier-Stokes equations is the efficient factorization/preconditioning of the resulting indefinite stiffness matrix. In this work, the formation of an indefinite matrix is avoided by using a pseudo compressibility technique in which an artificial term is introduced into the mass matrix. The artificial term is time dependent and disposed at a later stage once the steady state is reached. Using this approach, the resulting system of equations can then be solved iteratively with standard preconditioners. The non-linear convective term in the Navier-Stokes equations is linearized in

time. To diffuse the numerical oscillations which may occur in convection dominated flows, second-orderTaylor-Galerkinstabilization technique is used.

The entire solution procedure is encoded in C++ using object oriented programming. One of the special features of this FEM code is that it uses the exact integrals of the shape functions in order to improve the accuracy of the solution, as supposed to any numerical integration schemes. The solution procedure is validated using the benchmark computations for 3D steady incompressible flows.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

Navier-Stokes equations constitute one of the most fundamental fluid flow equations as most of the real life situations involving fluid motion are governed either by them or by one of their special cases. The Navier-Stokes equations are named in honor of two people—M.Navier from France and G.Stokes from England—who obtained the equations independently at the beginning of nineteenth century[3]. Since then Navier-Stokes equations are considered as the basis from which all the fluid mechanics studies begin[4].

Even though having an exact solution to the most general form of Navier-Stokes equations would be extremely useful for understanding the complex natures of fluid flow problem in various practical cases, so far it has eluded scientists and engineers. Several analytical solutions have been developed for the special cases of Navier-Stokes equations by weakening or removing the nonlinearity in the problem. These solutions are applicable for steady or unsteady cases of Coutte and Poiseuille flow[5]. There have also been few attempts to find a class of a solution for Navier-Stokes equation using potential function and transformed coordinates which looking at a larger picture would make it possible to obtain the global solution[6].

More success has come to those who have tried to obtain an approximate solution for Navier-Stokes equations using different numerical methods. Two such popular numerical methods, finite difference(FDM) and finite volume(FVM), are widely accepted by scientific community and also by commercial solver developers to solve

the Navier-Stokes equations approximately. The third approach—Finite Element Method(FEM)—although very popular in structural dynamics, hasn't been used for solving complex flow problems extensively. In this work, an effort has been made to solve the unsteady, incompressible, three dimensional Navier-Stokes equations using finite element method.

1.2 Finite Difference, Finite Volume and Finite Element Methods

The most burning issue before the start of this work was the choice of finite element method as solver for fluid dynamic problem. As mentioned earlier, finite element method has made an impact on solid mechanics field for long and is being considered as generally accepted method to solve continuum problems, same does not hold good when it comes to the fluid dynamics problems. The issue has been clearly addressed by Zienkiewicz in [7]. Finite element method, in spite of being computationally least expensive of the three, has not been considered as an alternative due to the success achieved in solving the flow problems by finite difference or finite volume methods and also due to the large investments in highly developed software which are making use of these methods. The comparison between these three methods can be seen in [8]. FEM and FVM both solve the integral form of the governing equations and thus have advantage over FDM in a sense that they satisfy the conservation laws inherently[9]. One of the main advantages FEM carries over other solvers is its ability to handle complex geometries with unstructured grids producing high accuracy solutions. The other significant advantage would be its inherent ability to incorporate natural boundary conditions[10]. Due to these advantages, it becomes an important task to try and have a method which will give improved solutions to Navier-Stokes equations.

1.3 Finite Element Method for Navier-Stokes Equations: Different Approaches

One can find three different approaches being used in literature to obtain the solution of Navier-Stokes equations using finite element method.

1. Stream Function Formulation
2. Stream Function and Vorticity Formulation
3. Primitive Variable (Velocity and Pressure) Formulation

The detailed study on these three approaches can be found in[11]. Third approach holds lot of advantages over the first two and hence been preferred by most of the authors. Comparison between these methods can be seen in[12, p. 390]. Another variation of primitive variable formulation —Penalty Method— is also very popular in which pressure is represented in terms of an additional parameter λ and continuity equation is dropped. Due to the simplicity and straight forward nature of primitive variable formulation, it is preferred in this work.

The primitive variable formulation is then encoded in C++ using object oriented programming. This program reads the input mesh file created by Ansys APDL in '.inp' format. After solving the governing equations iteratively using standard linear solvers, solution is written which can be read using Tecplot. Output results are compared with benchmark case of laminar flow around an obstacle and the formulation is validated.

1.4 Overview

This report has total five chapters. Chapter 2 explains the weak formulation of Navier-Stokes equations using method of weighted residuals. Chapter 3 talks about interpolation functions, coefficient matrices and finite element formulation. In chapter 4, some aspects of programming are discussed. Finally, in chapter 5, programming

results are validated and convergence study is carried out followed by conclusion and future work recommendations.

CHAPTER 2

WEAK FORMULATION OF INCOMPRESSIBLE NAVIER-STOKES EQUATION

2.1 Introduction

As mentioned in previous chapter, primitive variable formulation is used in this study to obtain a solution of Navier-Stokes equations using FEM. The formulation starts from the basic conservative laws i. e. conservation of mass (continuity), conservation of momentum (Navier-Stokes) and conservation of energy. For the purpose of this work, conservation of energy is excluded from the set of equations as heat transfer is not considered as a part of formulation. After choosing pressure and velocity as primitive variables, method of weighted residuals is applied in which the differential equations are converted into integral forms[12]. A Galerkin criteria for weight function is applied at this point and by using the integration by parts, weak form of governing equation is obtained.

2.2 Governing Equations and Method of Weighted Residuals (MWR)

This section provides a detailed derivation of 'weak form' of Navier-Stokes equations using MWR. Basic concept and application of this method can be found in literature[13, 14, 12]. It is important to note that MWR by itself has nothing to do with FEM and is just an alternate way of formulating finite element equations. Even though same solutions can be obtained by using other methods such as variational principle or minimum potential energy principle, MWR carries an advantage due to its easy implementation directly from governing differential equations and the availability of abundant weight functions.

The procedure starts by considering the suitable form of governing equations, in this case the conservative form of incompressible Navier-Stokes equations. All the notations used here are as per [3]. The incompressible continuity equation is given by equation 2.1,

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2.1}$$

The incompressible momentum equations in x, y and z directions are given by,

$$\rho \left[\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} \right] = \frac{\partial(\sigma_x - P)}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + \rho f_x \tag{2.2}$$

$$\rho \left[\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} \right] = \frac{\partial(\sigma_y - P)}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{zy}}{\partial z} + \rho f_y \tag{2.3}$$

$$\rho \left[\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} \right] = \frac{\partial(\sigma_z - P)}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \rho f_z \tag{2.4}$$

where, u, v ,w are velocity components in x, y and z directions respectively. t is time, τ_{ij} is the shear stress in j direction exerted on a plane perpendicular to i axis. Density of the fluid is denoted by ρ and μ is the dynamic viscosity. Pressure perpendicular to the face of fluid volume is denoted by p while f_x, f_y & f_z are the body forces per unit mass acting on the fluid element in x, y and z direction respectively. All the above equations are obtained after using an incompressibility condition ($\nabla.U = 0$) in the continuity and momentum equations.

For newtonian fluids, Stokes obtained the constitutive equations. For an incompressible flow ($\nabla.U = 0$), these equations are as shown below.

$$\sigma_x = 2\mu\frac{\partial u}{\partial x} \tag{2.5}$$

$$\sigma_y = 2\mu\frac{\partial v}{\partial y} \tag{2.6}$$

$$\sigma_z = 2\mu\frac{\partial w}{\partial z} \tag{2.7}$$

$$\tau_{xy} = \tau_{yx} = \mu \left[\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right] \tag{2.8}$$

$$\tau_{xz} = \tau_{zx} = \mu \left[\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right] \tag{2.9}$$

$$\tau_{yz} = \tau_{zy} = \mu \left[\frac{\partial w}{\partial y} + \frac{\partial y}{\partial z} \right] \tag{2.10}$$

After substituting constitutive equations in (2.2) to (2.4) we get the complete Navier-Stokes equations for incompressible flow.

$$-\rho \left[\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} \right] + \frac{\partial}{\partial x}\left[2\mu\frac{\partial u}{\partial x} - P \right] + \mu\frac{\partial}{\partial y}\left[\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right] + \mu\frac{\partial}{\partial z}\left[\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right] + \rho f_x = 0 \tag{2.11}$$

$$-\rho \left[\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} \right] + \frac{\partial}{\partial y}\left[2\mu\frac{\partial v}{\partial y} - P \right] + \mu\frac{\partial}{\partial x}\left[\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right] + \mu\frac{\partial}{\partial z}\left[\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right] + \rho f_y = 0 \tag{2.12}$$

$$-\rho \left[\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} \right] + \frac{\partial}{\partial z}\left[2\mu\frac{\partial w}{\partial z} - P \right] + \mu\frac{\partial}{\partial x}\left[\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right] + \mu\frac{\partial}{\partial y}\left[\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right] + \rho f_z = 0 \tag{2.13}$$

Suppose $\tilde{u}$, $\tilde{v}$, $\tilde{w}$ and $\tilde{P}$ are the approximate solutions of the governing equations leaving out a residual R as a result. For the sake of simplicity, these approximate solutions are denoted by u, v, w, and P here onwards. These approximate solutions are given by,

$$u = \sum_{i=1}^{n} \psi_i(x, y, z) u_i(t) \tag{2.14}$$

$$v = \sum_{i=1}^{n} \psi_i(x, y, z) v_i(t) \tag{2.15}$$

$$w = \sum_{i=1}^{n} \psi_i(x, y, z) w_i(t) \tag{2.16}$$

$$P = \sum_{j=1}^{m} \phi_j(x, y, z) P_j(t) \tag{2.17}$$

ψ and ϕ are the functions chosen to satisfy n and m number of unknowns respectively. These functions are known as 'Basis functions' or 'Shape functions' or 'Interpolation functions. Two different shape functions (unequal order) to obtain a correct pressure field in a solution[15]. This topic will be explained later in detail while defining the shape functions.

As per the procedure of MWR, two weight functions, Q & W are selected for continuity and momentum equations. The idea is to make the weighted average of residuals over the entire domain zero. This can be formulated as shown below.

$$\int_V Q \left[\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right] dV = 0 \tag{2.18}$$

$$\int_V W \Big[-\rho \left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} \right) + \frac{\partial}{\partial x}\left(2\mu \frac{\partial u}{\partial x} - P \right) + \mu \frac{\partial}{\partial y}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) + \mu \frac{\partial}{\partial z}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) + \rho f_x \Big] dV = 0 \tag{2.19}$$

$$\int_V W \Big[-\rho \left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} \right) + \frac{\partial}{\partial y}\left(2\mu \frac{\partial v}{\partial y} - P \right) + \mu \frac{\partial}{\partial x}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) + \mu \frac{\partial}{\partial z}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \rho f_y \Big] dV = 0 \tag{2.20}$$

$$\int_V W \Big[-\rho \left(\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} \right) + \frac{\partial}{\partial z}\left(2\mu \frac{\partial w}{\partial z} - P \right) + \mu \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) + \mu \frac{\partial}{\partial y}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \rho f_y \Big] dV = 0 \tag{2.21}$$

As mentioned earlier, MWR allows us to have a broad choice of weight functions. Generally for the purpose of finite element formulation, 'Galerkin Criterion' is

applied for the selection of weight functions. According to Galerkin Criterion, weight functions are chosen to be exact same as approximating functions or shape functions. For instance, for the purpose of continuity equation, weight function Q_j is same as shape function ϕ_j and for momentum equations, W_i is same as ψ_i.

Ths Galerkin criterion is used in equations (2.19) to (2.21) to replace the weight functions by shape functions and the expressions for these shape functions are obtained from equations (2.14) to (2.17). After rearranging, following expressions are obtained.

$$\int_V \phi \left[u\frac{\partial \psi^T}{\partial x} + v\frac{\partial \psi^T}{\partial y} + w\frac{\partial \psi^T}{\partial z} \right] dV = 0 \tag{2.22}$$

$$\begin{aligned} \int_V \left[\rho \left(\psi\psi^T \frac{\partial u}{\partial t} \right) \right] dV + \int_V \left[\rho \left(\psi\psi^T \frac{\partial \psi^T}{\partial x} u + \psi\psi^T \frac{\partial \psi^T}{\partial y} v + \psi\psi^T \frac{\partial \psi^T}{\partial z} w \right) u \right] dV \\ - \int_V \left[2\mu\psi \frac{\partial^2 \psi^T}{\partial x^2} u + \mu\psi \frac{\partial^2 \psi^T}{\partial y^2} u + \mu\psi \frac{\partial^2 \psi^T}{\partial z^2} u + \mu\psi \frac{\partial^2 \psi^T}{\partial y \partial x} v + \mu\psi \frac{\partial^2 \psi^T}{\partial z \partial x} w \right] dV \\ + \int_V \left[\psi \frac{\partial \phi^T}{\partial x} P \right] dV = \int_V \left[\psi \rho f_x \right] dV \end{aligned} \tag{2.23}$$

$$\begin{aligned} \int_V \left[\rho \left(\psi\psi^T \frac{\partial v}{\partial t} \right) \right] dV + \int_V \left[\rho \left(\psi\psi^T \frac{\partial \psi^T}{\partial x} u + \psi\psi^T \frac{\partial \psi^T}{\partial y} v + \psi\psi^T \frac{\partial \psi^T}{\partial z} w \right) v \right] dV \\ - \int_V \left[2\mu\psi \frac{\partial^2 \psi^T}{\partial y^2} v + \mu\psi \frac{\partial^2 \psi^T}{\partial x^2} v + \mu\psi \frac{\partial^2 \psi^T}{\partial z^2} v + \mu\psi \frac{\partial^2 \psi^T}{\partial x \partial y} u + \mu\psi \frac{\partial^2 \psi^T}{\partial y \partial z} w \right] dV \\ + \int_V \left[\psi \frac{\partial \phi^T}{\partial y} P \right] dV = \int_V \left[\psi \rho f_y \right] dV \end{aligned} \tag{2.24}$$

$$\begin{aligned} \int_V \left[\rho \left(\psi\psi^T \frac{\partial w}{\partial t} \right) \right] dV + \int_V \left[\rho \left(\psi\psi^T \frac{\partial \psi^T}{\partial x} u + \psi\psi^T \frac{\partial \psi^T}{\partial y} v + \psi\psi^T \frac{\partial \psi^T}{\partial z} w \right) w \right] dV \\ - \int_V \left[2\mu\psi \frac{\partial^2 \psi^T}{\partial z^2} w + \mu\psi \frac{\partial^2 \psi^T}{\partial x^2} w + \mu\psi \frac{\partial^2 \psi^T}{\partial y^2} w + \mu\psi \frac{\partial^2 \psi^T}{\partial x \partial z} u + \mu\psi \frac{\partial^2 \psi^T}{\partial y \partial z} v \right] dV \\ + \int_V \left[\psi \frac{\partial \phi^T}{\partial z} P \right] dV = \int_V \left[\psi \rho f_y \right] dV \end{aligned} \tag{2.25}$$

In order to ensure the continuity of field variable and its derivatives across the element boundaries, it is required to chose appropriate shape functions. Details about the order of continuity requirements that needs to be satisfied by the shape functions, can be found in [12, 16]. As a general rule, shape functions need to have C^n continuity of field variable to satisfy 'compatibility' condition across the element boundaries and C^{n+1} order of continuity for 'completeness' within the element, 'n+1' being the highest order of derivative appearing in the expression. It can be seen in above equations that the highest order of derivative appearing in the equation is two and hence the shape function satisfying the stringent continuity requirements (C^2) needs to be selected to ensure the continuous solution over entire domain. This difficulty can be resolved using integration by parts. Applying integration by parts reduces the order of the highest derivative appearing in the expression by one hence 'weakening' the continuity requirements for the shape function. Integration by parts in three dimensions is known as Gauss's theorem and is given by,

$$\int_V u(\nabla.v)\,dV = \int_A u(v.\hat{n})\,dA - \int_V v.\nabla u\,dV \tag{2.26}$$

Using Gauss's theorem for equations (2.17) to (2.25) and rearranging the terms, we obtain the 'Weak Form' of three dimensional Navier-Stokes equation.

$$\int_V \phi\left[\frac{\partial\psi^T}{\partial x}u + \frac{\partial\psi^T}{\partial y}v + \frac{\partial\psi^T}{\partial z}w\right]\,dV = 0 \tag{2.27}$$

$$\begin{aligned}\int_V \rho\left[\psi\psi^T\frac{\partial u}{\partial t}\right]\,dV + \int_V \rho\left[\psi(\psi^T u)\frac{\partial\psi^T}{\partial x} + \psi(\psi^T v)\frac{\partial\psi^T}{\partial y} + \psi(\psi^T w)\frac{\partial\psi^T}{\partial z}\right]u\,dV \\ + \int_V\left[2\mu\left(\frac{\partial\psi}{\partial x}\frac{\partial\psi^T}{\partial x}\right) + \mu\left(\frac{\partial\psi}{\partial y}\frac{\partial\psi^T}{\partial y}\right) + \mu\left(\frac{\partial\psi}{\partial z}\frac{\partial\psi^T}{\partial z}\right)\right]u\,dV \\ + \int_V\left[\mu\left(\frac{\partial\psi}{\partial x}\frac{\partial\psi^T}{\partial y}\right)v + \mu\left(\frac{\partial\psi}{\partial x}\frac{\partial\psi^T}{\partial z}\right)w\right]dV \\ - \int_V P\left(\frac{\partial\psi}{\partial x}\phi^T\right)\,dV = \int_V \psi\rho f_x\,dV\end{aligned} \tag{2.28}$$

$$\int_V \rho \left[\psi\psi^T \frac{\partial v}{\partial t}\right] \, dV + \int_V \rho \left[\psi(\psi^T u)\frac{\partial \psi^T}{\partial x} + \psi(\psi^T v)\frac{\partial \psi^T}{\partial y} + \psi(\psi^T w)\frac{\partial \psi^T}{\partial z}\right] v \, dV$$
$$+ \int_V \left[\mu\left(\frac{\partial \psi}{\partial x}\frac{\partial \psi^T}{\partial x}\right) + 2\mu\left(\frac{\partial \psi}{\partial y}\frac{\partial \psi^T}{\partial y}\right) + \mu\left(\frac{\partial \psi}{\partial z}\frac{\partial \psi^T}{\partial z}\right)\right] v \, dV$$
$$+ \int_V \left[\mu\left(\frac{\partial \psi}{\partial y}\frac{\partial \psi^T}{\partial x}\right) u + \mu\left(\frac{\partial \psi}{\partial y}\frac{\partial \psi^T}{\partial z}\right) w\right] dV$$
$$- \int_V P\left(\frac{\partial \psi}{\partial y}\phi^T\right) \, dV = \int_V \psi\rho f_y \, dV \qquad (2.29)$$

$$\int_V \rho \left[\psi\psi^T \frac{\partial w}{\partial t}\right] \, dV + \int_V \rho \left[\psi(\psi^T u)\frac{\partial \psi^T}{\partial x} + \psi(\psi^T v)\frac{\partial \psi^T}{\partial y} + \psi(\psi^T w)\frac{\partial \psi^T}{\partial z}\right] w \, dV$$
$$+ \int_V \left[\mu\left(\frac{\partial \psi}{\partial x}\frac{\partial \psi^T}{\partial x}\right) + \mu\left(\frac{\partial \psi}{\partial y}\frac{\partial \psi^T}{\partial y}\right) + 2\mu\left(\frac{\partial \psi}{\partial z}\frac{\partial \psi^T}{\partial z}\right)\right] w \, dV$$
$$+ \int_V \left[\mu\left(\frac{\partial \psi}{\partial z}\frac{\partial \psi^T}{\partial x}\right) u + \mu\left(\frac{\partial \psi}{\partial z}\frac{\partial \psi^T}{\partial y}\right) v\right] dV$$
$$- \int_V P\left(\frac{\partial \psi}{\partial z}\phi^T\right) \, dV = \int_V \psi\rho f_z \, dV \qquad (2.30)$$

In order to write the weak statement of Navier-Stokes equations in matrix form, some coefficient matrix formulae need to be defined.

1. Mass Matrix,
$$M = \int_V \rho\psi\psi^T \, dV \qquad (2.31)$$

2. Convective Matrix,
$$C(u, v, w) = \int_V \rho \left[\psi(\psi^T u)\frac{\partial \psi^T}{\partial x} + \psi(\psi^T v)\frac{\partial \psi^T}{\partial y} + \psi(\psi^T w)\frac{\partial \psi^T}{\partial w}\right] \, dV \qquad (2.32)$$

3. Diffusive Matrix,
$$K_{ij} = \int_V \mu\left(\frac{\partial \psi}{\partial x_i}\frac{\partial \psi^T}{\partial x_j}\right) \, dV \qquad (2.33)$$

4. Gradient Matrix,
$$Q_i = \int_V \frac{\partial \psi}{\partial x_i}\phi^T \, dV \qquad (2.34)$$

5. Force Vector,

$$F_i = \int_V \rho\psi f_i \, dV \tag{2.35}$$

After substituting above coefficient matrix formulae in equations (2.27) to (2.30) to obtain a matrix form of the weak statement.

$$\begin{bmatrix} M & 0 & 0 & 0 \\ 0 & M & 0 & 0 \\ 0 & 0 & M & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{P} \end{Bmatrix} + \begin{bmatrix} C(u,v,w) & 0 & 0 & 0 \\ 0 & C(u,v,w) & 0 & 0 \\ 0 & 0 & C(u,v,w) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ P \end{Bmatrix}$$

$$+ \begin{bmatrix} 2K_{11} + K_{22} + K_{33} & K_{12} & K_{13} & -Q_1 \\ K_{21} & K_{11} + 2K_{22} + K_{33} & K_{23} & -Q_2 \\ K_{31} & K_{32} & K_{11} + K_{22} + 2K_{33} & -Q_3 \\ Q_1^T & Q_2^T & Q_3^T & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ P \end{Bmatrix}$$

$$= \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ 0 \end{Bmatrix} \tag{2.36}$$

It becomes important to point out two aspects of equation (2.36),

1. Zero appearing in the diagonal of the mass matrix due to incompressibility constrain.
2. Possibility of unstable solution in convection dominated flows due to non linear terms appearing in the convection matrix.

These two conditions make it difficult to solve the incompressible Navier-Stokes equation using FEM. Techniques to handle these two difficulties are discussed in the next two sections.

2.3 Artificial/Pseudo Compressibility

As seen in equation (2.36), due to the incompressibility condition ($\nabla.\vec{u} = 0$) applied to the continuity equation, there exists a zero on the diagonal of mass matrix corresponding to the time derivative of pressure term. This condition makes the mass matrix singular. A method to handle this difficulty in incompressible Navier-Stokes equation was first developed by Alexander Chorin[17]. The scope of this method is not limited to finite element method and hence the method of 'Artificial Compressibility' has been successfully implemented in variety of incompressible viscous problems. The idea behind the technique is to convert an elliptical problem into hyperbolic problem by introducing an artificial term in continuity equation[18]. After changing the type of the problem, it can be solved by well known discretization techniques.

Using the technique of artificial compressibility, an alternate form of (2.1) can be written as,

$$\frac{\partial \rho}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2.37}$$

where, ρ is called artificial density and β is artificial compressibility such that $P = \rho/\beta$. Equation (2.37) is the artificial continuity equation. When the calculation progresses and becomes independent of time, it also becomes independent of β. Thus β can be disposed when solution converges to steady state[17]. Thus the weak form of continuity equation can be derived as shown below,

Applying the method of weighted residuals to artificial continuity equation,

$$\int_V Q \left(\beta \frac{\partial P}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) dV = 0 \tag{2.38}$$

Therefore,

$$\int_V \phi \left(\beta \phi^T \frac{\partial P}{\partial t} + \frac{\partial \psi^T}{\partial x} u + \frac{\partial \psi^T}{\partial y} v + \frac{\partial \psi^T}{\partial z} w \right) dV = 0 \tag{2.39}$$

Lets define a new matrix for this artificial equation.

Pressure Mass Matrix, $M_P = \int_V \beta \phi \phi^T dV$

Thus the equation (2.36) can be modified as,

$$\begin{bmatrix} M & 0 & 0 & 0 \\ 0 & M & 0 & 0 \\ 0 & 0 & M & 0 \\ 0 & 0 & 0 & M_P \end{bmatrix} \begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{P} \end{Bmatrix} + \begin{bmatrix} C(u,v,w) & 0 & 0 & 0 \\ 0 & C(u,v,w) & 0 & 0 \\ 0 & 0 & C(u,v,w) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ P \end{Bmatrix}$$

$$+ \begin{bmatrix} 2K_{11}+K_{22}+K_{33} & K_{12} & K_{13} & -Q_1 \\ K_{21} & K_{11}+2K_{22}+K_{33} & K_{23} & -Q_2 \\ K_{31} & K_{32} & K_{11}+K_{22}+2K_{33} & -Q_3 \\ Q_1^T & Q_2^T & Q_3^T & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ P \end{Bmatrix}$$

$$= \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ 0 \end{Bmatrix} \qquad (2.40)$$

It can be noted in equation (2.40) that mass matrix is no longer singular.

2.4 Taylor Galerkin Stabilization Technique

One key feature of incompressible Navier-Stokes equation is the presence of non-linear unsymmetrical convective term in the equation. As the Reynolds number starts increasing in high velocity flows, convective term starts dominating the flow field inducing oscillations in it, thus making the solution unstable. To handle this difficulty, Taylor-Galerkin(TG) stabilization technique has been employed in this work. TG technique was first presented by J.Donea in which an effective method to couple the time discretization with spatial Galerkin discretization was presented[19]. Purpose of TG method—also known as the finite element equivalent of Lax-Wandroff technique—is to introduce higher order numerical dissipation (diffusion or disper-

sion) to reduce the numerical oscillations. This method consists of writing higher order time derivative of taylor series in terms of spatial derivatives from governing PDEs. The original PDE is then recovered with additional dissipation by rearranging the terms and spatial discretization is then carried out by Galerkin method. Details about TG method and its application to linear,nonlinear equations can be seen in [19, 20, 21, 22, 23]. Here second order TG method is applied only to convective term in Navier-Stokes equations and stabilization matrix is computed. Derivation of this method for Navier-Stokes equations is provided in Appendix. Thus the stabilization matrix is written as,

$$K_{TG}(u,v,w) = \frac{\Delta t}{2}\int_V \psi\psi^T \Big[uu\frac{\partial\psi}{\partial x}\frac{\partial\psi^T}{\partial x} + uv\frac{\partial\psi}{\partial x}\frac{\partial\psi^T}{\partial y} + uw\frac{\partial\psi}{\partial x}\frac{\partial\psi^T}{\partial z} + uv\frac{\partial\psi}{\partial y}\frac{\partial\psi^T}{\partial x} + vv\frac{\partial\psi}{\partial y}\frac{\partial\psi^T}{\partial y} + vw\frac{\partial\psi}{\partial y}\frac{\partial\psi^T}{\partial z} + uw\frac{\partial\psi}{\partial z}\frac{\partial\psi^T}{\partial x} + vw\frac{\partial\psi}{\partial z}\frac{\partial\psi^T}{\partial y} + ww\frac{\partial\psi}{\partial z}\frac{\partial\psi^T}{\partial z}\Big]\, dV \qquad (2.41)$$

After calculating the stabilization matrix, the weak form of Navier-Stokes equation can be modified to,

$$\begin{bmatrix} M & 0 & 0 & 0 \\ 0 & M & 0 & 0 \\ 0 & 0 & M & 0 \\ 0 & 0 & 0 & M_P \end{bmatrix} \begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{P} \end{Bmatrix} + \begin{bmatrix} C(u,v,w) & 0 & 0 & 0 \\ 0 & C(u,v,w) & 0 & 0 \\ 0 & 0 & C(u,v,w) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ P \end{Bmatrix}$$

$$+ \begin{bmatrix} 2K_{11}+K_{22}+K_{33} & K_{12} & K_{13} & -Q_1 \\ K_{21} & K_{11}+2K_{22}+K_{33} & K_{23} & -Q_2 \\ K_{31} & K_{32} & K_{11}+K_{22}+2K_{33} & -Q_3 \\ Q_1^T & Q_2^T & Q_3^T & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ P \end{Bmatrix}$$

$$+ \begin{bmatrix} K_{TG}(u,v,w) & 0 & 0 & 0 \\ 0 & K_{TG}(u,v,w) & 0 & 0 \\ 0 & 0 & K_{TG}(u,v,w) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ P \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ 0 \end{Bmatrix} \quad (2.42)$$

Equation (2.42) is the final weak form of incompressible Navier-Stokes equations which will be discretized using piecewise approximation in following chapter.

CHAPTER 3

INTERPOLATION FUNCTIONS AND FINITE ELEMENT FORMULATION

3.1 Introduction

After obtaining the weak form of Navier-Stokes equations as explained in the previous chapter, the next step is to decide on the choice of shape functions to be used in finite element formulation. The accuracy and the computational efficiency of the solution largely depends on the choice of shape function. To get the continuous and accurate solutions, shape function needs to satisfy completeness and compatibility requirements. The availability of shape functions goes on decreasing with the increasing order of continuity requirements. Using integration by parts helps in lowering the degree continuity requirements of shape function to some extent as done in the derivation to the weak form of Navier-Stokes equations. It can be seen from the final form of the weak Navier-Stokes equations that both the shape functions—ϕ and ψ—should be C^1 continuous as the highest order of derivative appearing in the expressions is 1. According to the investigation carried out by P.Hood and C.Taylor, although it is possible to obtain an accurate velocity field using FEM for Navier-Stokes equation, it leads to certain inaccuracies in the pressure field[15]. After further investigation they found that accurate solutions can be obtained for both velocity and pressure fields by using unequal interpolation in such a way that the shape functions associated with velocity variables are one order higher than those associated with pressure. For this exact reason, two different weight functions, Q and W were selected for continuity and momentum equations. The detailed discussion on the selection of element shape

for geometry discretization and further finite element formulation is included in this chapter.

3.2 Tetrahedral Element

As finite element method carries an advantage over other numerical methods in terms of using unstructured grids to produce accurate solutions, straight sided tetrahedral element is selected in this problem to discretize the physical domain. As mentioned earlier, both velocity and pressure shape functions need to be at least C^1 accurate in order to satisfy compatibility and completeness requirements. For this purpose, first order (linear) shape function ϕ is selected for pressure variable and second order (quadratic) shape function ψ is selected for velocity variables. The derivation of shape functions for linear and quadratic tetrahedron in natural coordinates can be found in many books and articles[12, 16]. These shape functions are directly imported in this document without the derivation.

The quadratic shape function ψ is obtained from 10 node tetrahedron as shown in figure 3.1.

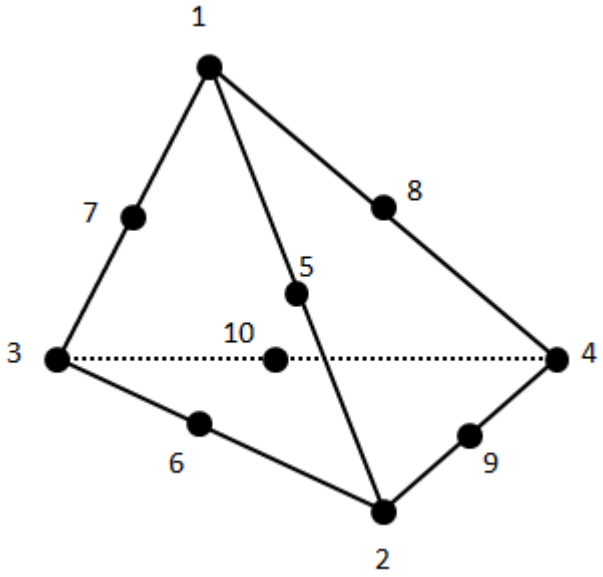


Figure 3.1. Qadratic Tetrahedral Element.

The quadratic shape function in natural coordinate L_i is given below,

$$\psi = \left\{ \begin{array}{c} L_1(2L_1 - 1) \\ L_2(2L_2 - 1) \\ L_3(2L_3 - 1) \\ L_4(2L_4 - 1) \\ 4L_1L_2 \\ 4L_2L_3 \\ 4L_1L_3 \\ 4L_1L_4 \\ 4L_2L_4 \\ 4L_3L_4 \end{array} \right\} \tag{3.1}$$

Similarly, linear shape function ϕ is obtained from a 4 node tet element as shown in figure 3.2.

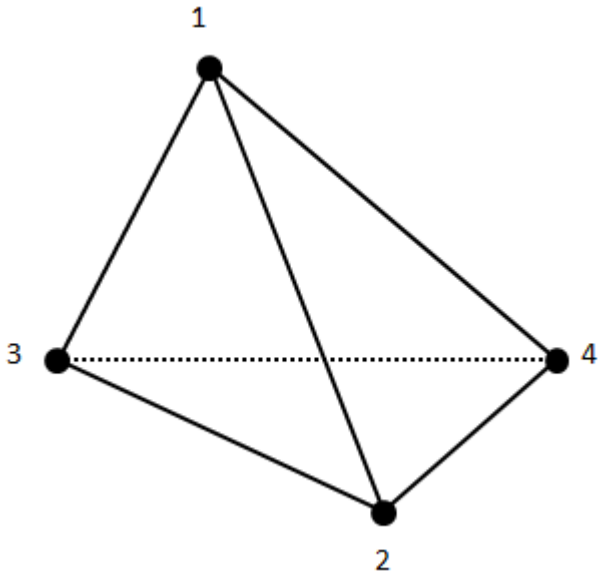


Figure 3.2. Linear Tetrahedral Element.

$$\phi = \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{Bmatrix} \tag{3.2}$$

The natural coordinate $L_1, L_2, L_3 \& L_4$ are functions of the cartesian coordinates.

$$L_1 = (a_1 + b_1 x + c_1 y + d_1 z)/(6V) \tag{3.3}$$

$$L_2 = (a_2 + b_2 x + c_2 y + d_2 z)/(6V) \tag{3.4}$$

$$L_3 = (a_3 + b_3 x + c_3 y + d_3 z)/(6V) \tag{3.5}$$

$$L_4 = (a_4 + b_4 x + c_4 y + d_4 z)/(6V) \tag{3.6}$$

The derivatives of shape functions are given by,

$$\frac{\partial L_i}{\partial x} = \frac{b_i}{6V} \qquad i = 1:4 \tag{3.7}$$

$$\frac{\partial L_i}{\partial y} = \frac{c_i}{6V} \qquad i = 1:4 \tag{3.8}$$

$$\frac{\partial L_i}{\partial z} = \frac{d_i}{6V} \qquad i = 1:4 \tag{3.9}$$

a_1, b_1, c_1 & d_1 coefficients are given as,

$$a_1 = \begin{vmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{vmatrix} \qquad b_1 = -\begin{vmatrix} 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \\ 1 & y_4 & z_4 \end{vmatrix}$$

$$c_1 = -\begin{vmatrix} x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \\ x_4 & 1 & z_4 \end{vmatrix} \qquad d_1 = -\begin{vmatrix} x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \end{vmatrix}$$

Other constants can be calculated by cyclic permutations of subscripts 1, 2, 3 &4 if defined counterclockwise.

Volume 'V' of an element can be found by using following formula.

$$V = \frac{1}{6}\begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix}$$

3.3 Finite Element Formulation

After gathering the necessary information about quadratic and linear shape functions, these shape functions are substituted in eqation (2.31) to (2.35) in order to find the weak form of governing equations given by (2.40). This task is carried out by using the integration formula for the tetrahedral shape functions given below.

$$\int_V L_1^\alpha L_2^\beta L_3^\gamma L_4^\delta \, dV = \frac{\alpha!\beta!\gamma!\delta!}{(\alpha+\beta+\gamma+\delta+3)!}(6V) \tag{3.10}$$

In order to make the task of integration simpler, shape functions and its derivatives are written as matrix-vector product such that all the coefficients are gathered in a matrix form while the vector is comprised of only pure functions of natural coordinates.

Shape function ψ given by equation (3.1) can also be written as $\{\psi\} = [A]\{R\}$ where [A] and {R} are given by,

$$[A] = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix} \qquad \{R\} = \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_1L_2 \\ L_2L_3 \\ L_1L_3 \\ L_1L_4 \\ L_2L_4 \\ L_3L_4 \end{Bmatrix}$$

Derivatives of the shape functions can also be found using this technique of separating the coefficients.

$$\frac{\partial \psi}{\partial x_i} = \frac{\partial \psi}{\partial L_1}\frac{\partial L_1}{\partial x_i} + \frac{\partial \psi}{\partial L_2}\frac{\partial L_2}{\partial x_i} + \frac{\partial \psi}{\partial L_3}\frac{\partial L_3}{\partial x_i} \tag{3.11}$$

where, dimension i changes from 1 to 3 for x, y and z. But $\{\psi\} = [A]\{R\}$ thus,

$$\frac{\partial\{R\}}{\partial x} = \begin{Bmatrix} 2L_1b_1 \\ 2L_2b_2 \\ 2L_3b_3 \\ 2L_4b_4 \\ L_1b_2 + L_2b_1 \\ L_2b_3 + L_3b_2 \\ L_1b_3 + L_3b_1 \\ L_1b_4 + L_4b_1 \\ L_2b_4 + L_4b_1 \\ L_3b_4 + L_4b_3 \end{Bmatrix}$$

But $\frac{\partial\{R\}}{\partial x}$ can be written as $\frac{\partial\{R\}}{\partial x} = [B]\{H\}$ where [B] and {H} are given by,

$$[B] = \begin{bmatrix} 2b_1 & 0 & 0 & 0 \\ 0 & 2b_2 & 0 & 0 \\ 0 & 0 & 2b_3 & 0 \\ 0 & 0 & 0 & 2b_4 \\ b_2 & b_1 & 0 & 0 \\ 0 & b_3 & b_2 & 0 \\ b_3 & 0 & b_1 & 0 \\ b_4 & 0 & 0 & b_1 \\ 0 & b_4 & 0 & b_2 \\ 0 & 0 & b_4 & b_3 \end{bmatrix} \qquad \{H\} = \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{Bmatrix}$$

[C] and [D] can be obtained by replacing b_i from [B] to c_i and d_i respectively. Thus the derivatives of {R} are given as,

$$\frac{\partial\{R\}}{\partial x} = [B]\{H\} \tag{3.12}$$

$$\frac{\partial\{R\}}{\partial y} = [C]\{H\} \tag{3.13}$$

$$\frac{\partial\{R\}}{\partial z} = [D]\{H\} \tag{3.14}$$

After defining the shape functions and their derivatives in appropriate forms, they can be substituted in coefficient matrices given by equations (2.31) to (2.35).

CHAPTER 4

FINITE ELEMENT PROGRAMMING

4.1 Introduction

In this chapter, few key featues of finite element programming have been explained. Even after having a correct finite element formulation, the accuracy and computational efficiency for the solver is largely influenced by these key programming aspects. A C++ object oriented programm is written using notepad++ and compiled using gnu g++ compiler on WINDOWS which comes as a part of CYGWIN.

4.2 Time Marching using Backward Euler (Implicit) Scheme

Equation (2.42) represents the elemental (local) form of finite element equation which can also me written as,

$$[m]\{\dot{u}\} + [K_e(u)]\{u\} = \{F_e\} \tag{4.1}$$

Time marching of above equation is carried out using implicit algorithm. Other time marching algorithms, their comparisons, advantages, limitations with implicit algorithms can be found in [3]. The non linear terms in $[K_e(u)]$ are linearized using the solution in previous time step.

By using the implicit algorithm,

$$[m]\left(\frac{u^{n+1} - u^n}{\Delta t}\right) + [K_e(u^n)]\{u^{n+1}\} = \{F_e\} \tag{4.2}$$

$$\therefore \left(\frac{m}{\Delta t} + [K_e(u^n)]\right) u^{n+1} = \{F_e\} + \frac{m}{\Delta t} u^n \tag{4.3}$$

Thus the modified stiffness matrix and force vectors are,

$$[K_e(u)] = \left(\frac{m}{\Delta t} + [K_e(u^n)]\right) \tag{4.4}$$

$$\{F_e\} = \{F_e\} + \frac{m}{\Delta t}u^n \tag{4.5}$$

Implicit method gives unconditional stability to the solution but at the expense of solving large system of linear equations for each time step. For more accurate time dependant solutions, higher order scheme such as Krank-Nicolson could prove more suitable.

4.3 Assembly of Global Stiffness Matrix using Sparse Matrix Libraries

Local stiffness matrix calculated after using an implicit algorithm is then assembled in a global system. The assembly becomes particularly complicated because of multiple degrees of freedom associated with every node. To add to the complications, 4 corner nodes of the quadratic tetrahedron carry 4 dofs while 6 midside nodes have 3 dofs each. Let N_1 be the number of corner nodes per element and N_2 be number of midside nodes per element then local (elemental) degreed of freedom (L_{dof}) can be calculated as,

$$\begin{aligned} L_{dof} &= 4N_1 + 3N_2 \\ &= 4(4) + 3(6) \\ &= 34 \end{aligned} \tag{4.6}$$

In similar fashion, if N_{1G} are the total corner nodes is the global system and N_{2G} are total midside nodes, global degrees of freedom (G_{dof}) are given as,

$$G_{dof} = 4N_{1G} + 3N_{2G} \tag{4.7}$$

Thus every local stiffness matrix is made up of 34x34 entries which are ordered in $u_1, u_2 \ldots u_{10}, v_1, v_2 \ldots v_{10}, w_1, w_2 \ldots w_{10}, p_1 \ldots p_4$ this fashion. In contrast, the assem-

bly of these local entries into the global system is arranged in $u_1, v_1, w_1, p_1 \ldots u_{N_{1G}}$, $v_{N_{1G}}, w_{N_{1G}}, p_{N_{1G}} \ldots u_n, v_n, w_n$ this way, n being total number of nodes.

Another key issue that arises during assembly process is the large memory requirements to store the global system. Incidentlly, most of the entries in this system are zeros making it a 'Sparse System'. Storing these large number of zeros is unnecessary and time consuming. To avoid this problem, an open source library 'SparseLib++' along with special matrix vector classes 'MV++'are effectively used[24, 25]. Using these libraries make it easier to solve the system of linear equations with minimum efforts and memory. After assembling the global system, factorization is carried out by using standard Incomplete LU Preconditioner and then the system is solved iteratively using Biconjugate Gradient (BiCG) method.

4.4 Program Outline

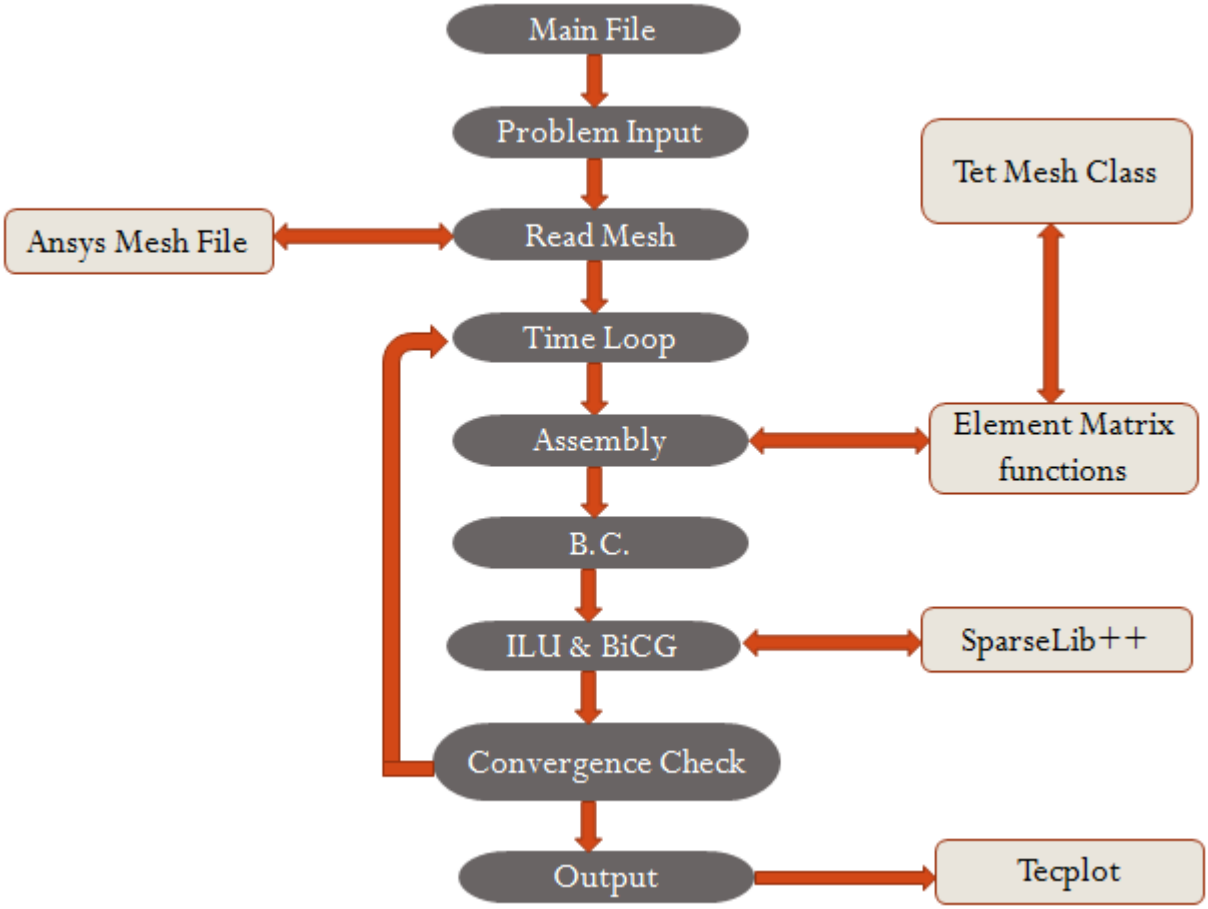


Figure 4.1. Simplified outline of FEM program.

CHAPTER 5

TESTING AND VALIDATION

5.1 Introduction

Before putting any new software to use, it needs to be validated with a benchmark problem. In this work, two validations are performed. First the FEM code is validated using a standard problem of flow in a cylindrical pipe and then using numerical benchmark cases of 3D laminar flow around an obstacle which was defined by DFG high priority research program *flow simulation with high performance computers* by Schäfer and Turek and which was studied by many authors such as[2, 1, 26]. In the second problem, objective is to find pressure drop across the obstacle and coefficients of drag and lift (C_D & C_L) on the surface of the obstacles. A slightly different approach has been adopted in present study and the code is validated only for ΔP across the obstacle. Drag and Lift coefficients are not evaluated but instead mass conservation is monitored to see if it is satisfying the continuity equation. An extensive validation study with the grid dependency test, even though desired, has not been carried out in this work. The purpose of this validation procedure is to find out whether the finite element solutions obtained have physical sense or not.

5.2 Poiseuille Flow

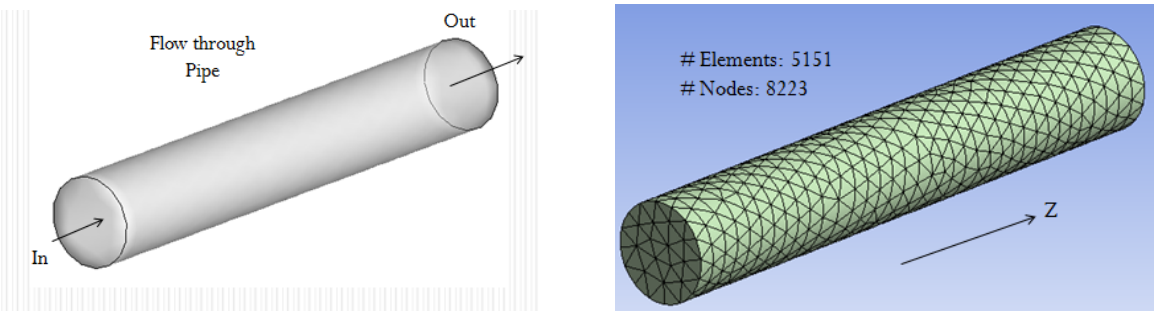
One of the simplest problem to validate the FEM code is flow through a cylinder. One can find analytical solutions for such a problem derived in cylindrical coordinates[27]. The velocity in z direction is given by formula (5.1).

$$w(r) = \frac{\Delta P \ D^2}{16\mu L}\left[1 - \left(\frac{2r}{D}\right)^2\right] \tag{5.1}$$

where, D is a diameter of cylinder and L is its length.

5.2.1 Problem Setup

A cylinder with diameter 0.4 m and length 2.5 m is created and meshed using Ansys Workbench as shown in figure 5.1. On inlet face, uniform velocity of 0.5 m/sec is given in the Z direction as a boundary condition keeping other components of velocity as zero.



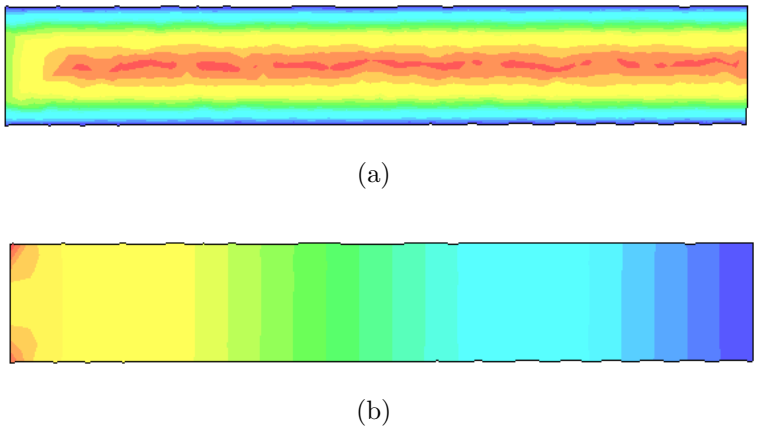
(a) (b)

Figure 5.1. Validation using Poiseuille flow (a) Cylindrical geometry for validation (b) Tetrahedral mesh on the surface of geometry.

This geometry is then subject to test using FEM code at $\Delta t = 0.1\ sec$ and $\beta = 5\ sec^2/m^2$ values. The results are presentated in next section.

5.2.2 Results for Poiseuille Flow

Figure 5.2 shows the pressure and velocity contours of Poiseuille flow on a section plane cut though the centre of cylinder along Z axis. In the velocity contours, it can be seen that the uniform inlet flow develops into the parabolic profile as it moves along the length. maximum velocity at the outlet (0,0,2.5) is measured in tecplot and required pressure drop to produce this velocity is compared using analytical formula and numerical values.



(a)

(b)

Figure 5.2. Result contours for poiseuille flow on section plane cut through the centre of cylinder along Z axis (a) Velocity contours with $w_{max} = 0.085\ m/sec$ and $w_{min} = 0\ m/sec$ (b) Pressure contours with $P_{max} = 0.024\ Pa$ and $P_{min} = 0\ Pa$.

For the maximum velocity of 0.081 m/sec at the end of a channel (0,0,2.5), the analytical and numerical pressure drops between (0,0,0) and (0,0,2.5) is validated as,

$$\Delta P_{Analytical} = 0.02025\ Pa$$

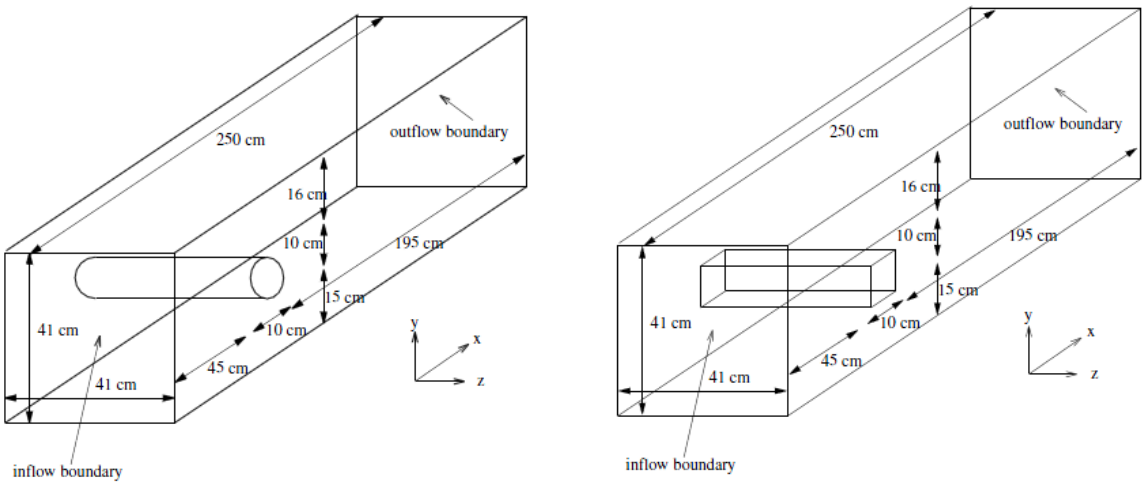
$$\Delta P_{Numerical} = 0.02124\ Pa$$

5.3 Numerical Benchmark Problems for Laminar Flow with Obstacle

In this section, FEM program will be validated using a numerical benchamark problems defined in DFG high priority research program as mentioned earlier.

5.3.1 Test Geometry and Mesh Setup

The validation is performed on two test models. Each model consists of a simple channel with a square cross section and has an obstacle in it. In model 1, the obstacle is a cylinder with circular cross-section while in model 2 it is a parallelepiped with square cross-section. Each model has three boundaries namely inlet, outlet and wall. The obstacle surfaces are also considered as part of walls. These model geometries with their orientations in space are shown in the figure 5.3. Note that the models are oriented along positive X axis. Both these models were created using 'design modeler' in Ansys Workbench.



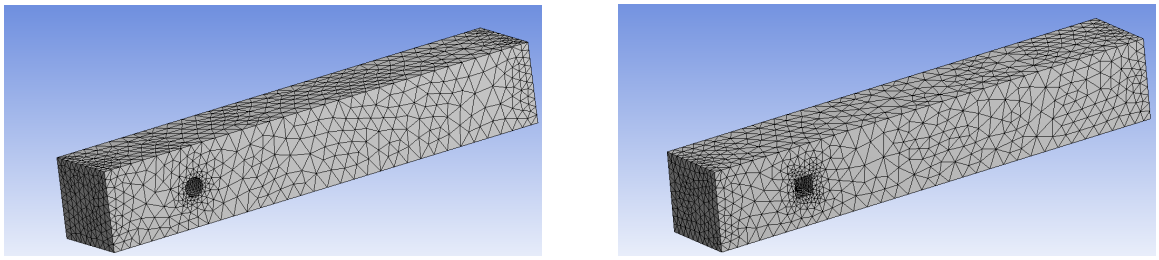
(a) (b)

Figure 5.3. Benchmark models with (a) Model-1:cylindrical obstacle and (b) Model-2:parellelepiped obstacle from[1].

After the model creation, they were divided in finite number of tetrahedral elements using Ansys Workbench 'meshing' module. Both the models were meshed using same meshing parameters but show different number of elements due to slight difference in their geometric shapes. Table 5.1 shows the meshing parameters used. All other parameters have default values given in Ansys Workbench. Figure 5.4 shows tetrahedral mesh in two models.

Table 5.1. Ansys Workbench meshing details for two benchmark models

Meshing Parameter	Selection
Meshing Method	Patch Conforming Tetrahedron
Advance Sizing Function	On Proximity and Curvature
Max Face Sizing	0.08 m
Max Tet Sizing	0.150 m
Element Size on Obstacle Surface	0.02 m
Midside Nodes	Kept
# Elements	Model-1: 18261, Model-2: 23220
# Nodes	Model-1: 27827, Model-2: 34793



(a) (b)

Figure 5.4. Tetrahedral mesh on (a) model-1 and (b) model-2 .

5.3.2 Validation

For both the configurations, same input quantities are used. Density of fluid is $\rho = 1kg/m^3$ and kinematic viscosity is $\nu = 10^{-3}m^2/sec$. Parabolic velocity profile is used as inlet boundary condition only in axial X direction. Y and Z components of velocity remain zero. Thus inlet condition is given by,

$$u(0, y, z) = 16U_m yz(H - y)(H - z)/H^4, \quad v = w = 0 \tag{5.2}$$

where, $H = 0.41m$ and $U_m = 0.45m/s$. These input quantities give $Re \approx 20$. Other input quantities used are time step $\Delta t = 0.1\ sec$ and artificial compressibilitiy $\beta = 5\ sec^2/m^2$.

For the purpose of validation, two quantities are computed.

1. Pressure difference ΔP between points (0.45, 0.2, 0.205) and (0.55, 0.2, 0.205).
2. Mass conservation i. e. $\int_A \vec{u}\, dA$ on inlet and outlet faces, difference between two being the error.

Both these validation operations are carried out by using a post processing tool 'Tecplot'. The pressure drop values are then compared with the ones available in literature[2, 1, 26] to get a sense whether the solution obtained is physical or not.

For model-1 with circular cross-section, reference values are published in [2] using boundary fitted higher order finite element methods as they proved to be the most accurate. Those values are computed using piecewise triquadratic elements for velocity discretization and discontinuous piecewise linear elements for pressure. For model-2 with square cross-section, reference values are taken from [1] which are computed using stabilized triquadratic elements for both velocity and pressure on structured meshes.

Table 5.2. Reference values published in [2] for model-1 with circular cross-section

# cells	dof	ΔP (Pa)
245760	7035840	0.170403
1966080	55666560	0.170779

Table 5.3. Reference values published in [1] for model-2 with square cross-section on structured grids

# cells	dof	ΔP (Pa)
78	3696	0.183495
624	24544	0.208050
4992	177600	0.177370
39936	1348480	0.176165

It can be seen from comparing the values from the result tables that the answers are over predicted in both the cases. Although different types of refinement techniques can be applied to the problem, no further effort has been made in this work to do so.

Different contour plots, streamline plots and residual convergence plots can be seen in figure 5.5 to 5.8

5.4 Convergence Study

To obtain a solution of incompressible Navier-Stokes equations using FEM, the method of artificial compressibility was used and the solutions for two models were validated in previous few sections. Although the validation proves the ability of FEM program to converge towards right solutions, it does not consider the amount of time and number of iterations required to get to the prescribed convergence limits. Convergence of the solution plays a very important role particularly in this work because of the modifications made to the continuity equations based on artificial

Table 5.4. Values obtained in present work for model-1 and model-2 using ILU Preconditioner and Bi-Conjugate Gradient linear solver

case	# elements	dof	ΔP (Pa)	$Q_{in} - Q_{out}$ (m^3/sec)	# iter	# time steps
model-1	18261	100612	0.2164	0.000313	25444	5085
model-2	23220	125238	0.2489	0.002018	32108	5161

compressibility to avoid the formation of indefinite system. As stated earlier, adding an artificial term, $\beta\frac{\partial P}{\partial t}$, to the continuity equation imparts some amount of hyperbolic nature to the problem which violates the incompressibility constraint. The solution obtained can be claimed right only if this time dependant artificial term becomes negligible and hence discarded at steady state. This artificial compressibility β is a free parameter and according to Chorin[17], can be related to the artificial speed of sound in compressible flows with small Mach number by the relation given in equation (5.3).

$$c = 1/\beta^{1/2} \tag{5.3}$$

Unfortunately, one can not find precise guidelines in lituerature on what range of values for β are required to be used to achieve a fast convergence. To damp out the artificial sound waves and hence to accelerate the convergence, an alternate form of Chorin's method is presented by J.D.Rashaw et al. in which an artificial bulk viscosity is introduced in the momentum equations which again vanishes at steady state[28]. In present work, we are making an effort to study what effects does this artificial compressibility has on the convergence speed and can the process be accelerated without adding the bulk viscosity.

5.4.1 Testing Setup

For the purpose of this convergence analysis, a simpler geometry which can be effectively meshed in fewer number of elements is preferred as it will allow for faster convergence and multiple test runs. The model is a very simple channel with square cross-section. The dimensions remain the same as per the models used in validation procedure. The obstacles are removed and hence becomes very simple to mesh. This simple channel geometry with its orientation is shown in figure 5.9. Note that its orientation in space is along positive Z axis.

Boundary conditions used for this study are given in table 5.5

Table 5.5. Boundary Conditions for Convergence Analysis

Boundary Face	Boundary Name	Boundary Condition
Inlet	Velocity Inlet	$u = 0$ $v = 0$ $w = 16U_m xy(H-x)(H-y)/H^4$
Outlet	Atmospheric Pressure Outlet	$P = 0$
Wall	No Slip	$u = 0$ $v = 0$ $w = 0$

where,U_m is the velocity amplitude, H=0.41m is the dimension of channel cross-section.

5.4.2 Pseudo Convergence Number (PCN)

To understand the effect of artificial compressibility on the solution convergence of Navier-Stokes equations, it becomes important to know in what form it appears in the weak statement of governing equations. In this work the the governing equations

used are in dimensional form and hence it becomes difficult to establish specific effects of different parameters on convergence because too many input parameters could get involved in the analysis. For resolving this difficulty, we define a new non dimensional number which takes into account the effects of artificial compressibility β, time step size Δt and material properties ρ and μ. This 'Pseudo Convergence Number (PCN)' then can be controlled to observe the effects of other input parameters such as grid size and boundary conditions on the convergence.

$$PCN = \nu \frac{\beta}{\Delta t} \tag{5.4}$$

where, ν denoted the kinematic viscosity and has dimensions m^2/sec while artificial compressibility from equation (5.3) has the dimensions of sec^2/m^2. This dimensionless number will be used as a variable to observe the effects of grid and boundary conditions on solution convergence.

5.4.3 Effect of Grid Size and Reynolds Number on Solution Convergence

In this section, effects of refining the mesh and changing the input velocity magnitude on convergence speed is studied at different values of PCN. In the first step, the convergence was tested for the same grid but three different Reynolds numbers and their convergence history was recorded. Reynolds numbers used are 10, 20 and 40. In the second step, similar testing was performed but this time for the same Reynolds number and three different grids. Simple tetrahedral grids without any local refinement are used in this study.

The two steps mentioned above produce four sets of results which are explained below. In each of these cases, PCN is varied to see the effect on other variables.

1. Case 1: In this case, same grid i. e. grid-1 is used for three different Reynolds numbers of 10, 20 and 40 to plot total number of iterations required for linear

Table 5.6. Three different grids used for convergence study

Grid	# Elements	# Nodes
Grid-1	2234	4250
Grid-2	4051	6469
Grid-3	6121	9612

solver to converge. Results obtained are shown in figure 5.10. It is clear from the figure that more iterations are required for the solver to converge and hence curve shift slightly upwards with increase in Reynolds number.

2. Case 2: In this case, same grid is used for three Reynolds numbers and variation in number of time steps is observed. It can be seen from figure 5.11 that the relationship is not very clear and it is hard to see any pattern in the limited range of PCN which was tested. Although the three curves seem to be following the order, the fluctuations in them make it hard to draw any conclusion.
3. Case 3: In this case, Reynolds number is kept constant at 20 and PCN was varied for three different grids to check the number of iterations required for convergence. Figure 5.12 shows that for the initial part of the curves till PCN=0.1 show a definite trend, a finer mesh requiring more number of iterations to converge than denser one. But for higher range of PCN, that trend is lost.
4. Case 4: This case deals with constant Reynolds number of 20 and different grid sizes to record the number of time steps required for convergence. Agin there is no definite trend found and fluctuations with different amplitude and different wavelengths are observed in figure 5.13.

It can be seen by studying these four cases that there certainly exiists a relationship between PCN and convergence speed, but the exact nature is complicated.

5.4.4 Elemental Reynolds Number

As seen in previous section, different grids and Reynolds numbers seem to be affecting the convergence speed. To combine the effect of two, we consider an 'Elemental Reynolds Number (ERe)' which is given by,

$$ERe = \frac{u_e h_e}{\nu} \tag{5.5}$$

where, u_e is the local velocity magnitude and h_e is the characteristic length of an element which is the ratio of volume of an element and its surface area. The idea behind using ERe is to be able to change PCN at elemental level to obtain faster convergence. Elemental Reynolds number combines the effect of grid size and boundary conditions together. Thus,

$$PCN = f(ERe) \tag{5.6}$$

where 'f' remain unknown function which needs to be determined. By using this relationship it would be possible to modify the value of PCN for every element when assembly of global system takes place. The value of PCN should be such that it would contribute towards speeding the convergence of global system. In an effort to find this unkown function 'f', variation in number of iterations and number of times steps is plotted against PCN for different values of ERe. It can be seen in figures 5.14 and 5.15. The total number of iterations seem to be converging towards some minimum value for different values of ERe, but the plot showing the time steps required is again a matter of concern. Unfortunately the range of PCN which was used for this study proved is too small to make any conclusions and hence definite relationship given by equation (5.6) remians unknown.

5.5 Concluding Remarks

This work presented an effective way to use Finite Element Method to solve three dimensional Navier-Stokes equations. The entire work was focused on developing a solver which would serve as a basic building block for more advanced and innovative techniques to be implemented in future. Method of artificial compressibility was effectively used to avoid the formation of indefinite system and hence factorization of linear system was performed using standard preconditioners. The entire formulation was then encoded using C++ programming language which makes it possible to run the program at high speeds. The solutions obtained were validated using benchmark case of three dimensional laminar flow.

Convergence analysis was performed to see which input parameters affect the convergence speed. A new dimensionless number 'Pseudo Convergence Number' was defined as $PCN = \nu\frac{\beta}{\Delta t}$ and its effect on convergence speed was observed for different grids and Reynolds numbers. It was found that the effect of different meshes and input velocities can be combined together in the form of 'Elemental Reynolds Number (ERe)'. It was proposed that the improved speeds of convergence can be achieved by substituting PCN as a function of ERe but the exact function remains unknown and requires future investigation.

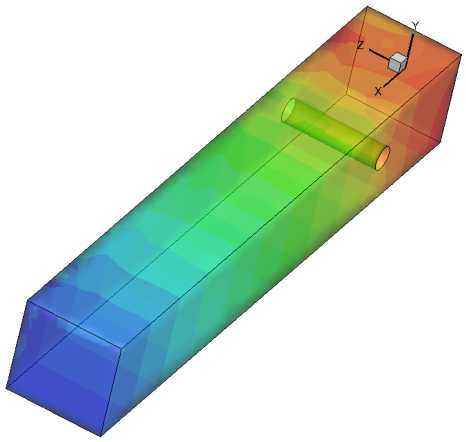
5.6 Future Work

No work is complete without the mention of possible future improvements. Two such high priority recommendations are listed below.

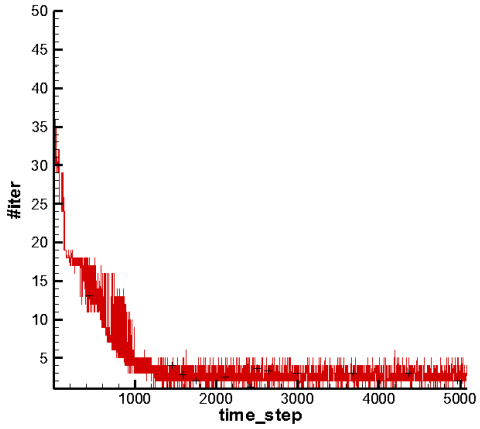
1. Dual Time Marching: In the present work, although unsteady form of governing equations are used, solution obtained are valid only at steady state when artificial compressibility can be ignored. To obtain unsteady solutions, dual

time marching is required. Two time variables, one in real time (Δt) and one in pseudo time ($\Delta \tau$) are used. For every real time step, steady state is achieved in pseudo time where artificial variables can be disposed safely without affecting the solutions.

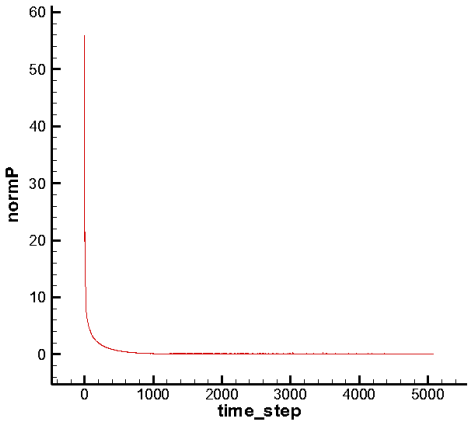
2. Non Dimensional Forms of Governing Equations: The governing equations used for FEM formulation are in dimensional form. Handling non dimensional numbers such as Re, ERe, and PCN which are used in this work, is difficult in dimensional form. Using non dimensional forms would be very useful for more effective convergence analysis.



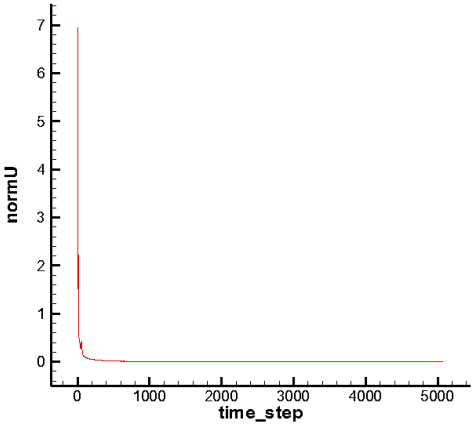
(a)



(b)

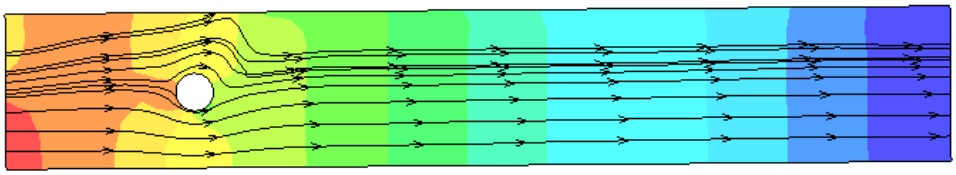


(c)

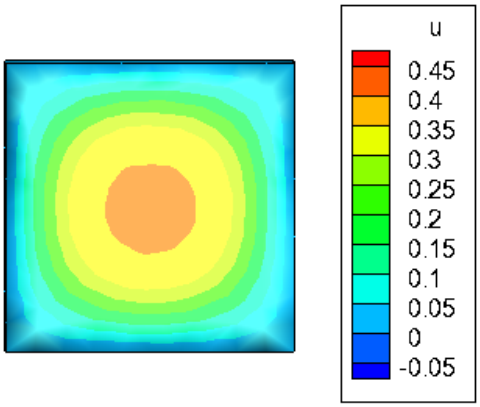


(d)

Figure 5.5. Solution results for model-1, (a) Contours of pressure on the surfaces of model-1, (b) Iteration history with time marching, (c) Convergence of $\frac{\beta}{\Delta t}|dP|$ with time marching, (d) Convergence of $|dU|$ with time marching .



(a)



(b)

Figure 5.6. Model-1 with (a) Streamlines drawn on the the plane showing pressure contours cutting through circular obstacle at z=0.205m (b) Parabolic boundary condition on inlet face .

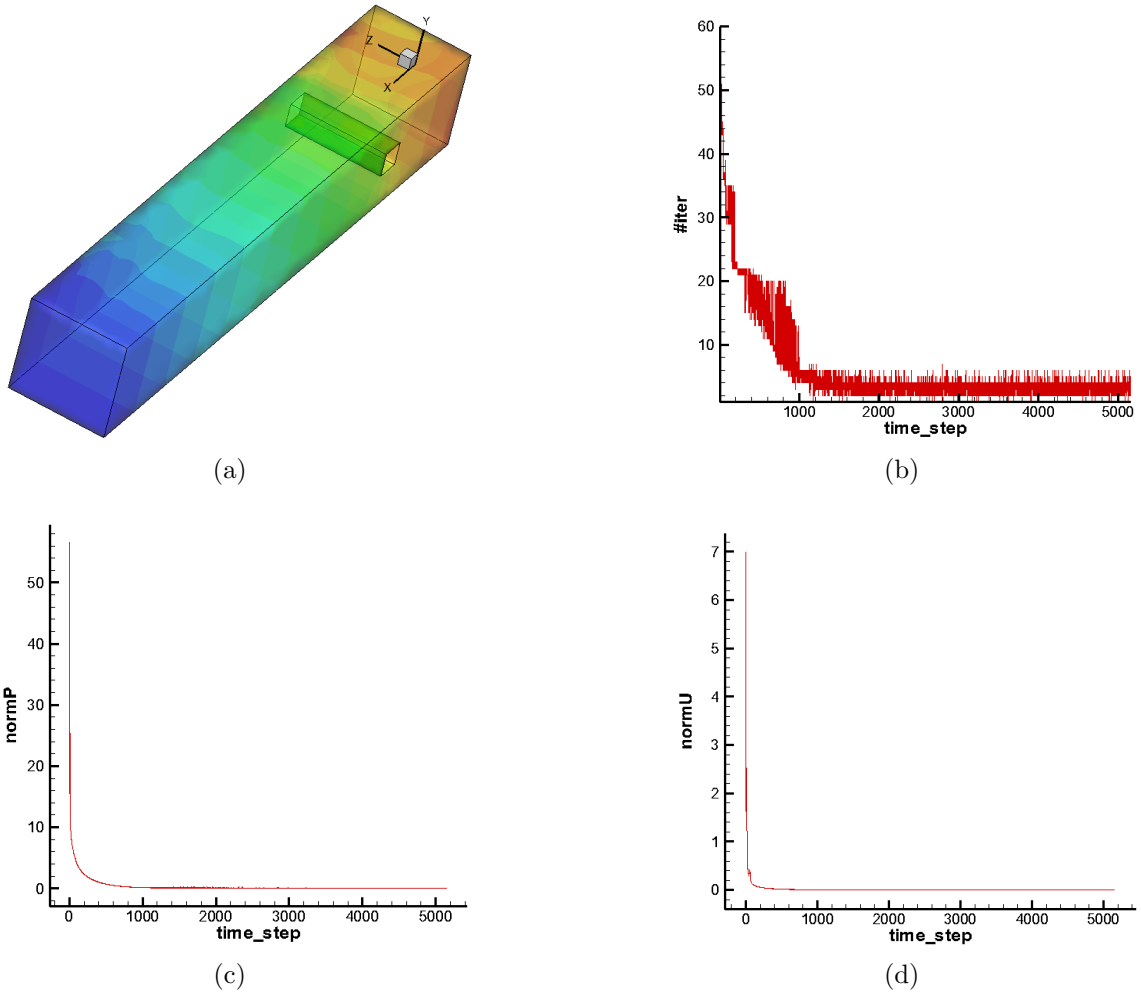
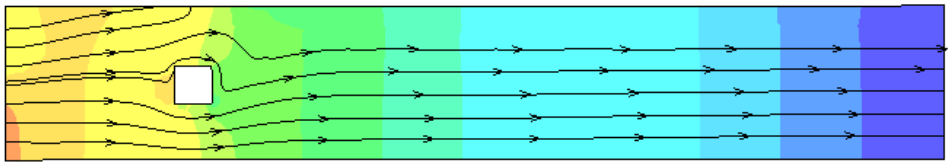
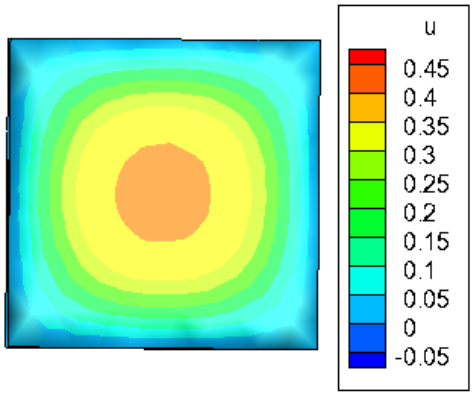


Figure 5.7. Solution results for model-2, (a) Contours of pressure on the surfaces of model-2, (b) Iteration history with time marching, (c) Convergence of $\frac{\beta}{\Delta t}|dP|$ with time marching, (d) Convergence of $|dU|$ with time marching .



(a)



(b)

Figure 5.8. Model-2 with (a) Streamlines drawn on the the plane showing pressure contours cutting through square obstacle at z=0.205m (b) Parabolic boundary condition on inlet face .

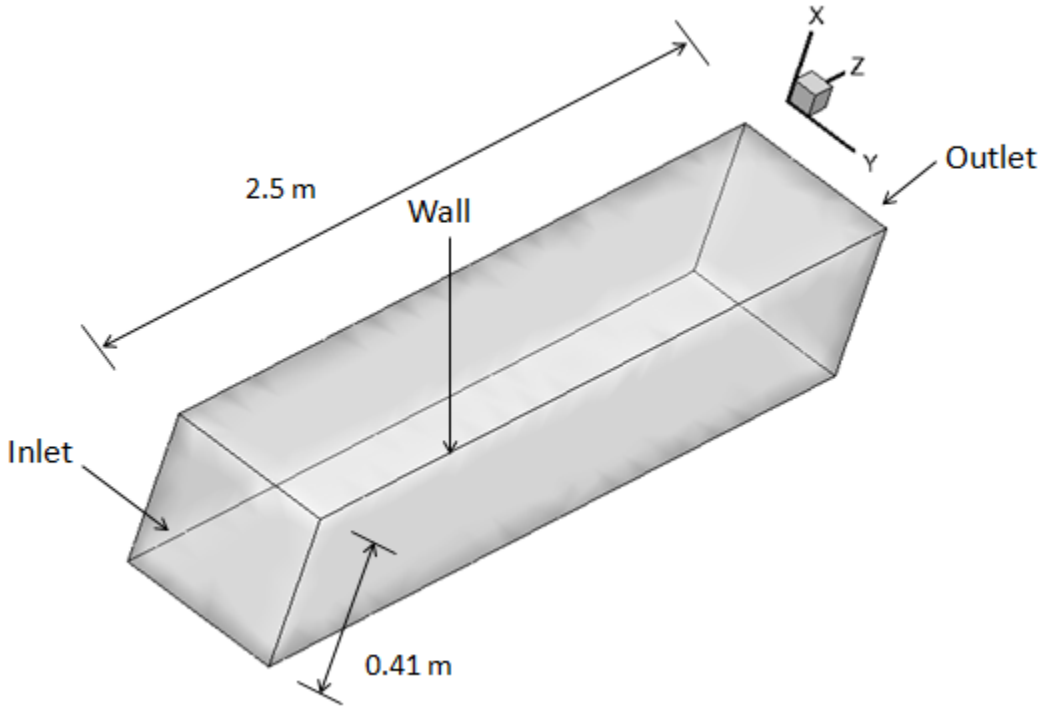


Figure 5.9. Channel geometry used for convergence study with its orientation in 3D space.

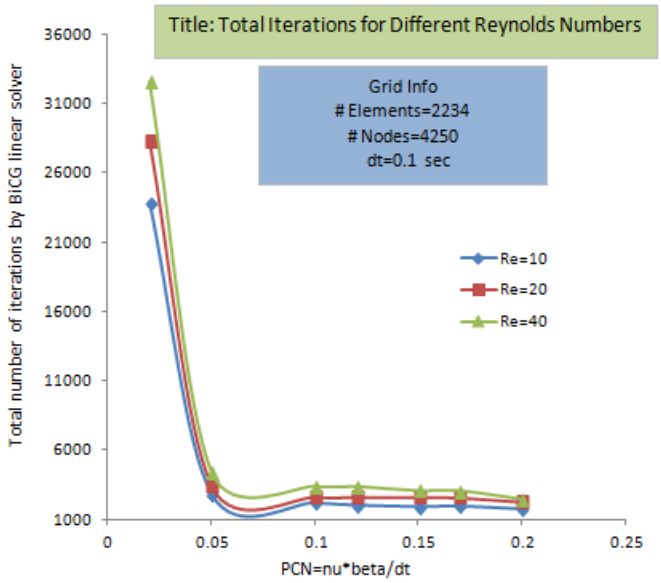


Figure 5.10. Variations in total number of iterations required for BiCG solver with ILU Preconditioner to converge for different Reynolds numbers on grid-1.

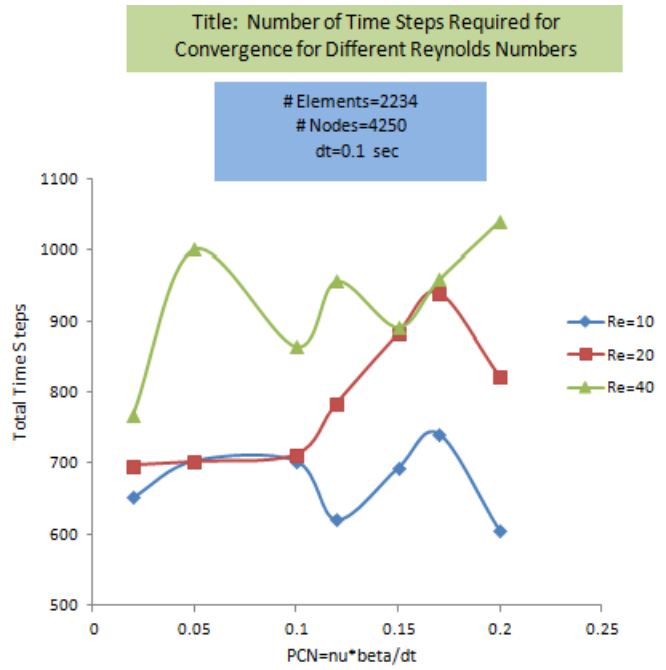


Figure 5.11. Variations in total number of time steps with $\Delta t = 0.1$ for different Reynolds numbers on grid-1.

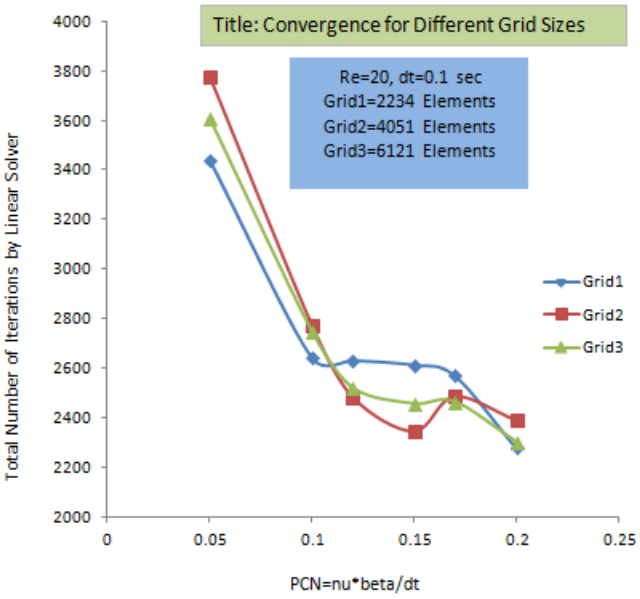


Figure 5.12. Variations in total number of iterations required for BiCG solver with ILU Preconditioner to converge for different grids with $Re = 20$.

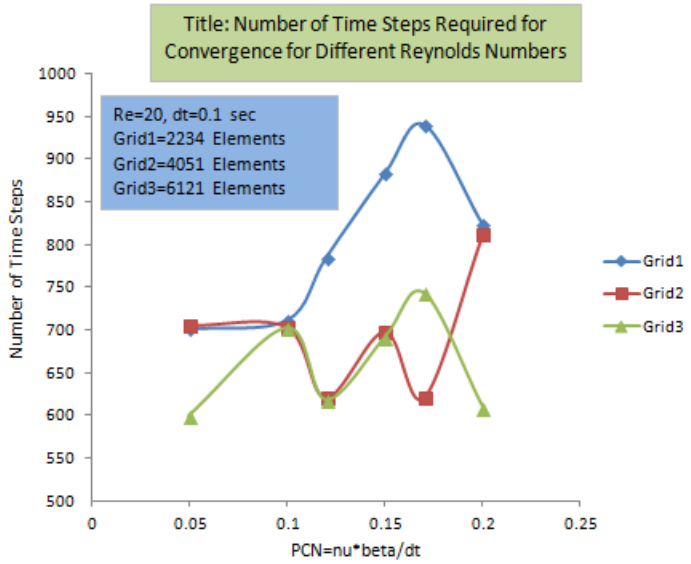


Figure 5.13. Variations in total number of time steps with $\Delta t = 0.1$ for different grids with $Re = 20$.

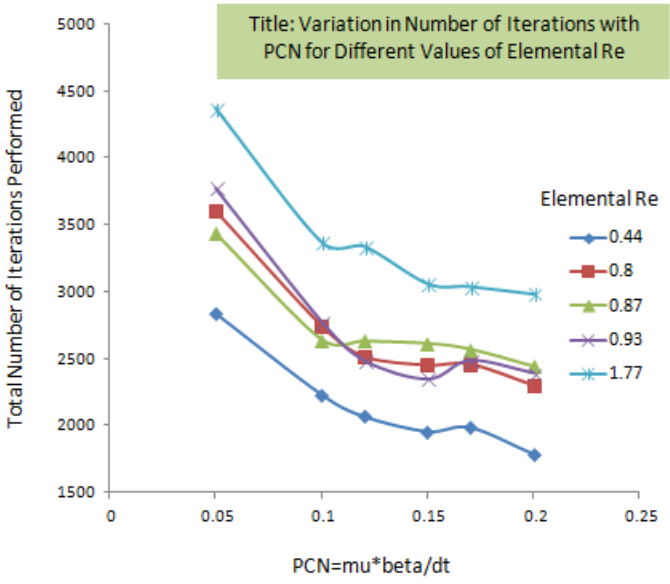


Figure 5.14. Variations in total number of iterations over a range of PCN for different values of ERe.

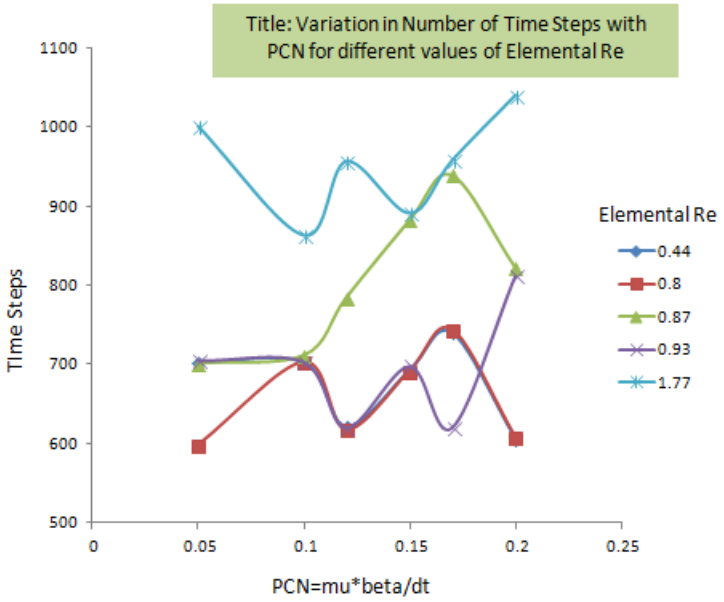


Figure 5.15. Variations in total number of time steps over a range of PCN for different values of ERe.

APPENDIX A

DERIVATIONS OF COEFFICIENT MATRICES

The coefficient matrices which needs to be substituted in equations (2.31) to (2.35) are derived in this appendix.

A.1 Preparation

Before finding the coefficient matrices, it becomes important to express the shape functions in suitable forms. This task is carried out in this section.

$$\{\psi\} = \begin{Bmatrix} L_1(2L_1-1) \\ L_2(2L_2-1) \\ L_3(2L_3-1) \\ L_4(2L_4-1) \\ 4L_1L_2 \\ 4L_2L_3 \\ 4L_1L_3 \\ 4L_1L_4 \\ 4L_2L_4 \\ 4L_3L_4 \end{Bmatrix} = \begin{Bmatrix} 2L_1^2-L_1 \\ 2L_2^2-L_2 \\ 2L_3^2-L_3 \\ 2L_4^2-L_4 \\ 4L_1L_2 \\ 4L_2L_3 \\ 4L_1L_3 \\ 4L_1L_4 \\ 4L_2L_4 \\ 4L_3L_4 \end{Bmatrix} \tag{A.1}$$

$\{\psi\}$ can also be written as $\{\psi\} = [A]\{R\}$ where,

$$[A] = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix} \qquad \{R\} = \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_1L_2 \\ L_2L_3 \\ L_1L_3 \\ L_1L_4 \\ L_2L_4 \\ L_3L_4 \end{Bmatrix}$$

This step separates the coefficients of natural coordinates into a square matrix and pure functions of natural coordinates into a vector. Integrations of shape functions which will follow after next few steps become easier due to this separation. The same procedure is followed for the derivatives of shape functions as well.

$$\frac{\partial \psi}{\partial x} = \frac{\partial \psi}{\partial L_1}\frac{\partial L_1}{\partial x} + \frac{\partial \psi}{\partial L_2}\frac{\partial L_2}{\partial x} + \frac{\partial \psi}{\partial L_3}\frac{\partial L_3}{\partial x} + \frac{\partial \psi}{\partial L_3}\frac{\partial L_4}{\partial x} \tag{A.2}$$

$$\frac{\partial \psi}{\partial y} = \frac{\partial \psi}{\partial L_1}\frac{\partial L_1}{\partial y} + \frac{\partial \psi}{\partial L_2}\frac{\partial L_2}{\partial y} + \frac{\partial \psi}{\partial L_3}\frac{\partial L_3}{\partial y} + \frac{\partial \psi}{\partial L_3}\frac{\partial L_4}{\partial y} \tag{A.3}$$

$$\frac{\partial \psi}{\partial z} = \frac{\partial \psi}{\partial L_1}\frac{\partial L_1}{\partial z} + \frac{\partial \psi}{\partial L_2}\frac{\partial L_2}{\partial z} + \frac{\partial \psi}{\partial L_3}\frac{\partial L_3}{\partial z} + \frac{\partial \psi}{\partial L_3}\frac{\partial L_4}{\partial z} \tag{A.4}$$

But $\{\psi\} = [A]\{R\}$ and hence,

$$\frac{\partial\{R\}}{\partial x} = \begin{Bmatrix} 2L_1b_1 \\ 2L_2b_2 \\ 2L_3b_3 \\ 2L_4b_4 \\ L_1b_2 + L_2b_1 \\ L_2b_3 + L_3b_2 \\ L_1b_3 + L_3b_1 \\ L_1b_4 + L_4b_1 \\ L_2b_4 + L_4b_1 \\ L_3b_4 + L_4b_3 \end{Bmatrix} = [B]\{H\} \tag{A.5}$$

$$\frac{\partial\{R\}}{\partial y} = \begin{Bmatrix} 2L_1c_1 \\ 2L_2c_2 \\ 2L_3c_3 \\ 2L_4c_4 \\ L_1c_2 + L_2c_1 \\ L_2c_3 + L_3c_2 \\ L_1c_3 + L_3c_1 \\ L_1c_4 + L_4c_1 \\ L_2c_4 + L_4c_1 \\ L_3c_4 + L_4c_3 \end{Bmatrix} = [C]\{H\} \tag{A.6}$$

$$\frac{\partial\{R\}}{\partial z} = \left\{ \begin{array}{c} 2L_1d_1 \\ 2L_2d_2 \\ 2L_3d_3 \\ 2L_4d_4 \\ L_1d_2 + L_2d_1 \\ L_2d_3 + L_3d_2 \\ L_1d_3 + L_3d_1 \\ L_1d_4 + L_4d_1 \\ L_2d_4 + L_4d_1 \\ L_3d_4 + L_4d_3 \end{array} \right\} = [D]\{H\} \tag{A.7}$$

where,

$$[B] = \begin{bmatrix} 2b_1 & 0 & 0 & 0 \\ 0 & 2b_2 & 0 & 0 \\ 0 & 0 & 2b_3 & 0 \\ 0 & 0 & 0 & 2b_4 \\ b_2 & b_1 & 0 & 0 \\ 0 & b_3 & b_2 & 0 \\ b_3 & 0 & b_1 & 0 \\ b_4 & 0 & 0 & b_1 \\ 0 & b_4 & 0 & b_2 \\ 0 & 0 & b_4 & b_3 \end{bmatrix} \tag{A.8}$$

$$[C] = \begin{bmatrix} 2c_1 & 0 & 0 & 0 \\ 0 & 2c_2 & 0 & 0 \\ 0 & 0 & 2c_3 & 0 \\ 0 & 0 & 0 & 2c_4 \\ c_2 & c_1 & 0 & 0 \\ 0 & c_3 & c_2 & 0 \\ c_3 & 0 & c_1 & 0 \\ c_4 & 0 & 0 & c_1 \\ 0 & c_4 & 0 & c_2 \\ 0 & 0 & c_4 & c_3 \end{bmatrix} \tag{A.9}$$

$$[D] = \begin{bmatrix} 2d_1 & 0 & 0 & 0 \\ 0 & 2d_2 & 0 & 0 \\ 0 & 0 & 2d_3 & 0 \\ 0 & 0 & 0 & 2d_4 \\ d_2 & d_1 & 0 & 0 \\ 0 & d_3 & d_2 & 0 \\ d_3 & 0 & d_1 & 0 \\ d_4 & 0 & 0 & d_1 \\ 0 & d_4 & 0 & d_2 \\ 0 & 0 & d_4 & d_3 \end{bmatrix} \tag{A.10}$$

$$\{H\} = \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{Bmatrix} \tag{A.11}$$

The linear shape function $\{\phi\}$ is same as $\{H\}$.

The formula for the exact integrations of shape functions in natural coordinates is given as,

$$\int_V L_1^\alpha L_2^\beta L_3^\gamma L_4^\delta \, dV = \frac{\alpha!\beta!\gamma!\delta!}{(\alpha+\beta+\gamma+\delta+3)!}(6V) \tag{A.12}$$

A.2 Mass and Pressure Mass Matrices

$$[M] = \int_V \rho\psi\psi^T \, dV \tag{A.13}$$

$$= \rho[A] \int_V \{R\}\{R\}^T \, dV \; [A]^T \tag{A.14}$$

$$[M] = \frac{1}{2520}\rho[A] \begin{bmatrix} 6 & 1 & 1 & 1 & -4 & -6 & -4 & -4 & -6 & -6 \\ 1 & 6 & 1 & 1 & -4 & -4 & -6 & -6 & -4 & -6 \\ 1 & 1 & 6 & 1 & -6 & -4 & -4 & -6 & -6 & -4 \\ 1 & 1 & 1 & 6 & -6 & -6 & -6 & -4 & -4 & -4 \\ -4 & -4 & -6 & -6 & 32 & 16 & 16 & 16 & 16 & 8 \\ -6 & -4 & -4 & -6 & 16 & 32 & 16 & 8 & 16 & 16 \\ -4 & -6 & -4 & -6 & 16 & 16 & 32 & 16 & 8 & 16 \\ -4 & -6 & -6 & -4 & 16 & 8 & 16 & 32 & 16 & 16 \\ -6 & -4 & -6 & -4 & 16 & 16 & 8 & 16 & 32 & 16 \\ -6 & -6 & -4 & -4 & 8 & 16 & 16 & 16 & 16 & 32 \end{bmatrix} [A]^T \tag{A.15}$$

$$[M_P] = \int_V \beta\phi\phi^T \, dV \tag{A.16}$$

$$= \beta \int_V \{H\}\{H\}^T \, dV \tag{A.17}$$

$$= \beta[G] \tag{A.18}$$

where,

$$[G] = \int_V \{H\}\{H^T\}\, dV \tag{A.19}$$

Therefore,

$$[M_P] = \beta \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix} \tag{A.20}$$

A.3 Gradient Matrix

$$[Q_i] = \int_V \frac{\partial \psi}{\partial x_i} \phi^T\, dV \tag{A.21}$$

$$= [A] \int_V \frac{\partial \{R\}}{\partial x_i} \phi^T\, dV \tag{A.22}$$

Therefore,

$$[Q_1] = [A][B][G] \tag{A.23}$$

$$[Q_2] = [A][C][G] \tag{A.24}$$

$$[Q_3] = [A][D][G] \tag{A.25}$$

A.4 Diffusive Matrix

$$[K_{ij}] = \int_V \mu \left(\frac{\partial \psi}{\partial x_i} \frac{\partial \psi^T}{\partial x_j} \right) dV \tag{A.26}$$

$$= \mu[A] \int_V \left(\frac{\partial \{R\}}{\partial x_i} \frac{\partial \{R^T\}}{\partial x_j} \right) dV \;\; [A]^T \tag{A.27}$$

$$[K_{11}] = [A][B][M][B]^T[A]^T \tag{A.28}$$

$$[K_{22}] = [A][C][M][C]^T[A]^T \tag{A.29}$$

$$[K_{33}] = [A][D][M][D]^T[A]^T \tag{A.30}$$

$$[K_{12}] = [A][B][M][C]^T[A]^T \tag{A.31}$$

$$[K_{21}] = [A][C][M][B]^T[A]^T \tag{A.32}$$

$$[K_{13}] = [A][B][M][D]^T[A]^T \tag{A.33}$$

$$[K_{31}] = [A][D][M][B]^T[A]^T \tag{A.34}$$

$$[K_{23}] = [A][C][M][D]^T[A]^T \tag{A.35}$$

$$[K_{32}] = [A][D][M][C]^T[A]^T \tag{A.36}$$

A.5 Force Vector

$$F_i = \int_V \rho\psi f_i \, dV \tag{A.37}$$

A.6 Convective Matrix

$$C(u,v,w) = \int_V \rho[\underbrace{\psi(\psi^T u)\frac{\partial\psi^T}{\partial x}}_{C_x} + \underbrace{\psi(\psi^T v)\frac{\partial\psi^T}{\partial y}}_{C_y} + \underbrace{\psi(\psi^T w)\frac{\partial\psi^T}{\partial w}}_{C_z}] \, dV \tag{A.38}$$

$$C_x = [A][A]^T \int_V [R][R]^T[A]^T u[H]^T \, dV[B]^T[A]^T \tag{A.39}$$

$$C_y = [A][A]^T \int_V [R][R]^T[A]^T v[H]^T \, dV[C]^T[A]^T \tag{A.40}$$

$$C_z = [A][A]^T \int_V [R][R]^T[A]^T w[H]^T \, dV[D]^T[A]^T \tag{A.41}$$

APPENDIX B

TAYLOR-GALERKIN FORMULATION

In this appendix, derivation of Taylor Galerkin technique for Navier-Stokes equation is shown.

For the purpose of simplicity and to avoid tediousness during derivation, vector form of Navier-Stokes equation is used. Although TG technique can be applied to the entire Navier-Stokes equation, only convective terms are considered in this work as stability issues arise due to them.

Taylor series expansion for $n + 1^{th}$ time step is,

$$u^{n+1} = u^n + \Delta t \frac{\partial u^n}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 u^n}{\partial t^2} + \dots \tag{B.1}$$

The incompressible Navier-Stokes equations considering only convective terms are given by,

$$\frac{\partial u}{\partial t} = -\vec{u}.\nabla\vec{u} \tag{B.2}$$

Second derivative of u with respect to t is obtained as,

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial t}(-\vec{u}.\nabla\vec{u}) \tag{B.3}$$

$$= -\vec{u}.\nabla\left(\frac{\partial u}{\partial t}\right) \tag{B.4}$$

$$= \vec{u}.\nabla(\vec{u}.\nabla\vec{u}) \tag{B.5}$$

Equation (B.4) is obtained by taking the ∇ operator inside assuming $\vec{u}$ as constant to avoid higher order terms.

Substituting (B.2) and (B.5) in (B.1) to recover the original equation and some additional terms.

$$\frac{u^{n+1} - u^n}{\Delta t} = -\vec{u}.\nabla u + \frac{\Delta t}{2}\ \vec{u}.\nabla(\vec{u}.\nabla u) \tag{B.6}$$

It can be seen by comparing (B.6) and (B.2) that the stabilization term obtained using TG technique in a strong form of Navier-Stokes equation is $\frac{\Delta t}{2}\ \vec{u}.\nabla(\vec{u}.\nabla\vec{u})$.

Thus the weak form of this stabilization term can be found by method of weighted residuals and applying integration by parts(Gauss's Theorem),

$$-\frac{\Delta t}{2} W \int_V \vec{u}.\nabla(\vec{u}.\nabla\vec{u}) = -\frac{\Delta t}{2} W \int_V (\vec{u}.\nabla\vec{u})(\nabla.\vec{u})\, dV \tag{B.7}$$

$$= \int_V W \left[\left(u\frac{\partial \vec{u}}{\partial x} + v\frac{\partial \vec{u}}{\partial y} + w\frac{\partial \vec{u}}{\partial z} \right) \left(\frac{\partial \vec{u}}{\partial x} + \frac{\partial \vec{u}}{\partial y} + \frac{\partial \vec{u}}{\partial z} \right) \right] dV \tag{B.8}$$

Thus the final expression of TG stabilization is given by,

$$\begin{aligned} K_{se}(u,v,w) = \frac{\Delta t}{2} \int_V \psi\psi^T \Big[& uu\frac{\partial\psi}{\partial x}\frac{\partial\psi^T}{\partial x} + uv\frac{\partial\psi}{\partial x}\frac{\partial\psi^T}{\partial y} + uw\frac{\partial\psi}{\partial x}\frac{\partial\psi^T}{\partial z} + \\ & uv\frac{\partial\psi}{\partial y}\frac{\partial\psi^T}{\partial x} + vv\frac{\partial\psi}{\partial y}\frac{\partial\psi^T}{\partial y} + vw\frac{\partial\psi}{\partial y}\frac{\partial\psi^T}{\partial z} + \\ & uw\frac{\partial\psi}{\partial z}\frac{\partial\psi^T}{\partial x} + vw\frac{\partial\psi}{\partial z}\frac{\partial\psi^T}{\partial y} + ww\frac{\partial\psi}{\partial z}\frac{\partial\psi^T}{\partial z} \Big] dV \end{aligned} \tag{B.9}$$

To calculate above expression, all the terms need to be integrated. One of such examples is shown below.

$$\int_V \psi\psi^T \left(uu\frac{\partial\psi}{\partial x}\frac{\partial\psi^T}{\partial x} \right) dV = \int_V [A]\{R\}\{R\}^T[A]^T uu[A][B]\{H\}\{H\}^T[B]^T[A]^T\, dV \tag{B.10}$$

$[B]$ is replaced by $[C]$ and $[D]$ to calculate the other terms where derivatives appear with respect to y and z respectively.

REFERENCES

[1] M. Braack and T. Richter, "Solutions of 3D Navier-Stokes benchmark problems with adaptive finite elements," *Computers and Fluids*, vol. 35, no. 4, pp. 372–392.

[2] V. John, "Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier Stokes equations," *International Journal for Numerical Methods in Fluids*, vol. 40, no. April, pp. 775–798, 2002.

[3] J. D. Anderson, *Computational fluid dynamics: the basics with applications*. McGraw-Hill, 1995.

[4] O. C. Zienkiewicz, R. L. Taylor, and P. Nithiarasu, *The Finite Element Method for Fluid Dynamics*. Butterworth-Heinemann, 2005.

[5] F. M. White, *Fluid Mechanics*. McGraw-Hill, 2010.

[6] G. Nugroho, A. M. Ali, and Z. a. Abdul Karim, "A class of exact solutions to the three-dimensional incompressible NavierStokes equations," *Applied Mathematics Letters*, vol. 23, no. 11, pp. 1388–1396, Nov. 2010.

[7] O.C.Zienkiewicz, "Finite Element Methods in Flow Problems," in *International Symposium on Finite Element Methods in Flow Problems*. Swansea, United Kingdom: UAH Press, 1974.

[8] W. Jiajan, "Solution to Incompressible Navier Stokes Equations by using Finite Element Method," Ph.D. dissertation, University of Texas at Arlington, Jan. 2010.

[9] J. C. Tannehill, D. A. Anderson, and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*. Taylor & Francis, 1997.

[10] Y. Yoshida and T. Nomura, "A Numerical Method for Finite Element Transient Analysis of Incompressible Navier-Stokes Eqations," *Theoretical and applied mechanics*, vol. 34, pp. 29–42, 1986.

[11] C. Taylor and P. Hood, "A Numerical Solution to the Navier-Stokes Equations Using the Finite Element Technique," *Comput. Fluids,*, vol. 1, pp. 73–100, 1973.

[12] K. H. Huebner, D. L. Dewhirst, D. E. Smith, and T. G. Byrom, *The Finite Element Method for Engineers*. John Wiley & Sons, 2001.

[13] B. A. Finlayson, "Weighted Residual Methods and their Relation to Finite Element Methods in Flow Problems," in *International Symposium on Finite Element Methods in Flow Problems*. Swansea, United Kingdom: UAH Press, 1974, pp. 13–19.

[14] S. Li, Y. Cheng, and Y.-F. Wu, "Numerical manifold method based on the method of weighted residuals," *Computational Mechanics*, vol. 35, no. 6, pp. 470–480, Dec. 2004.

[15] P. Hood and T. C, "Navier-Stokes Equations using Mixed Interpolation," in *International Symposium on Finite Element Methods in Flow Problems*. Swansea, United Kingdom: UAH Press, 1974, pp. 121–132.

[16] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: Its Basis And Fundamentals*. Butterworth-Heinemann, 2005.

[17] A. J. Chorin, "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, vol. 2, no. 1, pp. 12–26, Aug. 1967.

[18] P. Madsen and H. Schäffer, "A Discussion of Artificial Compressibility," *Coastal Engineering*, vol. 53, no. 1, pp. 93–98, Jan. 2006.

[19] J. Donea, "A Taylor-Galerkin method for convective transport problems," *International Journal for Numerical Methods in Engineering*, vol. 20, no. 1, pp. 101–119, Jan. 1984.

[20] J. Donea, H. Laval, S. Giuliani, and L. Quartapelle, "Taylor-Galerkin Method for Time Dependant Transport Problems," in *International Conference on Structural Mechanics in Reactor Technology.* Chicago, Illinois, U.S.A: Commission of the European Communities (EUR-8596) by North-Holland, 2. 2, Amsterdam, Neth, 1983, pp. 77–82.

[21] J. Donea and A. Huerta, *Finite Element Methods for Flow Problems.* John Wiley & Sons, 2003.

[22] B. Roig, "One-step TaylorGalerkin methods for convection diffusion problems," *Journal of Computational and Applied Mathematics*, vol. 204, no. 1, pp. 95–101, July 2007.

[23] T. J. Chung, *Computational Fluid Dynamics.* Cambridge University Press, 2010.

[24] R. Pozo, K. Remington, and A. Lumsdaine, "SparseLib++." [Online]. Available: http://math.nist.gov/sparselib++

[25] R. Pozo, "MV++ Numerical Matrix/Vector Classes in C++." [Online]. Available: http://math.nist.gov/mv++

[26] E. Bayraktar, O. Mierka, and S. Trek, "Benchmark Computations of 3D Laminar Flow Around a Cylinder with CFX , OpenFOAM and FeatFlow," *International Journal of Computatational Science and Engineerinring*, vol. 7, no. 7, pp. 253–266, 2012.

[27] B. R. Munson, D. F. Young, and T. H. Okiishi, *Fundamentals of Fluid Mechanics.* John Wiley, 1998.

[28] J. Ramshaw and V. Mousseau, "Accelerated artificial compressibility method for steady-state incompressible flow calculations," *Computers & Fluids*, vol. 18, no. 4, pp. 361–367, Jan. 1990.

BIOGRAPHICAL STATEMENT

Shrinivas G Apte was born in Pune, India, in 1987. He received his B.E. degree from Sinhgad Academy of Engineering, University of Pune, India, in 2009 and his M.S. degree from The University of Texas at Arlington in 2012, all in the field of Mechanical Engineering.