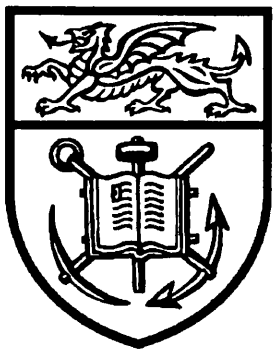
Computational methods for incompressible fluid flows with reference to interface modelling by an extended finite element method.

MacFadden, James

CIVIL AND COMPUTATIONAL ENGINEERING CENTRE

SCHOOL OF ENGINEERING

UNIVERSITY OF WALES SWANSEA



COMPUTATIONAL METHODS FOR INCOMPRESSIBLE FLUID FLOWS WITH REFERENCE TO INTERFACE MODELLING BY AN EXTENDED FINITE ELEMENT METHOD

BY

James MacFadden BEng, MSc

31st March 2006

Thesis submitted to the University of Wales
for the award of the degree of PhD

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Candidate: _____

Date: _____

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Candidate: _____

Supervisor: _____

Date: _____

Statement 2

I hereby consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organizations.

Candidate: _____

Date: _____

Acknowledgements

If I knew then what I know now I would have never embarked on this journey. Oh the naivety of youth! It is only through a lot of pain and sacrifice that this PhD. sits before you finished. I can only hope it you find it helpful.

Many people have supported me in this work and they deserve to be mentioned:

Firstly I would like to thank Professor Đorđe Perić, for giving me the opportunity to study for this PhD, for believing in me and for his unfailing support during this work.

To those studying and working in Swansea University Civil Engineering department, and whom are too numerous to mention. I thank you! Your input has helped sculpture this masterpiece.

To my parents, always there in times of need.

A mi mujer África. Dónde debaria empezar. Su apoyo en tiempos dificiles ha sido indefectible y sin su ayuda este PhD nunca habría visto la luz de día. Usted nunca podría sobreestimar su contribucíon a este PhD[1].

And finally how could I forget to thank my baby daughter, who does not yet know of my achievement. Thank you for not crying at night!

James

[1]To my wife África. Where should I begin. Your support in times of adversity has been unwavering, and without your help this PhD would have never seen the light of day. You could never overestimate your contribution to this PhD.

To María de África

Summary

In this thesis an implicit Semi-Discrete Stabilized eXtended Finite Element formulation has been successfully developed and implemented for laminar Newtonian incompressible fluid flows. In doing so we have contributed to the research into the field of incompressible fluid flows, multiphase flow and fluid-rigid body interaction.

The fluid flows are governed by the incompressible viscous Navier-Stokes equations, using a Finite Element formulation to model the fluid behaviour numerically. A Semi-Discrete time integration scheme was implemented, discretizing in space, leaving the system of ordinary differential equations to be integrated in time. Initially the classical Galerkin method is used to formulate the boundary value problem from the governing equations, however stability issues due to incompressibility and dominant advection terms force the implementation of the stabilized formulation, i.e. SUPG/PSPG. This approach gives greater flexibility in choice of velocity/pressure interpolations, such as equal order functions. The time integration schemes (Generalized α method and Generalized Midpoint rule) were compared and contrasted, with the Generalized α method demonstrating improved convergence. The highly nonlinear form of the governing equations required an implicit iterative solver and the Newton-Raphson procedure was chosen. Several tests were performed throughout the formulation of the boundary value problem to validate the implementation. The result, a robust, efficient and accurate unsteady incompressible Newtonian fluid formulation.

eXtended FEM was introduced by adding terms to the FEM formulation in a Partition of Unity framework. With the addition of complex solution procedures X-FEM was implemented and tested for multiphase and fluid-rigid body interaction, demonstrating the attractive qualities of this method.

Contents

Chapter 1

Introduction.

In this work we will consider the methods for modelling fluid mechanics problems using the Finite Element Method.

1.1 Fluid Mechanics.

Fluid mechanics problems have a considerable influence on modern engineering design and analysis that cannot be underestimated. We find fluid flow related problems in many areas of civil, chemical, mechanical, electrical, and aerospace engineering, such as in bridge design; pipe design; design of combustion engines; design of electronic circuits; and design of aircraft wings.

In fact fluid mechanics problems, including those with discontinuous interfaces, are so commonly encountered in engineering, that there is considerable commercial motivation to understand and the accurately predict fluid flows. However whilst the stimulus and numerical methods have been available for some time, it is only recently that the means to model such flows has become available. The advances in computer technology have moved powerful computational facilities from the domain of large institutions, to the desk of every researcher, and correspondingly there has been an explosion in fluid research. We could even suggest that the advancement in computational fluid mechanics matches that of improvements in computer technology.

The aim of this thesis is to contribute to current research into the field of fluid mechanics; to restrict the scope of this thesis we will only consider laminar Newtonian incompressible fluid flows. In particular this thesis will focus on the numerical methods used to accurately model laminar Newtonian incompressible fluid flows with reference to interface modelling, and develop a robust, accurate

and efficient numerical procedure for its solution.

In the following section there is a brief description of the numerical approach for this thesis, which then concludes with thesis structure.

1.2 Methodology.

1.2.1 Governing Equations.

Research into computational fluid mechanics is still very active and there are many physical relationships that are still not fully understood, such as the behaviour described by the full Navier-Stokes equations governing Newtonian fluids. The full Navier-Stokes equations are so complex that these equations are almost impossible to solve in their present state. Therefore to reduce the computational cost required to obtain an accurate solution it is common to treat certain variables as constants thereby reducing the overall nonlinearity of the equations. We will also only consider laminar problems as fluid problems such as turbulent flow are extremely difficult to model. Therefore in this work the laminar Newtonian incompressible viscous Navier-Stokes are employed as the governing equations.

1.2.2 Numerical Method.

There a various numerical methods available to solve partial differential equations, such as those found in fluid flow problems. These include Finite Difference, Finite Element, Boundary Element, Finite Element and Finite Volume methods. Each of these methods have advantages and disadvantages which make them suitable for certain problems but unsuitable for others. However for the sake of brevity only one method is chosen, namely the Finite Element Method, as the method can be extended in a Partition of Unity framework to include interface capturing functions to create the eXtended Finite Element Method. For further details of these numerical methods there are numerous texts on the subject, see Ferziger and Peric [31], Hughes [42], Zienkiewicz and Taylor [89], Wagner *et al* [86] and references within.

1.2.3 Solution Strategy.

The resulting boundary value problem is a strongly nonlinear mixed Galerkin formulation. The Newton-Raphson procedure is implemented to solve for the

unknowns, using a sparse solver to improve speed efficiency.

1.2.4 Stability and Time Integration.

The stability of a numerical method is of paramount importance, and to this end we employ the SUPG/PSPG stabilizing formulation, see Hughes *et al* [47] and Tezduyar *et al* [82]

For transient flow the temporal domain must also be considered in addition to the discretized spatial domain, provided by the Finite Element formulation. There are two recognised schemes which combine time integration schemes with spatial Stabilized Finite Element formulations, and they are separated into schemes which discretize the spatial domain and apply a discrete time integration scheme and those which discretize both spatial and temporal domains by Finite Elements, i.e. *Semi-Discrete* and *Space-Time* Finite Element Methods respectively, see Dettmer and Peric [27] and references within. The semi-discrete scheme is employed here.

1.2.5 Modelling interfaces.

Generally numerical simulations of the interfaces are based on two distinct methods, which approach the analysis from completely different directions. They are commonly known as Interface Tracking (IT) and Interface Capturing (IC) algorithms.

In Interface Tracking the mesh topology is designed to specifically to match the geometry of the interface e.g. placement of nodes on the actual line of interface. If the interface evolves in time or translates then the mesh will deform with the fluid, thereby allowing for accurate satisfaction of the conservation laws. However this can lead to excessive mesh distortion and computationally expensive remeshing. A commonly used Interface Tracking algorithm is the Arbitrary Lagrangian Eulerian scheme.

The Interface Capturing methods use (commonly) a fixed mesh, which does not necessarily conform with the interface, allowing instead, for the interface to cut elements at any point. These methods are particularly suited for evolving interfaces, the independence of the analysis from the mesh produces a method which is considerably more flexible than the Interface Tracking method. However the detachment of the solution from the mesh can lead to non physical mass transfer, which over long time integration problems can lead to significant errors.

To avoid problems there are various schemes to *sharpen* the solution; some of these appear to be just post processing methods and should be treated with some sceptism. The Interface Capturing method will be employed for this work, using the so called eXtended Finite Element Method, see Wagner *et al* [86], Stolarska *et al* [72], Chessa and Belytschko [17], and Sukumar [74, 73].

1.3 Thesis Structure.

The structure of the thesis is as follows:

Chapter 2: A Mathematical Introduction to Fluids Mechanics.
This chapter introduces the reader to the field of fluid mechanics, and in particular Newtonian fluids and the governing equations that model their behaviour. The governing equations, known as the Navier-Stokes equations are derived from the conservation laws and the constitutive equations to form the basis of the following chapters. We also briefly consider Stokes flow, a sub set of the full Navier-Stokes equations, and derive an analytical solution for the X-FEM chapters.

Chapter 3: Finite Element Methods for Fluid flows.
The General Finite Element method is presented here, prior to its use in Chapter 4.

Chapter 4: Stabilized Finite Elements Methods for fluid flows.
In this chapter we continue the mathematical description of the Finite Element Method with an indepth look at the classical Galerkin method and the problems associated with its use. We consider why the classical Galerkin FEM formulation, when applied to the steady state incompressible Navier-Stokes equations, fails for certain problem types. Two representative model problems i.e. advection-diffusion and Stokes flow equation are employed to demonstrate this. At this point we introduce *Stabilized* methods, as a means to overcome the numerical deficiencies. Finally the two model problems are used to show the improvement in the solution due the stabilized SUPG/PSPG method.

Chapter 5: Implicit Time Integration Schemes for the incompressible Navier-Stokes equations focusing on the Generalized-α Method.
Transient flows are addressed as a continuation to the Finite Element formulations

discussed previously in Chapters 3 & 4. Here various Semi-Discrete temporal integration schemes are developed in conjunction with the spatial Finite Element Method resulting in the *unsteady* incompressible viscous Navier-Stokes equations. Each scheme is assessed for its suitability and a comparison of methods is made through the use of numerical examples. Finally the numerical results and the observations made are discussed in detail.

Chapter 6: Modelling Interfaces in Fluid Mechanic Problems.
In this chapter we are concerned with the accurate modelling of the type of interfaces that can occur in fluid flows, e.g. multiphase fluids or fluid rigid body interaction. Of the various approaches available to model these types of problem, we concern ourselves only with eXtended Finite Element Method (X-FEM) and the process of enrichment that it entails. A simple 1D bar problem used demonstrate the effectiveness of the X-FEM formulation. We conclude this chapter by describing the actual processes of implementation, including local spatial refinement algorithms, search algorithms etc.

Chapter 7: X-FEM - Numerical Results.
In this chapter we present numerical examples to compare and contrasted X-FEM against FEM.

Chapter 2

A Mathematical Introduction to Fluids Mechanics.

The purpose of this chapter is to introduce the reader to the field of fluid mechanics, and in particular Newtonian[1] fluids and the governing equations that model their behaviour.

The chapter will be laid out in the following manner: initially a brief introduction to fluid mechanics is presented, which is followed by a more indepth look at the fundamental concepts that govern the physical systems under consideration. This will include the derivation of the governing equations from first principles, illustrating the assumptions made and mathematical techniques used. The resulting equations will be presented in full. Finally specific aspects of fluid flows particular to this work will be discussed and certain analytical solutions provided, for future reference in later chapters.

The field of fluid mechanics is a very large area of study and it would be impossible to cover every aspect of this area of study in this thesis. There are a great number of published works that are entirely dedicated to the subject of fluid mechanics. Therefore should the reader wish to obtain a greater understanding of the subject they should refer to Currie [24], Chorin and Marsden [20], Meyer [58], Serrin [67], Lamb [52] and Landau and Lifthtz [53] to name but a few.

[1]Newtonian fluids are the most common fluids on this planet, and includes air and water etc.

2.1 Introduction to Fluid Mechanics.

The derivation of the equations governing the motions of fluids is approached from two different directions:

The first approach is to study fluids at molecular level, as fluid matter is discrete at a microscopic level, treating the fluid as a collection of molecules. In this approach the motion is governed by the laws of dynamics and probability. This method is the most accurate depiction of the actual physics of a fluid; at present it is unable to deal with dense gases or liquids.

The second approach assumes that a fluid is a continuous matter, and the behaviour of individual molecules are ignored, i.e. that the properties of the fluid are continuous functions of time and position. Whilst this assumption works well for most fluids, it is unsuitable for a few special cases, e.g. thin gases, for shock, or high altitude near the edge of space, where pressure is extremely low, and in these particular situations the fluid is very discontinuous. For most macroscopic phenomena that occur in nature it is believed that this method is extremely accurate.

The first approach is the most accurate of the two options however the second of the two approaches will be used, as the method is considered to be a good approximation for all but a few specialized cases, none of which will be covered, and it is considerably less complex, as it avoids mathematical difficulties, such as those caused by probability etc. Therefore the purpose of this thesis it will be assumed that the fluid is a continuum, and for fluids the properties necessary to describe fluid behaviour are the *field variables* density, velocity, internal stresses and viscosity.

2.2 Fluid Flow - A Mathematical Overview.

In this thesis we are concerned with modelling fluid flows, and in particular Newtonian fluid flows which possess inertia, viscosity, and which are incompressible. In order to accurately describe the behaviour of Newtonian fluid flows through the medium of mathematics it is important that the governing equations are representative of the fluid response.

The system of equations which describes the full 3D, viscous nature of fluid are often extremely complex due to the influence of viscosity on the flow and are not open to mathematically rigorous analysis. However the inclusion of shear

stresses,[2] resulting from viscosity, into the equations of motion results in a system of equations that are recognized to provide the best approximation for the motion of a Newtonian fluid. This system of equations is known as the Navier-Stokes equations. The Navier-Stokes equations were first developed by Navier in 1831, with a more rigorous proof provided by Stokes in 1845, and have stood without modification since their first introduction. Sadly there is no known general solution of the Navier-Stokes, although the relatively recent advances in numerical methods have been extremely successful in obtaining accurate approximations of the solution.

In this section the full Navier-Stokes equations are derived; it should be noted that in particular this work is restricted to incompressible Newtonian fluids and the final set of equations will reflect this. Firstly the tools required to derive the Navier-Stokes equations are developed followed by the derivation of the equations from the conservation laws and constitutive equations.

2.2.1 Mathematical Methods.

In order to derive the Navier-Stokes equations, the knowledge of certain mathematical theorems and techniques are required. These are explained in reasonable detail within this section, however for a more in-depth understanding, the reader should consult standard mathematical texts, see Donea and Huerta [28], Lamb [52], and Abbott and Brasco [1].

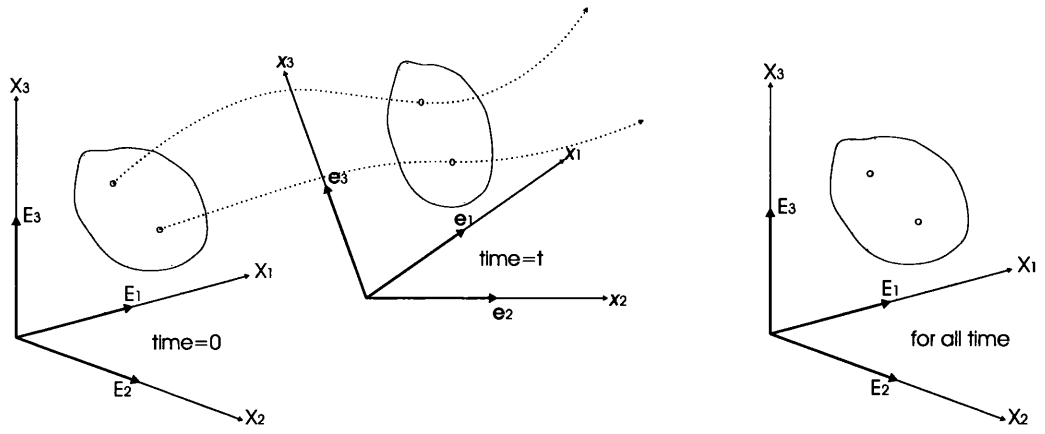
2.2.1.1 Material and Spatial Coordinate systems.

Before we continue we must first consider which coordinate system is most suited to the analysis:

[2] for a small fluid element

Time	Volume	Density
0	V	ρ_0
t	v	ρ

Time	Volume	Density
0	V	ρ_0
t	V	ρ



(a) Material (b) Spatial

Figure 2.1: Material and Spatial Descriptions

There exists two commonly used coordinate systems, see Figure 2.1, that may be used to describe the behaviour of the body whose motion is under consideration:

Material. The Material coordinate system, which is otherwise known as the Lagrangian reference system, moves with *(is attached to)* the particular mass of fluid under consideration. The conservation laws are applied to the mass of fluid, and whilst the global position, volume V and surface S may vary in time, there may not be any fluxes on the surface. For this reference frame the only independent variables are the initial spatial coordinates x_1^0, x_2^0, x_3^0 and time t^0

Spatial. The Spatial coordinate system, also referred to as the Eulerian reference system, is a stationary system. The fluid flows through a control volume and all observations, with regards to the behaviour of the fluid, are confined to within the boundaries of the control volume. The control volume is itself an arbitrary but fixed region of space with volume V and surface S remaining constant in time. The independent variables for this framework are the spatial coordinates x_1, x_2, x_3 and time t

2.2.1.2 Coordinate Systems for the Continuum Model.

Let β represent any of the continuum properties mentioned previously. The essential mathematical simplification of the continuum model is that the average properties in a volume element ∂V surrounding a point are considered to represent the point itself. This is equivalent to a fictitious continuous material, locally homogeneous, with a value of β at every point within the fluid at each instant of time. Thus the two coordinate systems, Material and Spatial, can be represented as:

Material (*Lagrangian*)

$$\beta = \beta(\boldsymbol{x}_p, t) \qquad \text{or} \qquad \beta = \beta(x_{1p}, x_{2p}, x_{3p}, t) \tag{2.1}$$

Spatial (*Eulerian*)

$$\beta = \beta(\boldsymbol{x}, t) \qquad \text{or} \qquad \beta = \beta(x_1, x_2, x_3, t) \tag{2.2}$$

2.2.1.3 Material Derivative.

The material Derivative is also known as the *substantial derivative*. Let β once again represent the value of some fluid property, i.e. velocity, density etc., and x_1,x_2 x_3 the coordinates of a Material point using the Spatial coordinate system. Calculate the rate of change of the value of β at this Material point[3].

$$d\beta = \frac{\partial \beta}{\partial x_1} dx_1 + \frac{\partial \beta}{\partial x_2} dx_2 + \frac{\partial \beta}{\partial x_3} dx_3 + \frac{\partial \beta}{\partial t} dt \tag{2.3}$$

dx_1, dx_2, dx_3 can be related to dt by:

$$dx_1 = u_1 dt \qquad dx_2 = u_2 dt \qquad dx_3 = u_3 dt \tag{2.4}$$

where $\boldsymbol{u}$ is the velocity field. At this point we introduce the term to denote differentiation following the motion of the fluid $\frac{D}{Dt}$. Substituting (2.4) into (2.3) the Material derivative term $\frac{D\beta}{Dt}$ in relation to the Spatial derivative is obtained as

$$\underbrace{\frac{D\beta}{Dt}}_{\text{Material Derivative}} = \underbrace{\frac{\partial \beta}{\partial x_1} u_1 + \frac{\partial \beta}{\partial x_2} u_2 + \frac{\partial \beta}{\partial x_3} u_3 + \frac{\partial \beta}{\partial t}}_{\text{Substantial Derivative}} \tag{2.5}$$

Equ (2.5) can be rewritten in vector form as:

$$\boxed{\frac{D\beta}{Dt} = \frac{\partial \beta}{\partial t} + \boldsymbol{u}(\nabla \cdot \beta)} \tag{2.6}$$

[3]This task is in concept essentially Lagrangian, but completed using the Eulerian reference system

2.2.1.4 Gauss's Theorem.

Gauss's theorem for the *scalar field* β states that:

$$\boxed{\int_{V(t)} \nabla \beta dV = \int_{S(t)} \beta \boldsymbol{n} dS} \tag{2.7}$$

where $\boldsymbol{n}$ is the unit vector normal to the surface.

2.2.1.5 Divergence Theorem.

The Divergence Theorem is an extension of Gauss's theorem for use with vector functions. The resulting equation for the *vector field* $\boldsymbol{\beta}$ is as follows:

$$\boxed{\int_{V(t)} \nabla \cdot \boldsymbol{\beta} dV = \int_{S(t)} \boldsymbol{\beta} \cdot \boldsymbol{n} dS} \tag{2.8}$$

This theorem is mainly used in order to move from volume integrals, to the easier to interpret surface integrals.

2.2.1.6 Control Volumes (CV).

The Control Volume[4] chosen is arbitrary, i.e. it has arbitrary limits of integration. The resulting equation for the *field variable* β is:

$$\int_V \mathcal{L}[\beta] dV = 0 \tag{2.9}$$

where $\mathcal{L}$ is a differential operator. In order to satisfy (2.9) the following differential form must also be true:

$$\boxed{\mathcal{L}[\beta] = 0} \tag{2.10}$$

This statement is only true if the CV is arbitrary.

2.2.1.7 Reynolds Transport Theorem.

The choice of which coordinate system to use is generally dependent on the type of problem, i.e. for fluid flows the Spatial system is more commonly used, however in solid mechanics both are used with equal success. However irrespective of which coordinate system is used it is often necessary, during formulation of the governing equations, to cross between the Material and Spatial systems. In such situations it is common to use the Reynolds Transport Theorem. This general

[4]The type of CV chosen is completely independent of the choice of the coordinate system

theorem relates derivatives in the Material framework to those in the Spatial framework.

The Reynolds Transport Theorem is a special case of the Leibnitz[5] Theorem, and is based upon the assumption that the arbitrary velocity of the moving volume is actually the velocity of a Material point:

$$\boldsymbol{w} = \boldsymbol{u} \tag{2.11}$$

Therefore the volume is no longer arbitrary and the volume becomes a Material Volume. The Reynolds Transport Theorem reads:

$$\boxed{\frac{D}{Dt}\int_V \gamma dV = \int_V \frac{\partial \gamma}{\partial t} dV + \int_V \nabla \cdot (\gamma \boldsymbol{u}) dV} \tag{2.12}$$

2.3 The Navier-Stokes equations - Derivation of the governing equations.

In this section the Navier-Stokes equations are derived from the conservation laws and constitutive equations is shown, using mathematical methods discussed previously. The equations are derived in tensor form and it is assumed that the reader has a good grasp of tensor algebra.

2.3.1 Conservation Laws

The equations that govern the motion of fluids follow the principles laid down by the Conservation Laws[6] - Mass, Momentum and Energy. The Mass, Momentum and Energy Conservation Laws are as follows:

Principle of the Conservation of Mass.
The rate of change of the mass in a Material region is zero.

Principle of the Conservation of Momentum.
The rate of change of momentum of a Material region is equal to the forces applied to it (*Newton's second law of motion*).

[5]For completeness the derivation of the Leibnitz's theorem is presented in Appendix A.

[6]The Conservation laws, have been observed to be valid when the length scales are sufficiently large, as to contain a large quantity of molecules.

Principle of the Conservation of Energy.
The rate of change of energy within a Material region is equal to the rate that energy is received by the region due to the total work done and heat (*Application of the First law of thermodynamics to fluid flows*).

The above laws will be systematically converted into mathematical form below.

2.3.2 Principle of Conservation of Mass

The mathematical representation of the Principle of Conservation of Mass can be written as:

$$\frac{d}{dt}m = 0 \tag{2.13}$$

where m is the mass of the body. The mass enclosed by Material volume, i.e. a Material region, can be described by the local value of density:

$$m = \int_{V(t)} \rho dV \tag{2.14}$$

where ρ is the density. Substituting (2.14) into (2.13):

$$\frac{D}{Dt}\int_{V(t)} \rho dV = 0 \tag{2.15}$$

Applying the Reynolds's Transport Theorem (2.12), derived in an earlier section, the Material derivative (2.15) can be converted into its Spatial form.

$$\frac{D}{Dt}\int_{V} \rho dV = \int_{V} \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u})\right] dV \tag{2.16}$$

The RHS of (2.16) is commonly known as the **Continuity Equation**. On the assumption that the volume is arbitrary, see Section 2.2.1.6, it follows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{2.17a}$$

$$\underbrace{\frac{\partial \rho}{\partial t} + \boldsymbol{u} \cdot \nabla \rho}_{\substack{\text{Substantial} \\ \text{Derivative (2.6)}}} + \rho(\nabla \cdot \boldsymbol{u}) = 0 \tag{2.17b}$$

$$\boxed{\frac{D\rho}{Dt} + \rho(\nabla \cdot \boldsymbol{u}) = 0} \tag{2.17c}$$

If the fluid is incompressible or nearly incompressible the Material derivative in (2.17c) will be equal to zero. If the first term of (2.17c) is zero then it follows that the second term must also be zero. The Continuity equation for an incompressible fluid is reduced to:

$$\boxed{\nabla \cdot \boldsymbol{u} = 0} \tag{2.18}$$

2.3.3 Principle of Conservation of Momentum.

The Principle of Conservation of Momentum states that the rate of change of momentum will be equal to the forces applied. There are two distinct groups of forces:

Body forces.

Forces due to external influences - such as *gravitational & electromagnetic* forces.

Surface forces.

Forces of stress - this is a consequence of the rest of the continuum acting on the surface of the body, includes *viscous & pressure* forces.

The body forces are defined as forces per unit volume dV - to determine the total body force $\rho \boldsymbol{g}$ is integrated over the whole volume V. The surface forces are due to stresses experienced on the surface of the body and are defined per unit surface area dS - these are called traction forces $\boldsymbol{h}$ and are integrated over the surface of the volume S.

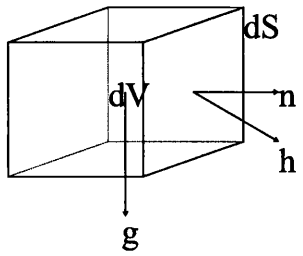


Figure 2.2: forces acting on a fluid element

where $\boldsymbol{g}$ is the acceleration due to gravity. The mathematical representation of the Principle of Conservation of Momentum can be written as:

$$\frac{D}{Dt}\int_{V(t)} \rho \boldsymbol{u} dV = \int_{S(t)} \boldsymbol{h} dS + \int_{V(t)} \rho \boldsymbol{g} dV \tag{2.19}$$

The forces acting on the surface of the body can be written in terms of the stress tensor $\boldsymbol{\sigma}$[7] and the unit vector normal to the surface $\boldsymbol{n}$

$$\boldsymbol{h} = \boldsymbol{\sigma} \cdot \boldsymbol{n} \tag{2.20}$$

[7]There are nine components of stress at any given point, one normal component and two shear components per coordinate plane" - *I.G.Currie, Fundamentals of Fluid Mechanics.*
σ_{11}, σ_{22}, σ_{33} normal components and σ_{12}, σ_{13}, σ_{21}, σ_{23}, σ_{31}, σ_{32} shear components.

- By substituting (2.20) into the surface integral on the RHS of (2.19), and applying the Divergence theorem, 2.2.1.5, the resultant term will be $\int_{V(t)} \nabla \cdot \boldsymbol{\sigma} dV$.

- By applying the Reynolds Transport Theorem to the Material derivative on the LHS of (2.19), it will be transformed into its Spatial form.

On completion of these two steps (2.19) can be rewritten using only volume integrals and Spatial terms. It follows that:

$$\int_{V(t)} \left[\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) \right] dV = \int_{V(t)} \nabla \cdot \boldsymbol{\sigma} dV + \int_{V(t)} \rho \boldsymbol{g} dV \tag{2.21}$$

Equ (2.21) can be rearranged into a form which resembles (2.9). Based on the assumption that the volume is totally arbitrary, then the following equation is also true:

$$\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{g} \tag{2.22}$$

By using the common tensor identities we note that:

$$\nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = (\boldsymbol{u} \cdot \nabla)\rho \boldsymbol{u} + (\nabla \cdot \boldsymbol{u})\rho \boldsymbol{u} \tag{2.23}$$

Therefore (2.22) can be expanded as:

$$\frac{\partial \rho}{\partial t}\boldsymbol{u} + \frac{\partial \boldsymbol{u}}{\partial t}\rho + (\boldsymbol{u} \cdot \nabla)\rho \boldsymbol{u} + (\nabla \cdot \boldsymbol{u})\rho \boldsymbol{u} = \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{g} \tag{2.24}$$

which can be rearranged in order to make the equation more recognizable by its parts:

$$\Big[\underbrace{\frac{\partial \rho}{\partial t} + (\nabla \cdot \boldsymbol{u})\rho}_{\text{Continuity Equation}} \Big] \boldsymbol{u} + \Big[\underbrace{\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}}_{\text{Substantial Derivative}} \Big] \rho = \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{g} \tag{2.25}$$

Finally the equation governing the change of momentum in a body of fluid, written in the Spatial coordinate system is as follows:

$$\boxed{\rho \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \right] = \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{g}} \tag{2.26}$$

2.3.4 Principle of Conservation of Energy.

Recall that the first law of thermodynamics, states:

Definition 2.1: The time rate of change of a material region's internal and kinetic energy is equal to the rate of heat transferred to the material region less the rate

of work done by the material region.

This can be mathematically stated as:

$$\frac{DE}{Dt} = \frac{DQ}{Dt} - \frac{DW}{Dt} \tag{2.27}$$

where $\frac{DE}{Dt}$ accumulation of internal energy

$\frac{DW}{Dt}$ rate of work done by external forces

$\frac{DQ}{Dt}$ heat transferred to the material volume

The following three blocks of equations will been used to expand the terms found in (2.27).

2.3.4.1 Accumulation of Internal Energy Term.

$$\frac{DE}{Dt} = \frac{D}{Dt}\int_{V(t)} \rho E dV \tag{2.28a}$$

where $E = e + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}$ (2.28b)

$E \rightarrow$ Total energy per unit mass

$e \rightarrow$ Internal energy per unit mass

$\frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u} \rightarrow$ Kinetic energy per unit mass

2.3.4.2 Work Done Term.

$$\begin{aligned} DW = D\boldsymbol{x}\cdot\boldsymbol{f} \quad &\Longrightarrow \frac{DW}{Dt} = \frac{D\boldsymbol{x}}{Dt}\cdot\boldsymbol{f} = \boldsymbol{u}\cdot\boldsymbol{f} \\ &= \int_{V(t)} \boldsymbol{u}\cdot\rho\boldsymbol{g}dV + \int_{S(t)} \boldsymbol{u}\cdot\boldsymbol{h}dS \end{aligned} \tag{2.29}$$

where $\boldsymbol{f} \rightarrow$ Force

$\boldsymbol{u} \rightarrow$ Velocity

$\rho\boldsymbol{g} \rightarrow$ External Body forces, see Figure 2.2

$\boldsymbol{h} \rightarrow$ Surface Traction forces, see Figure 2.2

2.3.4.3 Heat Transfer Term.

$$\frac{DQ}{Dt} = -\int_{S(t)} \boldsymbol{q} \cdot \boldsymbol{n} dS \tag{2.30a}$$

where $+/- \rightarrow$ Outflow/Inflow through the surface of the body

The integral of the conservation of energy for a moving volume can be written by combining (2.28a), (2.29), & (2.30a) into (2.27) to obtain (2.31a)[8] & (2.31b):

$$\left.\begin{matrix}\text{Accumulation of} \\ \text{Internal energy}\end{matrix}\right\{ \frac{D}{Dt}\int_{V(t)} \rho E dV = \int_{V(t)} \boldsymbol{u} \cdot \rho \boldsymbol{g} dV + \int_{S(t)} \boldsymbol{u} \cdot \boldsymbol{h} dS \Big\} \text{Work Done} \tag{2.31a}$$

$$-\int_{S(t)} \boldsymbol{q} \cdot \boldsymbol{n} dS \Big\} \text{Heat Transfer} \tag{2.31b}$$

As with the previous principles, i.e. Conservation of Mass & Momentum, the Reynolds Transport Theorem and the Divergence Theorem will transform the Material derivatives into Spatial derivatives, and convert surface integrals into volume integrals respectively.

2.3.4.4 Material Derivative to Spatial Derivative.

Applying the Reynolds Transport Theorem to (2.28a).

$$\frac{D}{Dt}\int_{V(t)} (\rho E) dV = \int_{V(t)} \frac{\partial}{\partial t}(\rho E) dV + \int_{V(t)} \nabla \cdot (\rho E \boldsymbol{u}) dV \tag{2.32a}$$

$$= \int_{V(t)} E\left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u})\right] dV \Big\} \text{Continuity Equation} \tag{2.32b}$$

$$+ \int_{V(t)} \rho\left[\frac{\partial E}{\partial t} + \boldsymbol{u} \cdot \nabla E\right] dV \Big\} \text{Substantial Derivative} \tag{2.32c}$$

$$= \int_{V(t)} \rho \frac{DE}{Dt} dV \tag{2.32d}$$

[8] Work done on fluid by the force is the negative of work done by the fluid

2.3.4.5 Surface Integrals to Volume Integrals.

Using the Divergence theorem on all surface integrals in (2.31).

$$\int_{S(t)} \boldsymbol{u} \cdot \boldsymbol{h} dS = \int_{V(t)} \nabla \cdot (\boldsymbol{\sigma u}) dV \tag{2.33a}$$

$$\int_{S(t)} \boldsymbol{q} \cdot \boldsymbol{n} dS = \int_{V(t)} \nabla \cdot \boldsymbol{q} dV \tag{2.33b}$$

The total energy per unit mass E is the sum of two terms, i.e. kinetic energy and internal energy. E will be replaced in all the following equations by (2.28b).

The assumption that the volume is arbitrary, as previously applied at (2.21), allows the rule defined by (2.9) to be used on the following equation. By applying the changes presented by (2.33) & (2.32) to (2.31) gives:

$$\rho \frac{D}{Dt}(e + \frac{1}{2}\boldsymbol{u} \cdot \boldsymbol{u}) = \boldsymbol{u} \cdot \rho \boldsymbol{g} + \nabla \cdot (\boldsymbol{\sigma u}) - \nabla \cdot \boldsymbol{q} \tag{2.34}$$

Expanding both sides of (2.34)

$$\begin{aligned} &\frac{\partial}{\partial t}(\rho e + \frac{1}{2}\rho \boldsymbol{u} \cdot \boldsymbol{u}) + \nabla \cdot \left[\left(\rho e + \frac{1}{2}\rho \boldsymbol{u} \cdot \boldsymbol{u} \right) \boldsymbol{u} \right] = \\ &\frac{\partial(\rho e)}{\partial t} + \nabla \cdot (\rho e \boldsymbol{u}) + \rho \boldsymbol{u} \cdot \frac{\partial(\boldsymbol{u})}{\partial t} + \rho \boldsymbol{u} \cdot (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \\ &\qquad \boldsymbol{u} \cdot \rho \boldsymbol{g} + \boldsymbol{u} \cdot (\nabla \cdot \boldsymbol{\sigma}) + \boldsymbol{\sigma} : \nabla^T \boldsymbol{u} - \nabla \cdot \boldsymbol{q} \end{aligned} \tag{2.35}$$

The first two terms on the LHS of (2.35) can be further expanded. Of the resulting four terms, two cancel as they form the Continuity equation:

$$\begin{aligned} &\rho \frac{\partial e}{\partial t} + \rho \boldsymbol{u} \cdot \nabla e + \rho \boldsymbol{u} \cdot \frac{\partial \boldsymbol{u}}{\partial t} + \rho \boldsymbol{u} \cdot (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \\ &\qquad \boldsymbol{u} \cdot \rho \boldsymbol{g} + \boldsymbol{u} \cdot (\nabla \cdot \boldsymbol{\sigma}) + \boldsymbol{\sigma} : \nabla^T \boldsymbol{u} - \nabla \cdot \boldsymbol{q} \end{aligned} \tag{2.36}$$

The equation derived from the Principle of Conservation of Momentum, (2.26), can be rearranged in terms of the time derivative of velocity (see below).

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} = -\rho(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{g} \tag{2.37}$$

Equ (2.37) can be used to simplify (2.35)

$$\rho \frac{\partial e}{\partial t} + \rho \boldsymbol{u} \cdot \nabla e = \boldsymbol{\sigma} : \nabla^T \boldsymbol{u} - \nabla \cdot \boldsymbol{q} \tag{2.38}$$

$\boldsymbol{\sigma}$ is symmetrical and therefore the term $\boldsymbol{\sigma} : \nabla^T \boldsymbol{u}$ can be rewritten as $\boldsymbol{\sigma} : \nabla \boldsymbol{u}$. The final mathematical form of the Principle of Conservation of Energy is:

$$\boxed{\rho \frac{\partial e}{\partial t} + \rho \boldsymbol{u} \cdot \nabla e = \boldsymbol{\sigma} : \nabla \boldsymbol{u} - \nabla \cdot \boldsymbol{q}} \tag{2.39}$$

2.3.5 Constitutive equations.

Kinematic equations and generic principles, e.g. the derived equations for the conservation laws, are very general formulations which broadly describe the response of material in question, however in order for them to be representative of the type of problem being studied it is important that the chosen system provides the correct response prediction. This requires, in short, the governing equations to be tailored to suit and be applicable to the particular material model, and provide a model of certain relations which are particular to the material and analysis, e.g. large strain material models are different to small strain material models. The constitutive equations are the manner in which material models are integrated with the kinematic equations to form the governing equations.

For the fluid flow case, there are various material models available, e.g. Newtonian, non-Newtonian, incompressible, compressible etc which allow the users to tailor the kinematic response to the particular fluid and flow conditions. The particular material model used throughout this thesis was developed to emulate viscous Newtonian incompressible flow. When the Newtonian flow constitutive equations are incorporated with the kinematic equations the resulting governing equation is commonly known as the Navier-Stokes equations.

The material model introduces further unknowns and they include the stress tensor $\boldsymbol{\sigma}$ with the deformation rate tensor $\boldsymbol{\varepsilon}$, and the heat-flux gradient $\boldsymbol{q}$ with the temperature gradient ∇T. These added unknowns, with associated equations are introduced below.

2.3.5.1 Stress-Strain rate constitutive equation.

For a Newtonian fluid[9] it is assumed that the stress tensor and the strain rate tensor are linearly related[10]. The stress-strain rate relationship is given as:

$$\boxed{\boldsymbol{\sigma} = -p\boldsymbol{I} + \lambda(\nabla \cdot \boldsymbol{u})\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u})} \tag{2.40}$$

where

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) = \nabla^S \boldsymbol{u} = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla^T \boldsymbol{u}) \tag{2.41}$$

[9]Newtonian fluids are the most commonly found fluids on this planet and they include water, air etc.

[10]All fluids which do not demonstrate linearity between the rate of deformation and stress, i.e. have nonlinear stress-strain relationship, are classified as a Non-Newtonian fluid.

and μ is the dynamic viscosity, λ is the second viscosity coefficient, $\boldsymbol{I}$ is used to denote the identity tensor and p the pressure. $\nabla^S \boldsymbol{u}$ is called the rate of deformation (or strain rate) tensor [11]. For the case of incompressible fluids, as shown earlier by (2.18) $\nabla \cdot \boldsymbol{u} = 0$ and consequently (2.40) can be written as:

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) \tag{2.42}$$

2.3.5.2 Heat flux - temperature gradient constitutive equation.

The flux term in the energy equation (2.30) is a consequence of heat transfer due to conduction, and is usually governed by Fouriers law, i.e. Fourier's law of heat conduction. Fourier's law states that heat flux is proportional to the local temperature gradient, and is given as:

$$\boxed{\boldsymbol{q} = \kappa \nabla T} \tag{2.43}$$

where κ is thermal conductivity.

2.4 The Full Navier-Stokes Equations.

Combined, the equations (2.43), (2.40), (2.39), (2.26) and (2.18) form what is known as the Navier-Stokes equations for incompressible fluid flow. For convenience they have been reproduced below:

$$\rho\left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}\right] = \nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{g} \tag{2.44a}$$

$$\frac{\partial \rho}{\partial t} + \rho(\nabla \cdot \boldsymbol{u}) = 0 \tag{2.44b}$$

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \lambda(\nabla \cdot \boldsymbol{u})\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) \tag{2.44c}$$

$$\rho\frac{\partial e}{\partial t} + \rho \boldsymbol{u} \cdot \nabla e = \boldsymbol{\sigma} : \nabla \boldsymbol{u} - \nabla \cdot \boldsymbol{q} \tag{2.44d}$$

$$\boldsymbol{q} = \kappa \nabla T \tag{2.44e}$$

2.5 The Incompressible Navier-Stokes equations.

Previously we have derived the full Navier-Stokes equations, see Sections 2.3 and 2.4. The reader will note that the full Navier-Stokes equations are extremely complex, reliant on many different variables, each increasing the nonlinearity of the overall equation. Such is the complexity that it is almost impossible to solve

[11] The operator ∇^S represents the symmetric tensor

this equation (2.44) in its present state. Therefore to reduce the computational cost required to obtain an accurate solution it is common to treat certain variables as constants.

The incompressible Navier Stokes equations model flows where the density is constant, however to reduce the nonlinearities, viscosities and thermal conductivity are also treated as constants i.e. $\rho\,\mu\,\lambda\,\kappa = \text{constant}$[12]. The incompressibility condition is usually connected to the Mach number[13] where it is assumed that for a small Mach number the variation in density is small, in comparison to the pressure variation, which is large. The justification lies with the energy equation, and a more indepth explanation can be found in Peyret and Taylor [64] and Cramer [23].

Remark 2.1: Incompressibility Constraint.
In Section 4.2 we will note that the incompressibility constraint requires special consideration when developing the Finite Element formulation. Incorrect consideration of incompressibility will lead to spurious and nonphysical oscillations in the pressure field.

When incompressibility is applied to (2.44b) it becomes:

$$(\nabla \cdot \boldsymbol{u}) = 0 \tag{2.45a}$$

$$\Rightarrow \boldsymbol{\sigma} = -p\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) \tag{2.45b}$$

This step results in the uncoupling of the energy equation[14] from the Navier-Stokes equations, allowing for a solution to be reached without regard to the temperature variation[15]. The temperature can be solved once $\boldsymbol{u}$ & p are found should the temperature be required.

We can also ignore that pressure p is a thermodynamic function of density and temperature because density is assumed constant and the energy equation is uncoupled from the system of equations. Pressure p can be treated as just another variable.

[12]There is no *a priori* reason to assume that constant density solutions correspond to constant $\mu\,\lambda\,\kappa$ solutions

[13]Flow speed divided by the speed of sound in the material

[14]This only possible due to constant viscosities.

[15]$\boldsymbol{u}$ & p can solved independently of the energy equation (2.44e), using (2.44a), (2.45a) & (2.45b), i.e. (2.44e) has been uncoupled.

2.5.1 Classification of Flow - Reynolds number, Laminar and Turbulent flows.

In order to determine the principles which are applicable to the particular problem being studied it is important to classify the flow type. The various types of flow lend themselves to being categorized in the following manner: *Laminar* or *Turbulent*, *Steady* or *Unsteady*, *Rotational* or *Irrotational* and *Uniform* or *Non-uniform*. For the purposes of this thesis, only the first two categories are of interest.

2.5.1.1 Reynolds number, and Laminar and Turbulent flow.

Laminar flow.

All the fluid particles proceed along parallel paths , with no transverse component of velocity.

Turbulent flow.

The progression of the fluid particles is irregular, with individual particles being subject to fluctuating transverse velocities so that the motion of the fluid is eddying. One consequence of the cross current velocities is a more uniform velocity distribution when compared with laminar flows. This is due to the interchange of momentum between fast moving fluid particles near the centre and the slower moving fluid particles near the walls. This also causes large energy loss in the flow.

The criterion which distinguishes laminar flow from turbulent flow, is the fundamental characteristic of flow and is given as the ratio between the inertial forces with viscous forces, universally known as the Reynolds number, a dimensionless term which is written as:

$$\mathrm{Re} = \frac{UD\rho}{\mu} \tag{2.46}$$

where Re, U, D, ρ and μ are the Reynolds number, characteristic velocity, characteristic length, density, and kinematic viscosity respectively.

It is possible to write a dimensionless form of the Navier-Stokes equations with Re as the only parameter[16], highlighting the characterizing effect of the Reynolds number on fluid flows. It is also apparent from Equ (2.46) that low

[16]This is achieved by non-dimensionalizing the velocity, pressure, spatial and temporal terms and replacing the kinematic viscosity by the inverse of Re

Reynolds number flows are dominated by viscous forces whereas high Reynolds number flows are inertia dominated.

It is observed from experimental data that low Reynolds number flows are *laminar* and high Reynolds number flows are generally *turbulent*, with the transition occurring at certain critical Reynolds number range. The point of transition is often denoted as Re_{crit}, and is completely dependent on the problem under consideration. It is generally accepted for flows through circular pipes, that the flow can be considered laminar for a $\mathrm{Re} < 2000$ and fully turbulent for a $\mathrm{Re} > 4000$.

Laminar flows are fully understood and for simple boundary conditions the velocity distribution can be analyzed by analytical means, however due to irregular nature turbulent flow has defied rigorous mathematical treatment, and solutions of turbulent flow rely on empirical or semi empirical relationships.

Because of the nature of laminar flows it is often appropriate to model laminar flows using a 2D framework, and laminar flow numerical simulations can give accurate results. In contrast turbulent flows contain within them 3D structures which are extremely complex and erratic, and on a particle scale appear to be completely random. Whilst it is still theoretically possible to model turbulent flow using the general Navier-Stokes equations, in practice this is rarely the possible. Modelling such a complex phenomenon would require such a fine discretization of the domain that the subsequent model would be too large to solve using current software and technology. It is for this reason that it is common practice to modify the governing equations to solve for mean flow and express turbulent effects by means of turbulent viscosity and use wall functions to model near wall effects. An in-depth review of turbulence research can be found in Ferziger and Peric [31].

Turbulence is not considered in this work, hence all numerical examples involve fluid flows with a Reynolds number in the laminar range.

2.5.1.2 Steady and Unsteady Flow.

Steady.

A flow is said to be steady when the conditions at any point are constant with respect to time. This definition leads to the conclusion that turbulent flow can not be truly steady. However it is convenient to assume that the main criterion is the general fluid motion, whilst erratic fluctuations are considered secondary effects. This allows the assumption that a flow with a constant discharge is steady.

Unsteady.

As a corollary it follows that a flow is unsteady if the conditions vary with time.

For many practical problems the flows can be considered to be steady. This is an advantage, as the time variable in unsteady flow complicates the analysis considerably, and in some cases unsteady flow can be reduced to steady flow by virtue of the principle of relative motion. In this work both steady and unsteady flows are considered.

2.5.2 Incompressibility and Convection.

The numerical solution of the Navier-Stokes equations for incompressible fluid flow is not a straightforward process, and before introducing the Finite Element techniques that will provide the basis of all numerical methods in this thesis, it is prudent to briefly mention the difficulties faced. In this section the difficulties which need to be addressed are discussed.

2.5.2.1 Convective terms.

Convection also poses a problem for those wishing to numerically solve the Navier-Stokes equations, due to the nonlinear and non symmetric convection terms found in the momentum equations, see (2.26). This problem becomes more apparent with an increasing flow Reynolds number, as high Reynolds number flows are convection dominated, and stabilization techniques must be applied to the governing equations in order to obtain a meaningful result. See Section 4.1.

2.5.2.2 Incompressibility.

The continuity equation for the incompressibility condition, see (2.17c), states that the velocity field must be divergence free. The constraint leads to certain numerical difficulties which must be addressed correctly, otherwise they become a source of numerical instability, adversely effecting the entire solution.

The requirement for divergence free velocity, paired with a pressure unknown that is not related to a constitutive equation, allows pressures inclusion[17] in the momentum equation to be used to satisfy the incompressibility constraint. Thus pressure becomes, in essence, a Lagrangian multiplier to enforce the incompressibility constraint. This leads to a coupling between the velocity and pressure

[17]introducing an additional degree of freedom

unknowns and it is the velocity/pressure relationship which is the origin of the difficulties faced.See Section 4.2.

There are various methods available which allow the difficulties encountered in incompressibility and convection to be circumvented. These methods are in the realm of Finite Elements and their implementation, and as such the actual mechanics of circumventing the issues mentioned above are discussed in a later and more appropriate section.

2.5.3 Stokes flow.

Stokes flow is a sub set of the full Navier-Stokes equations, in which the governing equations are written neglecting the time-dependent and convective terms, otherwise known as inertia terms. In this section Stokes flow will be covered in some depth as Stokes flow and the ability to produce an exact solution for this flow type plays an important part in chapters 6 & 7 of this thesis.

2.5.3.1 Stokes Flow - Analytical Solution - Flow past a rigid cylinder.

Flow round bodies at very small Reynolds numbers (*creeping motion*) is of interest to those studying flow around small particles. The flow is entirely laminar, no separation occurs and there is no disturbed wake. Stokes developed a mathematical solution for this case, in which it is assumed that viscous forces are large in comparison with the inertial forces. It should be noted however, that this assumption is only really applicable a finite distance from the particle, as at distances very far from the particle the velocity gradients are small and viscous forces becomes negligible. The effect of this will be discussed in detail at a later stage.

The analytical solution for this particular phenomenon, i.e. 2D flow past a particle, which is considered to be a rigid cylinder of infinite length - *normal to the axes*, in an infinite domain, will be used in a future chapter, therefore it is of interest and is briefly reviewed here. The Stokes equations for an incompressible, very viscous fluid are:

$$\nabla \cdot \boldsymbol{\sigma} = 0 \qquad \forall \boldsymbol{x} \in \Omega \tag{2.47}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \forall \boldsymbol{x} \in \Omega \tag{2.48}$$

or expanding (2.47):

$$-\nabla\rho + \mu\nabla^2\boldsymbol{u} = \mathbf{0} \tag{2.49}$$

The analytical solution will be found in terms of the stream function ψ written using planar polar coordinates r and θ. The velocity $\boldsymbol{u}$ is given as:

$$\boldsymbol{u} = \frac{1}{r}\frac{\partial\psi}{\partial\theta}\boldsymbol{e}_r - \frac{\partial\psi}{\partial r}\boldsymbol{e}_\theta \tag{2.50}$$

where $\boldsymbol{e}_r$ and $\boldsymbol{e}_\theta$ are unit vectors. Velocity in the coordinate directions are:

$$u_r = \frac{1}{r}\frac{\partial\psi}{\partial\theta}, \quad u_\theta = -\frac{\partial\psi}{\partial r} \tag{2.51}$$

The stream function ψ is given:

$$\nabla^4\psi = 0 \quad \text{or} \quad \nabla^2(\nabla^2\psi) = 0 \tag{2.52}$$

where

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial\theta^2} \tag{2.53}$$

Taking the centre of the cylinder to be the origin, where the velocity U far from the cylinder is uniform, the boundary conditions are:

$$u_r = u_\theta = 0, \qquad \text{at} \quad r = a \tag{2.54}$$

$$\Rightarrow \frac{\partial\psi}{\partial\theta} = 0, \quad \frac{\partial\psi}{\partial r} = 0 \tag{2.55}$$

$$u_r = U\cos\theta, \quad u_\theta = -U\sin\theta, \qquad \text{at} \quad r = \infty \tag{2.56}$$

$$\Rightarrow \frac{\partial\psi}{\partial\theta} = Ur\cos\theta, \quad \frac{\partial\psi}{\partial r} = U\sin\theta \tag{2.57}$$

where U is magnitude of the velocity along the axis $\theta = 0$. Based on the boundary conditions at infinity:

$$\psi = rU\sin\theta \quad \text{as} \quad r \to \infty \tag{2.58}$$

Concluding that the solution of Equ (2.52) will resemble Equ (2.58):

$$\psi(r,\theta) = UF(r)\sin\theta \tag{2.59}$$

Therefore Equ (2.52) reduces to:

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) - \frac{1}{r^2}\right]^2 F(r) = 0 \tag{2.60}$$

Integrating Equ (2.59) gives the solution for $\psi(r,\theta)$ in terms of A - D:

$$\psi(r,\theta) = \left(Ar^3 + Br\ln r + Cr + \frac{D}{r}\right)\sin\theta \tag{2.61}$$

where A - D are constants which are determined through the use of the prescribed boundary conditions. However there are no linear combination of ψ_1 & ψ_2 as $r \to \infty$ and this results in two different solutions. With no unique solution, the situation is known as Stokes Paradox.

It is possible to arrive at a useful solution by applying the exact wall boundary conditions and determining the unknowns using a better approximation to that provided by Stokes equations. This leads to the solution:

$$u_r = \frac{a^2}{2r}\sin\theta - \frac{1}{2}r - r\ln\left(\frac{r}{a}\right) \tag{2.62}$$

$$u_\theta = \frac{a^2}{2r}\cos\theta - \frac{1}{2}r + r\ln\left(\frac{r}{a}\right) \tag{2.63}$$

which can be rewritten for the cartesian coordinate system, i.e. u_x & u_y where

$$u_x = u_r\cos\theta - u_\theta\sin\theta \tag{2.64}$$

$$u_y = u_r\sin\theta + u_\theta\cos\theta \tag{2.65}$$

Finally we arrive at:

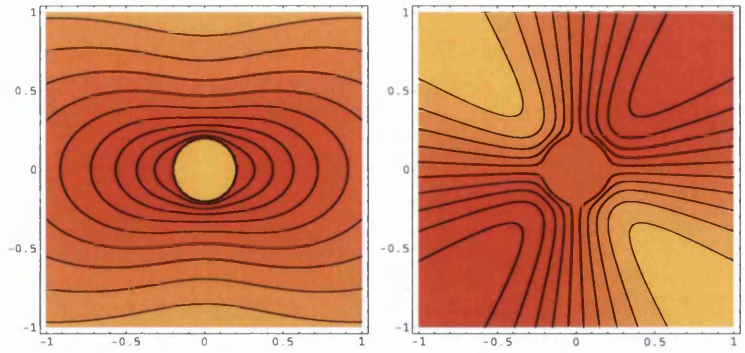
$$u_{1x} = \frac{(R^2 - r^2)\cos^2\theta + r^2\ln\left(\frac{r}{R}\right) + \frac{1}{2}(r^2 - R^2)}{r^2} \tag{2.66}$$

$$u_{1y} = \frac{(R^2 - r^2)\sin\theta\cos\theta}{r^2} \tag{2.67}$$

$$p = -\frac{2\mu\cos\theta}{r} \tag{2.68}$$

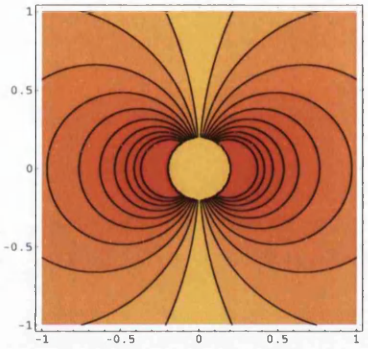
where R is the radius of the cylinder.

It should be noted once again that these functions are not applicable far from the cylinder, however it will be shown later that this constraint is not problematic. The functions can be presented graphically:



(a) u_{1x} (b) u_{1y}

Figure 2.3: u_x flow - Velocity Contour Plots



(a) p_1

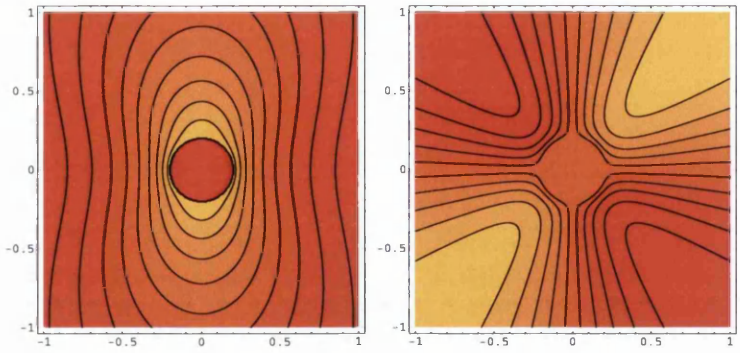
Figure 2.4: u_x flow - Pressure Contour Plots

The work so far has only considered flow in the x-direction, however if the process, (2.61)-(2.68) is repeated, such that the function for ψ, see (2.59), becomes $\psi(r,\theta) = F(r)\cos\theta$, then functions for flow in the y-direction are produced.

$$u_{2x} = -\frac{(R^2 - r^2)\sin\theta\cos\theta}{r^2} \tag{2.69}$$

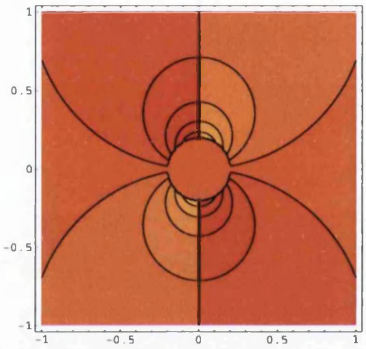
$$u_{2y} = \frac{(R^2 - r^2)\cos^2\theta - r^2\ln\left(\frac{r}{R}\right) + \frac{1}{2}(r^2 - R^2)}{r^2} \tag{2.70}$$

$$p = -\frac{2\mu\sin\theta}{r} \tag{2.71}$$



(a) u_{2x} (b) u_{2y}

Figure 2.5: u_y flow - Velocity Contour Plots



(a) p_2

Figure 2.6: u_y flow - Pressure Contour Plots

Assuming linear variation between flows in x-direction and y-direction these two sets of equations can prescribe velocity profiles for flow in any direction.

2.6 Conclusion.

In this chapter the basic equations that govern the physics of fluid flow have been derived by applying the basic conservation laws i.e. mass, momentum, and energy, to the fluid under consideration. The desired outcome were equations in the spatial or Eulerian coordinate system, however an arbitrary volume of fluid in a *material* or Lagrangian context was used, and then transformed to a spatial reference system through the use of the Reynolds transport theorem. The resulting equations were inadequate to accurately describe the behaviour of the fluid and further equations, known as the Constitutive equations were introduced at this point. The Constitutive equations are particular to the fluid type, therefore stress-strain rate relationship for Newtonian fluids and heat flux-temperature gradient relationship were used. Finally the equations that govern Newtonian fluid flow, i.e. the Navier-Stokes equations, were arrived at.

To conclude the chapter incompressible viscous Navier-Stokes equations, including flow classification, and a brief review of Stokes flow[18] with an analytical solution were discussed.

[18] A subset of the Navier-Stokes equations

Chapter 3

Finite Element Methods for Fluid Flows.

In the previous chapter the basic equations that govern fluid flow, in particular those which are used to describe incompressible viscous flow, were derived. The complexity of these equations negates the use of an analytical approach, thus the only option is to solve numerically. This chapter is focused on the mathematical background that forms the basis of the Finite Element Method, a numerical method which is commonly used to solve the Navier-Stokes equations in fluid flow problems.

The chapter is laid out in the following manner: initially there will be a short introduction to numerical methods, and in particular the FEM, which will be followed by some graphical examples of its applications in modern day engineering in general. In the main section the actual process of producing a FEM model from governing equations will be explained, including the use of variational principles to obtain the weak form of the governing equations, and the discretization of the domain into smaller subdomains. Finally the solution procedure is discussed, followed by a brief conclusion.

3.1 Brief Introduction to the Finite Element Method.

There a various methods available to solve Partial Differential Equations, such as those found in fluid flow problems, numerically and these include Finite Difference, Finite Element, Boundary Element and Finite Volume methods. Each of these methods have advantages and disadvantages which make them suitable for certain problems but unsuitable for others. For the sake of brevity we only

consider the method chosen to realize the ideas discussed within this thesis, i.e. the Finite Element Method, however should the reader wish to extend their understanding of numerical methods there are numerous texts on the subject, see Hughes [42], Zienkiewicz and Taylor [89] or Bathe [6].

A Finite Element solution.
Establish a set of governing algebraic equations which define the nature of the problem, and solve.

The Finite Element Method or FEM is only a relatively recent *discovery* but already it has had a profound effect on engineering. Since its first emergence in the 1950's-1960's it has grown from a mathematical idea to become a corner stone of modern day engineering, and yet its full potential is only partially realized. In this present day the FEM is widely used in commercial applications, e.g. structural dynamics, fluid flow, heat transfer, acoustics, magnetostatics, electrostatics, medicine, weather forecasting, to list but a few, and it is now possible to achieve an unprecedented level of understanding through their use.

3.1.1 Current Usage.

The applicability of the Finite Element Method to analyze many engineering problems is presented succinctly in picture format[1].

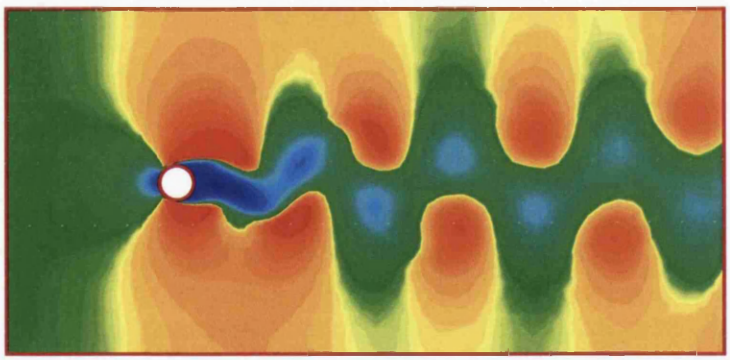
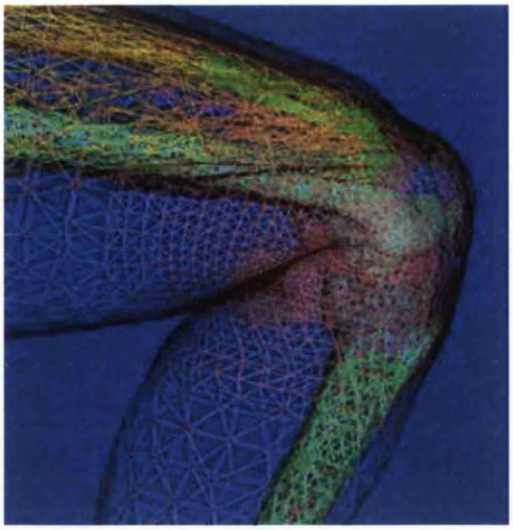


Figure 3.1: FE Analysis of flow past a circular cylinder

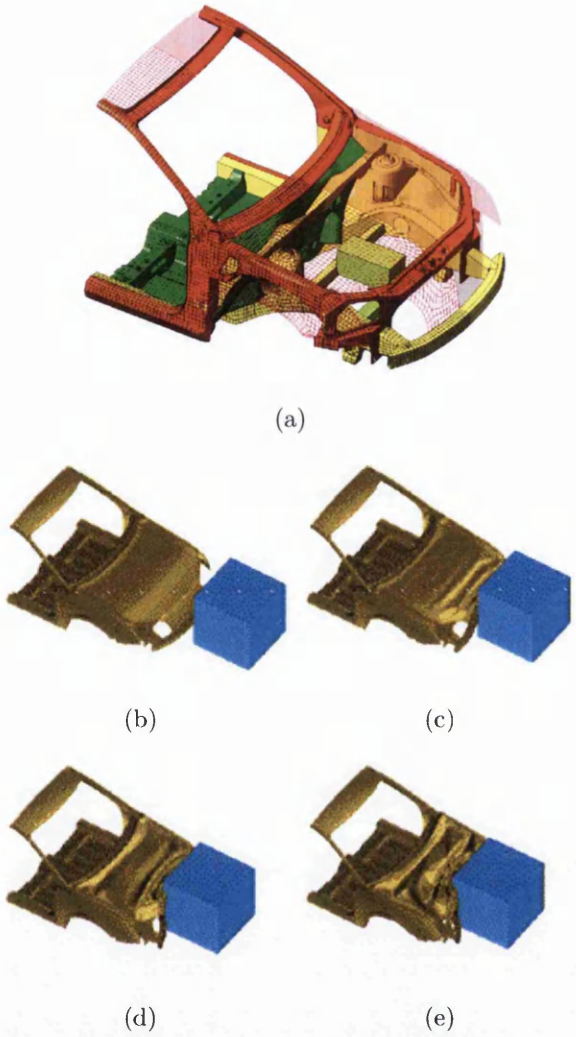
[1]Images 3.2, 3.3, 3.4 were obtained from www.hexa.ru.



(a)

(b) (c) (d)

Figure 3.2: FE Analysis of the human body. (a) Modelling a knee joint (b) actual bone (c)-(d) bone model



(a)

(b) (c)

(d) (e)

Figure 3.3: FE Analysis of car impact. (a) model of car (b)-(e) evolution of the impact in time

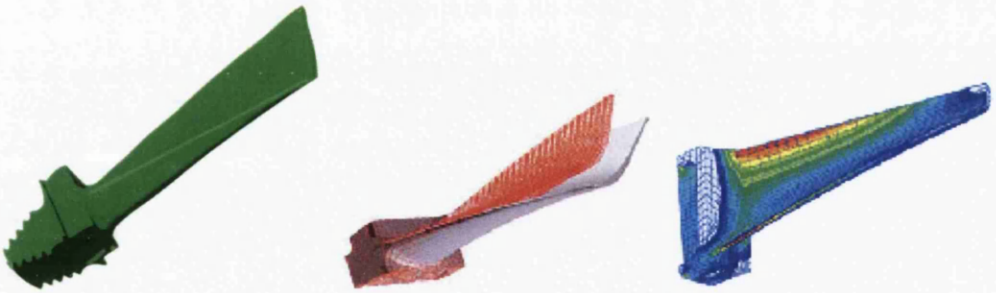


Figure 3.4: FE Analysis turbines blades. (a) Original mesh (b) deformation of blade (c) stress in the blade

3.2 Basics of Finite Element Modelling.

The essence of FEM is to solve a system equation to obtain approximate solutions of boundary value problems. FEM follows the following steps: 1) pre-processing of input data - continuous[2] functions and partial different equations are discretized to form vectors and matrices ; 2) computation to solve the matrix equation; and 3) post-processing of output results, to retrieve the solution from the discretization.

This section provides a basic introduction to Finite Element procedures, with emphasis placed on the particular methods associated with modelling the incompressible Navier-Stokes equations.

3.2.1 Weak Form of the Boundary Value Problem.

The process of spatial discretization by the Finite Element Method is based upon an integral form of the partial differential equations. The first task is the formulation of a (continuous) variational problem associated with the given partial differential equation and its boundary conditions. The method most often chosen for this process is the Weighted Residual method, and in particular the classical Galerkin method.

3.2.1.1 Variational Principles - The Weighted Residual Formulation.

Variational methods, of which the Weighted Residual method is a member, are the first step in the process of transforming the boundary value problem to a usable Finite Element formulation.

Definition 3.1: A 'Variational Principle' specifies a scalar quantity (functional) Π *which is defined by an integral form*[3]

$$\Pi = \int_{\Omega} F(\boldsymbol{u}, \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}, ..) d\Omega + \int_{\Gamma} E(\boldsymbol{u}, \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}, ..) d\Gamma$$

where $\boldsymbol{u}$ *is an unknown function and* F *and* E *are specified differential operators. The general rules for deriving variational principles (the unknown function) from non linear differential equations are complicated and even the tests necessary to establish the existence of such variational principles are not simple*[4].

[2] A function is defined over some region in space over which it is continuous - a FE formulation is a discrete approximation of this continuous function.

[3] Zienkiewicz and Taylor, The Finite Element Method, Vol 1.

[4] Zienkiewicz and Taylor, The Finite Element Method, Vol 1.

In essence the approach is to find the solution of the continuum problem ($\boldsymbol{u}$) such that the stationarity of Π is invoked, with respect to the state variables, i.e. $\delta\Pi = 0$ for all values of $\delta\boldsymbol{u}$. If such a variational principle can be found then it is possible to obtain an approximate solution, in the standard integral form, which is suitable for Finite Element Analysis.

The ability to solve boundary value problems which would otherwise have no classical solution, such as the Navier-Stokes equations, and the ability to take into account the boundary conditions in a manner which is considered more appropriate, e.g. for (4.3) the boundary condition becomes part of the weak form, are considered to be the main advantages of variational methods.

The standard variational approach commonly used for FEM analysis of fluid dynamics problems is the Weighted Residual method, and this method will be used throughout the remainder of this work.

There are numerous texts which give a detailed mathematical review, see Finlayson [32] and Mikhlin [59], of the background and justification of variational methods, however this knowledge is not necessary to be able to apply Weighted Residual method and as such will not be given here.

However it is important at this point to define the main components of the Weighted Residual method, i.e. the boundary conditions, *test* and *trial* functions and the spatial discretization of the domain.

3.2.1.2 The Dirichlet and Neumann boundary conditions.

The domain Ω is an open region of $\mathbb{R}^{n_{sd}}$ with a piecewise smooth boundary Γ, where n_{sd} is the number of spatial dimensions. The boundary Γ can be decomposed as:

$$\Gamma = \Gamma_g \cup \Gamma_h \quad \& \quad \emptyset = \Gamma_g \cap \Gamma_h \tag{3.1}$$

where the boundary Γ consists of two parts (3.1), the so called Dirichlet boundary conditions where the primary unknowns are prescribed and the Neumann boundary conditions where the derivatives of the primary unknowns are defined. The Dirichlet boundary is denoted by Γ_g and the Neumann boundary is denoted by Γ_h.

Prescribed velocities g on the surface Γ_g $$u = g \quad \forall \boldsymbol{x} \in \Gamma_g \tag{3.2}$$

Prescribed fluxes h on the surface Γ_h $$\boldsymbol{q}(u) \cdot \boldsymbol{n} = h \quad \forall \boldsymbol{x} \in \Gamma_h \tag{3.3}$$

where the solution variable u represents any physical field, such as the temperature distribution over the domain, $\boldsymbol{q}(u)$ is flux and a linear function of the first derivatives of u and $x \in \Omega \subset \mathbb{R}^{n_{sd}}$ and the outward unit normal vector, to the fluid surface, is $\boldsymbol{n}$. The unknown function u may also be a vector field, in which case the boundary conditions will be altered slightly.

3.2.1.3 Sobolev Spaces.

The spatial discretization by the Finite Element Method is based upon the discrete representation of the weak integral form of the partial differential equations. This requires that certain functions spaces be defined - known as Sobolev spaces. A Sobolev space is a Hilbert space and is denoted by $(\mathcal{H})$. For the Finite Element Method Sobolev spaces are used to approximate functions which are square integrable, and have square integrable first derivatives over the computational domain Ω, i.e. the trial and test functions. A more detailed account of Sobolev spaces can be found in Adams [2].

3.2.1.4 Trial and Test Functions.

The first step in the Weighted Residual method leading to a Finite Element discretization of the model problem is the formulation of the weak (variational) form of the boundary value problem. This is achieved through the use of the two sets of functions: *test* or *weighting* $(\mathcal{W})$ functions and *trial solution* $(\mathcal{V})$ functions.

The first set of functions, the so called *test* functions, are denoted by $\mathcal{W}$ and consists of all functions which are square integrable, have square integrable first derivatives over the computational domain Ω and vanish on the Dirichlet portion of the boundary. The Finite Element approximation space necessary for the correct discretization is:

$$\mathcal{W} = \{w \in \mathcal{H}^1(\Omega) \,|\, w = 0 \text{ on } \Gamma_g\} \tag{3.4}$$

The second set of functions, *trial*, are similar to the test functions, however these functions are required to satisfy the Dirichlet boundary condition on Γ_g. The FE approximation space is:

$$\mathcal{V} = \{u \in \mathcal{H}^1(\Omega) \,|\, u = u_g \text{ on } \Gamma_g\} \tag{3.5}$$

The sets $\mathcal{V}$ and $\mathcal{W}$ are approximated by the subsets $\mathcal{V}^h$ and $\mathcal{W}^h$ respectively, i.e. $\mathcal{V}^h \subset \mathcal{V}$ & $\mathcal{W}^h \subset \mathcal{W}$ and are in part characterized by the partition of the domain.

3.2.1.5 Partition of the domain.

The domain Ω can now be considered to be discretized into Finite Elements, where Ω is subdivided into n_{el} subdomains, which are denoted by Ω^e.

$$\Omega = \bigcup_{e=1}^{n_{el}} \Omega^e \quad \Omega^e \cap \Omega^f = \emptyset \quad \text{for } e \neq f \tag{3.6}$$

Each subdomain has a piecewise smooth boundary $\Gamma^e = \delta\Omega^e$, where subscript h indicates that the relevant variables belong to a finite dimensional space[5]. The test function $w^h \in \mathcal{W}^h$ vanishes on Γ_g whilst the trial function u^h is an approximation which lies in $\mathcal{V}^h$, and satisfies the boundary condition u_g^h on Γ_g, with a precision defined by h. The interpolation spaces can be defined as:

$$\mathcal{V}^h = \quad \{u^h \in \mathcal{H}^1(\Omega) \mid u^h \mid_{x \in \Omega^e} \in \mathcal{P}_k(\Omega^e),\ u^h \mid_{x \in \Gamma_g} = u_g\} \tag{3.7}$$

$$\mathcal{W}^h = \quad \{w^h \in \mathcal{H}^1(\Omega) \mid w^h \mid_{x \in \Omega^e} \in \mathcal{P}_k(\Omega^e),\ w^h \mid_{x \in \Gamma_g} = 0\} \tag{3.8}$$

where $\mathcal{P}_k(\Omega^e)$ is the space of all polynomials defined on Ω^e, complete to order $k \geq 1$.

3.2.1.6 Classical Galerkin Formulation.

One important Weighted Residual scheme is known as the Galerkin method or the Bubnov-Galerkin method, see Gresho et al [36] and Thomée [83]. The classical Galerkin method and variations of the classical Galerkin method will be adopted in the Finite Element work of this thesis.

Below the process of formulating a (continuous) variational problem associated with the given partial differential equation and its boundary conditions is shown for a model boundary value problem. Introducing Poissons equation:

$$-\nabla^2 u = s \quad \forall \boldsymbol{x} \in \Omega \tag{3.9}$$

where the quantity s represents a specified source term, and Ω is enclosed by a piecewise smooth boundary Γ.

The first step leading to the spatial discretization of the model problem (3.9) is the formulation of the weak (or variational) form of the boundary value problem. This is achieved by the multiplication of the governing equation with a test function w (see Section 3.2.1.4).

$$-\int_\Omega w \nabla^2 u d\Omega = \int_\Omega w s d\Omega \tag{3.10}$$

[5] h is the characteristic length

The integration of weak form (3.10) over the domain Ω, and the application of the divergence theorem results:

$$\begin{aligned} -\int_{\Omega} w\nabla^2 u d\Omega &= -\int_{\Omega} w(\nabla \cdot \nabla) u d\Omega \\ &= -\int_{\Omega} \left(\nabla \cdot (w\nabla u) - \nabla w \cdot \nabla u\right) d\Omega \\ &= \int_{\Omega} (\nabla w \cdot \nabla u) d\Omega - \int_{\Gamma} w(\boldsymbol{n} \cdot \nabla u) d\Gamma \end{aligned} \tag{3.11}$$

Considering the following

1. if $w \in \mathcal{W}$ then $w = 0 \,\text{on}\, \Gamma_g$, see (3.4)
2. $\boldsymbol{q}(u) \cdot \boldsymbol{n} = h$, see (3.3)

the weak form (3.10) becomes:

$$\int_{\Omega} (\nabla w \cdot \nabla u) d\Omega = \int_{\Omega} w s d\Omega + \int_{\Gamma} w h d\Gamma \tag{3.12}$$

Note that the use of the divergence theorem has allowed for the Neumann boundary condition to be introduced naturally. The form of (3.12) means that it can be written in the format:

$$a(w, u) = S(w) \tag{3.13}$$

where $a(w, u)$ and $S(w)$ are a bilinear and a linear form, respectively. Provided $u \in \mathcal{V}$, the Dirichlet boundary condition on Γ_g is satisfied, and the model problem can be stated as:

$$\text{find } u \in \mathcal{V} \text{ such that} \quad a(w, u) = S(w) \quad \forall w \in \mathcal{W} \tag{3.14}$$

Through the use of Lax-Milgram Lemma one can show that a weak solution $u \in \mathcal{V}$ of (3.14) is unique, however this proof is not shown here, see Donea and Huerta [28] for further details.

Thus one arrives directly at the classical Galerkin formulation of the original model, where the weak form (3.14) is restricted to the finite dimensional spaces $\mathcal{V}^h \subset \mathcal{V}$ and $\mathcal{W}^h \subset \mathcal{W}$. The weak form now reads:

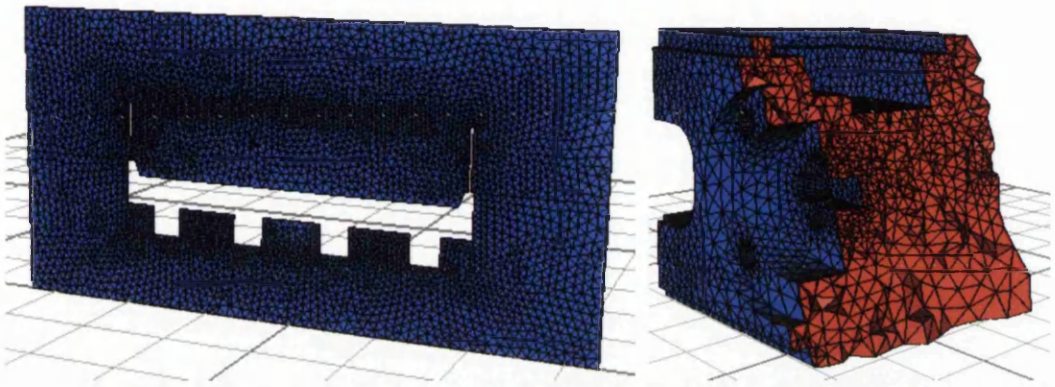
$$\text{find } u^h \in \mathcal{V}^h \text{ such that} \quad a(w^h, u^h) = S(w^h) \quad \forall w^h \in \mathcal{W}^h \tag{3.15}$$

where the approximate solution $u^h \approx u$ satisfies the Dirichlet boundary condition exactly whilst only satisfying the Neumann boundary condition in the *weak* sense.

3.2.2 Finite Elements.

The Finite Element Method generates an approximate solution using a systematic approach which consists of using finite dimensional subspaces for the trial and test functions. In the previous section the necessary mathematical background for these subspaces was provided, and in this section the discretization of the domain, and associated methods are discussed.

3.2.2.1 Discretization.



(a) 2D mesh - triangular elements (b) 3D mesh - tetrahedral elements

Figure 3.5: FE meshes

The subspaces Ω^e are obtained through the partition of the domain Ω, such that a mesh of non-overlapping elements is created. These elements, otherwise known as Finite Elements, can take the form of many different geometrical shapes, such as triangles, and quadrilaterals in 2D, Figure 3.5(a) and tetrahedra and hexahedra in 3D, Figure 3.5(b).

The work in this thesis is only concerned with 2D simulations and no further mention will be made of 3D modelling processes; the methodology is identical although 3D meshing is considerably more complex than that of 2D.

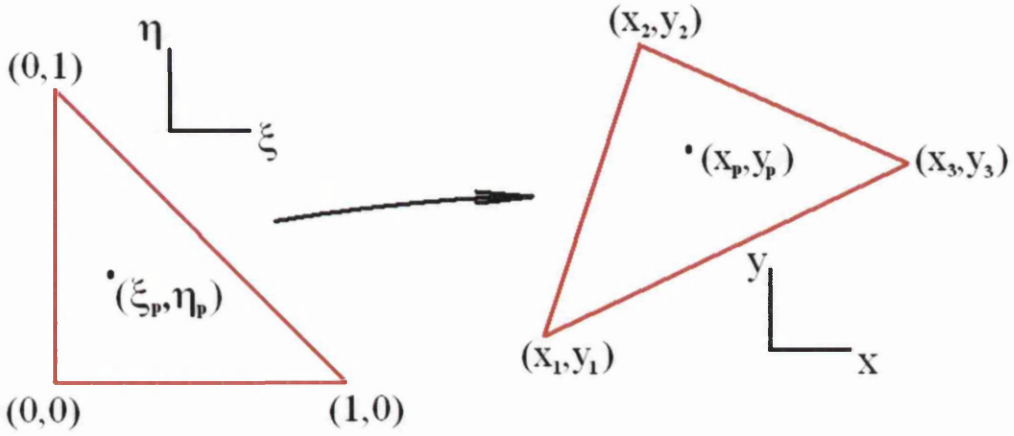
3.2.2.2 Mapping.

The Finite Element Method is a powerful tool because it is easily standardized for computer implementation. When the domain is discretized the result is a mesh of Finite Elements Ω^e which are all unique in location and size. Mapping is a technique that standardizes the calculations of a typical element irrespective of

its size or global location in the domain, by relating a Parametric element domain $\square$ to Finite Element domain Ω^e.

The map from parametric coordinates $\boldsymbol{\xi} \in \square$ to coordinates $\boldsymbol{x}^h \in \Omega^e$ is given by the relationship

$$\boldsymbol{x}^h = \boldsymbol{x}(\boldsymbol{\xi}) = \sum_I N_I(\boldsymbol{\xi}) \boldsymbol{x}_I \tag{3.16}$$



Parametric element domain $\square$ Finite Element Ω^e

Figure 3.6: Mapping transformation

Equ (3.16) for a 2D three noded triangular element (i.e. $n_{ne} = 3$), see Figure 3.6, can be written as:

$$x^h = \sum_{A=1}^{3} N_I(\xi, \eta) x_I \tag{3.17a}$$

$$y^h = \sum_{A=1}^{3} N_I(\xi, \eta) y_I \tag{3.17b}$$

or in matrix format:

$$\begin{bmatrix} x^h \\ y^h \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} \tag{3.18}$$

where the shape functions N_I are given as:

$$N_1 = 1 - \xi - \eta \quad N_2 = \xi \quad N_3 = \eta \tag{3.19}$$

3.2.2.3 Shape Functions.

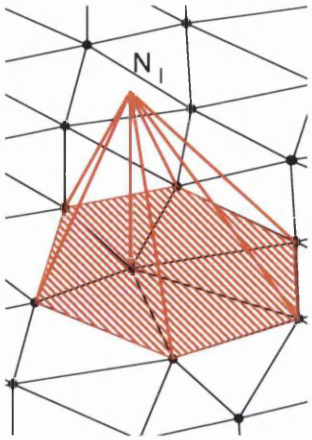


Figure 3.7: Graphical representation of shape function N_I in a 2D domain

The shape function N_I [6] is used to interpolate the nodal values w_I of a function $w^h \in \mathcal{W}^h$ within an element. The expression for w^h is given, over an element, as:

$$w^h = N_I(\boldsymbol{\xi})\, w_I \qquad \boldsymbol{\xi} \in \square \tag{3.20}$$

$$\Rightarrow \qquad w^h = N_I(\boldsymbol{x}^h)\, w_I \qquad x^h \in \Omega^e \tag{3.21}$$

where the value of N_I at the nodes, see Figure 3.7 is:

$$N_I(x_J) = \delta_{IJ} = \begin{cases} 1 & I = J \\ 0 & I \neq J \end{cases} \tag{3.22}$$

where δ_{IJ} is the Kronecker delta function, and $I, J = 1, 2, ...n_{ne}$.

3.2.2.4 Isoparametric Mapping.

An element is classified as isoparametric when the shape functions used to define the mapping between $\square \rightarrow \Omega^e$ and the function w^h are the same, i.e. the shape function that defines (3.16) also serves to define (3.21) . The isoparametric concept leads to an convenient framework for computer implementation and is generally attributed to Taig [75] and Irons [48].

[6] N_I a polynomial whose order is governed by the number of nodes per element

3.2.3 Solution Procedure.

The classical Galerkin formulation, see (3.15), reads:
For $\mathcal{W}^h \subset \mathcal{W}$ and $\mathcal{V}^h \subset \mathcal{V}$ find $u^h \in \mathcal{W}^h$ such that:

$$a(w^h, u^h) = S(w^h) \quad \forall w^h \in \mathcal{W}^h \tag{3.23}$$

Using the Finite Element discretizations, (3.21), for w^h and u^h the classical Galerkin formulation reads:

$$a(w^h, u^h) = \sum_{e=1}^{n_{el}} w_I k_{IJ}^e u_J = \boldsymbol{w} \cdot \boldsymbol{K}\boldsymbol{u} \tag{3.24a}$$

$$S(w^h) = \sum_{e=1}^{n_{el}} w_I f_I^e = \boldsymbol{w} \cdot \boldsymbol{F} \tag{3.24b}$$

where $\boldsymbol{w}$ and $\boldsymbol{u}$ are the n_{nd}-dimensional vectors of the nodal values of the test $\boldsymbol{w}^h$ and trial solution $\boldsymbol{u}^h$ functions respectively. The stiffness matrix $\boldsymbol{K}$ and external force vector $\boldsymbol{f}$ result from the assembly of the element contributions:

$$\boldsymbol{K} = \boldsymbol{A}_{e=1}^{n_{el}} \boldsymbol{k}^e \qquad \boldsymbol{F} = \boldsymbol{A}_{e=1}^{n_{el}} \boldsymbol{f}^e \tag{3.25}$$

where $\boldsymbol{A}$ represents the assembly operator[7] acting on the local element matrix and nodal vectors. Since $\boldsymbol{w}$ is arbitrary, (3.24) can be rewritten as

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{F} \tag{3.26}$$

and the original continuous field problem have been transformed into a system of equations in terms of discrete unknowns, which can be solved. For a system of nonlinear equations, the Galerkin formulation may be written as

$$a(w^h, u^h) = \sum_{e=1}^{n_{el}} w_I r_I^e(u_1, u_2, ..., u_{n_{ne}}) = \boldsymbol{w} \cdot \boldsymbol{R} \tag{3.27a}$$

$$S(w^h) = \sum_{e=1}^{n_{el}} w_I f_I^e = \boldsymbol{w} \cdot \boldsymbol{F} \tag{3.27b}$$

where $\boldsymbol{R}$ is a nonlinear function of $\boldsymbol{u}$ assembled from r_I^e as:

$$\boldsymbol{R} = \begin{bmatrix} r_1(\boldsymbol{u}) \\ \vdots \\ r_{n_{ne}}(\boldsymbol{u}) \end{bmatrix} \qquad \boldsymbol{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_{n_{ne}} \end{bmatrix} \tag{3.28}$$

Therefore

$$\boldsymbol{R}(\boldsymbol{u}) = \boldsymbol{F} \tag{3.29}$$

represents the nonlinear counterpart of (3.26)

[7] The addition of the element contributions to the appropriate locations in the global matrices and vectors

3.2.3.1 Newton-Raphson Iterative Solver

The Newton-Raphson iterative solver is possibly one of the most widely used and most rapidly convergent processes available for the solution of nonlinear equations, and is commonly found in standard texts on numerical procedures, including Bathe [6], and papers such as Stoer and Bulirsch [71], and Ortega and Rheinboldt [62]. The basic premise of the Newton Raphson scheme for nonlinear equations is to rewrite (3.29) in the form:

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{R} - \boldsymbol{F} \tag{3.30}$$

where the tangent stiffness $\boldsymbol{K}$ is given by:

$$\boldsymbol{K} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}} \tag{3.31}$$

and solve iteratively using the process described in Box.(1).

For $i = 1, 2, 3, ...$

1 Estimate $\boldsymbol{u}_{t+\Delta t}$ and calculate $\boldsymbol{R}_{t+\Delta t}$

2 Compute $\boldsymbol{R}_{t+\Delta t} - \boldsymbol{F}_{t+\Delta t}$

3 Check solution convergence. If $\| \boldsymbol{R}_{t+\Delta t} - \boldsymbol{F}_{t+\Delta t} \| \leqslant$ *tolerance* Then Goto 8

4 Compute tangent stiffness $\boldsymbol{K}_{t+\Delta t}^{(i-1)} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}$

5 Solve $\boldsymbol{K}_{t+\Delta t}^{(i-1)} \Delta \boldsymbol{u}^{(i)} = \boldsymbol{R}_{t+\Delta t} - \boldsymbol{F}_{t+\Delta t}$

6 Update $\boldsymbol{u}_{t+\Delta t}^{(i)} = \boldsymbol{u}_{t+\Delta t}^{(i-1)} + \Delta \boldsymbol{u}^{(i)}$

7 Goto 2

8 Exit

Box 1: Newton-Raphson procedure for the solution of nonlinear problem

3.3 Conclusion

It was noted in Chapter 2 that the complexity of the Navier-Stokes equations suited a numerical approach to the solution. In this chapter the numerical approach to solving partial different equations was developed in readiness for future chapters.

The chosen numerical method was the Finite Element Method (FEM), and for the sake of brevity no other method was discussed. The FEM however was developed in reasonable detail, from governing equations to FEM model, including the formulation of the weak form, discretization of the domain and the solution procedure.

Should the reader wish for more information on any subjects discussed within this chapter the author directs the reader to articles and books cited.

Chapter 4

Stabilized Finite Elements Methods for Fluid Flows.

Previously the mathematical methods and theories which form the basis of the Finite Element Method were discussed. In this present chapter we continue the mathematical description of the Finite Element Method with a more indepth look at the classical Galerkin method, as introduced previously.

It is well known that the classical Galerkin method is unsuitable for solving the incompressible Navier-Stokes equations. The solutions obtained using a Finite Element formulation generated by this method exhibit spurious oscillations, caused by inconsistencies in the velocity and pressure fields. It will be shown that spurious oscillations found in the velocity field are due to the convection term $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$ whilst the pressure instabilities are attributed to equal order interpolations of velocity and pressure. However the development of *stabilized* Galerkin methods will circumvent the difficulties typically associated with the classical Galerkin method. The end result will be a technique which maintains solution accuracy, facilitates more convenient FE approximations, whilst counteracting the inconsistencies mentioned earlier.

The order of the chapter will be a brief discussion concerning instabilities due to the convection term, i.e. velocity, followed by the development of the steady state 1D advection diffusion problem from governing equations to the weak form, including a brief numerical example. This will simply highlight the problems faced when modelling convection dominated flows. Afterwards the discussion will continue to oscillations in the pressure field and here the steady state Navier-Stokes equations will be used. Finally in this section the technique of stabilization will be covered and the previous examples will be reproduced to show the effect

of the stabilization on the solution results. For the sake of simplicity this chapter is restricted to steady state fluid flow, however the methodology shown here is equally applicable to the unsteady formulations discussed in later chapters.

4.1 Spurious Oscillations caused by Velocity.

It is well known that the classical Galerkin method leads to the best approximation in the energy norm for self adjoint problems [1] such as those typically found in solid mechanics. However this characteristic is lost when problems, such as those commonly found in fluid mechanics[2], are governed by non-symmetric operators. For problems involving highly non symmetric operators e.g. convection dominated problems, additional numerical difficulties arise. In these situations the classical Galerkin method is found to be inadequate and the solution tends to exhibit spurious and globally propagating oscillations, see Figure 4.1. The instabilities develop in areas where the boundary layer is strong and the inability of the classical Galerkin method to resolve this boundary layer causes the oscillations which spread through the entire domain.

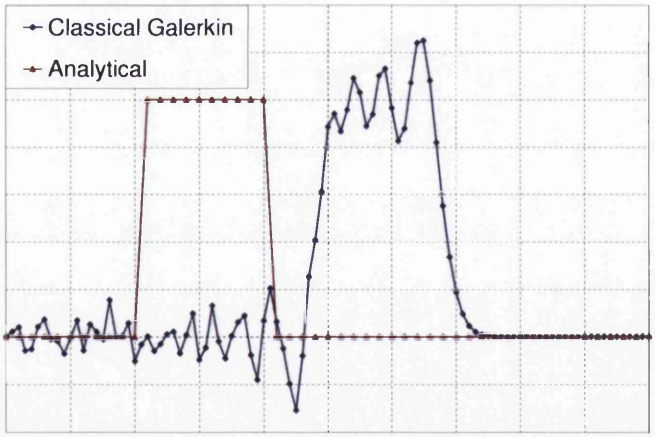


Figure 4.1: Numerical and analytical(displaced) solution for the case of a propagating rectangular wave, modeled using the 1D advection-diffusion equations

[1] Self-adjoint - An operator which is symmetrical

[2] For fluid problems poorly resolved internal and boundary layers can be identified as the original source of the oscillations.

There are various methods which can reduce or eliminate the oscillations, e.g. mesh refinement, ALE and stabilization. Mesh refinement in the affected areas is an obvious choice as the oscillations are caused by rapidly changing solutions, i.e. where the solution is not smooth, however this is not desirable due to the increase in problem size. ALE has been shown to improve the solution, as presented by Macfadden [56] for a simple 1D heat flow problem, but this method is computationally difficult to implement, and for $n_{dim} > 1$ can lead to excessive mesh distortion. Stabilization is a popular option, and one which uses a method often found in Finite Difference simulations, i.e. the addition of artificial diffusion[3]. The artificial diffusion balances the reduced diffusion of the classical Galerkin method, suppressing the oscillations; it can also cause loss of accuracy if too overdiffusive. Therefore care must be taken when choosing the level of artificial diffusion applied to avoid an incorrect solution.

In order to understand the underdiffusive nature of the classical Galerkin method the 1D Advection Diffusion problem is developed below, from governing equations to numerical example, using the Galerkin method.

4.1.1 FEM Formulation - 1D Steady State Advection Diffusion.

In order to model the problem the boundary value problem must be defined, and for numerical modelling this means the boundary value problem consists of a set of differential equations, a specification of the domain of interest, boundary conditions (BC's) and initial conditions (IC's). The domain of interest represents space over which the solution is sought and is fixed in space i.e. an Eulerian coordinate system, the boundary conditions represent prescribed values on the domain boundary and the initial conditions correspond to the prescribed values on the entire domain at the initial time step.

4.1.1.1 The Governing Equations.

Recalling the incompressible Navier-Stokes equations derived in Chapter 2, the steady state one dimensional version reads:

$$\rho(\not{a} + u_x u_{,x}) - (\mu u_{,x})_{,x} - q_x = 0 \tag{4.1}$$

[3]This method is applicable to the Finite Element Method due to the similar way which both methods approximate the differential operators

where u_x is a scalar unknown in the x spatial direction, μ is diffusion coefficient, q_x is the source term. Here the standard notation for derivatives are used:

$$u_x = u(x) \qquad \dot{u} = \frac{du}{dt} \qquad u_{,x} = \frac{du}{dx} \qquad u_{,xx} = \frac{d^2u}{dx^2} \tag{4.2}$$

The 1D steady state advection-diffusion equation is a linear form of the Navier-Stokes equation, see (4.1):

$$\cancel{\dot{u}} + \underbrace{(a_x u_{,x})}_{convection} - \underbrace{(\mu u_{,x})_{,x}}_{diffusion} - q_x = 0 \qquad \forall x \in \Omega \tag{4.3}$$

where a is a given free velocity field defined over the spatial domain. Note that the inherent nonlinearity due to the convection term of the Navier-Stokes equation has been removed. The equation (4.3) still contains additional nonlinearities which appear if any of the following terms are dependent on the scalar unknown u, e.g. diffusion μ, free velocity a. However in this section it is assumed that μ, a are independent, and therefore (4.3) is a linear differential equation.

It is sometimes useful to introduce, at this stage, the term

$$Pe^G = \frac{a_x L}{2\mu} \tag{4.4}$$

where the global Peclet number Pe^G is inversely proportional to μ[4]. We can rewrite (4.5) to give:

$$\cancel{\dot{u}} + \underbrace{\frac{2\mu\, Pe^G}{L}(u_{,x})}_{convection} - \underbrace{\frac{a_x L}{2\, Pe^G}(u_{,x})_{,x}}_{diffusion} - q_x = 0 \qquad \forall x \in \Omega \tag{4.5}$$

In consideration (4.10) demonstrates the fundamental difficulty faced in solving flow problems in general: as the Peclet number increases, i.e. $Pe^G > 1$, the solution becomes dominated by the convection term. However as the Peclet number decreases, i.e. $Pe^G < 1$, the problem becomes diffusion dominated. Therefore the analysis must be able to solve for the response initially governed by diffusion at low Peclet numbers, and convection for large Peclet numbers.

4.1.1.2 Domain and BC's.

The boundary conditions were described previously, Section 3.2.1.2, however they are briefly repeated here. Let $\Omega =]0, L[$ denote the spatial domain with a piecewise smooth boundary Γ. Considering only the Dirichlet boundary condition:

$$\text{Prescribed fluxes } g \text{ on the surface } \Gamma \qquad u = g \quad \forall x \in \Gamma \tag{4.6}$$

[4]If both L & a_x are constant, which is an acceptable assumption for this problem.

The boundary value problem is given as: find a value for u which satisfies (4.3) $\forall x \in \Omega$ and the prescribed boundary conditions.

4.1.1.3 Finite Element Spaces.

The Finite Element approximation spaces necessary for the correct discretization of the governing equations are defined as:

- $\Omega \subset \mathbb{R}^N$ represents the closure of the physical spatial domain, in N dimensions. Therefore $\Omega \cup \Gamma$, where Γ is the boundary.
- The spatial domain Ω is discretized into n_{el} Finite Elements, with characteristic length $\Delta x = x_{i+1} - x_i$
- For Semi-Discrete formulations the trial and test functions can be defined as:

  Trial
  $$\mathcal{V}^h = \{u^h \in H^1(\Omega),\, u^h_{x\in\Omega^e} \in P_k(\Omega^e),\, u^h|_{x\in\Gamma} = g\}$$
  Test
  $$\mathcal{W}^h = \{w^h \in H^1(\Omega),\, w^h_{x\in\Omega^e} \in P_k(\Omega^e),\, w^h|_{x\in\Gamma} = 0\}$$

  where H^1 denotes the usual Sobolev space of functions with square-integrable values and derivatives on Ω whilst $P_k(\Omega^e)$ is the space of all polynomials defined on Ω^e, complete to order $k \geq 1$.

4.1.1.4 Finite Element Discretization.

The next step in the Finite Element formulation is the discretization of the spatial domain[5]. As introduced previously the general Finite Element discretization applied to the field of the unknown variable can be written as:

$$u^h = \sum N_I(x) u_I \qquad w^h = \sum N_I(x) w_I \tag{4.7}$$

where the linear shape function N_I which denotes the shape function for node I, and is used to interpolate the nodal values of u_I, and its virtual counterpart w_I.

[5]Reduces a continuous-system mathematical model into a discrete idealization

4.1.1.5 Weak Form.

The process involves multiplying the governing equation (4.3) with the test function w, and the integration, over the spatial domain, of the resulting terms. The variational (weak) formulation of the boundary value problem (4.3), ignoring external forces terms, applying the classical Galerkin method is as follows:

$$G(u,w) = \int_{\Omega} w \cdot \left[\cancel{u} + (a_x u_{,x}) - (\mu u_{,x})_{,x} \right] dx = 0 \tag{4.8}$$

Applying the divergence theorem to the second term on the RHS and taking into account the Dirichlet boundary condition (4.6) [6] gives:

$$G(u,w) = \int_{\Omega} \left[w \cdot \cancel{u} + w \cdot (a_x u_{,x}) - w_{,x} \cdot (\mu u_{,x}) \right] dx = 0 \tag{4.9}$$

The final step of performing the Finite Element discretization of the weak form (4.9) leads to the following statement: Solve for $u^h \in \mathcal{V}^h$ such that $\forall w^h \in \mathcal{W}^h$

$$G(u^h, w^h) = \sum_{i=1}^{n_{el}} \left[\int_{x_i}^{x_{i+1}} w^h \cdot \cancel{u}^h + w^h \cdot (a_x u^h_{,x}) - w^h_{,x} \cdot (\mu u^h_{,x})\, dx \right] = 0 \qquad \forall x \in]0, L[\tag{4.10}$$

4.1.2 Numerical Example - Fixed Boundary Problem.

The 1D Steady State Advection Diffusion Fixed Boundary Problem is used to demonstrate the underdiffusive nature of the classical Galerkin method.

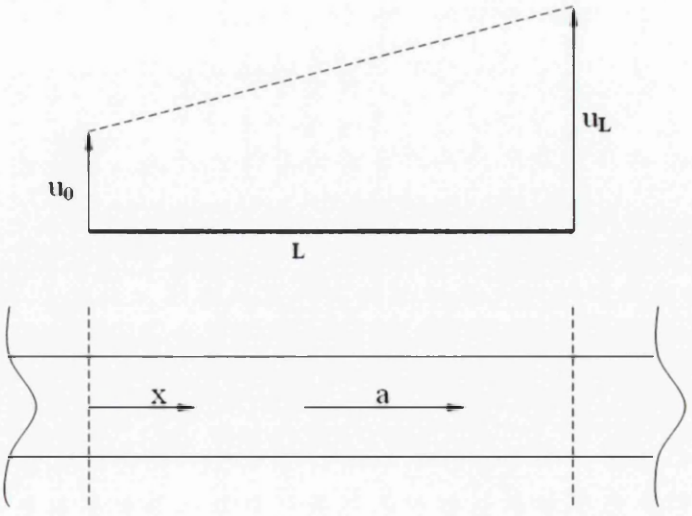


Figure 4.2: Advection-diffusion for 1D fixed boundary condition problem

[6] We assume that the flux on the surface is zero, i.e. $w \cdot [\mu u_{,x}] = 0$.

Model.

The solutions are obtained for $Pe^G = 1$, 10 100 with increasingly refined meshes (5, 10, 20, 50, 100 elements) using a 1D 2 noded element. The value of Pe^G is achieved by setting $a_x = 1$, $L = 1$ and varying $\mu = 0.005$, 0.05, & 0.5.

Geometry.

$$\Omega =]0, L[\tag{4.11}$$

Boundary Conditions.

$$u(0) = u_0 \qquad u(L) = u_L \tag{4.12}$$

Mesh.

$$\Delta x = \frac{1}{5}L, \frac{1}{10}L, \frac{1}{20}L, \frac{1}{50}L, \frac{1}{100}L \tag{4.13}$$

Exact Solution.

The exact solution[7] for the boundary value problem is given by:

$$\frac{u - u_0}{u_L - u_0} = \frac{e^{2\frac{Pe^G}{L}x} - 1}{e^{2Pe^G} - 1} \tag{4.14}$$

Numerical Results.

Figure 4.3 shows the plots for different values of Pe^G and Pe^e. These charts demonstrate the solution behaviour, and it is noticeable that for $Pe^e > 1.0$ the numerical solution is not comparable to the exact solution and the results are beset by nonphysical oscillation.

[7]Exact solution is derived in Appendix B

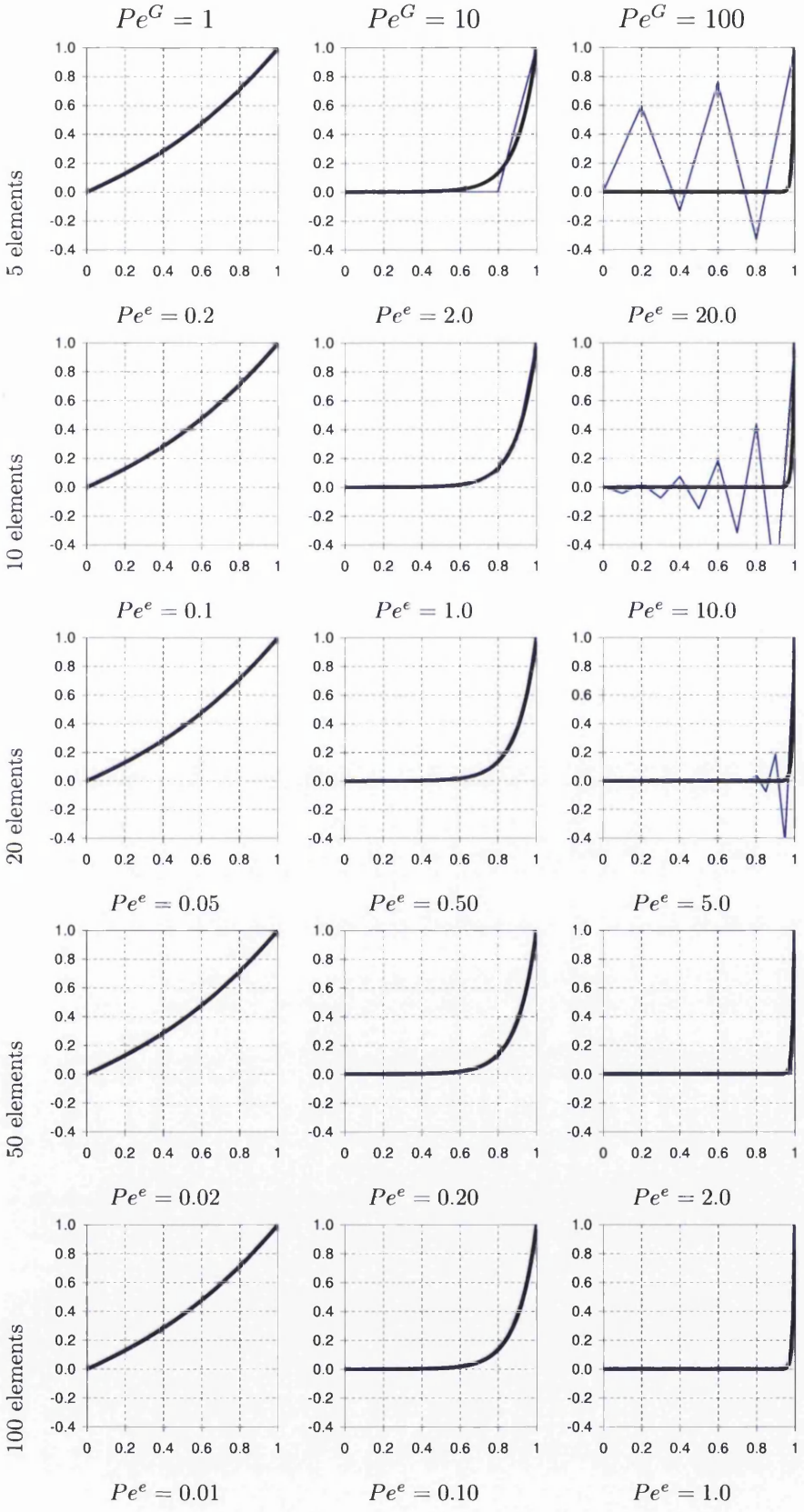


Figure 4.3: Comparison of FEM solution with exact solution for $Pe^G = 1$, 10, 100 for meshes with 5, 10, 20, 50 and 100 elements

Discussion.

In Figure 4.3 the results obtained from the solution of the model problem are presented, where Pe^G and Pe^e represent the global Peclet number and the local Peclet number respectively. Pe^e is defined by:

$$Pe^e = \frac{a_x \Delta x}{2\mu} \tag{4.15}$$

The above figure highlights the difficulties faced by the classical Galerkin method. The solutions which do not exhibit strong boundary layers are not affected by spurious oscillations, however those which do have strong boundary layers are affected and in these cases the solution is completely governed by the oscillations. It is apparent from Figure 4.3 that presence of a strong boundary layer is related to the value of Pe^G, however it can be seen that mesh refinement has a positive outcome on affected solutions. Therefore we can state that the numerical accuracy of the solution is governed by Pe^e. For $Pe^e \leq 1.0$ the results closely match the exact solution whereas solutions with $Pe^e > 1.0$ show spurious oscillations. Therefore the inference can be drawn that it is Pe^e that is the true measure of the solution and not Pe^G.

This conclusion can be further proved by the following simple example. Consider the equation for a 1D element, written in vector form:

$$\sum_{i=1}^{n_{el}} \begin{pmatrix} w_i & w_{i+1} \end{pmatrix} \cdot \left(\frac{a}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} + \frac{\mu}{\Delta x} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \right) \begin{pmatrix} u_i \\ u_{i+1} \end{pmatrix} = 0 \tag{4.16}$$

obtained by substituting in the equations for u, w and N into (4.10), where N for linear 1D shape function is given as:

$$N_i = \frac{x_{i+1} - x}{\Delta x} \qquad N_{i+1} = \frac{x - x_i}{\Delta x} \tag{4.17}$$

.

The nodal forces for two neighbouring elements are assembled into a global matrix and expanded and written in terms of u_{i-1}, u_i and u_{i+1}.

$$\left[Pe^e - 1 \right] u_{i+1} + 2u_i - \left[Pe^e + 1 \right] u_{i-1} = 0 \tag{4.18}$$

which can be rearranged making u_i the subject:

$$u_i = \frac{1 - Pe^e}{2} u_{i+1} + \frac{1 + Pe^e}{2} u_{i-1} \tag{4.19}$$

One will note that (4.19) is actually the central difference scheme for the governing equations; it is well known that this scheme is unsuitable for modelling advection dominated problems.

We demonstrate this by substituting in the values $u_{i-1} = 0$ and $u_{i+1} = 1$ into (4.19):

$$u_i = \frac{1 - Pe^e}{2} \times 1 + \frac{1 + Pe^e}{2} \times 0 \tag{4.20}$$

$$= \frac{1}{2}(1 - Pe^e) \tag{4.21}$$

The result is negative if $Pe^e > 1$!

Conclusion.

In conclusion the instabilities in the velocity field can be circumvented by mesh refinement, which in the case of the 1D Advection Diffusion problem is not an important factor, however for large scale 2D or 3D simulations this constraint is extremely computationally expensive and should be avoided where possible.

4.2 Incompressibility Constraint.

The dependent variables of the governing equations developed in Chapter 2 are velocity and pressure, which in a FE setting, the classical Galerkin formulation gives rise to what is termed *mixed methods*[8]. It is well known that the mixed Galerkin formulation requires careful consideration when applied to near or fully incompressible fluid flow problems. Taylor *et al* [76] found that the application of equal order interpolations lead to good velocity solutions yet the pressure solution was meaningless[9], see Figure 4.4. However when mixed interpolations[10] were employed, the solution for both fields were more accurate. Babuška [5] and Brezzi [14] produced a mathematical framework for understanding this behaviour, which specifies that the interpolating functions must satisfy certain stability conditions, i.e. the FE approximation spaces for pressure and velocity cannot be chosen independently. This is known as the Babuška & Brezzi (or inf-sup) condition. Detailed information, including the derivation of the proof and the significance of complying to the constraint in mixed-method formulations, can be found in Brezzi [15] and Girault and Raviart [35].

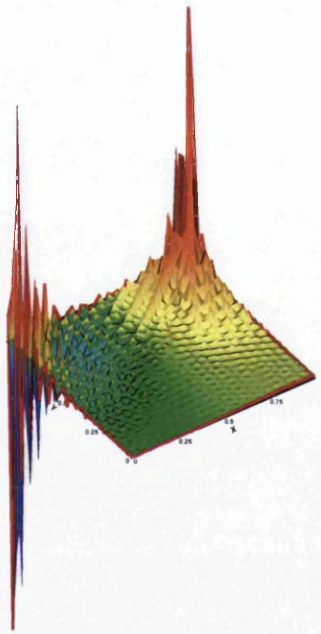


Figure 4.4: Pressure plot: Numerical solution for the lid driven cavity example, modelled using the classical Galerkin formulation on a refined mesh.

[8] FE approximation of two or more vector/scalar fields

[9] Equal order interpolations generally leads to a singular matrix

[10] Unequal interpolations

From a computational viewpoint the Babuška & Brezzi condition leads to inconvenient numerical procedures, in many cases seemingly natural combinations cause large pressure oscillations in the pressure field. These oscillations are prevalent throughout the pressure solution and unlike the oscillations caused by the convection term in the previous section even increased mesh refinement has no effect, see Figure 4.4. Various numerical techniques have been suggested to avoid the inconvenience of Babuška & Brezzi condition, and these will be discussed in great detail in a later section of this chapter, under the heading of Stabilization.

In order to highlight problems caused by the incompressibility constraint we will now look at the Stokes equations and a numerical example i.e. the lid driven cavity.

4.2.1 FEM Formulation - Stokes equation.

The Stokes equation is special case of the Navier-Stokes equations and is suited to the consideration of the incompressibility condition because the convection term, which proved to be unstable under certain conditions, see Section 4.1.1, has been removed. Therefore all solutions obtained using the Stokes equation will be unaffected by the spurious oscillations caused by convection.

4.2.1.1 The Governing Equations.

Recall the Stokes equations, see (2.47)-(2.48):

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \qquad \forall \boldsymbol{x} \in \Omega \tag{4.22}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \forall \boldsymbol{x} \in \Omega \tag{4.23}$$

4.2.1.2 Domain and BC's.

The domain Ω is an open region of $\mathbb{R}^{n_{sd}}$, therefore $\Omega \cup \Gamma$ where Γ is the piecewise smooth boundary and n_{sd} is the number of spatial dimensions. Considering only the Dirichlet boundary condition:

$$\text{Prescribed velocities } \boldsymbol{g} \text{ on the surface } \Gamma \qquad \boldsymbol{u} = \boldsymbol{g} \quad \forall \boldsymbol{x} \in \Gamma \tag{4.24}$$

where $\boldsymbol{x} \in \Omega \subset \mathbb{R}^{n_{sd}}$, and the outward unit normal vector, to the fluid surface, is $\mathbf{n}$. The boundary value problem consists of finding $\boldsymbol{u}$ and p which satisfy both Equ (4.22) & Equ (4.23) and the prescribed boundary condition (4.24).

4.2.1.3 Finite Element Spaces.

The Finite Element approximation spaces necessary for the correct discretization of the governing equations are defined as:

- The spatial domain Ω is discretized into n_{el} Finite Elements.
- For Semi-Discrete formulations the trial and test functions for the velocity field can be defined as:

  Trial

  $\mathcal{V}^h = \{\boldsymbol{u}^h \in H^1(\Omega)^{n_{dim}},\ \boldsymbol{u}^h|_{\boldsymbol{x}\in\Omega^e} \in P_k^{n_{dim}}(\Omega^e),\ \boldsymbol{u}^h|_{\boldsymbol{x}\in\Gamma_g} = \boldsymbol{g}\}$

  Test

  $\mathcal{W}^h = \{\boldsymbol{w}^h \in H^1(\Omega)^{n_{dim}},\ \boldsymbol{w}^h|_{\boldsymbol{x}\in\Omega^e} \in P_k^{n_{dim}}(\Omega^e),\ \boldsymbol{w}^h|_{\boldsymbol{x}\in\Gamma_g} = \boldsymbol{0}\}$

  whilst for the pressure field

  $\mathcal{P}^h = \{p^h |\, p^h \in C^0(\Omega) \cap \boldsymbol{L}_0^2(\Omega),\ p^h|_{\boldsymbol{x}\in\Omega^e} \in P_k(\Omega^e)\}$

  where $H^{1^{n_{dim}}}$ denotes the usual Sobolev space of functions with square-integrable values and derivatives on Ω, C^0 is the space of continuous functions on Ω, $\boldsymbol{L}_0^2$ denotes space of the square-integrable functions $\boldsymbol{L}^2(\Omega)$ modulo constants, whilst $P_k(\Omega^e)^{n_{dim}}$ is the space of all polynomials defined on Ω^e, complete to order $k \geq 1$ and n_{dim} is the number of degrees of freedom. Note: since there are no pressure boundary conditions imposed on the pressure field, the test and trial spaces for pressure coincide.

4.2.1.4 Finite Element Discretization.

As introduced previously the General Finite Element discretization applied to the velocity and pressure fields can be written as:

$$\boldsymbol{u}^h = \sum N_I(\boldsymbol{x})\boldsymbol{u}_I \qquad \boldsymbol{w}^h = \sum N_I(\boldsymbol{x})\boldsymbol{w}_I \tag{4.25}$$

$$p^h = \sum N_I(x)p_I \qquad q^h = \sum N_I(x)q_I \tag{4.26}$$

where the linear shape function N_I which denotes the shape function for node I, and is used to interpolate the nodal values of $\boldsymbol{u}_I$ & p_I, and its virtual counterparts $\boldsymbol{w}_I$ & q_I.

If the Finite Element spaces are chosen accordingly, i.e. as defined by the boundary value problem, see (4.31), then the approximations for velocity and

pressure will be computationally convenient equal order piecewise polynomials. There are distinct advantages to be gained from the use of equal order linear piecewise polynomials elements, as mentioned earlier, however there are certain limitations to this method and these are highlighted by the following numerical example.

4.2.1.5 Weak Form.

The *weak* form of (4.22) is reached by applying the classical Galerkin method, as presented in the section describing 1D Advection Diffusion. Ignoring external forces terms the weak form of the Stokes equation is given as:

$$G(\boldsymbol{u}, \boldsymbol{w}, p, q) = \int_{\Omega} \boldsymbol{w} \cdot \Big[-\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}, p) \Big] d\Omega + \int_{\Omega} q \cdot [\nabla \cdot \boldsymbol{u}] d\Omega = 0 \tag{4.27}$$

where $\boldsymbol{w}$ and q are velocity and pressure weighting or test functions respectively. Applying the Greens theorem to the integral of the diffusion term $\int_{\Omega} \boldsymbol{w} \cdot [\nabla \cdot \boldsymbol{\sigma}] d\Omega$ it follows:

$$\int_{\Omega} \boldsymbol{w} \cdot [\nabla \cdot \boldsymbol{\sigma}] d\Omega = -\int_{\Omega} \nabla \boldsymbol{w} : \boldsymbol{\sigma} d\Omega + \int_{\Gamma} \boldsymbol{w} \cdot \boldsymbol{\sigma} \boldsymbol{n} d\Gamma \tag{4.28}$$

Since the stress tensor $\boldsymbol{\sigma}$ is symmetric it follows that:

$$\int_{\Omega} \nabla \boldsymbol{w} : \boldsymbol{\sigma} d\Omega \equiv \int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{w}) : \boldsymbol{\sigma} d\Omega \tag{4.29}$$

Using the Dirichlet boundary condition, where $\boldsymbol{w} = 0$ on Γ, (4.28) can be rewritten by substituting in (4.29):

$$\int_{\Omega} \boldsymbol{w} \cdot [\nabla \cdot \boldsymbol{\sigma}] d\Omega = -\int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{w}) : \boldsymbol{\sigma} d\Omega \tag{4.30}$$

By substituting (4.30) into (4.27) the boundary value problem of (4.22)-(4.23) reads: Find $u^h \in \mathcal{V}^h$ and $p^h \in \mathcal{P}^h$ for all $w^h \in \mathcal{W}^h$ and $q^h \in \mathcal{P}^h$.

$$G(\boldsymbol{u}^h, \boldsymbol{w}^h, p^h, q^h) = -\int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{w}^h) : \boldsymbol{\sigma}(\boldsymbol{u}^h, p^h) d\Omega + \int_{\Omega} q^h \cdot [\nabla \cdot \boldsymbol{u}^h] d\Omega = 0 \tag{4.31}$$

4.2.2 Numerical Example - Lid Driven Cavity.

The Lid Driven cavity example is used to demonstrate the shortcomings of using equal order piecewise linear velocity-pressure elements:

Model.

The solution is obtained for Re = 100, with solution velocity, density and fluid viscosity as $u = 1$ $v = 0$, $\rho = 1$ and $\mu = 0.01$ respectively, with increasingly refined meshes, using a 2D 3 noded triangular element.

Geometry.

The cavity has unit depth and unit width, and the boundary conditions of the problem are represented in the diagram below:

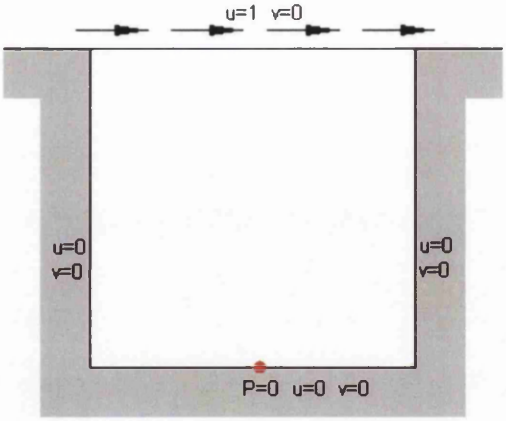


Figure 4.5: Model: Geometry and Boundary Conditions

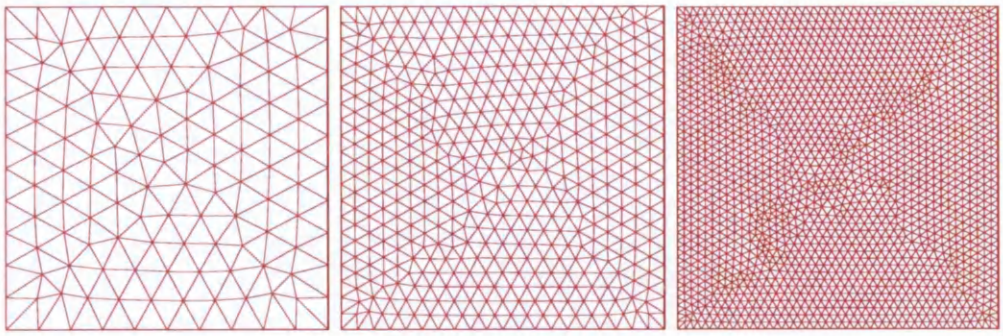
Boundary Conditions.

The velocity boundary conditions are given as zero for the entire boundary in both x and y directions, with the exception of the top boundary (including the corner nodes)[11] which is prescribed a unit horizontal velocity. Pressure is fixed at an arbitrary point, which for this example, is midpoint of the bottom boundary.

Mesh.

In order to highlight the failure of progressive mesh refinement to produce a stable solution the example is carried out for 3 mesh sizes of increasing refinement, i.e. 232 (910) &{3604} elements and 137 (496) & {1883} nodes, see Figure 4.6.

[11] Known as *flow past a 'leaky' cavity*

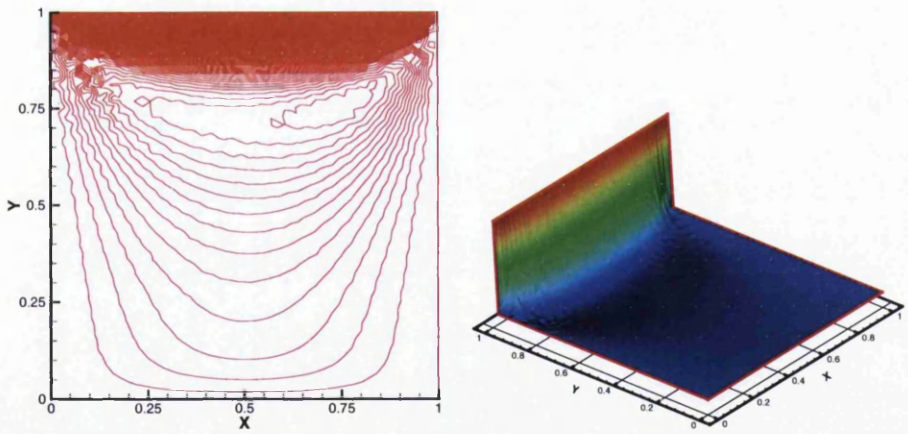


(a) 232 elements (b) 910 elements (c) 3604 elements

Figure 4.6: Meshes: 232 (910) and {3604} elements and 137 (496) and {1883} nodes

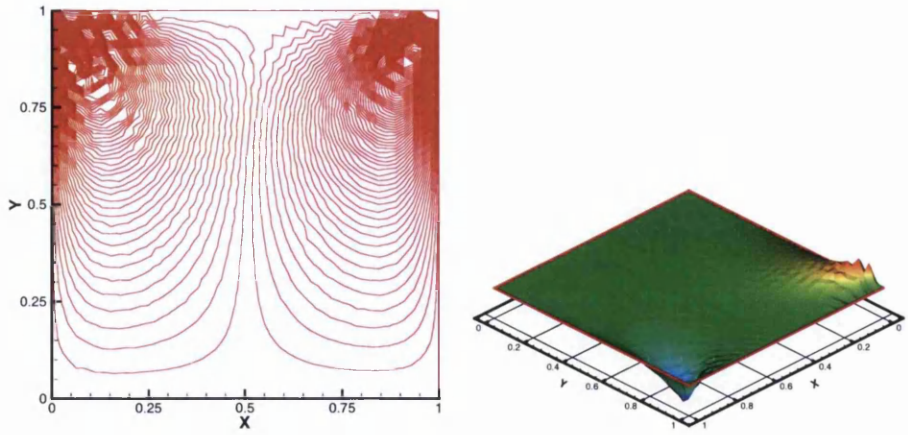
Numerical Results.

Figure 4.7→Figure 4.9 show the solution results for the lid driven 'leaky' cavity. The figures represent the results for 3604 element mesh for velocity and pressure, for both 2D and 3D plots.



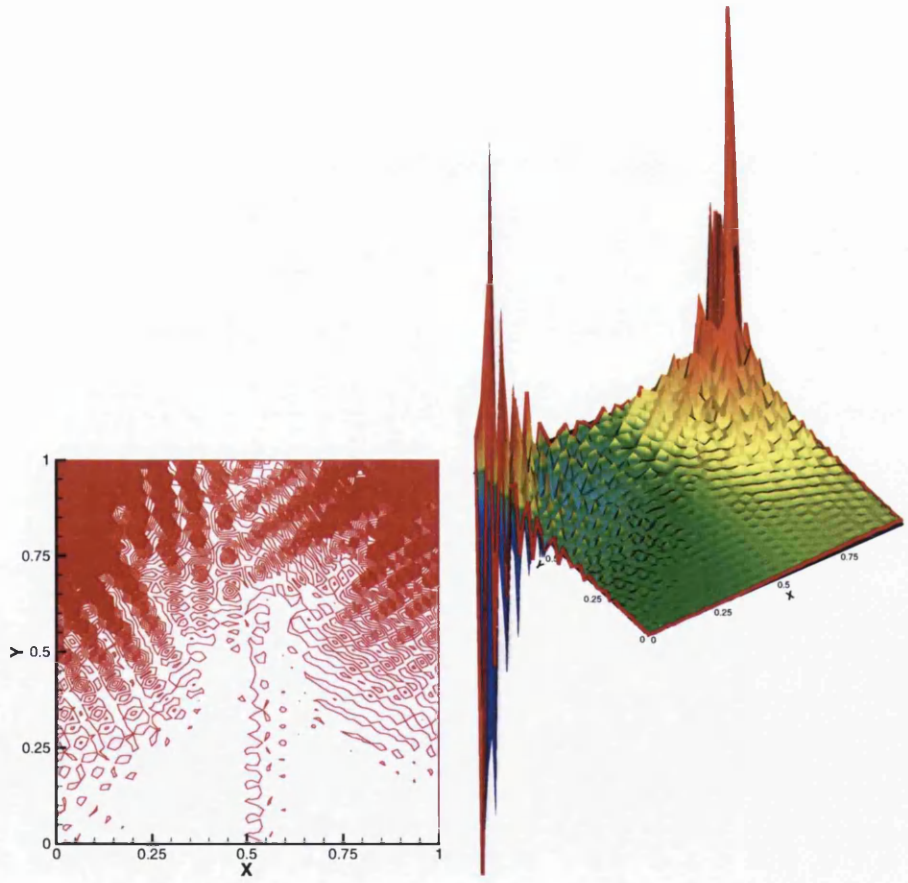
(a) 2D (b) 3D

Figure 4.7: Velocity in x-direction u for lid driven cavity example for 3604 element mesh



(a) 2D (b) 3D

Figure 4.8: Velocity in y-direction v for lid driven cavity example for 3604 element mesh



(a) 2D (b) 3D

Figure 4.9: Pressure p for lid driven cavity example for 3604 element mesh

Discussion.

Figure 4.9 show the results from the simulation for the largest of the 3 meshes, i.e. 3604 elements. It is clear to see that the pressure field is affected by the spurious oscillations, and it was noted that the pressure solution showed no signs of improvement through the use of mesh refinement.

The results for the velocity plots are seemingly not affected by the oscillations in the pressure field, as demonstrated in Figure 4.7 & Figure 4.8. This observation remains unchanged irrespective of the mesh density. However this result is expected as the Stokes equations lack the *convection* terms which previously spoiled the velocity field for the 1D advection diffusion example, see Section4.1.

As mentioned previously the inability of the classical Galerkin method to properly resolve the pressure field is entirely due its failure to satisfy the Babuška-Brezzi condition. If the velocity and pressure finite element spaces are not chosen in accordance to the Babuška-Brezzi constraint then the solutions using convenient velocity-pressure interpolations, such as those used here, will suffer and the good stability and convergence properties of the Galerkin method will be lost.

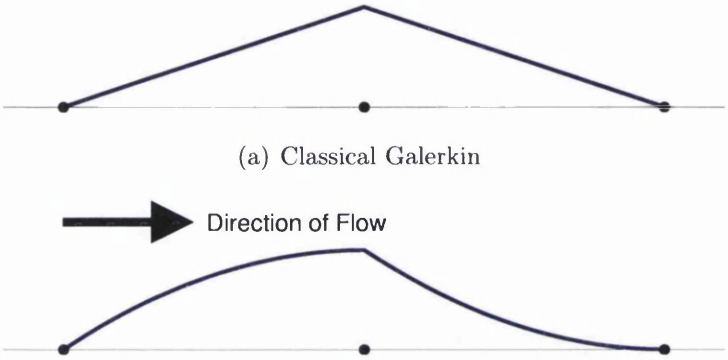
Conclusion.

As shown in the previous two sections, the classical Galerkin method suffers from instabilities which can lead to spurious oscillations in both velocity and pressure fields. The root of the problem is its inability i) to properly resolve rapidly changing interior and boundary layers in the velocity field and ii) to properly consider the Babuška & Brezzi condition, leading to problems in the pressure field. It has been shown that certain techniques, such as mesh refinement, can reduce or cancel entirely the oscillations in the velocity field, however nothing has been presented so far in this work that allows us to circumvent the Babuška & Brezzi condition.

In the next section a technique will be developed that will allow solution of the incompressible fluid dynamics problems. This method is commonly known as the Stabilized Finite Element Method and is present here.

4.3 Stabilization.

The problems encountered from the use of the classical Galerkin method has lead to considerable interest into stabilized methods. In 1975 Zienkiewicz *et al* [88] first discussed the shortcomings of the classical Galerkin method and introduced the first Finite Element stabilization schemes, known as the Upwind Finite Element formulation. This method was based upon a modified weighting functions which weighted the upstream element more than the downstream element, i.e. modelled in the form of a Petrov-Galerkin formulation, see Figure 4.10:



(a) Classical Galerkin

(b) Petrov-Galerkin

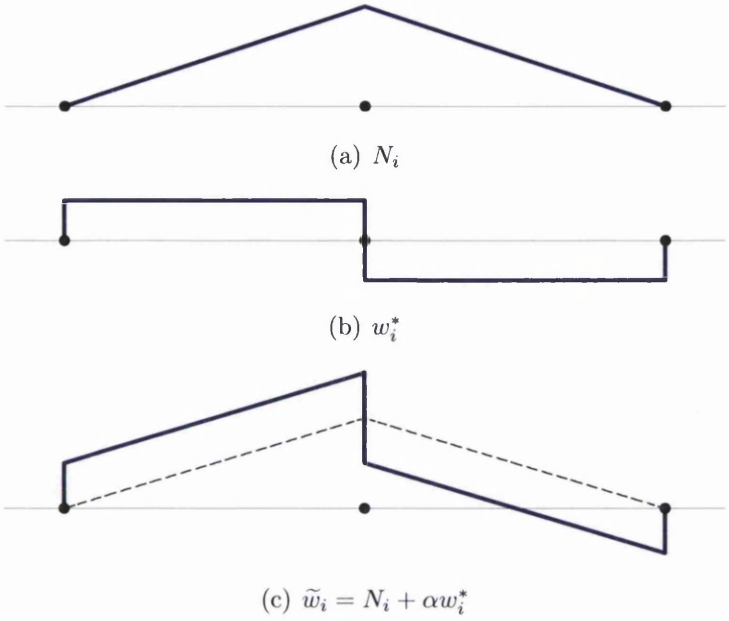
Figure 4.10: Representative 1D Shape functions

For the Petrov-Galerkin formulation the weighting function may be selected from a different class of functions than the approximate solution, i.e. Galerkin formulation $w_i = N_i$ whilst for the Petrov-Galerkin formulation $w_i \neq N_i$ or more specifically:

$$\widetilde{w}_i = N_i + \alpha w_i^* \quad \text{where} \quad \int_{\Omega_e} w_i^* dx = \mp \frac{h^e}{2} \tag{4.32}$$

where $\widetilde{w}_i$ is the Petrov-Galerkin weighting function and h^e is the characteristic length of the element. See Figure 4.11.

However the initial attempts in creating a workable Petrov-Galerkin scheme where not completely successful, with the method suffering severe shortcomings when it was applied to all but the most simple simulations. Since the initial forays into creating workable stabilization methods there has been further work, carried out by a number of authors, including Codina [22], Galeo and Gomes [34], Hughes *et al* [43] and Heinrich *et al* [38], and references within. The result was a practically convenient modification of the classical Galerkin variational form of the problem.



(a) N_i

(b) w_i^*

(c) $\widetilde{w}_i = N_i + \alpha w_i^*$

Figure 4.11: Streamline-upwinding/Petrov-Galerkin shape function. (a)+(b)=(c)

This method adds artificial diffusion to combat the oscillations caused by a dominant convection term and artificially compensates for the circumvention of the Babuška-Brezzi condition. The end result is stabilization which suppresses the oscillations both in the velocity and pressure fields and allows for the use of equal order $\boldsymbol{u}$-p interpolations, normally unstable within the Galerkin framework, but which become convergent in the stabilization framework enhancing stability and keeping consistency.

Among those most commonly used stabilization methods are the Streamline Upwind/Petrov-Galerkin (SUPG) in combination with the Pressure Stabilizing/Petrov-Galerkin (PSPG) and Galerkin Least Squares (GLS), see Hannani [37]. In the following section it will be shown that whilst the SUPG and GLS methods are different, they coincide for linear piecewise elements, such as those used within this work.

4.3.1 Velocity Stabilization - SUPG.

The spurious oscillations in the velocity field were observed by Brooks *et al* [16] who noted that they had no physical meaning. Further studies into this phenomenon by Donea [28] suggest that the classical Galerkin framework produces a system that is under diffusive. Donea transformed the Galerkin discretization back into a partial differential equation and compared the result with the origi-

nal governing equation. He found that the classical Galerkin scheme introduces a truncation error in the diffusion operator and this truncation error has a negative value for all values of the Peclet number. Therefore he concluded that a modified version of the original governing equations was actually being solved by the classical Galerkin method, one with a reduced diffusion coefficient. It was also proved that this error increased as the value of Pe grew and in this situation no stable solution is guaranteed.

The most obvious solution to this problem is to re introduce the lost diffusion into the Galerkin formulation artificially. The process of adding artificial diffusion[12]is not as simple as it may seem. The value of this artificial diffusion will vary over the domain and is dependent on the local values of velocity, diffusion and mesh size, however an incorrect value will result in an over diffusive or under diffusive solution which are both undesired results.

Hughes and Brookes noted that artificial diffusion caused unwanted and excessive crosswind diffusion[13] in 2D and 3D simulations, and in a series of papers Hughes and Brooks [43] proposed methods which allowed for the artificial diffusion to be applied only in streamline direction i.e. in the direction a_x and not transversely. This observation lead to the concept of Streamline Upwinding (SU) schemes. Initially the SU weighting function was added only to the convective term and it was found to be accurate for 1D problems but accuracy was still an issue for multidimensional problems. In a subsequent paper by Hughes and Brooks [44] this technique was applied to all terms in the equation to obtain a consistent formulation and the Streamline Upwinding/Petrov-Galerkin method was born.

4.3.1.1 Streamline Upwinding/Petrov-Galerkin Formulation.

Consider the weak formulation of the 1D advection diffusion equation, as introduced in Section 4.1.1.5, with added artificial diffusion:

$$\sum_{i=1}^{n_{el}} \int_{x_i}^{x_{i+1}} \left[w^h \cdot \dot{x}^h + w^h \cdot (a_x u^h_{,x}) + (\mu + \mu_{art}) w^h_{,x} \cdot u^h_{,x} \right] dx = 0 \qquad (4.33)$$

where it is assumed that $\mu_{art} = \mu_{art}(a^e_x, \mu, \Delta x)$, and that $\mu_{art} \to 0$ as $\Delta x \to 0$ in order to maintain consistency of the original equations, see Johnson and Saranen [51] and Hirsch [39].

[12]Otherwise known as balancing diffusion

[13]Diffusion normal to the streamlines

Applying the artificial diffusion only in the streamline direction the formulation (4.33) becomes:

$$\sum_{i=1}^{n_{el}} \int_{x_i}^{x_{i+1}} \left[w^h \cdot \dot{\not{u}}^h + w^h \cdot (a_x u^h_{,x}) + \mu w^h_{,x} \cdot u^h_{,x} + \tau a_x w^h_{,x} \cdot a_x u^h_{,x} \right] dx = 0 \quad (4.34)$$

This is known as the Streamline diffusion method. Because of the inclusion of a_x into the artificial diffusion term, μ_{art} becomes τ in order to account for the inclusion of time $\Rightarrow \tau = \tau(t, a^e_x, \mu, \Delta x)$. Note that it is important to maintain the consistency of the scheme, and due to the choice of linear elements, τ must tend to zero as the spatial mesh is refined i.e. $\tau \rightarrow 0$ as $\Delta x \rightarrow 0$, see Jansen [49] and Droux *et al* [29]. Should Semi-Discrete linear Finite Element interpolations be used then (4.33) can be categorized as a special case of:

$$\sum_{i=1}^{n_{el}} \int_{x_i}^{x_{i+1}} \left[w^h \cdot \dot{\not{u}}^h + w^h \cdot (a_x u^h_{,x}) + \mu w^h_{,x} \cdot u^h_{,x} \right] dx \quad (4.35)$$

$$+ \sum_{i=1}^{n_{el}} \int_{x_i}^{x_{i+1}} \left[\tau_{SUPG} \left(a_x w^h_{,x} \right) \left[\dot{u}^h + a_x u^h_{,x} + \mu u^h_{,xx} \right] \right] dx = 0 \quad (4.36)$$

Equ (4.35) can rewritten in condensed form:

$$\sum_{i=1}^{n_{el}} \int_{x_i}^{x_{i+1}} \widetilde{w}^h \left[\dot{\not{u}}^h + a_x u^h_{,x} + \mu u^h_{,xx} \right] dx = 0 \quad (4.37)$$

where

$$\widetilde{w}^h = \underbrace{w^h}_{\text{Galerkin Weighting}} + \underbrace{\tau_{SUPG} \left(a_x w^h_{,x} \right)}_{\text{UpwindWeighting}} \quad (4.38)$$

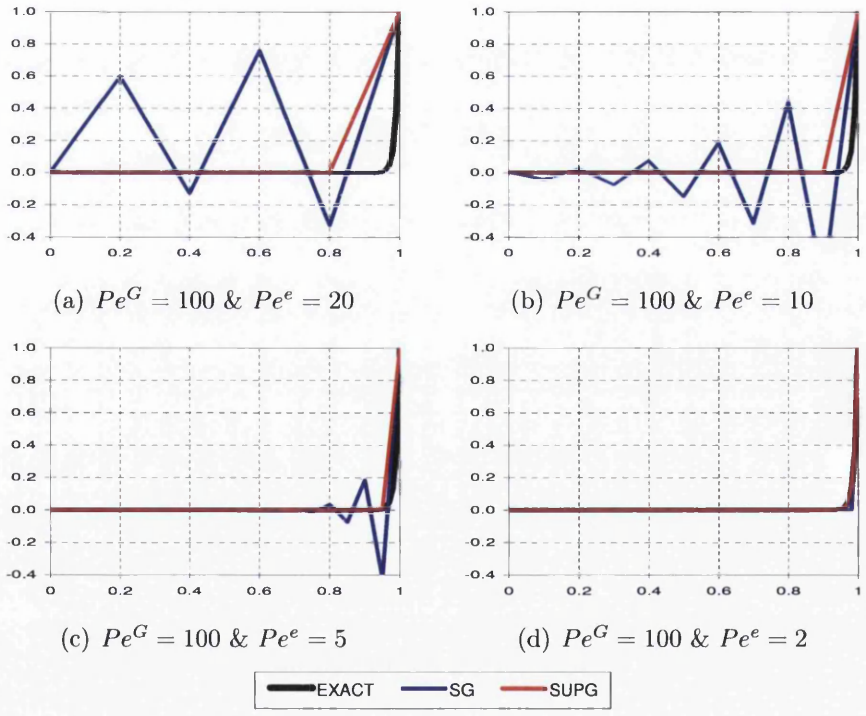
When (4.37) is compared with the Petrov-Galerkin formulation (4.32) it is obvious that (4.37) is a Petrov-Galerkin formulation. However to recognize the fact that the artificial diffusion is only added in the direction of the flow the formulation (4.35) is often known as the Streamline-upwinding/Petrov-Galerkin method.

The SUPG formulation is applied in the repeat of the numerical example, see Section 4.1.2, to demonstrate the advantages of the SUPG stabilization.

4.3.2 Numerical Example - Stabilized 1D Advection Diffusion Fixed Boundary Problem.

In Section 4.1 the classical Galerkin method was shown to be deficient in the analysis of convection dominated problems and this was proved conclusively by 1D Advection-Diffusion example that followed. The solutions from the example were rendered useless by oscillations which had no physical meaning; it was noted that positive results, in comparison to the exact solution of the problem, could be obtained by reducing the element size. However this is not a necessarily the best method as this leads to an increase in computational time.

In a continuation of the 1D Advection-Diffusion example, Section 4.1.2, Figure 4.3 is repeated here with the SUPG stabilization term included in the boundary value problem. Only the results which suffered from insufficient numerical damping, i.e. $Pe > 1$ are presented. The results are conclusive of the improvement derived from including the SUPG term into the classical Galerkin formulation.



(a) $Pe^G = 100$ & $Pe^e = 20$

(b) $Pe^G = 100$ & $Pe^e = 10$

(c) $Pe^G = 100$ & $Pe^e = 5$

(d) $Pe^G = 100$ & $Pe^e = 2$

Figure 4.12: Solution improvement of SUPG Stabilization: Comparison of methods - Classical Galerkin & Streamlined-Upwinding / Petrov-Galerkin with Exact solution

4.3.3 Pressure Stabilization - PSPG.

The instabilities found within the pressure field if the Babuška & Brezzi condition is disregarded were discussed in Section 4.2. It was shown that the classical Galerkin method is unable to satisfy the Babuška & Brezzi condition when convenient equal order $\boldsymbol{u} - p$ interpolations are used. The result from the numerical example i.e. Stokes equations - lid driven cavity, showed that the pressure field was spoiled by spurious oscillations and even mesh refinement proved to be ineffective in their removal.

In this section pressure stabilizing methods are used to circumvent of the Babuška & Brezzi condition, and it is shown that it is possible, through their application, to remove the instabilities caused by incorrect consideration of the Babuška & Brezzi condition.

4.3.3.1 Pressure Stabilizing/Petrov-Galerkin Formulation.

Hughes *et al* [45] considered the difficulties faced when solving the Stokes equations, and correctly understood that the key requirement was the satisfaction of a stability condition and that it involved both pressure and velocity interpolations. He suggested a new formulation which not only satisfied the Babuška & Brezzi condition, and possessed better stability properties than the classical Galerkin for the Stokes equations but also allowed for equal order interpolations to be used. The proposed formulation, given by Hughes reads: Find $u^h \in \mathcal{V}^h$ and $p^h \in \mathcal{P}^h$ for all $w^h \in \mathcal{W}^h$ and $q^h \in \mathcal{P}^h$.

$$\sum_{i=1}^{n_{el}} \int_{\Omega_e} \widetilde{\boldsymbol{w}}^h \cdot \left[\nabla p^h - 2\mu \nabla \cdot \boldsymbol{\varepsilon}(\boldsymbol{u}^h)\right] + q^h \cdot [\nabla \cdot \boldsymbol{u}^h] d\Omega = 0 \tag{4.39}$$

where

$$\widetilde{\boldsymbol{w}}^h = \underbrace{\boldsymbol{w}^h}_{\text{Galerkin Weighting}} + \underbrace{\tau_{PSPG} \nabla q^h}_{\text{PressureWeighting}} \tag{4.40}$$

By virtue of the fact that the weighting function is $\boldsymbol{w}^h$ with an added term, the above formulation can be classified as a Petrov-Galerkin method, and is commonly known as Pressure Stabilizing/Petrov-Galerkin method or PSPG.

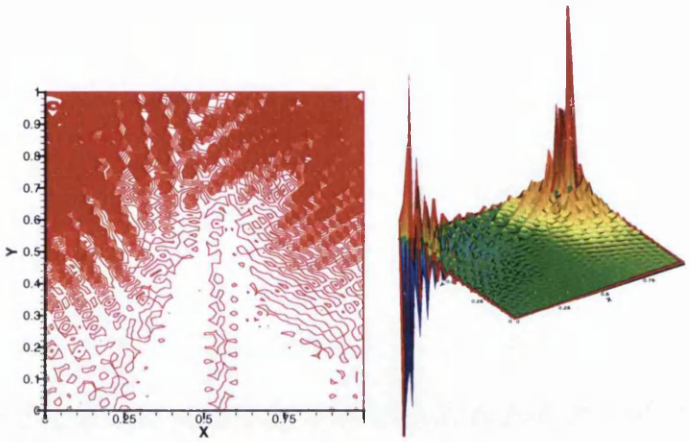
Therefore in a similar manner to the advection diffusion problem, the improved formulation represents the *classical Galerkin formulation + stabilizing term* and $\tau_{PSPG} \rightarrow 0$ as $h^e \rightarrow 0$ in order to maintain consistency.

Details of the implementation of this procedure are given later in Chapter 5.

4.3.4 Numerical Example - Stabilized Lid Driven Cavity.

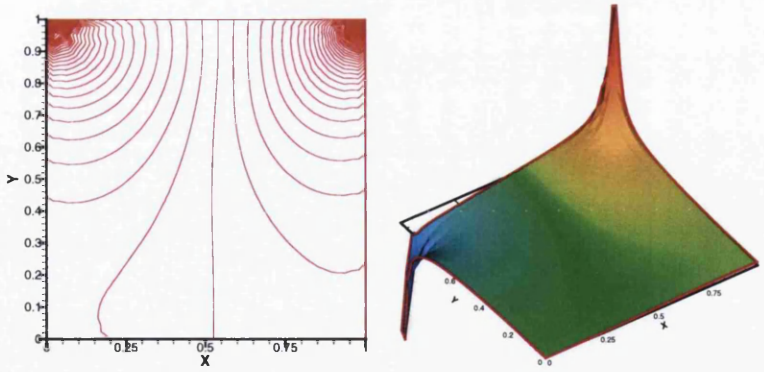
The example introduced earlier, see section 4.2.2, as a show case of the problems caused by incorrect consideration of the Babuška-Brezzi condition is presented briefly to qualify previous statements that stabilization is capable of successfully suppressing the oscillations in the pressure field.

In a repeat of the example introduced in Section 4.2.2, the pressure plot using the stabilized formulation is shown below. In order for a direct comparison a non stabilized pressure plot is also shown:



(a) (b)

Figure 4.13: Pressure plot using a refined mesh without stabilization



(a) (b)

Figure 4.14: Pressure plot using a refined mesh for the stabilized PSPG formulation

4.3.5 Galerkin Least Squares Formulation - GLS.

The conceptual framework of the Galerkin/Least Squares method was originally developed by Hughes and Brooks, [16], [46] as a generalized SUPG method for higher order elements, as part of an extended investigation into stabilized methods. In the Galerkin/Least Squares method the classical Galerkin formulation is supplemented by the Least Squares term of the residual of the governing equations. Significantly the method allows for the use of low and equal order interpolation of the pressure and velocity fields, thus providing an efficient FE framework whilst preserving the accuracy of the Galerkin method and enhanced stability for advection dominated problems. In particular see Hughes *et al* [46] and Brooks *et al* [16] regarding stabilization for advection dominated problems and Hughes [45] for pressure stabilization.

The Galerkin/Least Squares(GLS) variational formulation is obtained by adding a further term to the classical Galerkin variational formulation. The extra term is a weighted Least Squares term of the governing equation and is added as shown below for the case of the full Navier-Stokes equations:

$$G_{GLS}(\boldsymbol{u}^h, \dot{\boldsymbol{u}}^h, \boldsymbol{w}^h, p^h, q^h) = G(\boldsymbol{u}^h, \dot{\boldsymbol{u}}^h, \boldsymbol{w}^h, p^h, q^h) \tag{4.41}$$

$$+ \sum_{i=1}^{n_{el}} \int_{x_i}^{x_{i+1}} L\tau R^h \, dx = 0 \tag{4.42}$$

$$L(\boldsymbol{w}^h, q^h) = \rho\Big[\dot{\boldsymbol{w}}^h + (\boldsymbol{w}^h \cdot \nabla)\boldsymbol{w}^h\Big] - \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{w}^h, q^h) \tag{4.43}$$

$$R^h(\boldsymbol{u}^h, p^h) = \rho\Big[\dot{\boldsymbol{u}}^h + (\boldsymbol{u}^h \cdot \nabla)\boldsymbol{u}^h\Big] - \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}^h, p^h) \tag{4.44}$$

where $R^h(\boldsymbol{u}^h, p^h)$ is the residual of the strong form of the problem (4.3), $L(\boldsymbol{w}^h, q^h)$ is the Linear differential operator and τ is the stabilization parameter, and is regarded as the weighting function of the Least Squares term. Note that the Least Squares term tends to zero as $\boldsymbol{u}^h \rightarrow \boldsymbol{u}$ or $\Delta \boldsymbol{x} \rightarrow 0$ and the variational formulation is consistent for any value of τ, however this is only true for quadratic or higher order elements. The elements used for both 1D and 2D simulations in this work are linear, and in this situation the second derivatives disappear, therefore to maintain the consistency of the formulation τ must tend to zero as the spatial mesh is refined, see Jansen [49] and Droux *et al* [29] for further details regarding the consistency of this scheme. As a result the diffusive part vanishes. In addition the Semi-Discrete formulation means that $\dot{\boldsymbol{w}}$ must also be omitted.

4.3.6 SUPG/PSPG Formulation.

In section 4.3.1 the SUPG method was introduced as possible solution to the oscillatory behaviour caused by convection dominated flows as discussed in Section 4.1. Subsequently the PSPG method was shown to circumvent the Babuška & Brezzi 4.2 successfully. Tezduyar [81] suggested that the two formulations could be combined to form the SUPG/PSPG method, also see Tezduyar [77] for a review of Stabilized Finite Element Methods for the incompressible Navier-Stokes equations

Here the combined SUPG/PSPG formulation is presented for the case of the full incompressible viscous Navier-Stokes equations.

$$G_{SUPG/PSPG}(\boldsymbol{u}^h, \dot{\boldsymbol{u}}^h, \boldsymbol{w}^h, p^h, q^h) = G(\boldsymbol{u}^h, \dot{\boldsymbol{u}}^h, \tilde{\boldsymbol{w}}^h, p^h, q^h) \tag{4.45}$$

where $\tilde{\boldsymbol{w}}$ is the modified SUPG/PSPG weighting function. Equ (4.38) and (4.40) combine to give:

$$\tilde{\boldsymbol{w}}^h = \underbrace{\boldsymbol{w}^h}_{\text{Galerkin Weighting}} + \underbrace{\tau_{SUPG}\,\rho(\boldsymbol{w}^h \cdot \nabla)\boldsymbol{u}^h}_{\text{Upwind Weighting}} + \underbrace{\tau_{PSPG}\nabla q^h}_{\text{Pressure Weighting}} \tag{4.46}$$

Equ (4.46) is often rewritten in a shortened version i.e. $\tilde{\boldsymbol{w}}^h = \boldsymbol{w}^h + \boldsymbol{\delta}^h + \boldsymbol{\epsilon}^h$, with τ_{SUPG} and τ_{PSPG} replaced by τ_u and τ_p respectively to represent velocity and pressure stabilization, thus allowing for independent control of the velocity and pressure stabilization:

$$\boldsymbol{\delta}^h = \tau_u \rho(\boldsymbol{w}^h \cdot \nabla)\boldsymbol{u}^h \tag{4.47}$$

$$\boldsymbol{\epsilon}^h = \tau_p \nabla q^h \tag{4.48}$$

where $\boldsymbol{\delta}^h$ is the specific velocity weighting function and $\boldsymbol{\epsilon}^h$ is the specific pressure weighting function.

4.3.7 SUPG/PSPG or GLS.

The reader will note that the SUPG/PSPG and the GLS formulations become identical if the Finite Element interpolation is linear, as the diffusion term in the Least Squares term disappears and if $\tau_u = \tau_p$.

Both formulations exhibit good convergence and stability properties, and in comparison to the classical Galerkin method both prove to be superior, as was shown by Johnson *et al* [51].In this thesis only linear elements are considered and in this situation both formulations coincide. However for simplicity the formulation will be called under one name i.e. SUPG/PSPG.

4.3.7.1 Stabilization Parameter.

The correct choice of stabilization parameter τ is imperative if good solution accuracy is to be assured, and numerous texts can be found on this subject, including Johnson and Saranen [51], Dettmer and Peric *et al* [27] and Donea and Huerta [28]. τ is defined by:

$$\tau = \frac{h}{2|\boldsymbol{u}^h|}\zeta \tag{4.49}$$

where

$$\zeta = \frac{\beta_1}{\sqrt{1+\left(\frac{\beta_1}{\beta_2 \mathrm{Re}^h}\right)^2}} \tag{4.50}$$

and the element Reynolds number Re^h:

$$\mathrm{Re}^h = \frac{h|\boldsymbol{u}^h|\rho}{2\mu} \tag{4.51}$$

where the variables β_1 and β_2 are scaling parameters, and $\boldsymbol{u}^h$ represents the fluid velocity in the element whilst h equals the characteristic length, which for 2D $h = 2\sqrt{A/\pi}$ is adopted, for element areas A.

Both Johnson and Saranen [51] and Hughes *et al* [45] have considered the requirements that govern the choice of τ in order to achieve the best rate of convergence. Johnson and coworkers only considered the convection-dominated limits case of the Navier-Stokes equations and they concluded that the order of τ is $O(h^2)$. Whilst Hughes considered the limits for diffusion dominated flows and arrived at conclusion that τ should be in the order of $O(h)$. The manner in which this criteria is met is described below:

$$\lim_{Re^h \to 0} \zeta = \beta_2 Re^h \qquad \text{Diffusion limit} \qquad \Rightarrow \tau = \frac{\beta_2}{4\mu}h^2 = O((h)^2) \tag{4.52}$$

$$\lim_{Re^h \to \infty} \zeta = \beta_1 \qquad \text{Convection limit} \qquad \Rightarrow \tau = \frac{\beta_1}{2\mu\rho}h = O(h) \tag{4.53}$$

Equ (4.52) demonstrates how the parameters β_1 and β_2 limit the behaviour of ζ. For this thesis the values of β_1 & β_2[14] are defined as:

$$\beta_1 = 30 \qquad \beta_2 = 100 \tag{4.54}$$

Therefore the values of τ_u and τ_p are defined as:

$$\tau_u = \frac{h}{2|\boldsymbol{u}^h|}\frac{1}{\sqrt{1+\left(\frac{3}{\mathrm{Re}^h}\right)^2}} \quad , \quad \tau_p = \frac{h}{2|\boldsymbol{u}^h|}\frac{\beta_1}{\sqrt{1+\left(\frac{3\beta_2}{\mathrm{Re}^h}\right)^2}} \tag{4.55}$$

Details of the implementation of this procedure are given in Chapter 5.

[14]The effect of varying β was noted to be small unless the values of β are at the extremes.

4.4 Numerical Example - Stabilized Lid Driven Cavity.

The second example will be for a flow past a '*non leaky*' cavity using the stabilized Navier-Stokes equations for incompressible viscous flow, and the results will be compared directly to those obtained by Ghia *et al* [15]. The purpose of this example is to highlight the excellent solutions obtained through the use of the PSPG/SUPG method for relatively coarse meshes.

Model.

In order to compare the results with those given by Ghia *et al* the simulation was carried out for comparable Reynolds numbers. Both u and ρ are fixed at unity and fluid viscosity μ was varied in order to regulated Re, i.e. $\mu = 0.01 \Rightarrow Re = 100$, $\mu = 0.0025 \Rightarrow Re = 400$, $\mu = 0.001 \Rightarrow Re = 1000$, $\mu = 0.0002 \Rightarrow Re = 5000$, & $\mu = 0.0001 \Rightarrow Re = 10000$. The problem was modeled with increasingly refined meshes, using a 2D 3 noded triangular element.

Geometry.

The cavity has unit depth and unit width, and the boundary conditions of the problem are represented in the diagram below:

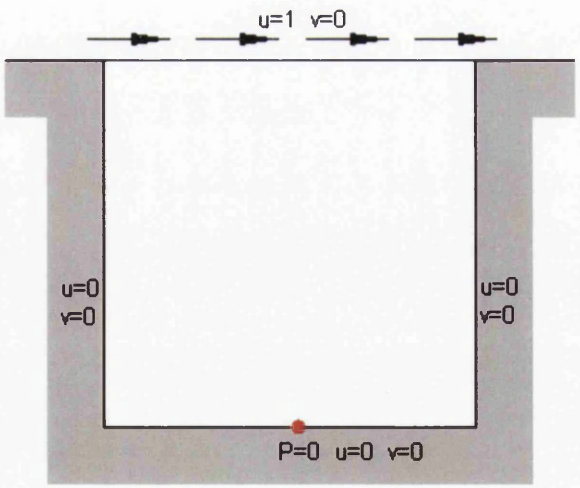


Figure 4.15: Model: Geometry and Boundary Conditions

[15] Ghia's results are regularly used to bench mark results for flow past a 'non leaky' cavity - produced using a Finite Difference scheme on very dense meshes

Boundary Conditions.

The velocity boundary conditions are given as zero for the entire boundary in both x and y directions, with the exception of the top boundary (excluding the corner nodes)[16] which is prescribed a unit horizontal velocity. Pressure is fixed at an arbitrary point, which for this example, is midpoint of the bottom boundary.

Mesh.

3 mesh sizes of increasing refinement, i.e. 3604, (8076) and [15818] elements and 1883, (4159) and [8076] nodes were used, see Figure 4.16.

(a) 232 elements (b) 910 elements (c) 3604 elements

Figure 4.16: Meshes: 3604 (8076) and {15818} elements and 1883 (4159) and {8076} nodes

Numerical Results.

Figure 4.17 shows the pressure and velocity contour plots for Re = 100, 1000, 10000 for the 'non-leaky' lid driven cavity example.

Figure 4.18 shows the velocity profile for u and v for the horizontal and vertical axes of symmetry respectively. They are presented and compared with results by Ghia.

[16] Known as *flow past a 'non leaky' cavity*

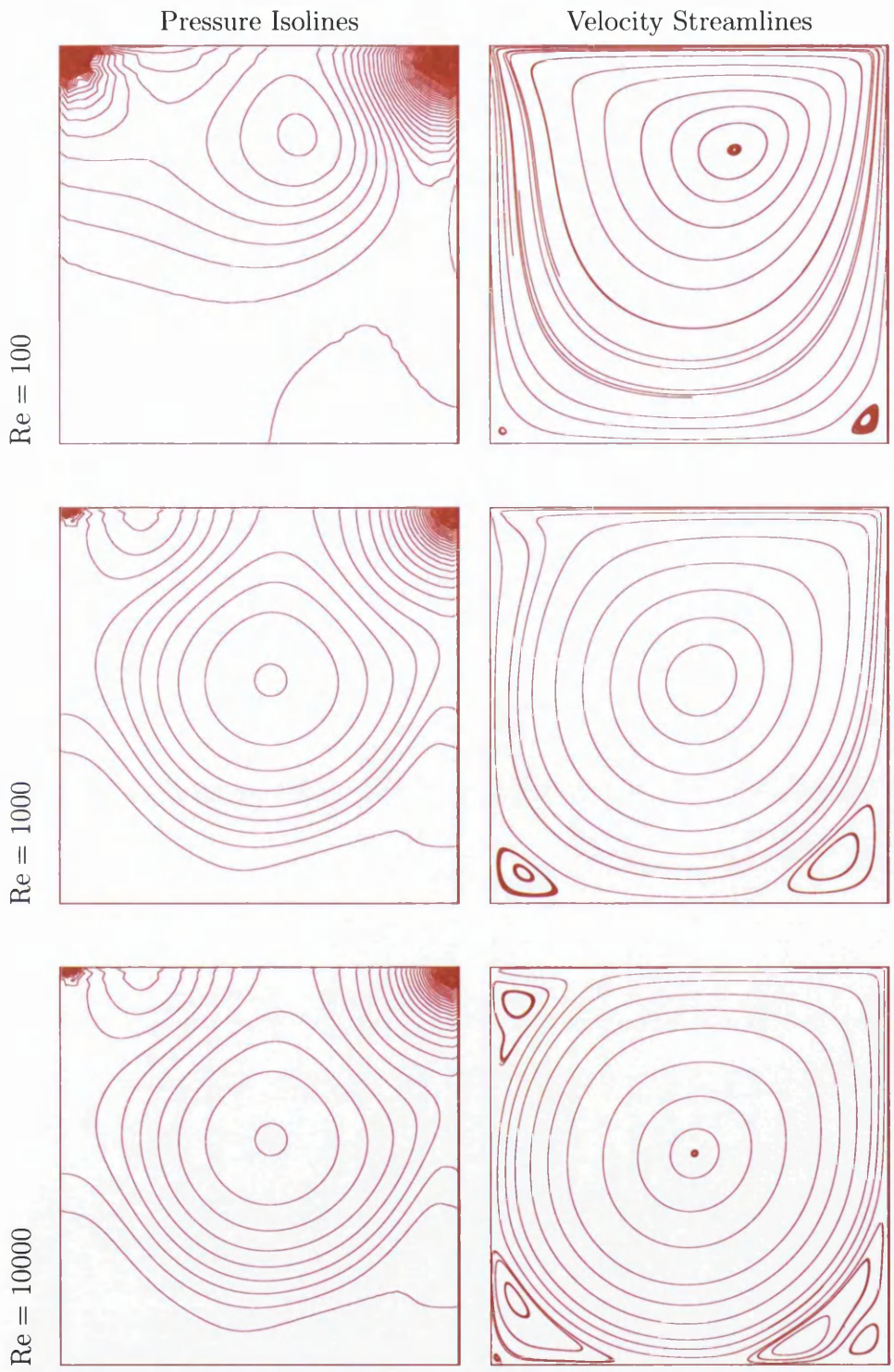
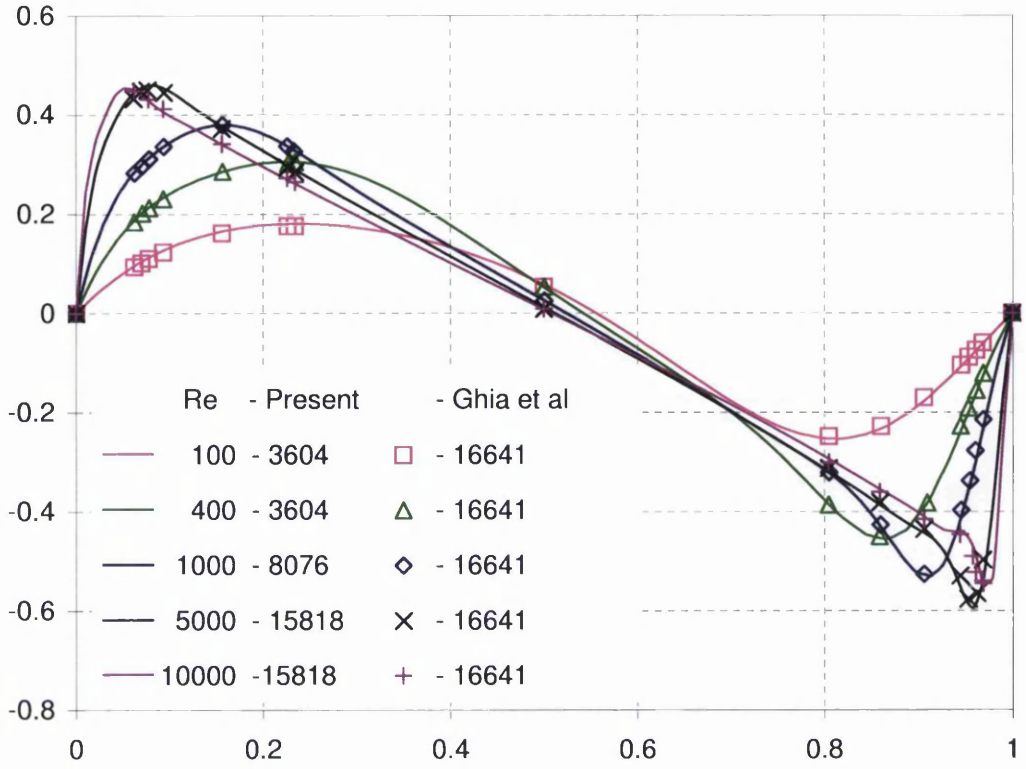
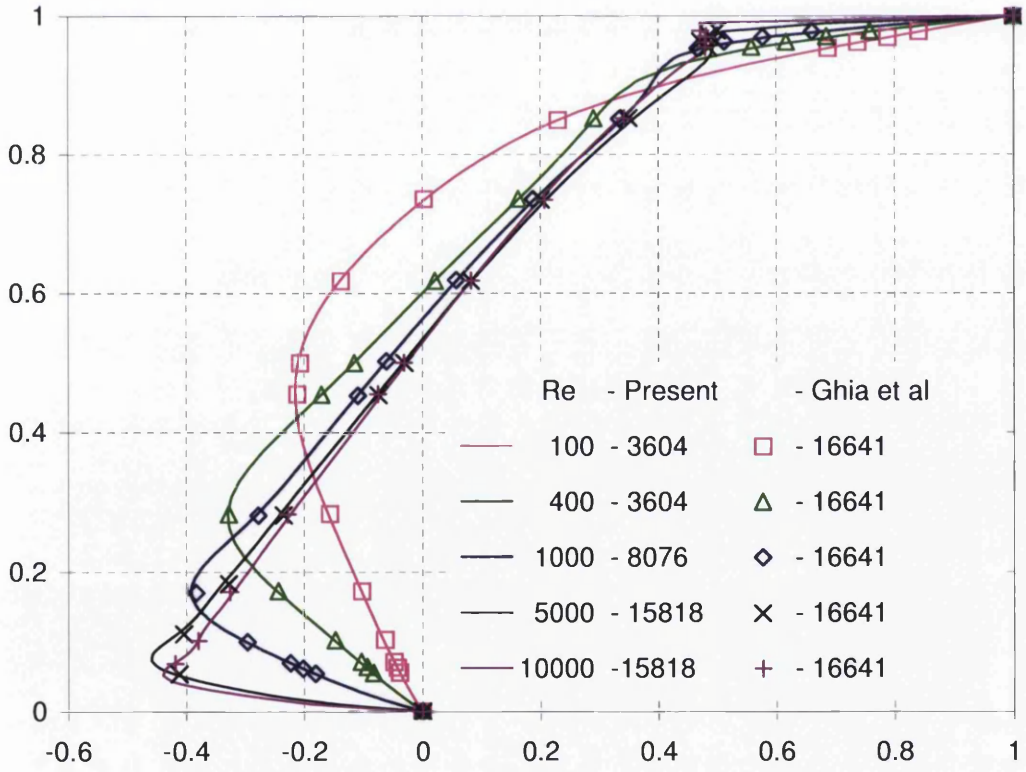


Figure 4.17: Pressure and Velocity contour plots for the lid driven cavity flow problem for an 'non leaky' cavity for $\mathrm{Re} = 100, 1000, 10000$



(a) y-velocity v profile along the horizontal axis of symmetry



(b) x-velocity u profile along the vertical axis of symmetry

Figure 4.18: Comparison of numerical results performed using the SUPG/PSPG formulation with those obtained by Ghia *et al*

Discussion.

It is clear from Figure 4.6 that there is a marked improvement in solution due to the implementation of the stabilizing SUPG/PSPG formulation, when compared to solutions were stabilization is absent. Both the velocity and pressure contour plots are free of the non-physical oscillations which were prevalent in the classical Galerkin solution seen previously in Section 4.1 & (4.2). In addition to this Figure 4.18 demonstrates that the accuracy of the solution is unaffected by the SUPG/PSPG formulation, as the velocity plots in x and y axes are comparable to the results obtained by Ghia. In addition to this the meshes used are, in some cases, considerably coarser than those used by Ghia, and a reduced level of accuracy could be expected.

Conclusion.

The results from this example highlight the benefits of the SUPG/PSPG formulation over the classical Galerkin method, i.e. circumvention of the Babuška & Brezzi condition, resulting in an improved pressure field. In addition added diffusion in the streamline direction counteracts the effects of the under diffusive classical Galerkin method, with no loss of accuracy.

4.5 Conclusion.

The focus of this chapter has been the classical Galerkin method, its application to the incompressible viscous Navier-Stokes equations, and the techniques that have been developed, namely stabilized methods, to circumvent the inherent and undesirable effects of the classical Galerkin method.

We have shown that the classical Galerkin formulation is under diffusive for convection dominate problems and does not satisfy the Babuška & Brezzi condition when equal order interpolations are used, causing non physical oscillations to occur in both the velocity and pressure fields respectively. Two numerical examples were presented *[Stokes flow problem and 1D advection diffusion problem]* to validate these statements.

A *stabilized* formulation was introduced to circumvent the Babuška & Brezzi condition and correct other failings of the classical Galerkin method. The stabilization (SUPG/PSPG) was added in a Petrov-Galerkin framework and was shown to be successful in controlling non physical oscillations in two numerical examples *[1D advection diffusion, lid driven cavity]*.

In the next chapter the stabilized formulation will be developed for unsteady flows.

Chapter 5

Implicit Time Integration Schemes for the incompressible Navier-Stokes equations focusing on the Generalized-α Method.

Most fluid flow problems can be classified as *transient*, however the computational expense of modelling transient problems, e.g. in particular turbulence, has restricted research. With the ever increasing computational capability of modern day computers the expense is no longer prohibitive and in response to this, research is being focused more and more on unsteady flows, and in particular temporal integration schemes.

In this chapter transient flows are addressed as a continuation to the Finite Element formulations discussed previously in Chapters 3 & 4. Here various temporal integration schemes are developed in conjunction with the spatial Finite Element Method for the solution of *unsteady* incompressible viscous Navier-Stokes equations. Each scheme will be assessed for its suitability and a comparison of methods will be made through the use of numerical examples, which are discussed in detail at the end of this chapter.

5.1 Introduction.

In transient flows it is often necessary to integrate a broad range of temporal and spatial scales over large periods of time, and it has long been realized that a successful time integration scheme needs to possess a certain degree of algorithmic

damping. In particular the scheme needs to control high frequency modes which are often poorly resolved by the spatial discretization of the domain, and can lead to non-physical instabilities if left undamped. However, as previously mentioned in Section 4.3, the application of damping to reduce or nullify non-physical instabilities is a complicated process, and one that must be carefully considered. If left undamped poorly resolved high frequency can cause the iterative solution to collapse, however too much damping and important lower frequency modes will also be damped and the dynamics of the resolved scales will not be faithfully represented. Therefore a time integration scheme featuring controllable numerical dissipation is required, and they are discussed presently.

There are various implicit methods which can be employed to integrate the Navier-Stokes equations in time, and they can be divided into 2 separate categories, *Discrete* time integration and *Time* Finite Element Methods. Within each category there are a variety of schemes, however these schemes are often problem specific with substantially varying properties, performing well for certain problems, yet failing on one or more type of problem. These schemes are unsuitable candidates for an all purpose scheme, and it is uncommon to find a general all purpose scheme that performs well for all problem types. In this work we intend to introduce a time integration scheme, known as the Generalized-α method, as an all purpose scheme suitable for fluid flows.

5.1.1 Combined Time Integration/Spatial Finite Element formulations.

In Chapter 4 we introduced the steady state formulations. In this chapter time integration schemes are presented, and accordingly we introduce an extra domain i.e. the temporal domain. Therefore in addition to a discretized spatial domain the temporal domain must also be considered. There are two recognised schemes which combine time integration schemes with spatial Stabilized Finite Element formulations, and they are separated into schemes which discretize the spatial domain only and those which discretize both spatial and temporal domains, i.e. *Semi-Discrete* and *Space-Time* Finite Element Methods respectively.

Semi-Discrete Finite Element Methods.

Semi-Discrete methods only discretize the spatial domain using the Finite Element Method, whilst leaving a system of ordinary differential equations to be integrated in time. For non moving domains (see Space-Time Finite Element Methods) this temporal integration scheme is considered to be the most appropriate, see Jansen *et al* [50] and Dettmer and Peric *et al* [27] and references within.

Space-Time Finite Element Methods.

In addition to the spatial discretization found in the Semi-Discrete method, the Space-Time method also discretizes the temporal domain. In reality this means that the solutions spaces and weighting functions are given a temporal dependence as well as a spatial dependence. Both Shakib *et al* [68], [69] and Tezduyar *et al* [78] have proposed and analysed Space-Time Finite Element Methods, and found that whilst the solutions were accurate, the extra computational cost could not be justified for standard simulations. However for simulations which involve moving domains, deforming bodies etc, the added cost is considered to be justifiable because the Space-Time formulation allows for a consistent method with which to track the moving boundary.

5.1.2 Stability of Time Integration.

The choice of the integration method is often dependent upon the problem analysed and the reasons for these choices lies in the stability constraints of the phenomenon being analysed. Generally for computational fluid dynamics explicit schemes are used whilst for structural dynamics implicit schemes are preferred. However when fluids and solids are modelled within the same problem, such as for fluid-structure interaction it is sometimes difficult to decide which approach is most suitable.

Explicit Schemes.

For explicit schemes the stability criteria state that, for a linear convection equation, the Courant (C) number must be lower than one, i.e.

$$C = \boldsymbol{u}\Delta t/\Delta \boldsymbol{x} \leqslant 1 \quad \text{where} \quad \boldsymbol{x} = x_i \;\&\; \boldsymbol{u} = u_i \quad \text{for} \quad i = 1,2,3 \qquad (5.1)$$

which simply stated means that the flow cannot advance more than one element spacing during one timestep, in any given coordinate direction[1]. However the timestep restriction imposed by the stability condition can sometimes be stringent compared to the time scales required by the solution. In addition to this explicit schemes require large amounts of data storage to store the history of the solution but do not require factorization of the stiffness matrix to advance in time which is a distinct advantage.

Implicit Schemes.

Implicit methods have the advantage of being typically unconditionally stable in terms of linear analysis, however they require the factorization of the stiffness matrix (at least once every timestep[2]) and therefore require considerably more computational time.

Because the critical time step for an explicit integration scheme of the fluid response is often larger than the critical time step required by the structural response, it is common, for fluid structure interaction problems, to couple different integration operators. In this manner conditionally stable explicit schemes are used to integrate the fluid response whilst using unconditionally stable implicit schemes for the structural response. However difficulties can arise when choosing which method to use, and how they should be coupled. And in addition to this an unconditionally stable implicit scheme is required by incompressible flow simulations for the pressure equations. Therefore in order to avoid such confusion and to deal with the pressure constraints we will implement an Implicit method.

The chosen Implicit time integration method for this work will be Generalized-α method for all unsteady flow simulations. The Generalized-α method is an implicit method introduced as an extension of the so called HHT scheme by

[1] Although this general rule is a governing factor in most explicit schemes there are exceptions to this, namely the Runge-Kutta scheme which violates this restriction within the order of magnitude.

[2] Only when Δt is not constant

Chung and Hulbert [21]. It was initially considered for structural problems only, however Jansen *et al* [50] used this scheme to model the stabilized Navier-Stokes equations with success and this has confirmed by Dettmer *et al* [27]. In this work we intend to confirm the applicability of the Generalized-α method for a number of different fluid dynamics problems.

The *Generalized midpoint rule* will also be considered here in order to provide a comparison of schemes. A scalar model problem will used to present their properties.

5.2 Scalar Model Problem.

In this section a scalar model problem will be presented and this will form the basis for the analysis of the time integration schemes in later sections.

The model problem can be written as a first order ordinary differential equation:

$$\dot{y} = \lambda y \tag{5.2}$$

The initial value problem, consists of finding a function $y = y(t)$ which satisfies (5.2) and the initial condition given by $y(0) = y_0$, where y_0 is given. The exact solution is written as:

$$y = y_0 e^{\lambda t} \tag{5.3}$$

Therefore it follows, from (5.3), that

$$|y_{n+1}| < |y_n| \qquad \lambda > 0 \tag{5.4}$$

$$|y_{n+1}| = |y_n| \qquad \lambda = 0 \tag{5.5}$$

where y_n and y_{n+1} are used to approximate $y(t_n)$ and $y(t_{n+1})$, $t_{n+1} > t_n$ and $\Delta t = t_{n+1} - t_n$. The amplification factor $\mathbb{A}$ is given as:

$$\mathbb{A} = \frac{y_{n+1}}{y_n} \quad \Rightarrow \quad \mathbb{A} = e^{\lambda \Delta t} \tag{5.6}$$

Considering (5.4) & (5.5) the amplification factor has the following condition:

$$|\mathbb{A}| \leqslant 1 \tag{5.7}$$

where $|\mathbb{A}|$ is the spectral radius and is often denoted by ρ.

$$\rho = |\mathbb{A}| \tag{5.8}$$

To make the model problem (5.2) reflect both the advective and diffusive phenomena λ - the eigenvalue associated with a chosen mode, is a complex number, i.e.

$$\lambda = -\xi + iw \tag{5.9}$$

where the difference between w and ξ and their numerical counterparts w^h and ξ^h represents the amount of numerical dispersion and numerical damping, respectively.

From this point the scalar model problem will be used to compare the two timestepping schemes and their respective properties.

5.2.1 Generalized Midpoint Rule.

Introducing at this point the Generalized Midpoint rule for integrating (5.2) from t_n to t_{n+1}.

$$y_{n+\gamma} = \gamma y_{n+1} + (1-\gamma)y_n \tag{5.10}$$

$$\dot{y}_{n+\gamma} = \frac{y_{n+1} - y_n}{\Delta t} \tag{5.11}$$

where $0 \leqslant \gamma \leqslant 1$. Substituting (5.10) & (5.11) into (5.2):

$$y_{n+1} - y_n - \lambda\Delta t\left[\gamma y_{n+1} + (1-\gamma)y_n\right] = 0 \tag{5.12}$$

Rearrangement of (5.12) into the equation for the amplification factor results in:

$$\mathbb{A}^h = \frac{1 + (1-\gamma)\lambda\Delta t}{1 - \gamma\lambda\Delta t} \tag{5.13}$$

Similarly by inserting (5.13) & (5.9) into the equation for the spectral radius the following equation is obtained[3]:

$$\rho^h = \sqrt{\frac{(1-(1-\gamma)\xi\Delta t)^2 + ((1-\gamma)w\Delta t)^2}{(1+\gamma\xi\Delta t)^2 + (\gamma w\Delta t)^2}} \tag{5.14}$$

5.2.1.1 Stability.

It is important that the timestepping scheme is stable in order to ascertain convergence of the algorithm, i.e. $y_n \to y(t_n)$ as $\Delta t \to 0$. Therefore if (5.4) & (5.5) are taken to be the stability conditions, then on evaluation of (5.13) it is found that *unconditional stability* is achieved when:

$$\lim_{\Delta t\lambda \to \infty} \mathbb{A}^h = \frac{1-\gamma}{\gamma} \quad \Rightarrow \quad \gamma \geqslant \frac{1}{2} \tag{5.15}$$

[3]Calculation by Mathematica

For values of $\gamma < \frac{1}{2}$ stability is only achieved for a given value of λ when an upper bound value of the time step is imposed - this is called *conditional stability.* However it can be shown that the time step is proportional to h^2, where h is the mesh size parameter, and this is an inconvenient constraint which is best avoided. Therefore unconditionally stable algorithms are generally preferred. The limit of ρ, for unconditional stability, is given as:

$$\rho_\infty^h = \lim_{\Delta t \to \infty} \rho^h = \frac{1-\gamma}{\gamma} \tag{5.16}$$

More usefully (5.16) can be rearranged to make γ the subject

$$\gamma = \frac{1}{1+\rho_\infty^h} \tag{5.17}$$

where $0 \leqslant \rho_\infty^h \leqslant 1$. When $\rho_\infty^h = 1$ the method is known as the *Trapezoidal Method* and when $\rho_\infty^h = 0$ the *Backward Euler Method*

5.2.1.2 Convergence.

Once again considering the model problem (5.2), written with force term F:

$$\dot{y} - \lambda^h y - F = 0 \tag{5.18}$$

If (5.18) is discretized in both spatial and temporal domains, by substituting in (5.6), (5.10), (5.11) and (5.13):

$$y_{n+1} - \mathbb{A}^h y_n - \frac{\Delta t}{1-\gamma\Delta t\lambda} F_{n+\gamma} = 0 \tag{5.19}$$

Replacing the approximations y_{n+1} and y_n with their exact counterparts, $y(t_{n+1})$ and $y(t_n)$ respectively:

$$y(t_{n+1}) - \mathbb{A} y(t_n) - L_n = \Delta t \cdot \tau \tag{5.20}$$

where $L_n = \frac{\Delta t}{1-\gamma\Delta t\lambda^h} F_{n+\gamma}$ and τ represents the truncation error.

Using a finite Taylor expansion on $y(t_{n+1})$ and $y(t_n)$, in terms of $y(t_{n+\gamma})$, and the model equation to eliminate time derivatives of $y(t_{n+\gamma})$ it is possible to arrive at the following expression[4],

$$\tau = (1-2\gamma)\mathcal{O}(\Delta t^1) + \mathcal{O}(\Delta t^2) \tag{5.21}$$

It is obvious from (5.21) that the Generalized Midpoint rule is first-order accurate method, except for the special case where $\gamma = \frac{1}{2}$ or $\rho_\infty^h = 1$, then the second order accuracy is achieved.

[4]The expansion and resulting algebraic manipulation is time consuming and is not included here, see Hughes [42] for further details

Remark 5.1: Trapezoidal rule.
For convection dominated flow[5]*, where* $\rho_\infty^h = 1$ *(or very near 1),* $\lambda\Delta t \to \infty$ *and* $\mathbb{A} \cong -1$ *the solution exhibits a 'sawtooth' pattern which manifests itself frequently in computations, see Figure 5.1. For such cases it is often appropriate to use a value which is strictly less than one.*

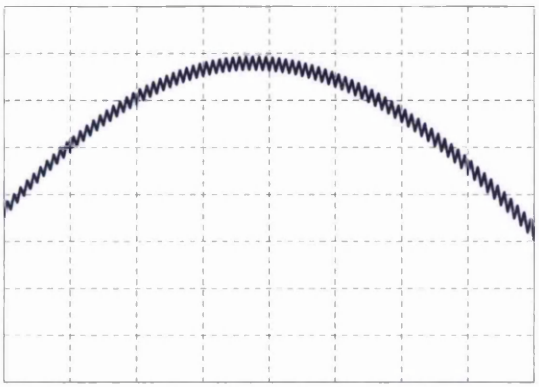


Figure 5.1: Numerical example:- Flow past a circular cylinder. Plot C_L against time for Re = 400 $\Delta t = 0.0025$ $\rho_\infty^h = 1$ with a medium dense mesh

5.2.2 Generalized-α Method.

We introduce at this point the Generalized-α method for integrating (5.2) from t_n to t_{n+1} where $t_{n+1} = t_n + \Delta t$.

$$\dot{y}_{n+\alpha_m} = \lambda y_{n+\alpha_f} \tag{5.22}$$

$$y_{n+1} = y_n + \Delta t \dot{y}_n + \Delta t \gamma (\dot{y}_{n+1} - \dot{y}_n) \tag{5.23}$$

$$\dot{y}_{n+\alpha_m} = \dot{y}_n + \alpha_m (\dot{y}_{n+1} - \dot{y}_n) \tag{5.24}$$

$$\dot{y}_{n+\alpha_f} = y_n + \alpha_f (y_{n+1} - y_n) \tag{5.25}$$

(5.22)→(5.25) can be combined to produce a system which takes the form

$$\boldsymbol{a}\boldsymbol{y}_{n+1} = \boldsymbol{b}\boldsymbol{y}_n \qquad \longrightarrow \qquad \boldsymbol{y}_{n+1} = \boldsymbol{c}\boldsymbol{y}_n \tag{5.26}$$

where the amplification matrix $\boldsymbol{c} = \boldsymbol{a}^{-1}\boldsymbol{b}$ and the solution vector at t_n is given by $\boldsymbol{y}_n = (y_n, \Delta\dot{y}_n)^T$. (5.26) is rewritten in matrix format as

$$\begin{bmatrix} y_{n+1} \\ \Delta t \dot{y}_{n+1} \end{bmatrix} = \frac{1}{d} \begin{bmatrix} \alpha_m - (\alpha_f - 1)\gamma\lambda\Delta t & \alpha_m - \gamma \\ \lambda\Delta t & \alpha_m - 1 + \alpha_f \lambda \Delta t (1-\gamma) \end{bmatrix} \begin{bmatrix} y_n \\ \Delta t \dot{y}_n \end{bmatrix} \tag{5.27}$$

[5]The effect is less for linear problems

where $d = \alpha_m - \alpha_f \gamma \lambda \Delta t$. It is fairly straightforward to show that if $\boldsymbol{y}_n = \boldsymbol{A}\boldsymbol{y}_{n-1}$ is combined with (5.26) and the rate terms, i.e. $\dot{y}_{n+1}$, $\dot{y}_n$ & $\dot{y}_{n-1}$ are removed then it follows that, see Hughes [42]:

$$y_{n+1} = tr(\boldsymbol{c})y_n - det(\boldsymbol{c})y_{n-1} \tag{5.28}$$

Using a finite Taylor expansion on $y(t_{n+1})$ and $y(t_{n-1})$, in terms of $y(t_n)$ in time, in a similar manner to the analysis of the Generalized Midpoint rule, it is possible to arrive at the following expression which assures second order accuracy:

$$\gamma = \frac{1}{2} + \alpha_m - \alpha_f \tag{5.29}$$

Rearrangement of (5.29), and taking into account (5.6) gives:

$$\mathbb{A}^2 - tr(\boldsymbol{c})\mathbb{A} + det(\boldsymbol{c}) = 0 \tag{5.30}$$

which is the characteristic equation for $\boldsymbol{c}$ (see Chung and Hulbert [21] and Zienkiewicz and Taylor [89]), it is possible to find eigenvalues of $\boldsymbol{c}$ to prove stability however this is an enormous task when it would be far simpler to find the stability constraints through the use of limiting values. The Routh-Hurwitz criterion defines the stability constraint.

Remark 5.2: Routh-Hurwitz criterion.
A characteristic polynomial is given below:

$$c_0 z^2 + c_1 z + c_2 = 0 \tag{5.31}$$

The Routh-Hurwitz criterion for stability requires that the modulus of the roots of a stability polynomial or of the associated amplification matrix be less or equal to one or more simply:

$$c_0 > 0 \quad c_1 \geqslant 0 \quad c_2 > 0 \tag{5.32}$$

The limiting values are given as $\Delta t \to 0$ and $\Delta t \to \infty$

$$\lim_{\Delta t \to 0} \mathbb{A}^h = \left\{ 1 - \frac{1}{\alpha_m}, 1 \right\} \tag{5.33}$$

$$\lim_{\Delta t \to \infty} \mathbb{A}^h = \left\{ \frac{2(\alpha_m - \alpha_f) - 1}{2(\alpha_m - \alpha_f) + 1}, 1 - \frac{1}{\alpha_f} \right\} \tag{5.34}$$

Therefore the requirements for unconditional stability are:

$$\alpha_m \geqslant \alpha_f \geqslant \frac{1}{2} \tag{5.35}$$

Once again it is convenient to specify the stability parameters in terms of the spectral radius ρ_∞^h, where:

$$\rho_\infty^h = max(|\mathbb{A}_1|, |\mathbb{A}_2|) \tag{5.36}$$

Therefore it follows from (5.34) that:

$$\alpha_m = \frac{1}{2}\frac{3-\rho_\infty^h}{1+\rho_\infty^h} \qquad \alpha_f = \frac{1}{1+\rho_\infty^h} \tag{5.37}$$

where $0 \leqslant \rho_\infty^h \leqslant 1$. The Generalized-$\alpha$ method has the ability to control the high frequencies damping (within resolution) exactly, for $\rho_\infty^h = 0.0$ the method will remove the highest frequencies, yet for $\rho_\infty^h = 1.0$ it has the property of introducing no damping regardless of the timestep. However $\rho_\infty^h = 1.0$ is equivalent to the Trapezoidal rule which as noted previously suffers occasionally from 'sawtoothing'. If the desired effect is no damping then ρ_∞^h should be given a value close to one but which is not one.

5.3 Step-by-step solution algorithms.

The time integration algorithms i.e. Generalized-α method and the Generalized Midpoint rule, yield incremental time stepping algorithms. The time stepping procedure is explained briefly for both algorithms below.

The scalar model problem, see Section 5.2, is reintroduced in a form which is commonly found in fluid flow problems, i.e. in a nonlinear form:

$$\dot{y} = \lambda(y)\, y \tag{5.38}$$

Following the same steps as applied previously for the linear form we arrive at:

$$y_{n+1} = \mathbb{A}^h(y_{n+1}, y_n)\, y_n \tag{5.39}$$

The dependence of $\mathbb{A}^h$ on the unknown variable produces an implicit scheme for the solution of y_{n+1}.

By the same process the nonlinear time integration schemes also produce implicit solution equations. To solve implicit equations requires an implicit solver,

and a commonly used solver is the Newton Raphson iterative solver, as described previously in Chapter 3. Both Generalized-α method and the Generalized Midpoint rule use this procedure, however due to the difference in time integration used by the two methods the solution procedures are different.

Generalized Midpoint Rule for the Model Problem.

Equations (5.10) & (5.11) are both written in terms of y_{n+1} and y_n, therefore:

$$G(y_{n+1}, y_n) = 0 \tag{5.40}$$

The Newton-Raphson procedure for updating the next timestep are given as:

---

1. Solve $G(y_{n+1}, y_n) = 0$ for y_{n+1}
2. Update $y_n \leftarrow y_{n+1}$
3. Goto 1.

Generalized-α Method for the Model Problem.

The process is similar to that given for the Midpoint rule. Equ (5.22), (5.23), (5.25) and (5.24) are all required to produce equations for the solution at t_{n+1} i.e. $\dot{y}_{n+1}$ and y_{n+1}. The method is shown below:

Equ (5.23) is rearranged to produce

$$\dot{y}_{n+1} = \frac{1}{\gamma \Delta t} y_{n+1} - \frac{1}{\gamma \Delta t} y_n - \frac{(1-\gamma)}{\gamma} \dot{y}_n \tag{5.41}$$

which is substituted into (5.24) to remove the dependence on the unknown term $\dot{y}_{n+1}$:

$$\dot{y}^{(i)}_{n+\alpha_m} = \frac{\alpha_m}{\gamma \Delta t} y^{(i)}_{n+1} - \frac{\alpha_m}{\gamma \Delta t} y_n + (1 - \frac{\alpha_m}{\gamma}) \dot{y}_n \tag{5.42}$$

Equ (5.42) and (5.25) give (5.22) in terms of the only unknown y_{n+1} and this gives:

$$G(y_{n+1}, y_n, \dot{y}_n) = 0 \tag{5.43}$$

As a result; one extra variable $\dot{y}_n$ is required for the solution at the next time step. This addition leads to a slightly more complex iterative procedure:

1. Solve $G(y_{n+1}, y_n, \dot{y}_n) = 0$ for y_{n+1}
2. When Residual $\leqslant$ tol calculate $\dot{y}_{n+1}$ with (5.41)
3. Update $\dot{y}_n, y_n \leftarrow \dot{y}_{n+1}, y_{n+1}$
4. Goto 1

5.3.1 Comparison of Schemes.

Integration Method	Stability	Order of Accuracy	ρ_∞^h	Memory Requirements
Backward Euler	Unconditionally Stable	1	0.0	y_n
Trapezoidal rule	Unconditionally Stable	2	1.0	y_n
Generalized Midpoint rule	Unconditionally Stable	1-2	user defined	y_n
Generalized-α method	Unconditionally Stable	2	user defined	y_n , $\dot{y}_n$

Table 5.1: Comparison of time stepping algorithms

Remark 5.3: Storage requirements.
The Generalized-α method has larger storage requirements in comparison to the other schemes - the number of stored variables from the previous time step has doubled.

Dettmer and Peric [27], with some Backward Euler analysis carried out by Shakib & Hughes [68], conducted an indepth study of each time integration method combined with the Stabilized Finite Element Method using a Fourier analysis and the conclusions drawn are compiled here:

- Convergence is assured for $\mu > 0$ as long as $\Delta x \rightarrow 0$, $\Delta t \rightarrow 0$ and $\tau \rightarrow 0$
- for pure advection i.e. $\mu = 0$ the methods converge for any value of τ

- The accuracy of the solution is mesh dependent and is irrespective of the method used.
- Spatial discretization has an effect on numerical damping when $\tau > 0$.
- τ is a characterizing parameter and may effect the accuracy of the solution if incorrectly chosen.
- the choice of a time dependent τ is ill advised, as reduction of the timestep leads to a reduction in numerical damping for unresolved high frequencies.

Dettmer concluded that the parameter ρ_∞^h is the determining factor for the damping characteristics of two schemes. He also showed that the numerical damping effect on the lower frequencies is less for the Generalized-α method than it is for Generalized Midpoint rule.

5.4 Implementation of Timestepping in FEM.

In this section the stabilized 1D Advection Diffusion and Navier Stokes equations will be combined with the two time integration schemes in preparation for the numerical example which will follow.

5.4.1 1D Advection Diffusion.

The weak form of the stabilized steady 1D advection diffusion equations were previously developed, in Section 4.3.1. The unsteady formulation is easily obtained, through the inclusion of the time derivative (previously crossed out). The stabilized unsteady formulation is written here for convenience:

$$G(w,u,\dot{u}) = \sum_{i=1}^{n_{el}} \int_{x_i}^{x_{i+1}} \left[w^h \cdot \dot{u}^h + w^h \cdot (a_x u^h_{,x}) + \mu w^h_{,x} \cdot u^h_{,x} \right] dx \qquad (5.44)$$
$$+ \sum_{i=1}^{n_{el}} \int_{x_i}^{x_{i+1}} \left[\tau_u \left(a_x w^h_{,x}\right) \left[\dot{u}^h + a_x u^h_{,x} + \mu u^h_{,xx} \right] \right] dx = 0$$

where μ, a_x, u^h, $\dot{u}^h$, τ_u, and w^h are diffusion, free velocity in the x spatial direction, the trial function of the scalar unknown, the time derivative of the scalar unknown, the stabilization parameter and the test function respectively. For further discussion regarding the stabilization, discretization etc, please refer to Chapter 4. Note that the expression (5.44) is semi discrete, i.e. spaces has been

Integration Method	a_1	a_2	a_3	a_4	a_5	a_6
Backward Euler	1	0	0	$\frac{1}{\Delta t}$	$\frac{1}{\Delta t}$	0
Trapezoidal rule	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{\Delta t}$	$-\frac{1}{\Delta t}$	0
Generalized Midpoint rule	γ	$1-\gamma$	0	$\frac{1}{\Delta t}$	$-\frac{1}{\Delta t}$	0
Generalized-α method	α_f	$1-\alpha_f$	0	$\frac{1}{\Delta t}\frac{\alpha_m}{\gamma}$	$-\frac{1}{\Delta t}\frac{\alpha_m}{\gamma}$	$1-\frac{\alpha_m}{\gamma}$

Table 5.2: Timestepping Parameters $a_1 \rightarrow a_6$ described in terms of α_f, α_m and γ

discretized, whilst time remains continuous.

Equ (5.44) can rewritten in condensed form:

$$\sum_{i=1}^{n_{el}} \int_{x_i}^{x_{i+1}} \widetilde{w}^h \left[\dot{u}^h + a_x u^h_{,x} + \mu u^h_{,xx} \right] dx = 0 \tag{5.45}$$

where

$$\widetilde{w}^h = \underbrace{w^h}_{\text{Galerkin Weighting}} + \underbrace{\tau_u \left(a_x w^h_{,x} \right)}_{\text{UpwindWeighting}} \tag{5.46}$$

and τ_u, developed in Section 4.3.7.1, has been rewritten for the 1D case. τ_u for 1D advection-diffusion problems is often referred to as τ_{space}, see Shakib and Hughes [68], however for consistency the term τ_u will be used throughout to define the velocity stabilization parameter,

$$\tau_u = \frac{\Delta x}{2a_x} \frac{1}{\sqrt{1 + \left(\frac{3}{Pe^h}\right)^2}} \qquad \text{where} \qquad Pe^h = \frac{2a_x \Delta x}{2\mu} \tag{5.47}$$

In order to emphasize the method used to incorporate time integration into the weak form, (5.44) has been re written with $\dot{u}^h$ and u^h replaced by $\dot{u}^h_\beta$ and u^h_α:

$$G(w, u, \dot{u}) = \tag{5.48}$$
$$\sum_{i=1}^{n_{el}} \int_{x_i}^{x_{i+1}} \left[w^h \cdot \dot{u}^h_\beta + w^h \cdot \left(a_x u^h_{\alpha,x} \right) + \mu w^h_{,x} \cdot u^h_{\alpha,x} + \tau_u \left(a_x w^h_{,x} \right) \left[\dot{u}^h_\beta + a_x u^h_{\alpha,x} \right] \right] dx = 0$$

where $\dot{u}^h_\beta$ and u^h_α represent linear combinations of $\dot{u}^h_n$, u^h_n and u^h_{n+1}:

$$u^h_\alpha = a_1 u^h_{n+1} + a_2 u^h_n + a_3 \dot{u}^h_n \tag{5.49}$$

$$u^h_\beta = a_4 u^h_{n+1} + a_5 u^h_n + a_6 \dot{u}^h_n \tag{5.50}$$

For each time integration scheme the coefficients a_i, where $i = 1 \rightarrow 6$, have a definite value which is determined by the time step size and spectral radius, and these are given in Table 5.2. The parameters γ, α_m, and α_f in Table 5.2 are defined by:

$$\alpha_m = \frac{1}{2}\frac{3-\rho^h_\infty}{1+\rho^h_\infty} \qquad \alpha_f = \frac{1}{1+\rho^h_\infty} \qquad \gamma = \frac{1}{2} + \alpha_m - \alpha_f \tag{5.51}$$

5.4.2 2D Incompressible viscous Navier-Stokes.

5.4.2.1 The Governing Equations.

Recall the full incompressible viscous Navier-Stokes equations:

$$\rho\left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right] - \nabla\cdot\boldsymbol{\sigma} = 0 \tag{5.52}$$

$$\nabla\cdot\boldsymbol{u} = 0 \tag{5.53}$$

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) \tag{5.54}$$

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T) \tag{5.55}$$

5.4.2.2 Domain and BC's.

The domain Ω is an open region of $\mathbb{R}^{n_{sd}}$, therefore $\Omega \cup \Gamma$ where Γ is the piecewise smooth boundary and n_{sd} is the number of spatial dimensions.

Prescribed velocities g on the surface Γ_g $\qquad \boldsymbol{u} = \boldsymbol{g} \quad \forall \boldsymbol{x} \in \Gamma_g$ (5.56)

Prescribed fluxes h on the surface Γ_h $\qquad \boldsymbol{\sigma}\cdot\boldsymbol{n} = \boldsymbol{h} \quad \forall \boldsymbol{x} \in \Gamma_h$ (5.57)

where $\boldsymbol{x} \in \Omega \subset \mathbb{R}^{n_{sd}}$, and the outward unit normal vector, to the fluid surface, is $\boldsymbol{n}$.

5.4.2.3 Finite Element Spaces.

The Finite Element approximation spaces necessary for the correct discretization of the governing equations are defined as: The spatial discretization occurs in a finite dimensional space $H^1(\Omega)^{n_{dim}}$ defined as:

$$H^1(\Omega)^{n_{dim}} = \{\phi^h | \phi^h \in C^0(\Omega), \phi^h|_{\boldsymbol{x}\in\Omega^e} \in P_k^{n_{dim}}\}$$

where ξ denotes the set of elements resulting from the Finite Element discretization of the computational domain Ω into Ω^e subdomains. The trial and test functions for the velocity field, $\mathcal{V}$ and $\mathcal{W}$ respectively are as follows:

$$\mathcal{V}^h = \{\boldsymbol{u}^h \in H^1(\Omega)^{n_{dim}},\ \boldsymbol{u}^h|_{\boldsymbol{x}\in\Omega^e} \in P_k^{n_{dim}}(\Omega^e),\ \boldsymbol{u}^h|_{\boldsymbol{x}\in\Gamma_g} = \boldsymbol{g}\}$$
$$\mathcal{W}^h = \{\boldsymbol{w}^h \in H^1(\Omega)^{n_{dim}},\ \boldsymbol{w}^h|_{\boldsymbol{x}\in\Omega^e} \in P_k^{n_{dim}}(\Omega^e),\ \boldsymbol{w}^h|_{\boldsymbol{x}\in\Gamma_g} = \boldsymbol{0}\}$$

whilst for the pressure field

$$\mathcal{P}^h = \{p^h |\ p^h \in C^0(\Omega) \cap \boldsymbol{L}_0^2(\Omega),\ p^h|_{\boldsymbol{x}\in\Omega^e} \in P_k(\Omega^e)\}$$

where $H^{1^{n_{dim}}}$ denotes the usual Sobolev space of functions with square-integrable values and derivatives on Ω, C^0 is the space of continuous functions on Ω, $\boldsymbol{L}_0^2$ denotes space of the square-integrable functions $\boldsymbol{L}^2(\Omega)$ modulo constants, whilst $P_k(\Omega^e)^{n_{dim}}$ is the space of all polynomials defined on Ω^e, complete to order $k \geq 1$ and n_{dim} is the number of degrees of freedom. Note: since there are no pressure boundary conditions imposed on the pressure field, the test and trial spaces for pressure coincide.

5.4.2.4 Finite Element Discretization.

As introduced previously the General Finite Element discretization applied to the velocity and pressure fields can be written as:

$$\boldsymbol{u}^h = \sum N_I(\boldsymbol{x})\boldsymbol{u}_I \qquad \boldsymbol{w}^h = \sum N_I(\boldsymbol{x})\boldsymbol{w}_I \tag{5.58}$$
$$p^h = \sum N_I(x)p_I \qquad q^h = \sum N_I(x)q_I \tag{5.59}$$

5.4.2.5 Weak Form

We can write the stabilized Petrov-Galerkin formulation, in terms of $\boldsymbol{u}_\alpha^h$ & $\dot{\boldsymbol{u}}_\beta^h$ [6], as: Find $u^h \in \mathcal{V}^h$ and $p^h \in \mathcal{P}^h$ such that $\forall w^h \in \mathcal{W}^h$ and $\forall q^h \in \mathcal{P}^h$ [7].

[6] As shown previously in Section 5.4.1

[7] The integral of diffusion term $\int_\Omega \boldsymbol{w} \cdot [\nabla \cdot \boldsymbol{\sigma}] d\Omega$ is simplified to $\int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{w}^h) : \boldsymbol{\sigma}^h d\Omega$ through the use of the Greens theorem, as demonstrated in Section 4.2.1.5.

$$
\begin{aligned}
G(\boldsymbol{u}_\alpha^h, \dot{\boldsymbol{u}}_\beta^h, \boldsymbol{w}^h, p_{n+1}^h, q^h) = & \int_\Omega \boldsymbol{w}^h \cdot \rho \left(\dot{\boldsymbol{u}}_\beta^h + \boldsymbol{u}_\alpha^h \cdot \nabla \boldsymbol{u}_\alpha^h \right) d\Omega - \int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{w}^h) : \boldsymbol{\sigma}(\boldsymbol{u}_\alpha^h, p_{n+1}^h) d\Omega \\
& + \int_\Omega q^h \cdot [\nabla \cdot \boldsymbol{u}_\alpha^h] d\Omega - \int_{\Gamma_h} \boldsymbol{w}^h \cdot \boldsymbol{h} d\Gamma \\
& + \sum_{e=1}^{nel} \int_{\Omega^e} \boldsymbol{\delta}^h [\rho(\dot{\boldsymbol{u}}_\beta^h + (\boldsymbol{u}_\alpha^h \cdot \nabla)\boldsymbol{u}_\alpha^h) - \nabla p_{n+1}^h)] d\Omega \\
& + \sum_{e=1}^{nel} \int_{\Omega^e} \boldsymbol{\epsilon}^h [\rho(\dot{\boldsymbol{u}}_\beta^h + (\boldsymbol{u}_\alpha^h \cdot \nabla)\boldsymbol{u}_\alpha^h) - \nabla p_{n+1}^h)] d\Omega = 0
\end{aligned}
\tag{5.60}
$$

where $\dot{\boldsymbol{u}}_\beta^h$ and $\boldsymbol{u}_\alpha^h$ represent linear combinations of $\dot{\boldsymbol{u}}_n^h$, $\boldsymbol{u}_n^h$ and $\boldsymbol{u}_{n+1}^h$:

$$
\boldsymbol{u}_\alpha^h = a_1 \boldsymbol{u}_{n+1}^h + a_2 \boldsymbol{u}_n^h + a_3 \dot{\boldsymbol{u}}_n^h \tag{5.61}
$$

$$
\boldsymbol{u}_\beta^h = a_4 \boldsymbol{u}_{n+1}^h + a_5 \boldsymbol{u}_n^h + a_6 \dot{\boldsymbol{u}}_n^h \tag{5.62}
$$

and $\boldsymbol{\delta}^h$ is the specific velocity weighting function and $\boldsymbol{\epsilon}^h$ is the specific pressure weighting function:

$$
\boldsymbol{\delta}^h = \tau_u \rho (\boldsymbol{w}^h \cdot \nabla) \boldsymbol{u}_\alpha^h \tag{5.63}
$$

$$
\boldsymbol{\epsilon}^h = -\tau_p \nabla q^h \tag{5.64}
$$

The stress tensor $\boldsymbol{\sigma}$, and the deformation rate tensor $\boldsymbol{\varepsilon}$ are:

$$
\boldsymbol{\sigma}(\boldsymbol{u}_\alpha^h, p_{n+1}^h) = -p_{n+1}^h \boldsymbol{I} + 2\mu \boldsymbol{\varepsilon}(\boldsymbol{u}_\alpha^h) \tag{5.65}
$$

$$
\boldsymbol{\varepsilon}(\boldsymbol{u}_\alpha^h) = \nabla^S \boldsymbol{u}_\alpha^h = \frac{1}{2}(\nabla \boldsymbol{u}_\alpha^h + \nabla^T \boldsymbol{u}_\alpha^h) \tag{5.66}
$$

and μ, ρ, $\boldsymbol{u}^h$, p^h, τ_u, τ_p and $\boldsymbol{w}^h$ are diffusion, density, velocity vector, the velocity and pressure stabilization parameters and the test function respectively.

A set of nonlinear equations are obtained in terms of $\boldsymbol{u}_{n+1}$ & p_{n+1} by introducing (5.61)→(5.65) into (5.60) and for each timestep these equations are solved using the Newton-Raphson procedure as described in Section 5.2.

5.5 Numerical Examples.

A total of three numerical examples will be used in the validation of the techniques described in the this chapter. They have been chosen as the best method of highlighting the advantages and disadvantages for each technique discussed, and they are generally regarded to be standard benchmarks for fluid flow problems.

- 1D Advection-Diffusion - Propagation of a rectangular wave.
- 2D Navier-Stokes - Flow past a 2D cylinder.

The numerical examples are briefly discussed here.

Propagation of a rectangular wave.
This example is an extension of the 1D advection diffusion examples from Chapter 4, i.e. the fixed boundary problem is replaced by a propagating rectangular wave. These examples are considered to be the steady and unsteady versions of the same problem, i.e. modelling a jump in the main solution variable.

The main outcome from this example is the demonstration that stabilization is effective, in a manner which can be easily demonstrated graphically, for transient problems as it is for steady state problems.

Flow past a 2D Cylinder.
The viscous incompressible flow past a bluff body is a well recognised natural phenomena, and can be found in a variety of engineering applications, such as in heat exchangers, flow around chimneys, bridge piers, stays, antennae, and arrays of nuclear fuel rods, etc. Consequently the theoretical and practical significance has lead to investigations into the physical mechanisms, arising from the very complex behaviour and the interactive loading between the body and the viscous fluid. These mechanisms have been studied in great detail both experimentally and, recently, also numerically. It is for this reason that this type of problem is perfectly suited to fulfill the requirements of this chapter.

In order to compare the different formulations discussed within this chapter two examples of newtonian incompressible viscous flow past bluff bodies are used to perform the analysis, and they are:

- flow past a circular cylinder, and
- flow past a square cylinder with near wall effects

The two examples can be classified as *Flow past a 2D Cylinder* which is the general description for the series of problems of which the two above belong, yet they can be considered to be sufficiently different to warrant their inclusion.

These examples are suitable for the purposes of this section because there is considerable literature available and they are simple problems to setup and run effectively. In addition to this the results themselves are suited to graphical representation and they are commonly used, as in this thesis, to benchmark new methods.

These three examples will be used to consider the effects to solution accuracy, speed, and stability caused by altering the solution algorithm.

In order that these simulations can be replicated the exact details of each benchmark test are recorded in full.

5.5.1 Programs - Introduction to 1DFLOW & FEAP.

In order to conduct the research effectively it was obvious that the techniques proposed would need to be validated against benchmark tests. Not only are these techniques relatively new, untried & untested, but the complexity of the benchmark tests necessitate the use of computers. As mentioned earlier in this chapter, the Finite Element Method's growth in popularity is a direct result of improvements made in the field of high powered computers, and software. It is for this reason that there are now many commercial Finite Element software packages available. However it was realized from the beginning that it would not be possible to conduct the research for this PhD using any of the standard commercial packages available, as their capabilities did not extend into all aspects of this research. The other options were to either 'design and build' a FEM application, or find a application that could be easily tailored to requirements.

It was apparent that actually writing a FE software application from initial conception to final product would be time consuming, whereas tailoring a basic 'all round' FE application would be relatively simple. It therefore became necessary to find a FE software package that would be easy to modify and be open access, i.e. a freely available source code.
The following FE applications have used to run the simulations for this thesis: 1DFLOW and FEAPpv.

5.5.1.1 1DFLOW.

This software application was designed and written by the author from concept to perform 1D heat transfer simulations. Its initial capabilities included various timestepping schemes, namely General Midpoint rule, Backward Euler, and the Trapezoidal rule implemented in conjunction with the Predictor-Corrector time integration scheme, see Shakib and Hughes [68], Eulerian and Arbitrary Lagrangian Eulerian (ALE) solutions, stabilization in the form of Streamline Upwinding Petrov-Galerkin, and Gaussian elimination solver. As this work evolved further additions have been made, and include Enrichment for fluid structure interaction, augmenting timestepping to include the Generalized-α method. Also added were Newton-Raphson iterative scheme to solve non-linear simulations and a Sparse & Bandwidth solver to improve solution times, all of which were required by the addition of Enrichment.

1DFLOW is used as the basis of all previous and forthcoming one dimensional simulations.

5.5.1.2 FEAPpv - Finite Element Analysis Program *personal version.*

FEAP is a general purpose Finite Element Analysis program, designed for research and educational use. It is was originally written by the research group at Berkeley headed by R.L. Taylor to accompany the reference books 'The Finite Element Method', by O.C. Zienkiewicz and R.L. Taylor. Since its development FEAP has grown in size and capabilities, and now two releases exist, FEAP and FEAPpv. FEAP is now a licensed software application, whereas FEAPpv is still distributed freely for educational and research purposes.

FEAPpv perfectly suited the requirements for this PhD. It is easily understood due to the extensive manuals accompanying the program, it contains a simple graphical interface, and it has been designed assuming the user would wish to increase its capabilities. In addition to this the source code is freely available and regularly updated with new routines. As such FEAPpv v6 is used as the basis of all two dimensional simulations.

To the standard FEAPpv we have added numerous capabilities. They include all the capabilities of 1DFLOW for two dimensions and added to this are meshing capabilities, and a new graphical interface. Some of the additions are enrichment based, and are explained in Chapter 5.

FEAPpv is used as the basis of all previous and forthcoming two dimensional simulations.

5.6 Numerical Example - Propagation of a Rectangular Wave.

The Propagation of a Rectangular wave is often used to validate the implementation of time integration schemes for 1D Advection Diffusion problems. It will be employed to demonstrate and confirm the conclusions drawn previously in this chapter.

Model.

In this example both pure advection ($\mu = 0$ & $Pe^h = \infty$) and advection-diffusion ($\mu = 0.00005$ & $Pe^h = 10$) cases will be considered. The numerical results are be classified by time integration scheme, stabilization factor and the element Courant number C^h, given by:

$$C^h = \frac{\Delta t}{\Delta x} a_x \tag{5.67}$$

The mesh and a_x are both constant for all solutions, therefore in order to vary the Courant number Δt is varied.

Geometry.

$$\Omega =]0, L[\tag{5.68}$$

Boundary Conditions.

$$u(0,t) = 0 \qquad u(L,t) = 0 \tag{5.69}$$

Initial Conditions.

$$u(x,0) = 1 \qquad c \leqslant x \leqslant d \qquad (c = 0.1L, d = 0.2L) \tag{5.70}$$

$$u(x,0) = 0 \qquad \text{elsewhere} \tag{5.71}$$

Mesh.

$$\Delta x = \frac{1}{1000} L \tag{5.72}$$

Exact Solution - Pure Advection.

For this example the diffusion parameter $\mu = 0$, see (4.5) for 1D advection diffusion equation.

$$u(x,t) = 1 \qquad c + a_x t \leqslant x \leqslant d + a_x t \tag{5.73}$$

$$u(x,t) = 0 \qquad \text{elsewhere} \tag{5.74}$$

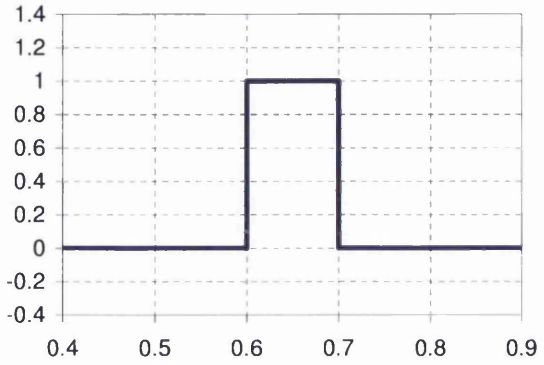


Figure 5.2: Exact solution for the Pure advection case

Exact Solution - Advection Diffusion.

For this example the diffusion parameter is given a nominal value $\mu = 0.00005$.

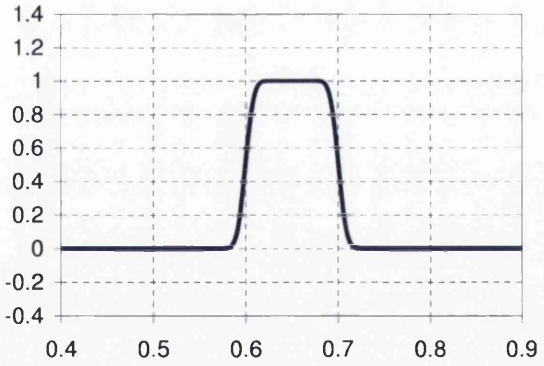


Figure 5.3: Exact solution for the advection diffusion case

Numerical Results.

Figure 5.4 shows the plot of $u(x,t)$ at $t = 0.5s$. Comparison is provided of the classical Galerkin formulation with the SUPG formulation for the pure advection and advection diffusion cases for $C^h = 0.01, 0.10, 1.00$. The results are obtained by using the Generalized-α method $\rho_\infty^h = 0.0$, and $n_{elem} = 1000$.

Figure 5.5 shows the plot of $u(x,t)$ at $t = 0.5s$. Comparison is provided of the classical Galerkin formulation with the SUPG formulation for the pure advection and advection diffusion cases for $C^h = 0.01, 0.10, 1.00$. The results are obtained by using the Generalized-α method $\rho_\infty^h = 0.5$, and $n_{elem} = 1000$.

Figure 5.6 shows the plot of $u(x,t)$ at $t = 0.5s$. Comparison is provided of the classical Galerkin formulation with the SUPG formulation for the pure advection and advection diffusion cases for $C^h = 0.01, 0.10, 1.00$. The results are obtained by using the Generalized Midpoint rule $\rho_\infty^h = 0.0$, and $n_{elem} = 1000$.

Figure 5.7 shows the plot of $u(x,t)$ at $t = 0.5s$. Comparison is provided of the classical Galerkin formulation with the SUPG formulation for the pure advection and advection diffusion cases for $C^h = 0.01, 0.10, 1.00$. The results are obtained by using the Generalized Midpoint rule $\rho_\infty^h = 0.5$, and $n_{elem} = 1000$.

Figure 5.8 shows the plot of $u(x,t)$ at $t = 0.5s$. Comparison is provided of the classical Galerkin formulation with the SUPG formulation for the pure advection and advection diffusion cases for $C^h = 0.01, 0.10, 1.00$. The results are obtained by using the Generalized-α method $\rho_\infty^h = 1.0$, and $n_{elem} = 1000$.

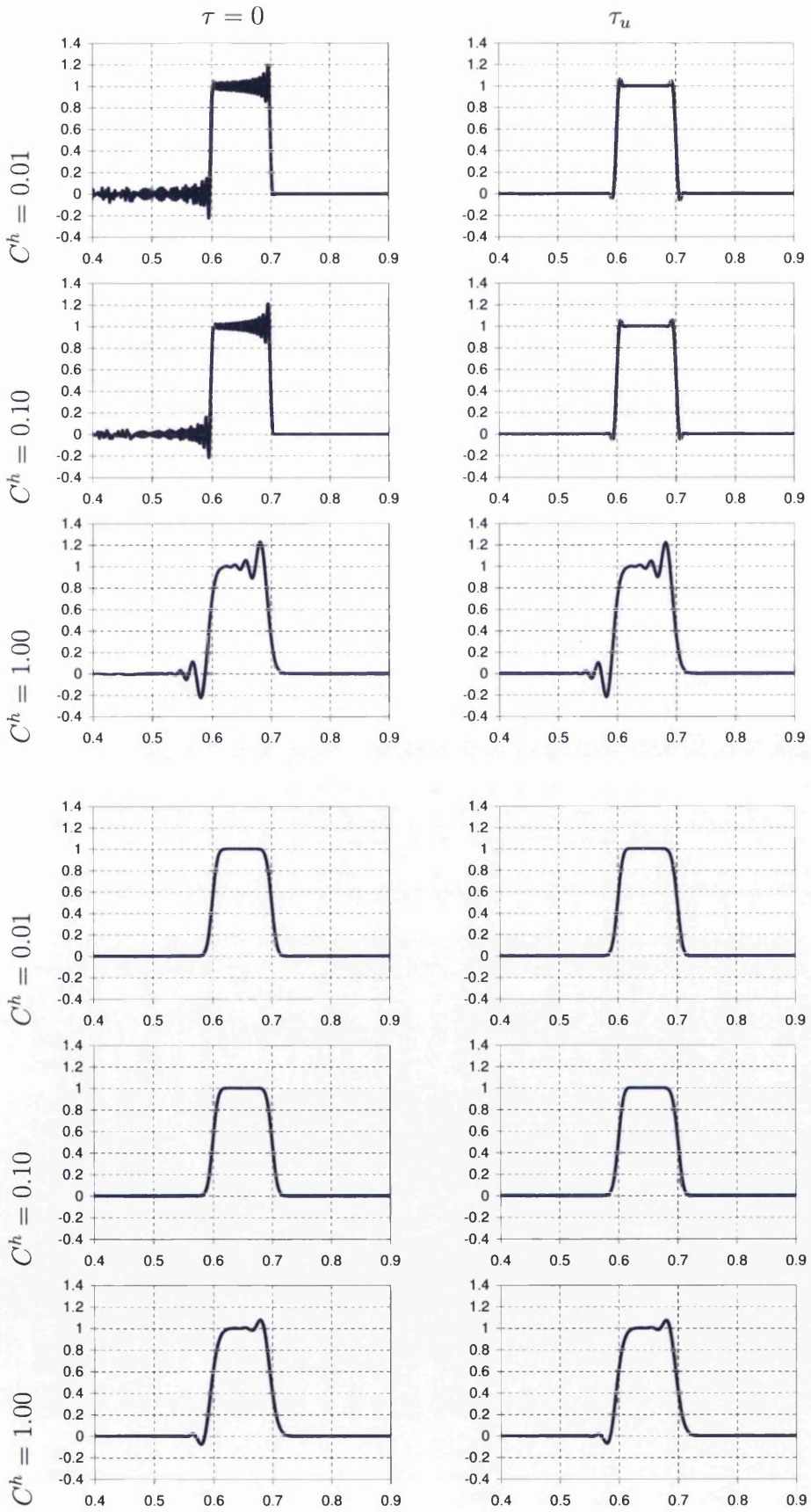


Figure 5.4: Solution for the Generalized-α method $\rho_\infty^h = 0.0$, at t=0.5s for 1000 element mesh

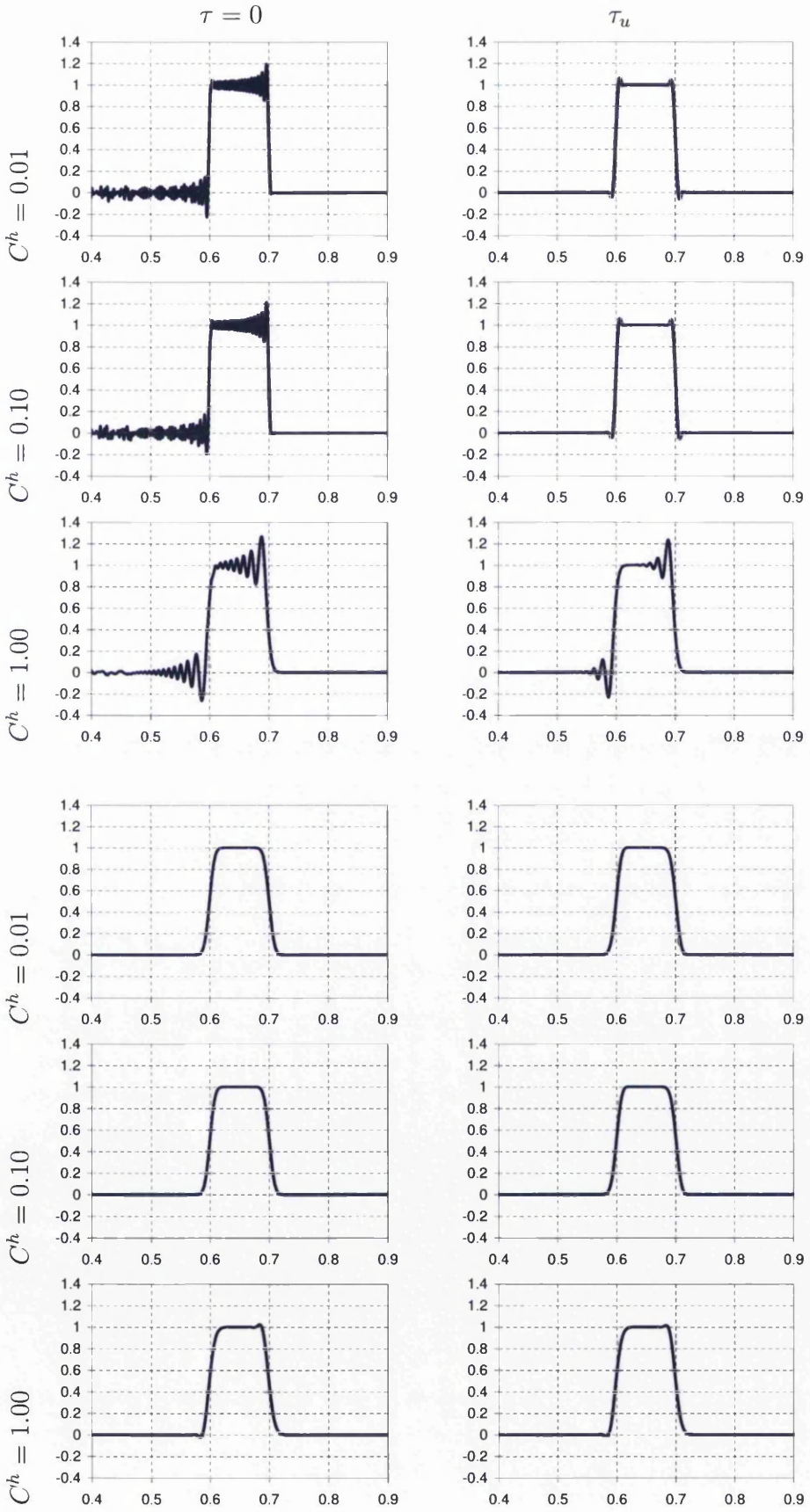


Figure 5.5: Solution for the Generalized-α method $\rho_\infty^h = 0.5$, at t=0.5s for 1000 element mesh

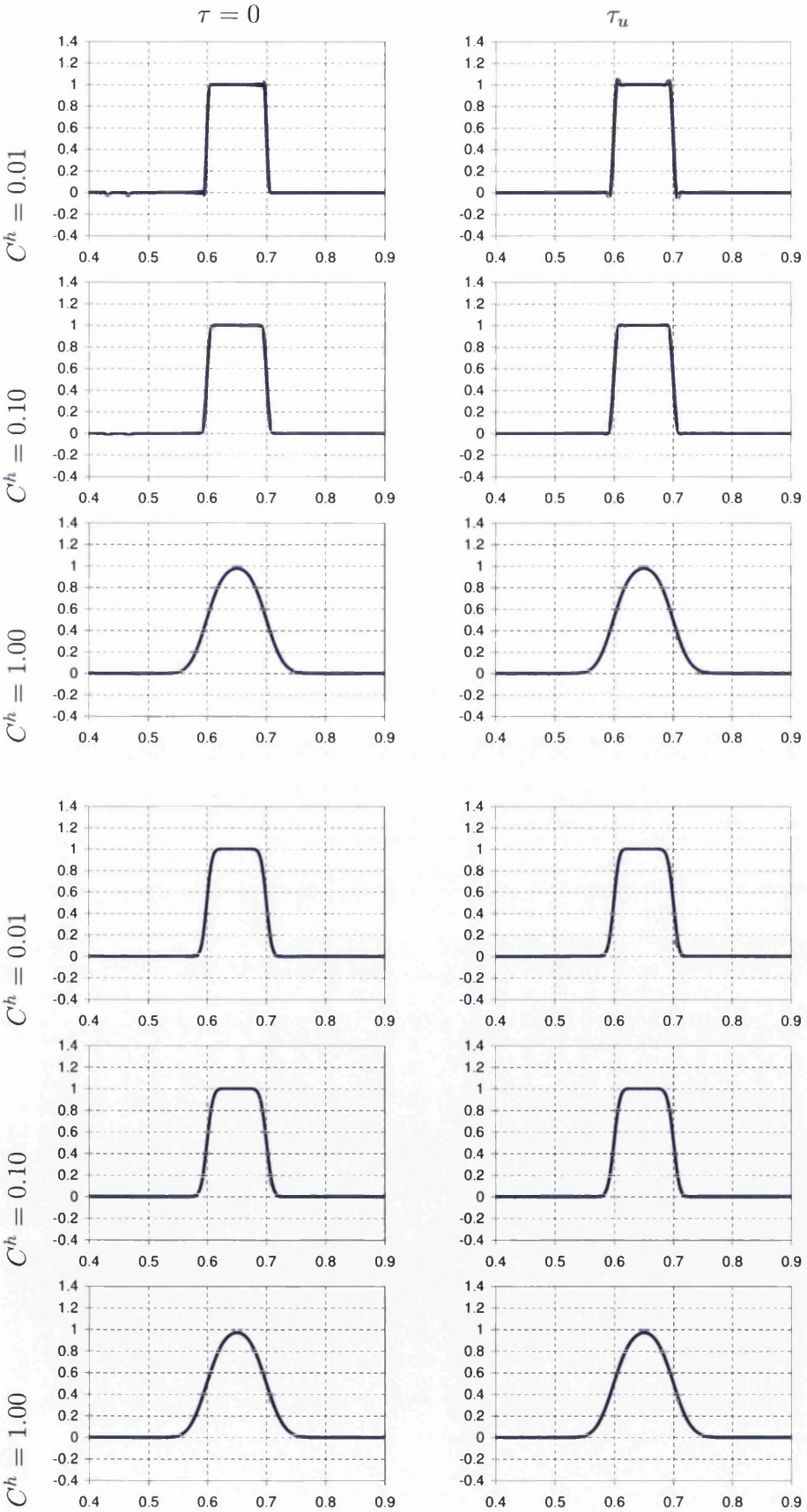


Figure 5.6: Solution for the Backward Euler method/Generalized Midpoint rule $\rho_\infty^h = 0.0$, at t=0.5s for 1000 element mesh

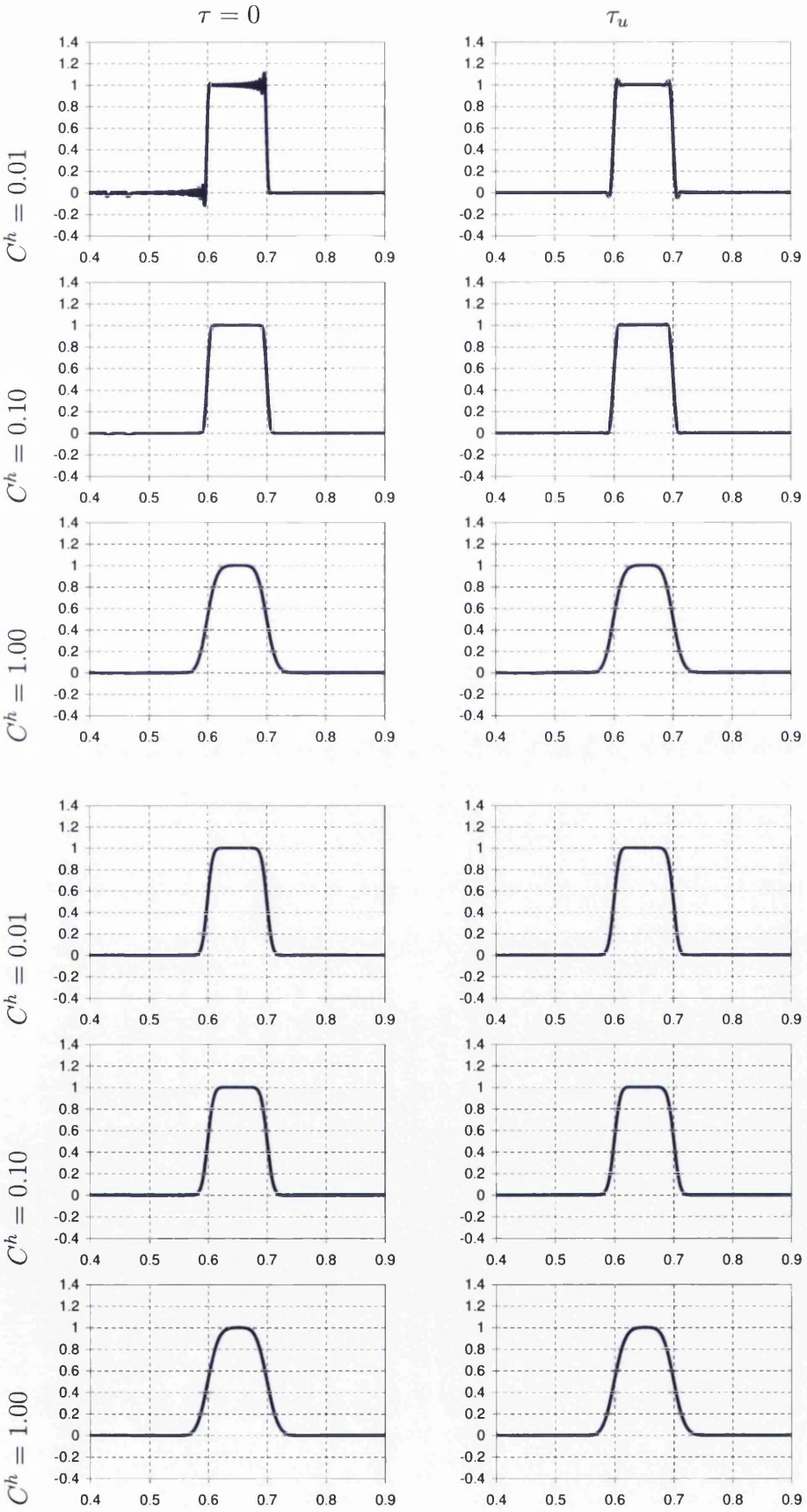


Figure 5.7: Solution for the Generalized Midpoint rule $\rho_\infty^h = 0.5$, at t=0.5s for 1000 element mesh

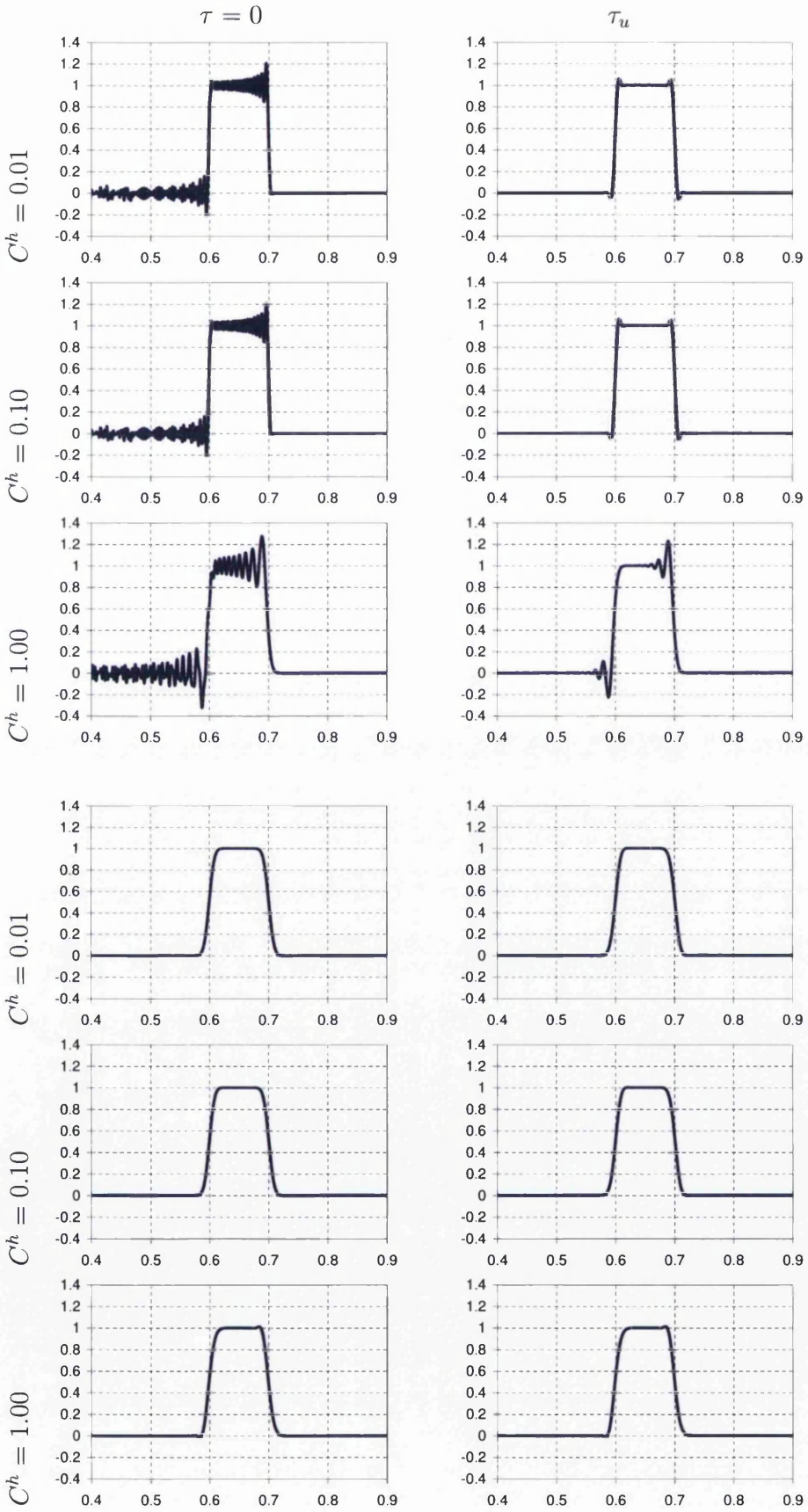


Figure 5.8: Solution for the Generalized Midpoint $\rho_\infty^h = 1.0$, Trapezoidal rule or Generalized-α method $\rho_\infty^h = 1.0$, at t=0.5s for 1000 element mesh

Discussion.

The following conclusions have been drawn from the results of the numerical simulation of 1D propagating rectangular wave example.

Mesh Dependency.

The time integration schemes all tend to the same solution as $\Delta t \rightarrow 0$ for the same mesh. Therefore the solution is mesh dependent.

Convergence.

The Generalized-α method converges to the solution faster than the Generalized Midpoint rule. This is apparent in a comparison of Figure 5.4 and Figure 5.5 with Figure 5.6 and Figure 5.7. The mesh dependent solution is achieved when $C^h = 0.1$ for the Generalized-α method i.e. $\text{Solution}_{C^h=0.1} \approx \text{Solution}_{C^h=0.01}$ whilst this point is only reached by Generalized Midpoint rule solution when $C^h = 0.01$. This property can be attributed to the second order accuracy in time.

Damping.

The unstabilized Generalized-α method, for similar ρ^h, appears to gives solutions which suffer from excessive high frequency oscillations whilst the Generalized Midpoint rules appears to be artificially diffusive. In both cases the oscillations appear to worsen as $C^h \rightarrow 0$.

This leads to the conclusion that as $C^h \rightarrow 0$ the solution tends to the mesh dependent solution which exhibits a more pronounced interface, which, as discussed previously in Section 4.1, lead to greater oscillations. In the case of the Generalized Midpoint rule the artificial diffusion introduced by the scheme manages to suppress these oscillations to some extent. However the Generalized-α method appears to be victim of its own success, i.e. improved convergence properties leads to greater spurious oscillation.

It was also noted that if the diffusion coefficient is appreciable, then artificial diffusion properties of the Generalized Midpoint rule has little or no effect on the solution.

Stabilization.

The difference in the solution between the classical Galerkin formulation and that of the SUPG can be attributed to the correct choice of stabilizing parameter τ_u.

Advection diffusion.

For this case the classical Galerkin formulation is comparable to the stabilized formulations. The effect of diffusion on the solution is to dampen or 'smooth' the interface, negating the problems caused with the convection term, such that the spurious oscillations that affected the pure advection case are reduced.

5.7 Flow past a circular cylinder.

Flow around a circular cylinder has been well documented and is considered one of the classical problems of fluid dynamics - to gain complete understanding of the fluid dynamics of the flow requires an understanding of fundamental subjects, such as the boundary layer, the free shear layer, and the dynamics of the vortices.

In this section there will be a brief introduction into the background of this example, followed by a review of the literature available, and finally a numerical simulation will be performed to compare the methods developed in previous chapters with the results discussed and conclusions drawn.

5.7.1 Background.

It is always difficult to pinpoint precisely the date and author of first *discovery*, and this seems the case with the periodic phenomenon associated with flow past a circular cylinder. Although the effect of wind causing vibrations in wires (Aeolian tones) have been known for some time, it was Strouhal's experimental observations (1878) that first correctly identified the physics of the problem. The experiments confirmed that the frequency of vibration was dictated by the velocity of the air flow not the elastic properties of the wire and two years later Rayleigh deduced that the frequency was actually Reynolds number dependent.

However it was not until the early 20^{th} century that the importance of, what is now known as, the von Karman vortex street became appreciated - previous observations were only concerned themselves with the relation between vibration frequency and flow velocity; notably in the 15^{th} century Leonardo da Vinci produced drawings showing the vortex formation, although they were sketched symmetrically. In 1911 von Karman produced his famous theory of the vortex street stimulating long lasting interest in the subject.

5.7.2 Description of Flow dynamics.

For small Re the flow is symmetrical; as Re increases it becomes increasingly asymmetrical and at Re $\simeq$ 6 closed eddies begin to form behind the cylinder. The inline length of the eddies region grows linearly with the Reynolds number, and as Re approaches Re_{crit} (Re $\simeq 30-40$), oscillations appear within the eddies region. The onset of vortex shedding ensues at $\mathrm{Re}_{crit} \simeq 49$ when the closed eddies behind the cylinder become detached to form a vortex street. For the range of

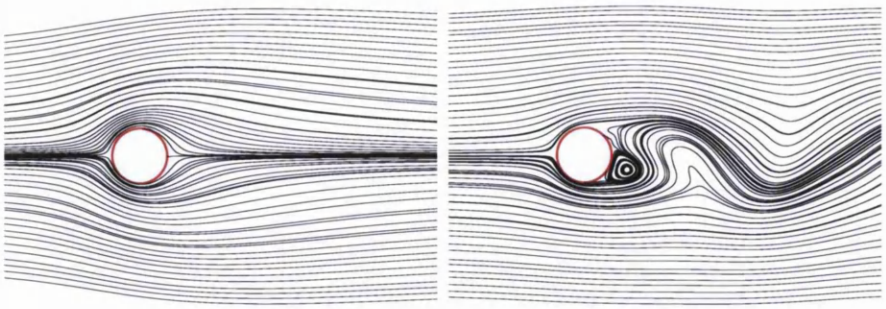
$49 \leqslant \mathrm{Re} \leqslant 150$ vortices are generated with almost perfect periodicity, which are characterized by dimensionless frequency (Strouhal number) which increases smoothly from $0.12 - 0.19$. Whilst the vortices are formed at fixed intervals within this Reynolds range, irregularities can appear downstream of the cylinder as interaction between the vortices occur. Experimentally it has been noted that somewhere between $\mathrm{Re} = 150$ and $\mathrm{Re} = 194$ the generation of the vortices becomes irregular and we see the onset of 3D structures in the wake, from this point forward there are very few qualitative changes in the flow up to $\mathrm{Re} = 10^4$. At high Reynolds numbers there is overall periodicity much like the vortex street, but the detailed motion of the fluid is increasingly random, i.e. turbulence is present.

An exact solution for this problem only exists for infinitesimal Reynolds numbers, Oseen [63], for larger Re we become reliant on close approximations. There are increasingly accurate numerical calculations based on direct approximations of the Navier-Stokes equations for the flow regime of attached eddies. For the vortex street in roads have been made into understanding the phenomena, but it is only recently that it has been possible to accurately deduce the general paths of the fluid elements; as yet no analytical solutions has been found. There have been attempts to model interaction between vortices using the Navier-Stokes equations, but the results show great sensitivity to flow history and the imposed boundary conditions, even if they are far from the cylinder.

5.7.3 Initial work.

The initial work carried out to define the different flow regimes of flow past a circular cylinder was by Roshko [65]. Further work has since been carried to investigate and correlate this work, such as Bloor [11]. Within the Reynolds number range of $0 \rightarrow +10^4$ there are five distinct regimes, and these are categorized as:

$0 < \mathrm{Re} < 6$	Unseparated Flow
6 to $15 < \mathrm{Re} < 49$	Fixed pair of Föppl vortices in wake.
$49 < \mathrm{Re} < 150$	Laminar vortex street
$150 < \mathrm{Re} < 300$	Transition to turbulence in vortex street
$300 < \mathrm{Re} < 3 \times 10^5$	Turbulent vortex street



(a) Laminar flow (b) Vortex shedding

Figure 5.9: Streamlines around a circular cylinder for different regimes

Since Roshko's [66] initial research there has been considerable work into flow past a cylinder and its effects, most experimental, some numerically based. Many of these investigations have studied the relationship between the Reynolds number and the frequency of the vortex shedding as defined by the Strouhal number ($Sr = fA/u_\infty$ where f is the shedding frequency, u_∞ is the free stream velocity and A is the diameter of the cylinder). In particular the detailed review by Williamson [87], is considered to be a defining study into this relationship, firmly establishing the existence of a universal Reynolds-Strouhal number relationship from experimental evidence, after accounting for experimental errors such as cylinder vibration, end effects, and shear in the oncoming flow free stream etc.

Such has been the interest into this field that it can now be sub divided into more specific areas, as the focus moves from gaining an understanding of overall picture, as investigated by Roshko in his early work, to focusing on distinct areas of this complex topic separately. These sub divisions include the study of arrays of cylinders, wake, near wall effects, spring mounted cylinders for inline/cross flow vibration, vibration damping and boundary layers to name a few. In this section we are interested in benchmarking various timestepping schemes, and flow past a circular cylinder is suitable vehicle with which to achieve this aim, therefore the focus will remain on the overall dynamics of the flow.

5.7.4 3D Structures.

Advancements have been made in the understanding of wake vortex dynamics due to studies into 3D phenomena which have led to explanations for long misunderstood phenomena which have always been assumed to have 2D origins. Williamson [87] discusses the advances made in the understanding of the onset of

3D phenomena and states that the onset of instability is Re=49 and at Re=190-260 a 3D wake transition develops. However it is standard practice in numerical literature to treat the flow as 2D for Re $>$ 190, whilst recognizing that the 2D calculations will not be truly representative of the actual flow at these Reynolds numbers. Also having reviewed other such studies it has been noted that the drag and lift coefficients will be slightly larger than the values one would expect from a 3D study.

5.8 Numerical Example - Flow past a circular cylinder.

Model.

In this example the viscous incompressible flow past a circular cylinder in an infinite domain is considered. The inflow is assumed to be uniform at infinity and is denoted as u_∞, with diameter of the cylinder denoted A, and density and the viscosity of the fluid ρ and μ respectively.

The simulations performed will use differing mesh densities, timesteps and timestepping methods for Reynolds numbers of 100, 400, & 1000. In order to obtain these Re numbers the following values were used $\rho = 1.0$, $\mu = 0.01$ for velocities $u_\infty = 1.0$, $u_\infty = 4.0$ and $u_\infty = 10.0$.

We should note that the long term solution for this problem is periodic vortex shedding which subjects the cylinder to oscillating lift and drag forces which act normal and parallel to the flow, F_L & F_D respectively, at a frequency f_s [8]. The non-dimensional lift C_L and drag C_D coefficients, and the Strouhal number Sr are defined as:

$$\mathrm{Re} = \frac{\rho u_\infty A}{\mu} \qquad Sr = \frac{fA}{u_\infty} \tag{5.75}$$

$$C_D = \frac{2F_D}{\rho u_\infty^2 A} \qquad C_L = \frac{2F_L}{\rho u_\infty^2 A} \tag{5.76}$$

Geometry.

The computational domain is often only an approximation of the actual domain, and as noted in an earlier section, the closeness of the computational domain to the object can have great effects on the solution accuracy. In addition to

[8]The lift force oscillation frequency is half the frequency of the drag force oscillation. It is standard practice to use the frequency of the lift force.

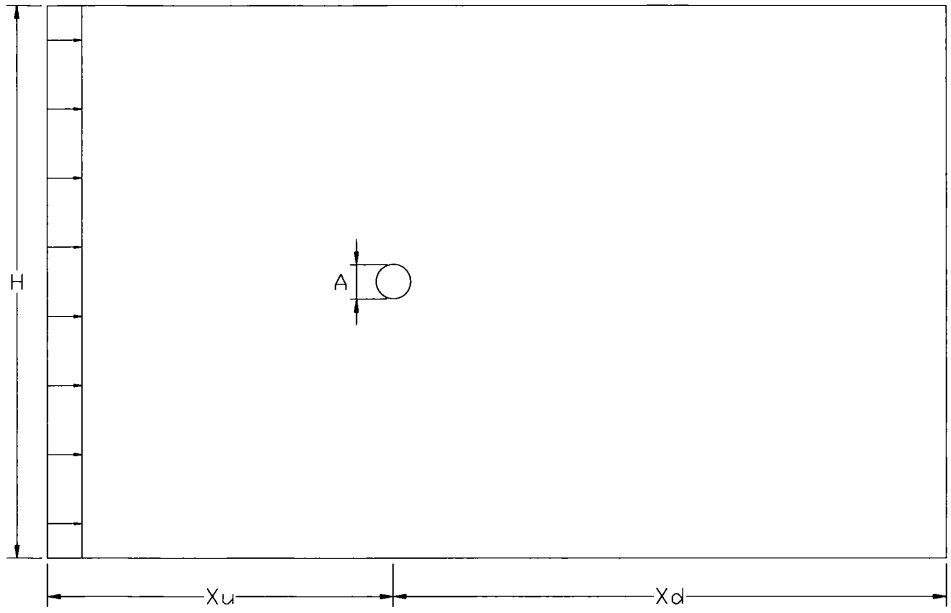


Figure 5.10: Geometry of the domain

this many types of boundary condition are only appropriate when placed at a sufficiently large distance away from the region where accuracy is paramount. However in the process of meeting these requirements we inadvertently adversely affect the computational cost. In practice the designer of the example must weigh the advantages and disadvantages and decide which is a more critical restraint: -accuracy or -time. In most cases it is appropriate for a balance to be reached where both restraints are adequately satisfied.

There have been many investigations into the most appropriate computational domain for this type of example, i.e. flow past a circular cylinder in an infinite domain. The two factors to consider are the position of the cylinder in regard to the (1) inflow/outflow boundaries and (2) lateral boundaries.

Inflow/Outflow Boundary.

There have been various papers which have considered solution accuracy with regards to varying the distances x_u and x_d, e.g. Tezduyar *et al* [82] and Behr *et al* [8]. It was found, with the commonly used traction free boundary condition applied to the outflow boundary, that solution quantities such as Strouhal number, drag and lift, were insensitive to variation in the distance x_d. However if x_d was less than 6 cylinder diameters, marked differences, including lose of periodicity and tending to steady state, were noted. There is much less information regard-

ing the length x_u, though it is regularly placed as close as 5 cylinder diameters from the cylinder, as used by Dettmer and Peric *et al* [27].

Using trial values for x_u and x_d the inflow and outflow boundaries were located at 5 and 16 diameters respectively from centre of the cylinder - there is little of interest in the region of flow upstream of the cylinder and the need to accurately represent the wake requires boundary placement further downstream than the critical distance found by Behr and Tezduyar [8] [82]. Sensitivity tests were performed varying the initial lengths of x_u and x_d and the investigation suggests that the effects due to the outflow boundaries become negligible beyond 16 diameters. However it was found that the inflow boundary distance was situated too close and should be placed further away in order that the results are more representative of the problem. for an improved solution for little increased computational cost the distance x_u has been chosen as cylinder diameters. Therefore $x_u = 10A$ and $x_d = 16A$.

Lateral Boundary.

Behr *et al* [7], and Lei *et al* [54] have all considered the effects of the lateral boundary on the solution quantities i.e. Strouhal number, and drag/lift coefficients. Behr in his paper discusses the problems of using a poorly constructed domain to benchmark timestepping schemes. He compares a commonly used transient method i.e. flow past a circular cylinder, as the benchmarking example for Re = 100. He concludes that the effect of a near lateral boundary is far greater than the effects due to a near inflow/outflow boundary and require extra care in their placement. The data presented by Behr suggests that in order to remove the effects of the lateral boundary on the shedding phenomena the boundary should be placed, at least, 8 cylinder diameters from the cylinder; too close and the primary solution parameter, Strouhal number, will be artificially high. It is possible to envisage a situation where a dissipative timestepping scheme (lowering the Strouhal number) is combined with closely spaced lateral boundaries (increasing the Strouhal number) to produce shedding characteristics which are comparable to experimental observations. Therefore in an investigation, such as in this thesis, where the frequency is used as an accuracy indicator, boundary placement is as important as mesh refinement.

Dettmer and Peric [27] in his comparable study of timestepping schemes has chosen closely spaced lateral boundaries for the simulation of flow past a circular cylinder. As for the inflow/outflow, we can compare results from simulations

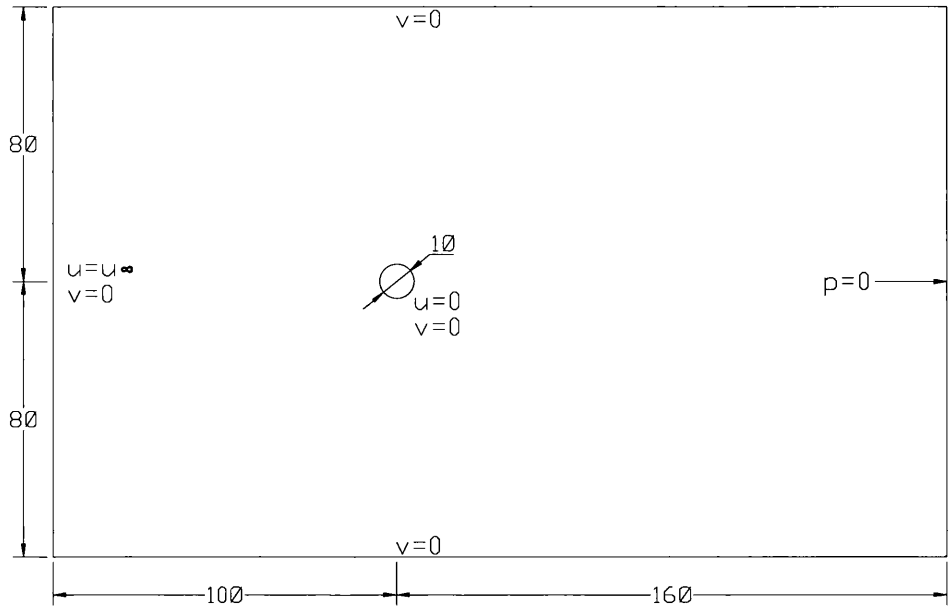


Figure 5.11: The problem setup - the domain with the boundary conditions - u, v, and p are the velocity in the x and y directions and pressure respectively

carried out using similarly constructed meshes with larger lateral spacing. Once again the improvement in the Strouhal number is worth the computational cost.

In conclusion the cylinder is centred vertically in the domain and placed horizontally such that there is maximum accuracy whilst preserving efficiency. The lateral, inflow and outflow boundaries are located at 8,10, and 16 diameters from the centre of the cylinder respectively. A slight imperfection in the geometry has been introduced into the model to initiate unsteady flow. This has been achieved by moving one of the lateral boundaries by 0.1% of the overall width of the domain, in order to place the cylinder slightly *off-centre* and remove the symmetrical nature of the geometry.

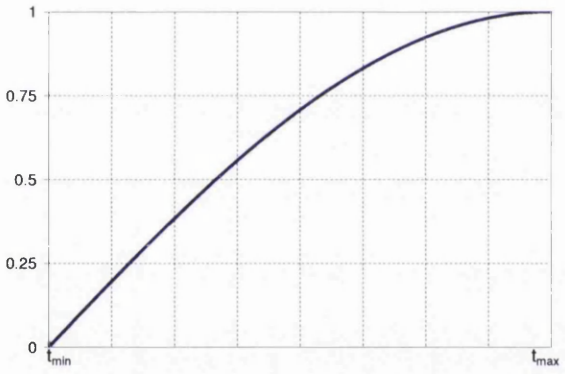
Boundary Conditions.

The Navier-Stokes problem must be completed with suitable initial and boundary conditions to form a well-posed initial boundary value problem. In the following section the initial and boundary conditions that form the initial boundary value problem for this problem are explained.

Velocity Boundary Conditions.

A uniform velocity is assigned to the inflow boundary. The outflow boundary we specify a Neumann-type boundary condition for velocity which corresponds to a zero viscous stress tensor. This has been chosen such that the flow crossing the boundary is not affected by the presence of the boundary, therefore vortices approaching the outflow boundary cross the boundary undisturbed by its presence. There is a no-slip and no penetration condition on the cylinder wall and finally a zero normal velocity at the lateral boundaries.

The inflow velocity u_∞ was reached by smoothly increasing the velocity component u at the inflow boundary from $0 \rightarrow u_\infty$ over a set time period. This was achieved using the section of the sine curve shown below as a loading curve, see Figure 5.12. The loading time ranges for Re = 100, 400 & 1000 are [0,4], [0,1] and [0,0.4] respectively.



$$\sin\left[a(t - t_{min})\right] \quad \text{where} \quad a = \frac{\sin^{-1}(1)}{(t_{max} - t_{min})}$$

Figure 5.12: Loading Curve

Pressure Boundary Conditions.

No initial condition need be set for the pressure unknown. This is a direct consequence of the absence of a time derivative for pressure in the governing equations. Pressure also only needs to be specified at one point within the domain and defined with an arbitrary value, in order to provide a unique solution. This is due the fact that pressure is only present by its gradient in the governing equations. The placement of the fixed pressure node is also arbitrary, therefore for these examples pressure has been fixed on the outflow boundary.

Mesh.

The 3 meshes were produced for this example using ELFEN software (1896, -3006-, *5436* elements and 997, -1569-, *2811* nodes). All were generated using the same geometries, applying the boundary conditions discussed earlier, each with a different mesh density. To ensure an optimal solution the mesh near to the bluff body was refined in order to model the dynamics of the problem accurately.

For the investigation into the effects of the domain size on the solution it was important to remove any dependance on near field refinement, therefore the larger mesh was constructed using the smaller mesh. See Figure 5.13

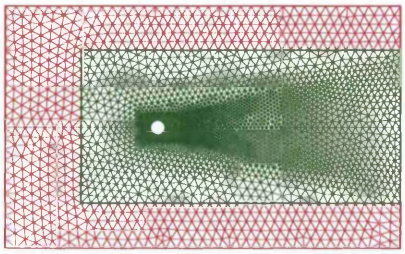
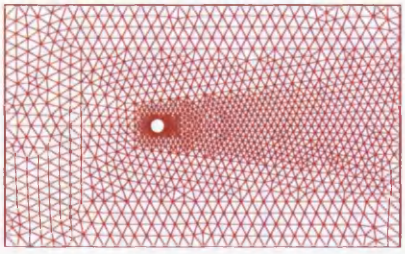
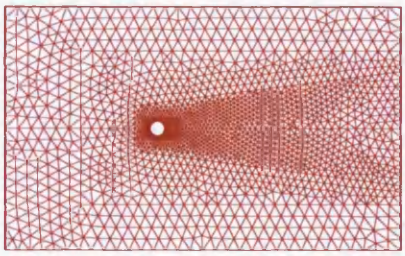


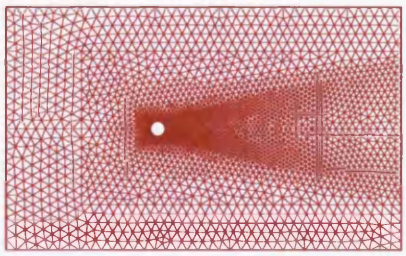
Figure 5.13: Nested Mesh



(a) 1896 elements



(b) 3006 elements



(c) 5436 elements

Figure 5.14: Meshes for the flow past a circular cylinder.

Numerical Results.

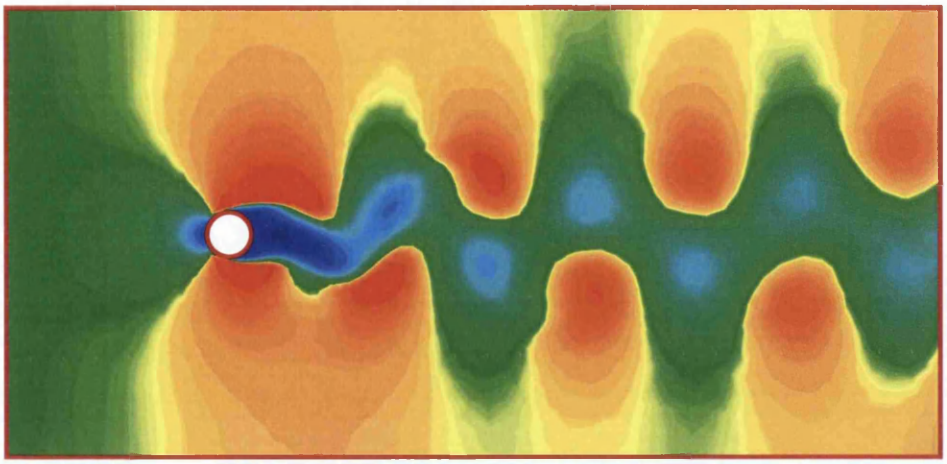
Figure 5.15 are velocity/pressure plots for the solution Re = 225. These plots clearly demonstrate the vortices shedding from behind the cylinder.

Figure 5.16 $\rightarrow$ Figure 5.18 document the evolution in time of the lift and drag coefficients, for different Reynolds number.

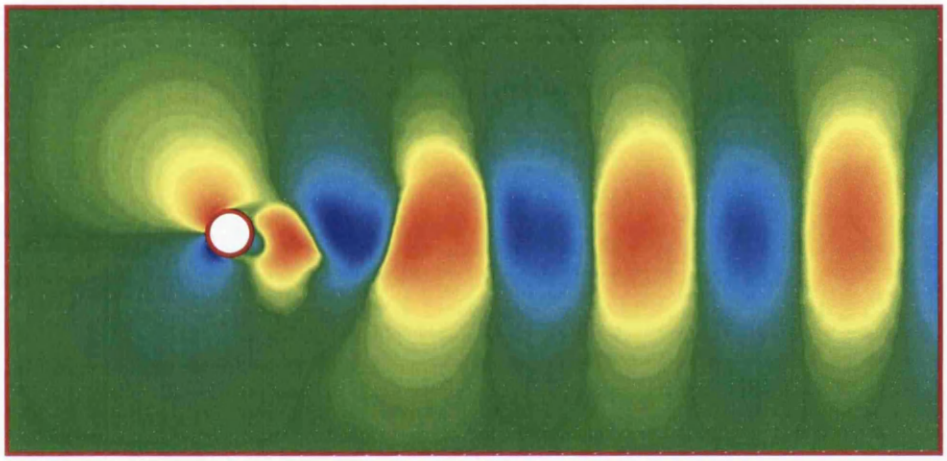
Figure 5.19 shows a close up of the pressure isolines for one complete oscillation, whilst Figure 5.20 $\rightarrow$ Figure 5.21 shows the evolution of the lift coefficient, with corresponding pressure isolines.

Figure 5.22 $\rightarrow$ Figure 5.24 are the full documentation of the numerical results recorded for this example. A comparison of time stepping schemes (Generalized-α method and Generalized Midpoint rule) for different Reynolds number and mesh densities (1896, -3006-, *5436* elements and 997, -1569-, *2811* nodes), in terms of Strouhal number and the amplitude of lift coefficient. Note that for Re = 1000 the coarse mesh is not represented as the Newton-Raphson scheme fails.

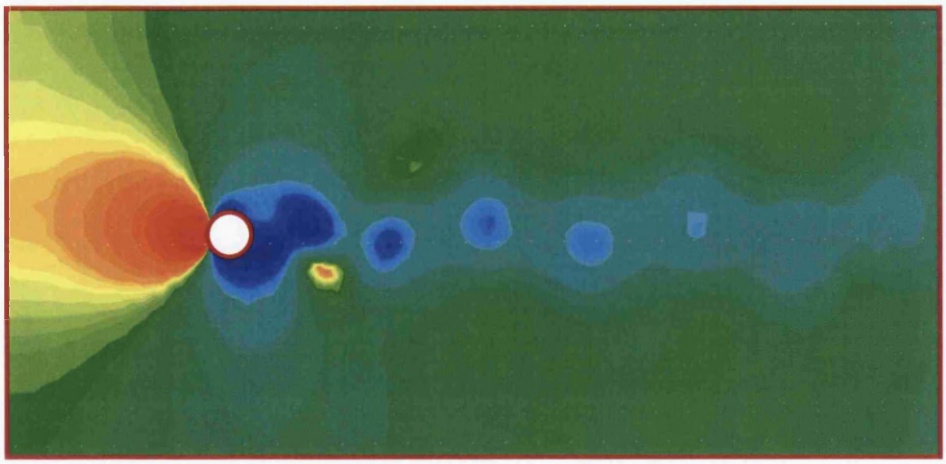
Figure 5.25 show the Strouhal and lift coefficients as the spatial refinement increases. To determine the effect of spatial refinement the example was repeated for Re= 100, 400 and 1000, for domains with high density meshes, i.e. 10182, (16449), [20243] elements and 5224, (8393), [10281] nodes. The timesteps were small, such that the solutions had asymptotically converged as $\Delta t \rightarrow 0$ and could be considered to be independent of the time integration scheme.



(a) u

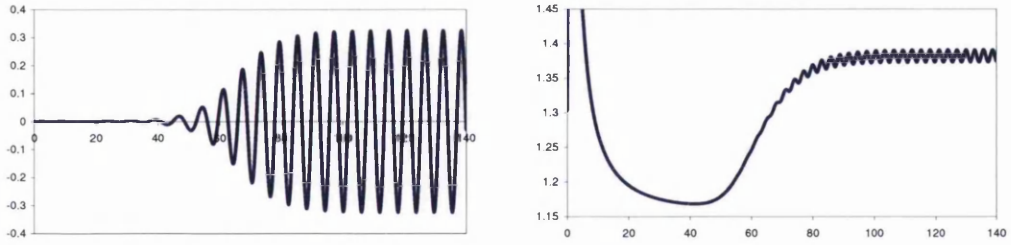


(b) v



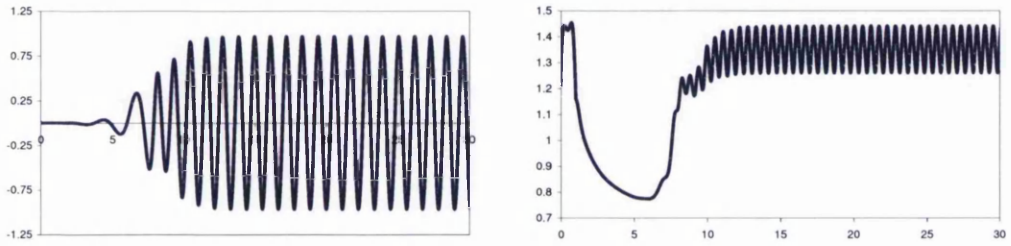
(c) p

Figure 5.15: Velocity/Pressure contour plots - Re = 225, Generalized-α method $\rho_\infty^h = 0.9$, $n_{elem} = 5436$



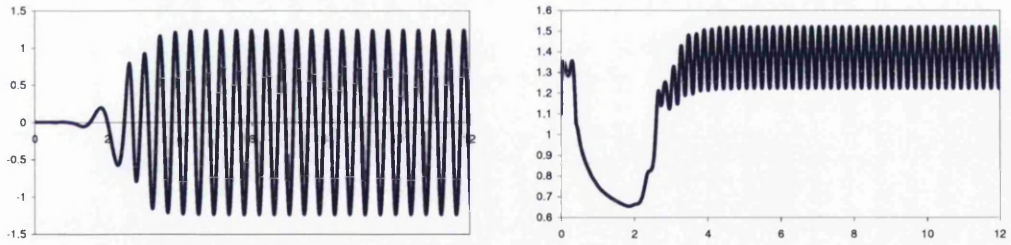
(a) C_L (b) C_D

Figure 5.16: Evolution of Lift C_L and Drag C_D coefficients in time. Re = 100, $\Delta t = 0.01$, Generalized-α method $\rho_\infty^h = 0.9$, $n_{elem} = 5436$



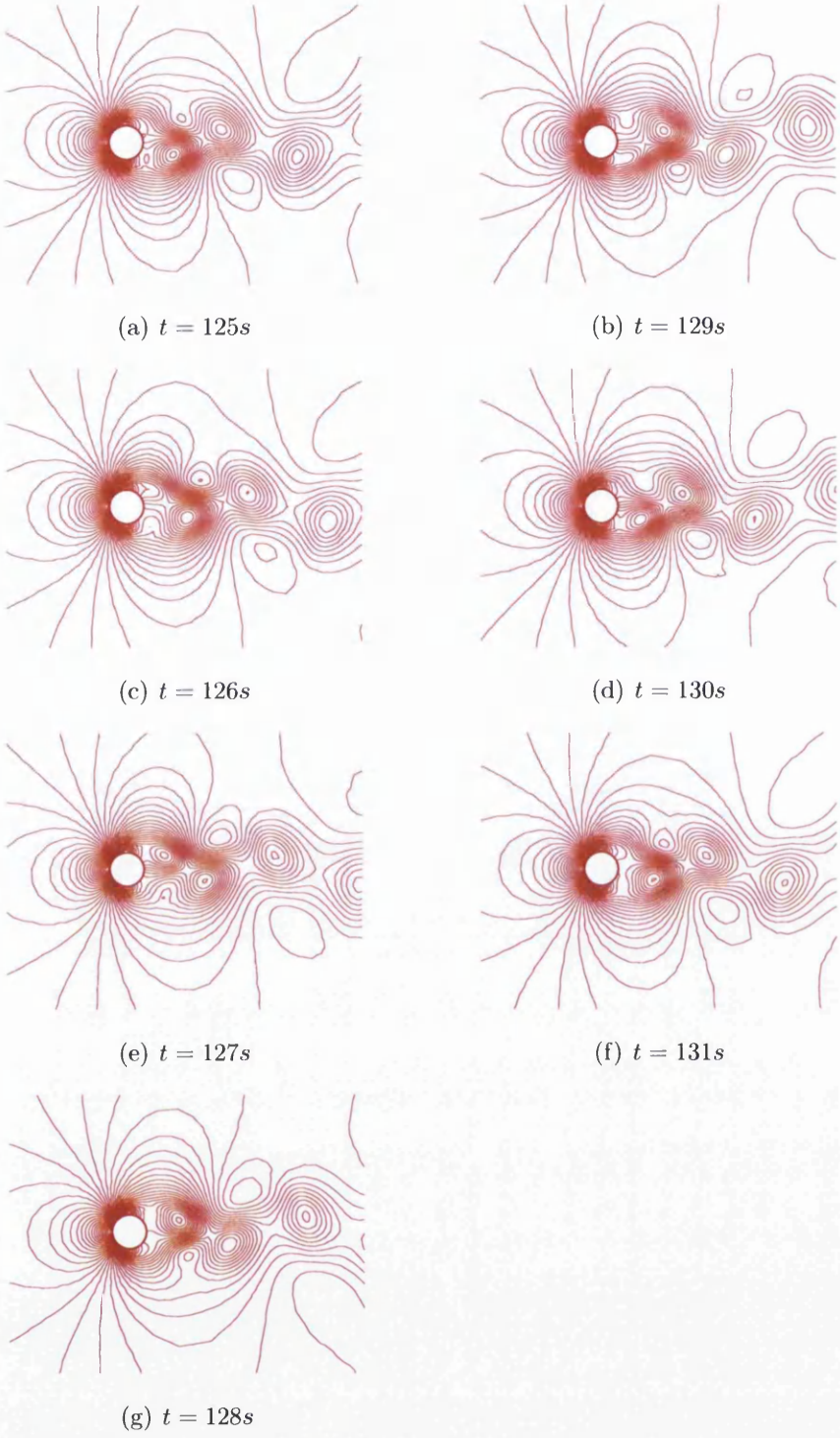
(a) C_L (b) C_D

Figure 5.17: Evolution of Lift C_L and Drag C_D coefficients in time. Re = 400, $\Delta t = 0.0025$, Generalized-α method $\rho_\infty^h = 0.9$, $n_{elem} = 5436$



(a) C_L (b) C_D

Figure 5.18: Evolution of Lift C_L and Drag C_D coefficients in time. Re = 1000, $\Delta t = 0.001$, Generalized-α method $\rho_\infty^h = 0.9$, $n_{elem} = 5436$



(a) $t = 125s$ (b) $t = 129s$

(c) $t = 126s$ (d) $t = 130s$

(e) $t = 127s$ (f) $t = 131s$

(g) $t = 128s$

Figure 5.19: Pressure Contours - One complete oscillation Re = 100, $\Delta t = 0.1$, Generalized-α method $\rho_\infty^h = 0.9$, $n_{elem} = 5436$

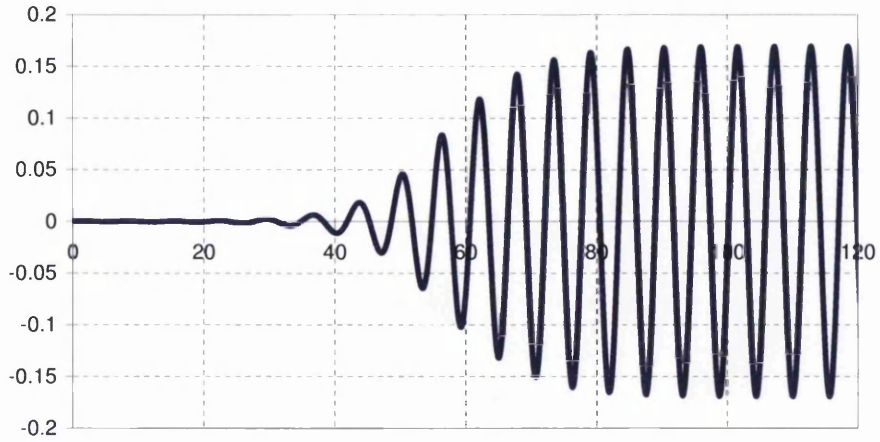
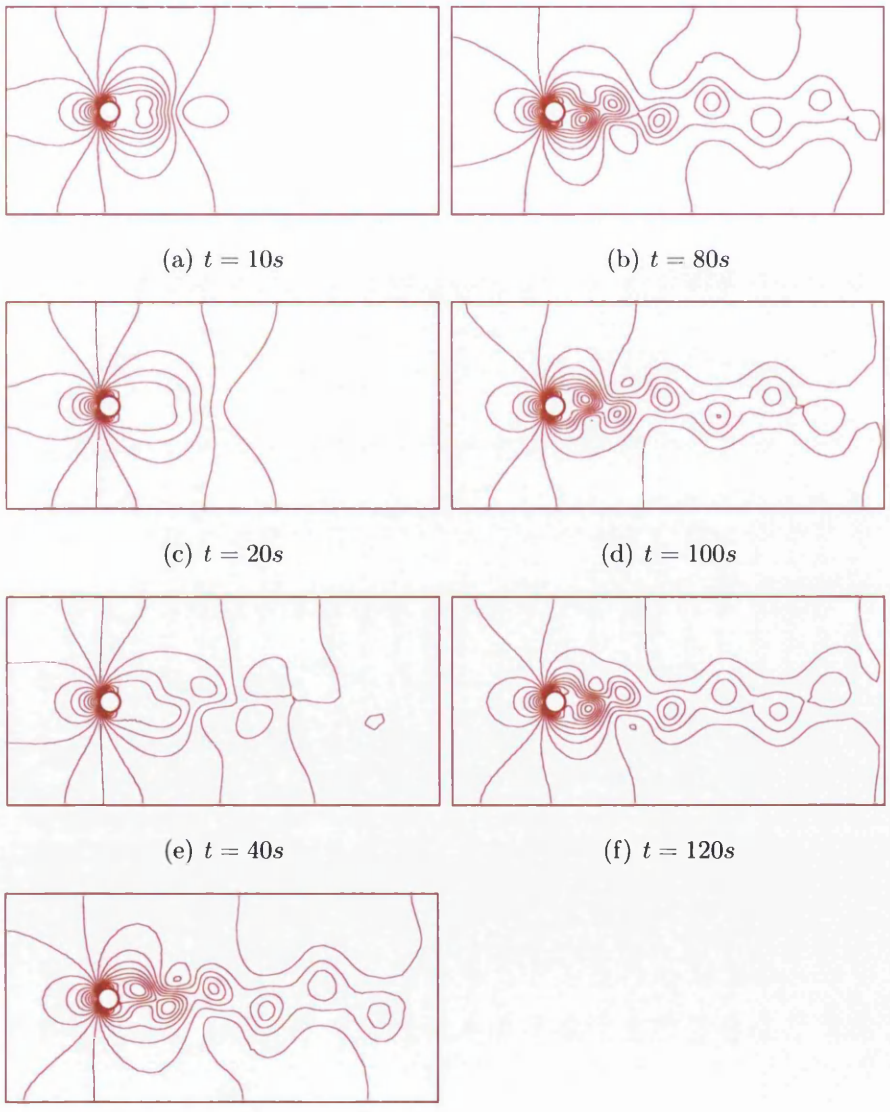


Figure 5.20: C_L against time - Re = 100, $\Delta t = 0.1$,Generalized-α method $\rho_\infty^h = 0.9$, $n_{elem} = 5436$



(a) $t = 10s$

(b) $t = 80s$

(c) $t = 20s$

(d) $t = 100s$

(e) $t = 40s$

(f) $t = 120s$

(g) $t = 60s$

Figure 5.21: Pressure Contours - Re = 100, $\Delta t = 0.1$, Generalized-α method $\rho_\infty^h = 0.9$, $n_{elem} = 5436$

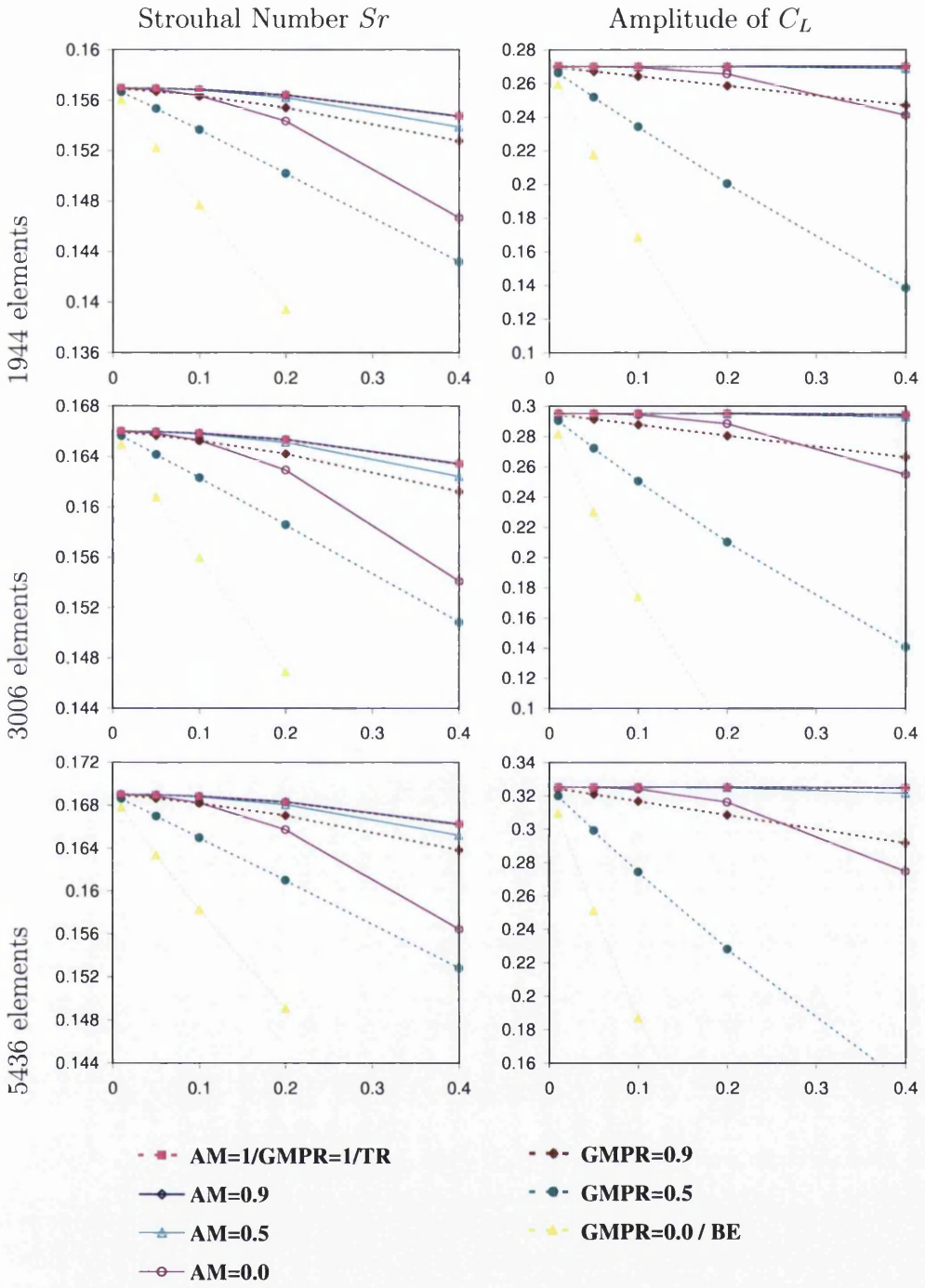


Figure 5.22: Re=100 - Convergence of Strouhal number St and Lift coefficient C_L. AM - Generalized-α method, GMPR - Generalized Midpoint rule, BE - Backward Euler, TR -Trapezoidal rule

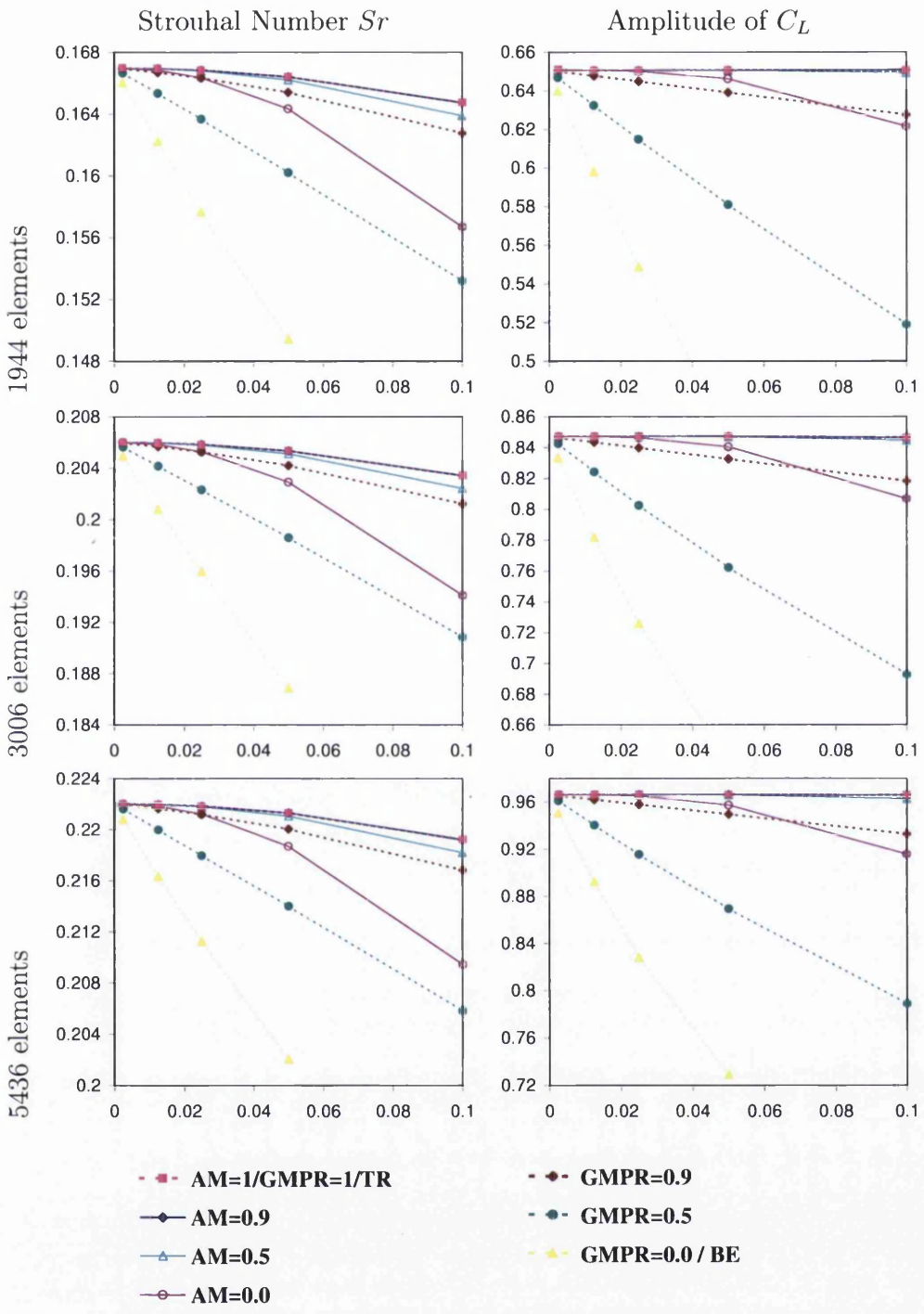


Figure 5.23: Re=400 - Convergence of Strouhal number St and Lift coefficient C_L. AM - Generalized-α method, GMPR - Generalized Midpoint rule, BE - Backward Euler, TR -Trapezoidal rule

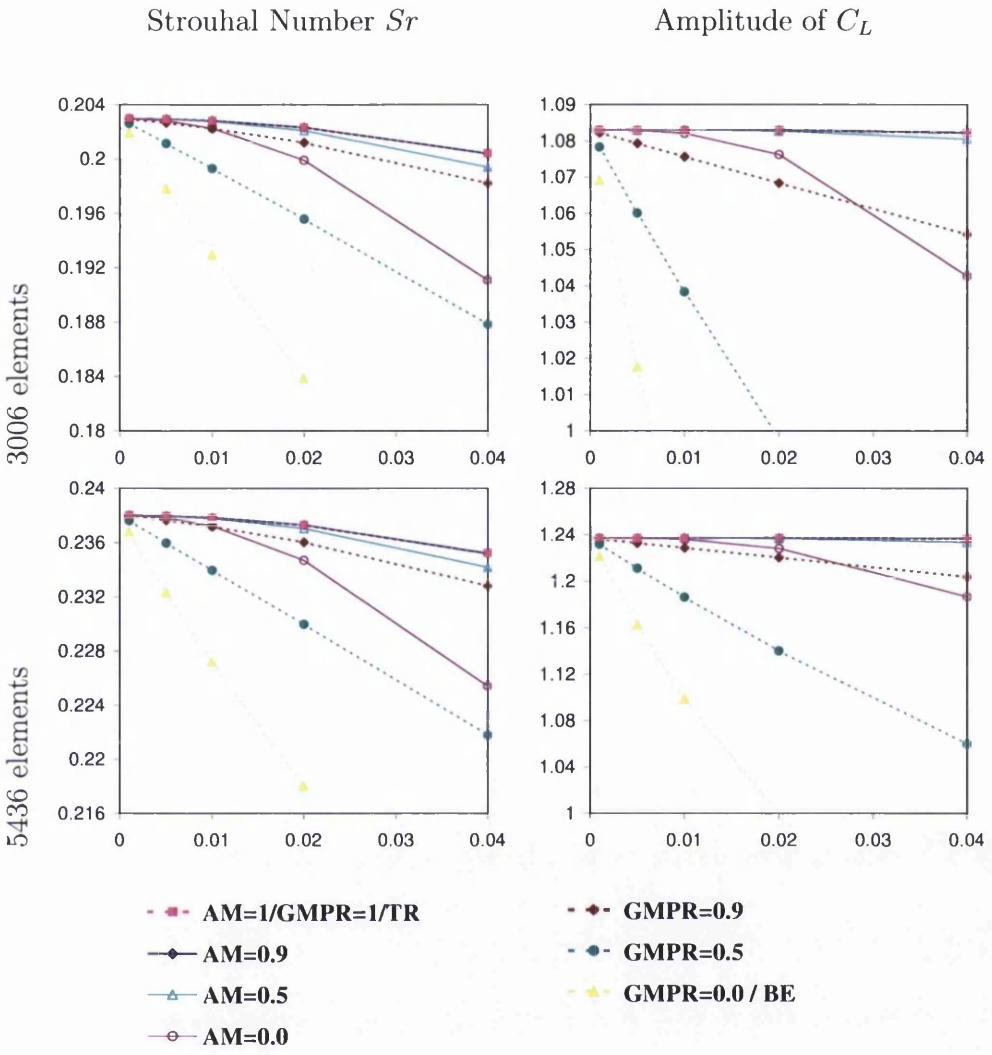


Figure 5.24: Re=1000 - Convergence of Strouhal number St and Lift coefficient C_L. AM - Generalized-α method, GMPR - Generalized Midpoint rule, BE - Backward Euler, TR -Trapezoidal rule

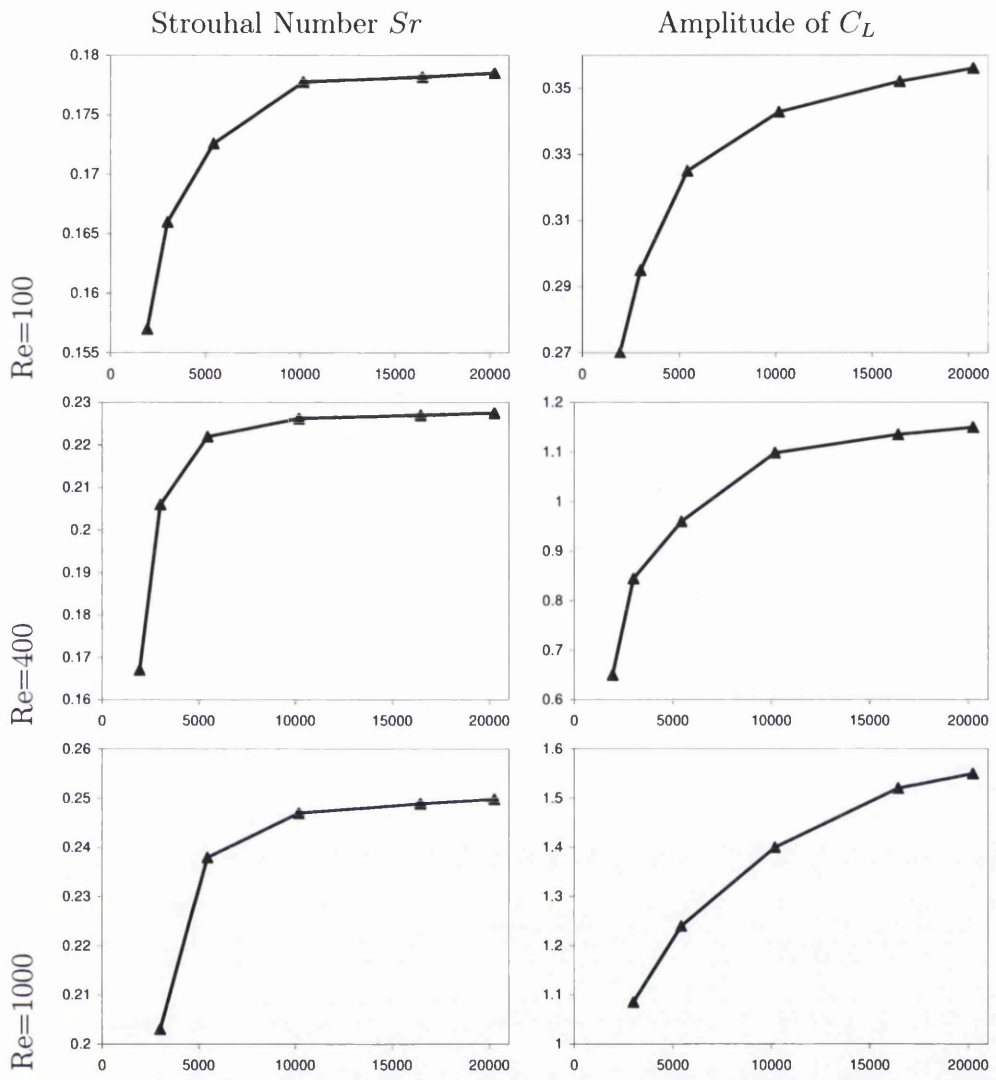


Figure 5.25: Convergence of the Strouhal number and lift coefficient as $h \rightarrow 0$ for meshes of 1896, -3006-, *5436*, {10182}, (16449), [20243] elements and 997, -1569-, *2811*, {5224}, (8393), [10281] nodes

Discussion.

Because the 1D Advection Diffusion equations are a simplified version of the Navier Stokes equations it is expected that they will share many common solution attributes. Therefore the conclusions drawn from 1D propagating rectangular wave can also be drawn for, and indeed have been confirmed by the solutions for flow past a circular cylinder. These conclusions are reiterated briefly below:

Mesh Dependency.

Figure 5.22→Figure 5.24 highlight the conclusion made previously that the solutions are mesh dependent for coarse, and irrespective of the time stepping algorithm, the solutions tend to a certain result as $\Delta t \rightarrow 0$. However, Figure 5.25 demonstrates that the solution becomes mesh independent as the spatial refinement increases, i.e. a plateau is reached as $h \rightarrow 0$.

Convergence.

Figure 5.22→Figure 5.24 also demonstrate the improved convergence properties of the Generalized-α method in comparison to the Generalized Midpoint rule, here shown by the shallower gradient of the Generalized-α method plots.

It appears from the previous two examples that a value of $\rho_\infty^h = 1.0$ gives better convergence than solutions where ρ_∞^h is close to zero. However in Section 5.2 the problems associated with $\rho_\infty^h = 1.0$ were discussed, and it was recommended that a value close to 1.0 be used. Therefore it is the author's considered opinion that $\rho_\infty^h = 0.9$ is the most suitable compromise. The results plotted in Figure 5.22→Figure 5.24 include data for the Generalized-α methods for $\rho_\infty^h = 0.9$. There seems to be little or no effect on the convergence properties when compared to the values for $\rho_\infty^h = 1.0$, therefore $\rho_\infty^h = 0.9$ has the advantages yet none of the disadvantages associated with $\rho_\infty^h = 1.0$.

Pressure and velocity stabilization.

The pressure field is free of the spurious oscillations which were apparent in the unstabilized lid driven cavity problem. The velocity field is also free of non physical oscillations for realistic time steps.

5.9 Flow past a square cylinder with near wall effects.

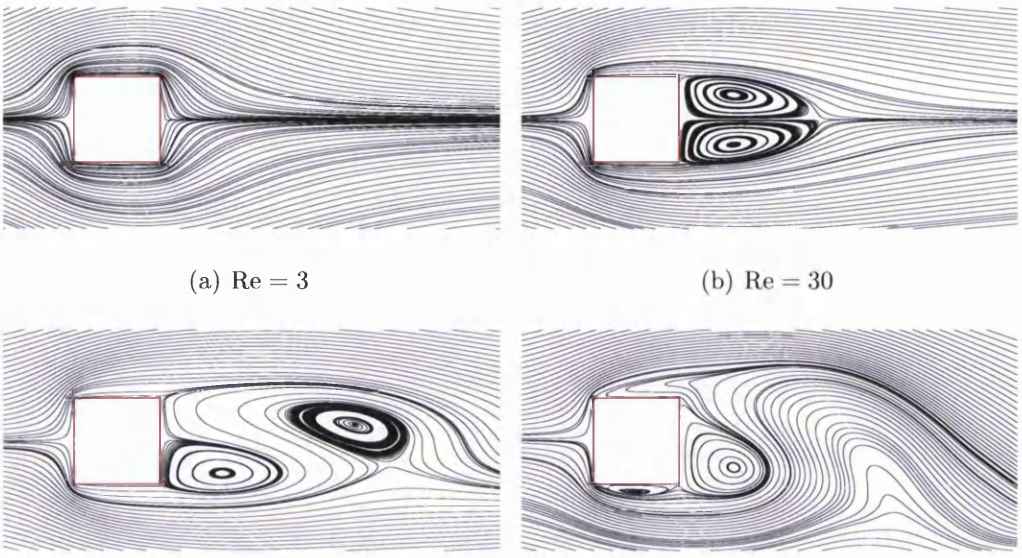
Flow past a square cylinder with near wall effects is an investigation into the problem of confined flows with a built in obstruction. This type of problem experiences the same phenomenon as the free flow example, see Section 5.7. However when a cylinder is placed in the vicinity of a solid wall the mean structure and dynamic behavior of the wake alter significantly from the behaviour exhibited by the same flow for an infinite domain. The close proximity of the wall suppresses the oscillatory behaviour of the fluid as boundary layer[9] effects begin to dominate.

The analysis of flow past square cylinders with near wall effects has a dominant role to play in many technical applications, such as building aerodynamics and cooling of electronics etc. Most research has been focused towards high Re flows i.e. $\mathrm{Re} > 10^4$, for circular cylinders, Lei *et al* [54] etc; or for square cylinders, see Bosch *et al* [12], Durao *et al* [30] and Martinuzzi *et al* [57]. However a few studies have considered the influence of near wall effects for laminar flow, see Breuer *et al* [13], Davis and Moore [25], Li and Humphrey [55] and Turki *et al* [84].

In this section a brief discussion is given regarding previous investigations into the field of flow past a square cylinder focusing on the influence of the wall proximity on the Strouhal number, and lift & drag coefficients. These will be followed by a numerical simulation for flows past a square cylinder, using the Generalized-α method for 3 blockage ratios $\left(\frac{1}{4}, \frac{1}{6}, \frac{1}{8}\right)$. Finally the results and conclusion will be discussed in detail.

[9]The boundary layer is a region where the velocity drops rapidly to zero to satisfy the no slip condition at the lateral boundary. The direct effects of viscosity are felt only within the boundary layer.

5.9.1 Description of Flow Dynamics.



(a) Re = 3 (b) Re = 30

(c) Re = 60 (d) Re = 200

Figure 5.26: Streamlines around a square cylinder for different Reynolds numbers for $\frac{1}{8}$ blockage ratio.

Flow past a confined square cylinder has various distinct regimes which are Re dependent in a similar manner to free flow past circular cylinder. The distinguishing feature between the two problems is that for confined flow the value of Re for each regime change will be partially dependent on the blockage ratio.

For small Re (ie < 1) viscous forces dominate and no separation occurs at the cylinder surface. Increasing the Re the flow first separates from the cylinder at the trailing edges, and forms a closed steady recirculation region consisting of 2 symmetric vortices behind the body. As Re increases to Re_{crit} the vortices increase in size and strength until at Re_{crit} the vortices start to detach and we begin to observe von Karman vortex street. The strength (or circulation) of these vortices is directly related to the periodic loading of the cylinder and the mixing rate of the near wake. A further increase in Re and the flow begins to separate from the leading edges of the body, and the circulation is generated at the point of separation. The onset of this phenomenon is not clearly defined in literature but has been assessed at $\mathrm{Re} \approx 100 - 150$ for a $\frac{1}{8}$ blockage, see Breuer *et al* [13], Okajima [61], Franke *et al* [33]. At this point the simulation can still be classed as 2D. However beyond this point we will start to see 3D structures in the wake, and Franke and Breuer provide guidance, suggesting that $\mathrm{Re} \geq 300$ is the limit

beyond which 3D structures should be expected and the transition to turbulence begins in the free shear layers. It is for this reason the Re range studied here will be between Re_{crit} and $\mathrm{Re} = 300$.

It should be noted that due to the defined points of separation for flow around a square cylinder that aerodynamics coefficients are less dependent on Re than circular structures.

5.9.2 Initial Work.

In comparison to the free flow problem, the near wall effects problem requires a further two parameters to describe the fluid behaviour, i.e. inflow profile and blockage ratio β. Davis *et al* [25, 26] demonstrated that the inflow velocity profile had a appreciable effect on the frequency of the vortex shedding. He also noted that a correctly applied theoretical velocity profile may not produce a fully developed laminar flow profile, therefore careful consideration is needed before direct comparisons of numerical and experimental data are made. The second parameter, the blockage ratio β is defined as $\beta = \frac{A}{D}$ where A is the cylinder width, and D is the square distance between the lateral boundaries. The change in flow from free flow to confined flow is marked when the critical blockage ratio β_{crit} is reached and any further increase in β causes a suppression in the vortex shedding.

Davis *et al* [26] investigated flow past a square cylinder with near wall effects for a large range of Re for blockage ratios of $\beta = \frac{1}{4}$ and $\beta = \frac{1}{6}$, both numerically and experimentally. Davis found, and has since been confirmed by Breuer *et al* [13], Turki *et al* [84] to name a few, that the proximity of the walls leads to increased drag coefficient and Strouhal number.

5.10 Numerical Example - Flow past a square cylinder with near wall effects.

Model.

In this example the viscous incompressible flow past a square cylinder with near wall effects is considered. The blockages ratios are $\beta = \frac{1}{4}, \frac{1}{6}, \frac{1}{8}$ for $\mathrm{Re} = 60 - 300$ using the Generalized-α method. The inflow is assumed to be parabolic at infinity such that the flow matches fully developed laminar flow upstream of the blockage. The maximum velocity is denoted as $u_{\max}$, with width of the cylinder denoted A,

and density and the viscosity of the fluid ρ and μ respectively.

The simulations performed will use varying Re from $\mathrm{Re}_{crit} \rightarrow 300$[10] (timesteps and u_{max} will vary accordingly). Timestepping method will be the Generalized-α method for $p^h_\infty = 0.9$. In order to obtain the values of Re the following values were used $\rho = 1$, $\mu = 0.01$.

We should note that the long term solution for this problem is periodic vortex shedding which subjects the cylinder to oscillating lift and drag forces which act normal and parallel to the flow, F_L & F_D respectively, at a frequency f_s [11]. The non-dimensional lift C_L and drag C_D coefficients, and the Strouhal number Sr are repeated here:

$$\mathrm{Re} = \frac{\rho u_\infty A}{\mu} \qquad Sr = \frac{fA}{u_\infty} \tag{5.77}$$

$$C_D = \frac{2F_D}{\rho u_\infty^2 A} \qquad C_L = \frac{2F_L}{\rho u_\infty^2 A} \tag{5.78}$$

Geometry.

The only geometrical consideration in this example is the distance of the inflow and outflow boundaries from the square cylinder. Because of the effect an incorrect inflow velocity profile may have on the final outcome it is imperative that it is correctly positioned whilst taking numerical cost into account.

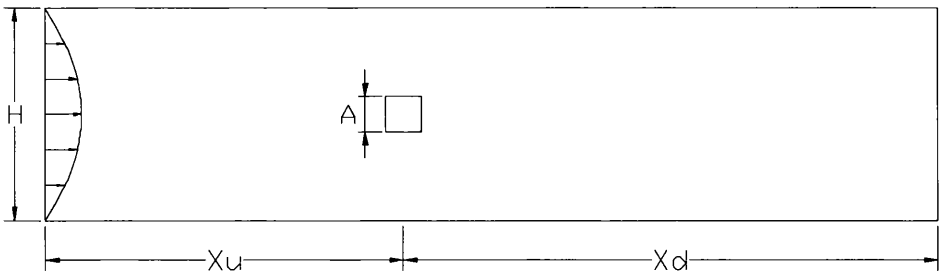


Figure 5.27: Domain

Inflow/Outflow Boundary.

Turki *et al* [84], Breuer *et al* [13] and the author considered solution accuracy with regards to varying the distances x_u and x_d. A series of sensitivity tests were performed in accordance with Turki and it was noted that for $x_u > 10$ widths and $x_d > 15$ widths negligible gains were made.

[10] Turki *et al* [84] provide guidance on the value of Re_{crit} for the blockage ratio

[11] The lift force oscillation frequency is half the frequency of the drag force oscillation. It is standard practice to use the frequency of the lift force.

Lateral Boundary.

The lateral boundary was placed at 2 widths, 3 widths, and 4 widths either side of the centre of the cylinder for the blockage ratios $\frac{1}{4}$, $\frac{1}{6}$, $\frac{1}{8}$ respectively.

In conclusion the cylinder is centred vertically in the domain and placed horizontally such that there is maximum accuracy whilst preserving efficiency. A slight imperfection in the geometry has been introduced into the model to initiate unsteady flow. This has been achieved by moving one of the lateral boundaries by 0.1% of the overall width of the domain, in order to place the cylinder slightly *off-centre* and remove the symmetrical nature of the geometry.

Boundary Conditions.

The Navier-Stokes problem must be completed with suitable initial and boundary conditions to form a well-posed initial boundary problem. In the following section the initial and boundary conditions that form the initial boundary value problem for this problem are explained.

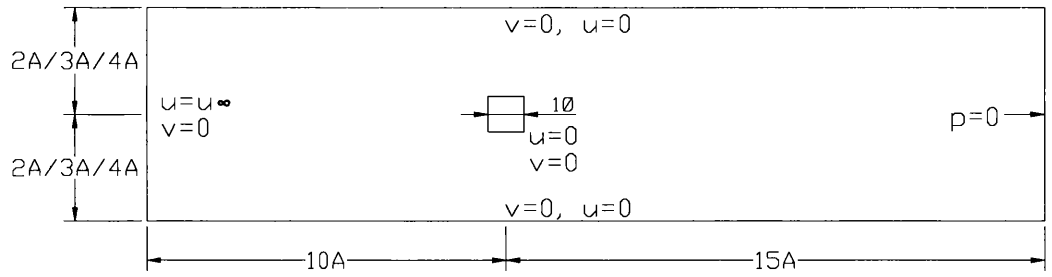


Figure 5.28: The problem setup - the domain with the boundary conditions - u, v, and p are the velocity in the x and y directions and pressure respectively

Velocity Boundary Conditions.

A parabolic velocity profile is assigned to the inflow boundary, where the fluid velocity u is given as a function of the u_{max} and the distance from the centre of flow y,

$$u = u_{\infty}\left(1 - \frac{y^2}{D^2}\right) \tag{5.79}$$

The outflow boundary is specified as Neumann-type boundary condition for velocity which corresponds to a zero viscous stress tensor. This has been chosen such that the flow crossing the boundary is not affected by the presence of the

boundary, therefore vortices approaching the outflow boundary cross the boundary undisturbed by its presence. There is a no-slip and no penetration condition on the cylinder wall and at the lateral boundaries.

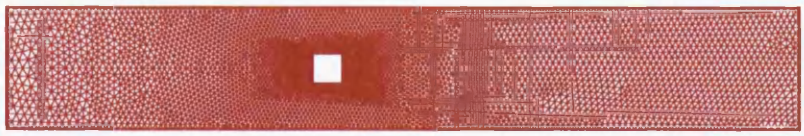
The inflow velocity u_{max} was reached by smoothly increasing the velocity component u at the inflow boundary from $0 \rightarrow u_{max}$ over a set time period, in a process described previously in Figure 5.12.

Pressure Boundary Conditions.

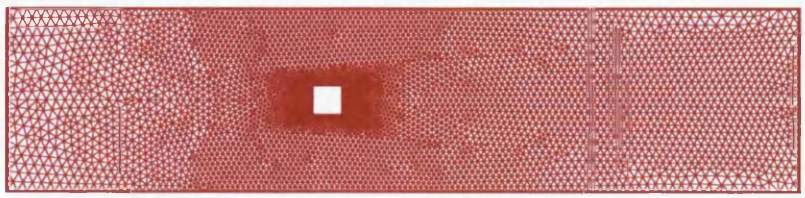
The pressure boundary is defined as previously described for the *flow past a circular cylinder* example, see Section5.7. It is fixed arbitrarily at the midpoint of the outflow boundary.

Mesh.

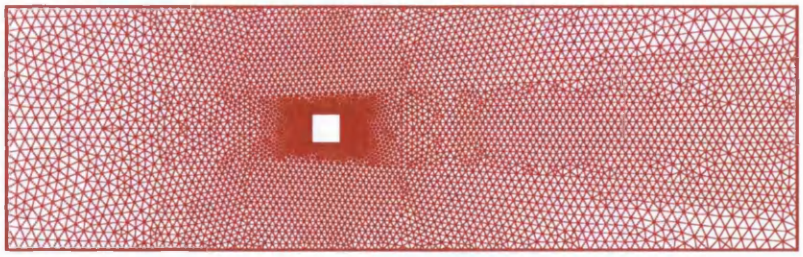
The 3 meshes were produced for this example using ELFEN software ($\frac{1}{4}$, $\frac{1}{6}$, $\frac{1}{8}$ blockages, with 9454, (9920), [9510] elements and 4933, (5140), and [4908] nodes). Each mesh had a different geometries to represent the changing blockage values, whilst maintaining the width of the square cylinder. The boundary conditions were applied as prescribed previously. To ensure an optimal solution the mesh near to the cylinder (160 nodes in total to define the cylinder) was refined in order to model the dynamics of the problem accurately. In addition to refined meshing around the cylinder refinements are also placed at the lateral boundary near the blockage in order to improve the resolution of the solution.



(a) $\frac{1}{4}$ blockage



(b) $\frac{1}{6}$ blockage



(c) $\frac{1}{8}$ blockage

Figure 5.29: Meshes for flow past a square cylinder with near wall effects for $\frac{1}{4}$, $\frac{1}{6}$, $\frac{1}{8}$ blockages $\Rightarrow$ 9454, (9920), [9510] elements and 4933, (5140), and [4908] nodes

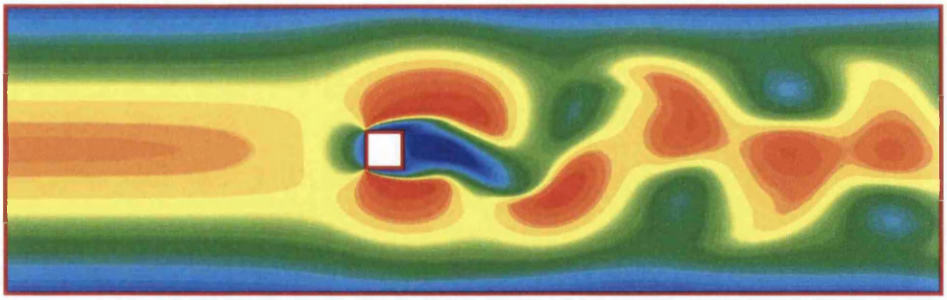
Numerical Results.

Figure 5.30→Figure 5.31 are velocity/pressure plots for the solution Re = 200 for $\frac{1}{4}$, $\frac{1}{8}$ blockages. These plots clearly demonstrate the vortices shedding from behind the cylinder, and the restrictive effect of the reducing the distance between the exterior walls.

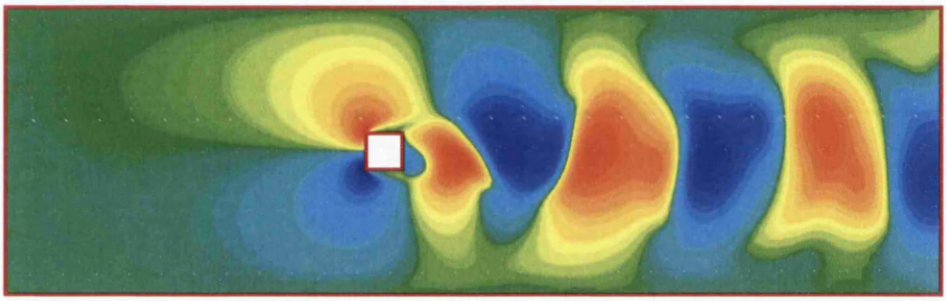
Figure 5.32 shows the evolution of the lift coefficient for the $\frac{1}{4}$, $\frac{1}{6}$, $\frac{1}{8}$ blockage ratios and free flow, demonstrating the effect of blockage on lift.

Figure 5.33 shows the evolution of pressure isolines for the $\frac{1}{4}$ blockage.

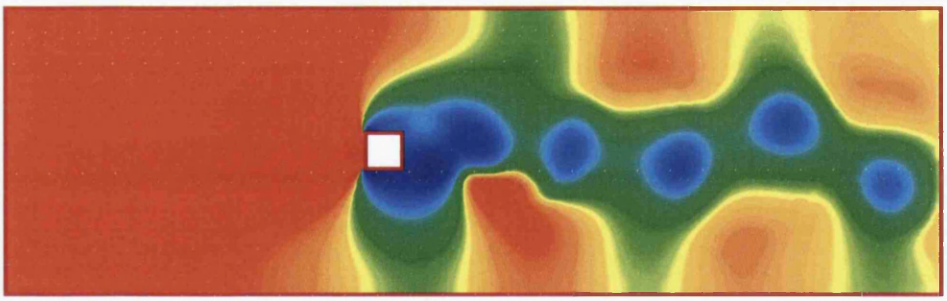
Figure 5.34→Figure 5.36 show the numerical results recorded for this example, Strouhal numbers, lift and drag coefficients against the Reynolds number, plotted along side data from Breuer *et al* [13] and Turki *et al* [84]. The plots for the $\frac{1}{4}$ blockage ratio diverges from Turki's observed results at Re $\approx$ 200, however Turki's results for Strouhal numbers, lift and drag coefficients seem to show a change in flow at Re $\approx$ 200 which is not captured in the results obtained in this work. The most obvious explanation is the large discrepancy in spatial refinement of the mesh used here compared to Turki's meshes.



(a) u

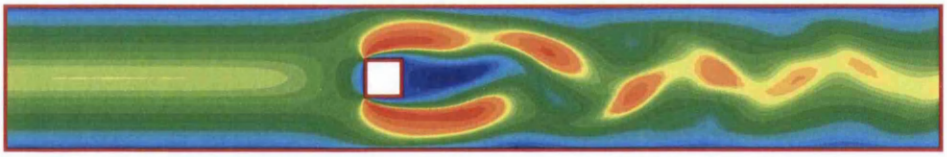


(b) v

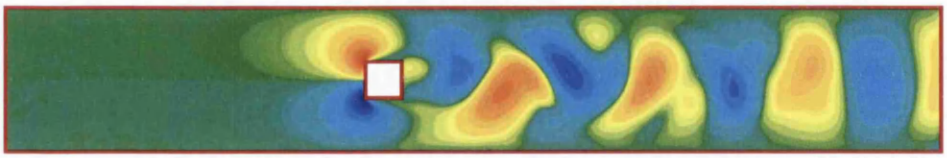


(c) p

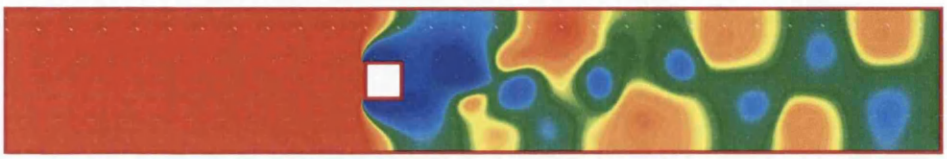
Figure 5.30: $\frac{1}{8}$ blockage - Velocity/Pressure contour plots - Re $= 200$, Generalized-α method $\rho_\infty^h = 0.9$, $n_{elem} = 5436$



(a) u

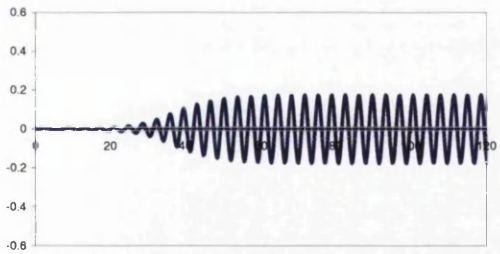


(b) v

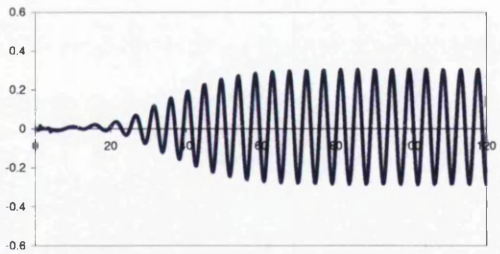


(c) p

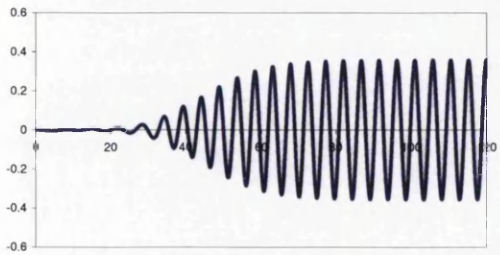
Figure 5.31: $\frac{1}{4}$ blockage - Velocity/Pressure contour plots - Re = 200, Generalized-α method $\rho_\infty^h = 0.9$, $n_{elem} = 5436$



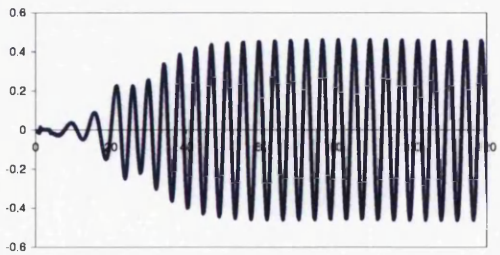
(a) $\frac{1}{4}$ Blockage



(b) $\frac{1}{6}$ Blockage



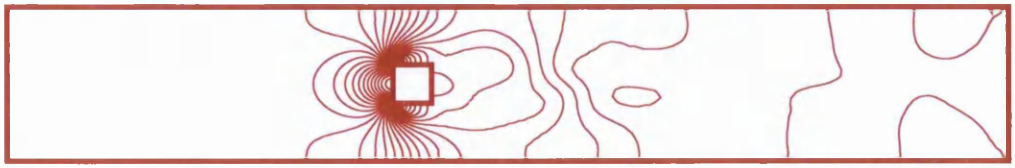
(c) $\frac{1}{8}$ Blockage



(d) Free flow

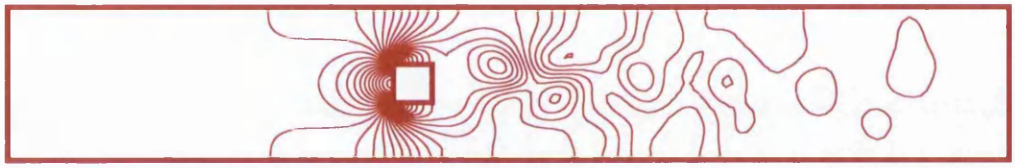
Figure 5.32: Evolution of lift C_L coefficients in time. Re = 150, $\Delta t = 0.01$, Generalized-α method $\rho_\infty^h = 0.9$

(a) Streamlines $t = 300s$

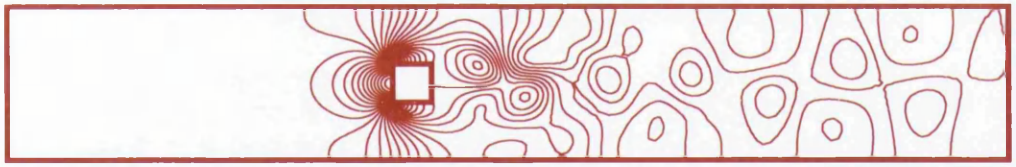


(b) $t = 1s$

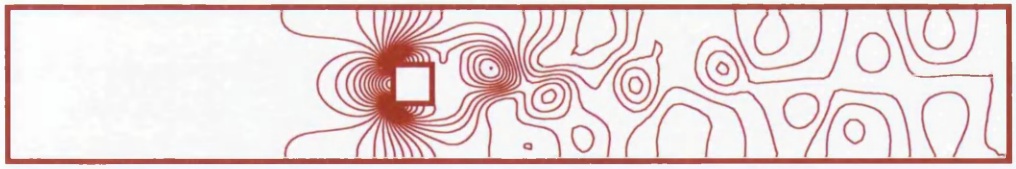
(c) $t = 5s$



(d) $t = 15s$

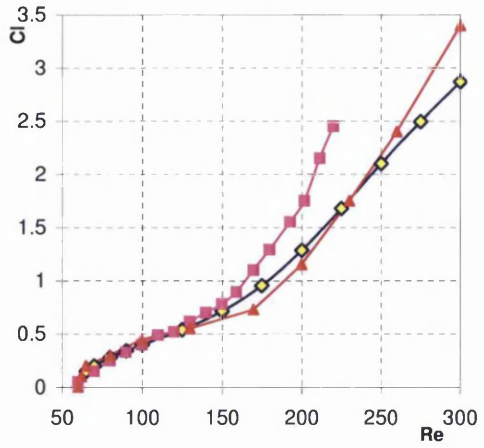


(e) $t = 40s$

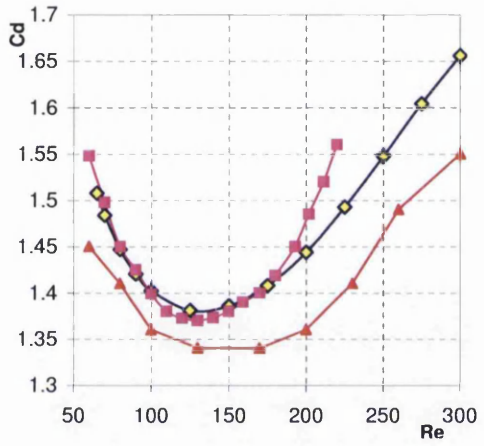


(f) $t = 60s$

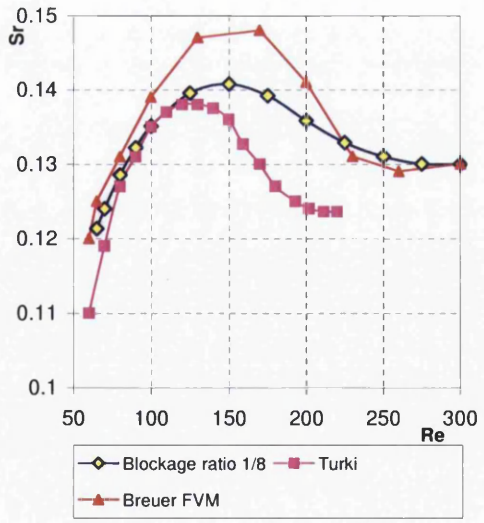
Figure 5.33: (a) Streamlines, (b)-(f) Evolution of pressure Isolines - Re = 150, $\Delta t = 0.01$, Generalized-α method $\rho_\infty^h = 0.9$, $\frac{1}{4}$ Blockage



(a) $C_L^{max} - C_L^{min}$

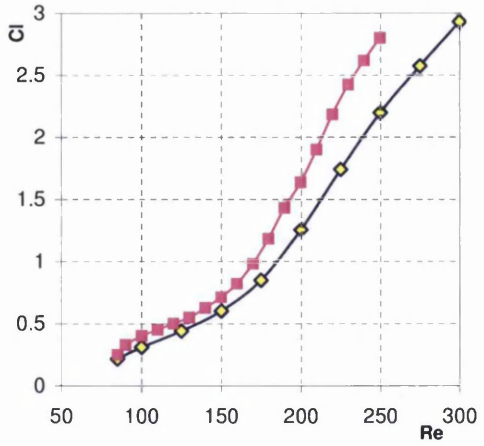


(b) Time averaged C_D

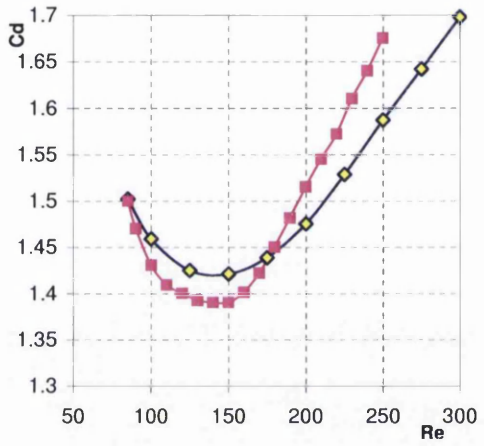


(c) Sr

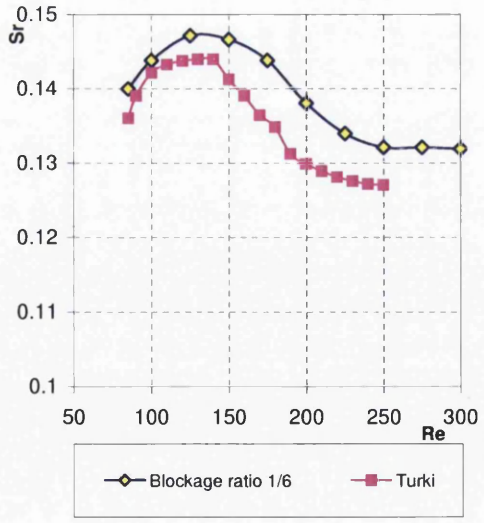
Figure 5.34: Comparison of $\frac{1}{8}$ blockage results for Strouhal number Sr, lift C_L and drag C_D coefficients. Author - $\Delta t = 0.1 \rightarrow 0.01$, 4908 nodes, 160 nodes over cylinder surface. Turki *et al* [84] - $\Delta t = 0.01$, 9394 nodes, 400 nodes over cylinder surface



(a) $C_L^{max} - C_L^{min}$

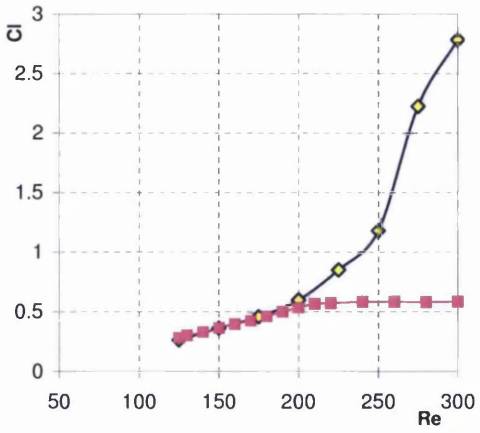


(b) Time averaged C_D

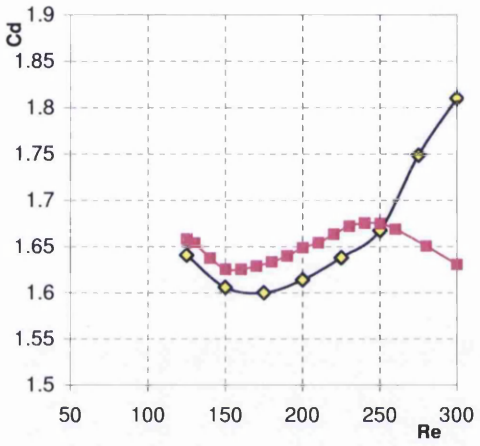


(c) Sr

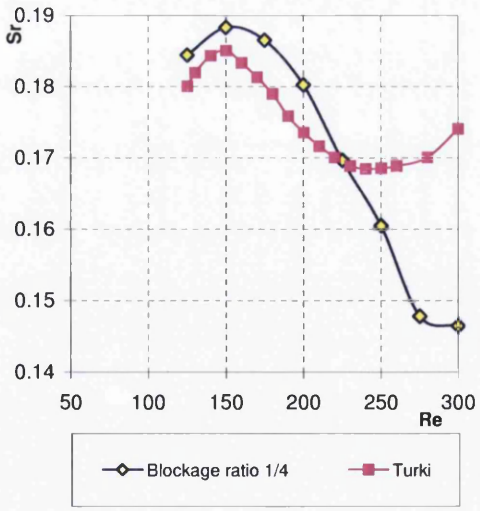
Figure 5.35: Comparison of $\frac{1}{6}$ blockage results for Strouhal number Sr, lift C_L and drag C_D coefficients. Author - $\Delta t = 0.1 \rightarrow 0.01$, 5140 nodes, 160 nodes over cylinder surface. Turki *et al* [84] - $\Delta t = 0.01$, 8174 nodes, 400 nodes over cylinder surface



(a) $C_L^{max} - C_L^{min}$



(b) Time averaged C_D



(c) Sr

Figure 5.36: Comparison of $\frac{1}{4}$ blockage results for Strouhal number Sr, lift C_L and drag C_D coefficients. Author - $\Delta t = 0.1 \rightarrow 0.01$, 4933 nodes, 160 nodes over cylinder surface. Turki *et al* [84] - $\Delta t = 0.01$, 6954 nodes, 400 nodes over cylinder surface

Discussion.

This example has reaffirmed the conclusions drawn for *Propagating rectangular wave* and *Flow past a circular cylinder.* The Generalized-α method $\rho_\infty^h = 0.9$ has produced results which are comparable with data provided by Turki *et al* [84] and Breuer *et al* [13], and stabilization has given solutions that are free of spurious oscillation.

The solution results plotted for the Strouhal number, lift and drag coefficients, see Figure 5.34-Figure 5.36, can be favourably compared with data obtained from Turki and Breuer. However the results for the $\frac{1}{4}$ blockage for Re $>$ 200 are inconsistent with Turki's observations. It is the authors considered opinion that the inconsistency is due to differences in the construction of the domain, and not due to stabilization nor timestepping algorithm.

5.11 Conclusions.

In this chapter we have focused on the stabilized unsteady incompressible viscous Navier-Stokes equations, and the time integration schemes that are fundamental to the modelling unsteady flows, showcasing three numerical examples which demonstrate the solution properties.

From the steady state FEM formulation developed in Chapter 4, we included the time dependent terms of the Navier-Stokes equations that had otherwise been ignored, to arrive at the formulation for unsteady flow. The inclusion of time dependent terms required a suitable method to model both the spatial and temporal domains and we chose a method commonly known as Semi-discrete time integration schemes to perform this task. Semi-Discrete methods only discretize the spatial domain using the Finite Element Method, whilst leaving a system of ordinary differential equations to be integrated in time. Discretization of the spatial domain has been dealt with previously however we required a time integration scheme to model the temporal domain, such as the Generalized-α method and the Generalized Midpoint rule. Both the Generalized-α method and the Generalized Midpoint rule were developed and implemented into the FEM formulation and the solutions were obtained for 3 different numerical examples. The observations from the numerical examples are summarized here.

- Irrespective of the time integration scheme, all 3 examples converge to a mesh dependent solution, however as refinement increase the variance between solutions decreases, until a plateau is reached.

- The Generalized-α method converges to the solution faster than the Generalized Midpoint rule. However it was noted that both schemes are identical when $\rho_\infty^h = 1.0$, otherwise known as the Trapezoidal rule.

- It was determined that $\rho_\infty^h = 1.0$ could lead to *Sawtoothing* therefore it is more practicable to use a value of ρ_∞^h which is distinctly not 1.0. We chose $\rho_\infty^h = 0.9$ as it maintains the advantages of $\rho_\infty^h = 1.0$ yet does not suffer the disadvantages.

- The pressure field is free of the spurious oscillations which were apparent in the unstabilized lid driven cavity problem. The velocity field is also free of non physical oscillations for realistic time steps.

- The Newton-Raphson performed well, with quadratic convergence for all timesteps and mesh densities, unless the timestep was either too large or the spatial discretization too coarse then the procedure failed.

From these observations we can conclude that the Generalized-α method is an efficient time integration scheme, which when used in conjunction with stabilization and the Newton-Raphson procedure produces a solution procedure that is robust.

In the examples for chapters 3 and 4 we encountered problems which contained interfaces in one of the solution variables, however the interface was largely ignored as the focus was concentrated elsewhere. In the next chapter the work focuses on modelling fluid flow problems that contain interfaces and the methods that can be integrated into the stabilized formulation in an attempt to model the interface more accurately.

Chapter 6

Modelling Interfaces in Fluid Dynamic Problems.

Interfaces occur in many physical problems, and are evident in both fluid and solid dynamics problems, e.g. material interfaces, strain localization, shocks, moving surfaces, crack propagation, multiphase fluid flow etc. and, of course, are the major phenomenon found in fluid-structure interaction problems. Modelling problems with interfaces has proven to be challenging concept, and one which is considerably more complex to model. Often the interface will be the focus of the analysis, i.e. such as in crack propagation, and an accurate representation of the interface remains the major obstacle in the simulation of this class of problem.

In this chapter we are concerned with the accurate modelling of the type of interfaces that can occur in fluid flows, e.g. multiphase fluids or fluid structure interaction. There are various approaches to this problem, which are briefly explained, however we only concern ourselves with the eXtended Finite Element Method (X-FEM) and the process of enrichment that it entails. Initially we discuss the development of X-FEM from a General Finite Element Method with added terms in a partition of unity framework, followed by a simple 1D bar problem as a demonstration of its capability to model interfaces accurately. Recognizing that X-FEM is considerably more complicated to implement than FEM we conclude this chapter by describing the actual processes of implementation, including local spatial refinement algorithms, search algorithms etc.

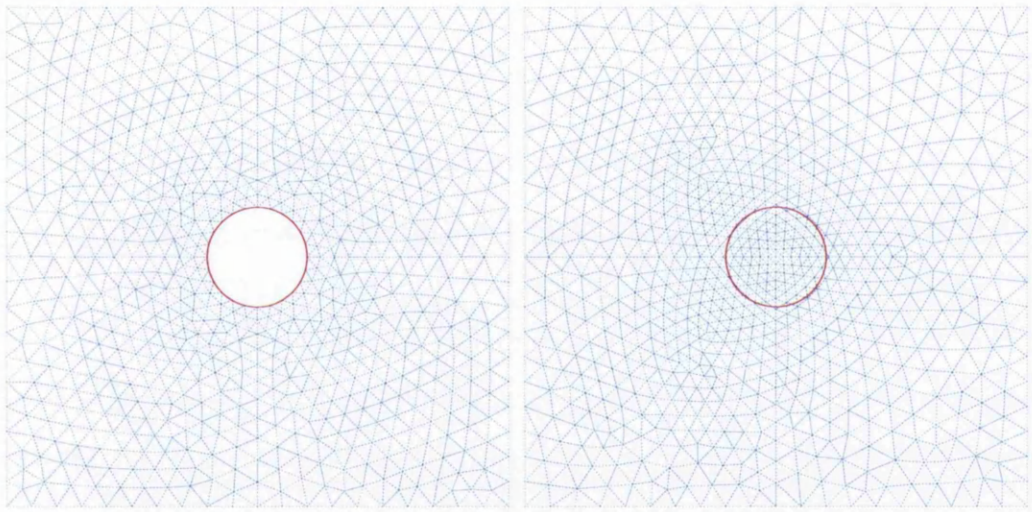
6.1 Introduction.

In modelling problems with interfaces the difficulties encountered are twofold:

i) capturing the interface accurately

ii) knowing the position of the interface at all times

In order to achieve a successful result the numerical algorithm needs to take account of the above criteria. Generally numerical simulations of the interfaces are based on two distinct methods, which approach the analysis from completely different directions. They are commonly known as Interface Tracking (IT) and Interface Capturing (IC) algorithms[1] are illustrated in Figure 6.1 and these are explained in brief below:



(a) Interface Tracking (b) Interface Capturing

Figure 6.1: Examples of Interface modelling methods

6.1.1 Interface Tracking.

In Interface Tracking the mesh topology is designed specifically to match the geometry of the interface e.g. placement of nodes on the actual line of the interface[2]. Should the problem include an evolving interface, i.e. shock, or if the interface is moving, i.e. free surface, then the mesh quickly becomes outdated, and continued

[1]Methods also used to model interfaces include Meshless methods but these are not discussed here

[2]Numerical simulation using General Finite Element Methods have difficulty capturing these interfaces with any accuracy if they cut elements

accuracy can no longer be guaranteed. Commonly in these situations the mesh is altered to reflect the ever changing geometry of the problem. If the displacements of the interface are small it may be possible to match the movement of interface by updating the mesh without changing the connectivities. However this process can lead to extremely deformed meshes and requires other considerations to be taken into account as the numerical simulation would no longer be strictly Eulerian, but Arbitrary Lagrangian Eulerian (ALE). In these situations, i.e. when the deformation of the mesh becomes too large for the ALE approach to work, it is often necessary to perform remeshing i.e. generating a new set of nodes and connectivities. Remeshing is expensive, requiring a complete regeneration of the mesh, projection of data from the old mesh to the new mesh and adapting visualization techniques during postprocessing, all of which are costly and time consuming, even more so if the requirement for remeshing is at every timestep.

Interface Tracking generally guarantees a level of accuracy but sacrifices the efficiency of the analysis and should be avoided wherever possible if any other reliable method exists. Examples of Interface Tracking algorithms are Arbitrary Lagrangian Eulerian methods, see Huerta *et al* [41] and deforming Space-Time Finite Element formulations, see Tezduyar *et al* [79, 80].

6.1.2 Interface Capturing.

The Interface Capturing methods use (commonly) a fixed mesh, which does not conform to the interfaces present[3] allowing, instead, for the interface to cut elements at any point. The interface is marked by function which is itself discontinuous or the gradient is discontinuous, defining two distinct values: this function is commonly known as an *Enrichment* function. The process of using enrichment functions to define the interface places heavy reliance on the resolution of the mesh to capture the interface accurately, and often highly refined meshes are required to achieve an acceptable level of accuracy - even so this often produces results which are not as accurate as those obtained through Interfacing Tracking methods. To avoid large meshes whilst maintaining accuracy Enhanced Discretization Interface Capturing Techniques (EDICT) have been introduced. EDICT avoids the use of excessively refined meshes by refining the mesh at the interface only, thus giving improved definition of the interface without the added

[3]The independence of the analysis from the mesh produces a method which is considerably more flexible when considering moving interfaces

computational cost.

Examples of Interface Capturing algorithms are Volume of Fluid methods, see [40], and Level Set methods, see Sukumar [73], Stolarska *et al* [72] and Chessa and Zienkiewicz [17].

In this chapter the focus turns to the modelling of arbitrary interfaces found commonly in fluid dynamics problems, using an Interface Capturing method, commonly known as the eXtended Finite Element Method (X-FEM).

The main attraction of X-FEM is the ability to use an Eulerian mesh, where the interface causing the effect being studied is not physically present in the mesh, e.g. in flow past a cylinder, the cylinder is represented by an absence of mesh, i.e. normally a hole is left in the mesh to represent the cylinder, see Figure 6.1. X-FEM in contrast enables the interface to be placed *on top* of the mesh mathematically, and this allows for certain problems to be solved where otherwise it would require substantial and computationally expensive re-meshing, and postprocessing.

Following a brief introduction into X-FEM, there will be a more detailed look at the method, which will include a brief 1D example and the formulation of the method. Finally the implementation of X-FEM into FEAP will be discussed and examples of the method produced with concluding remarks.

6.2 eXtended Finite Element Methods (X-FEM).

X-FEM is the General Finite Element Method with added terms in a Partition of Unity Method (PUM) framework.

The eXtended Finite Element Method (X-FEM), first proposed by Belytschko and Black [9] and Möes *et al* [60], is the union of the General Finite Element Method (GFEM) and additional functions, in a framework provided by the Partition of Unity Method (PUM), see Babuška *et al* [5]. This process allows for some aspects of the functional behaviour of the solution field, which are know *a priori*, to be included in the analysis, thereby enriching the solution. In addition many of the techniques and methods commonly associated with FEM can be applied to X-FEM. Therefore the methods and formulations previously discussed in Chapter 3, 4 & 5 i.e. the Stabilized Finite Element Method and time integration schemes, are applicable to X-FEM.

Belytschko *et al* [10] first considered the possibilities of modelling interfaces using PUM in the X-FEM structure, and this method was first applied to discontinuity analysis for solid mechanics problems, specifically, for the analysis of crack propagation. Since then X-FEM has been used in many classes of problems, including 3D crack modelling - Sukumar *et al* [74], in particulate Stokes flow - Wagner *et al* [86], whereas discontinuities in derivatives have been applied by Chessa *et al* [18] for solidification problems.

In order to understand the mathematical background of X-FEM a brief overview of the PUM is given here.

6.2.1 Partition of Unity Enrichment (PUM).

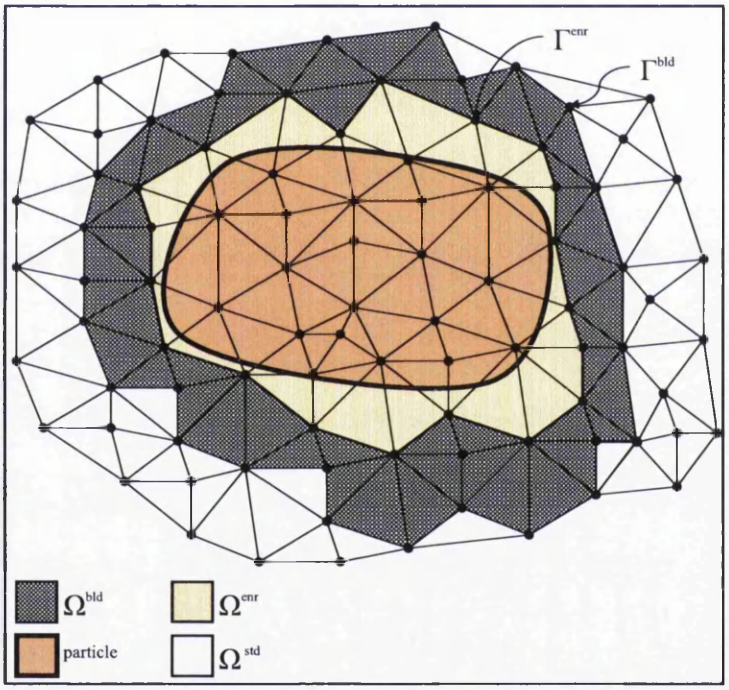


Figure 6.2: Typical discretization of a domain with enrichment. Illustrates the three distinct types of domain found in a local enrichment scheme, i.e. standard domain, enriched domain and finally the transitional domain

$$\Omega^{std} \subset \Omega, \quad \Omega^{enr} \subset \Omega, \quad \Omega^{bld} \subset \Omega, \tag{6.1}$$

The first type is the standard domain Ω^{std}, and in this domain the elements are not enriched, i.e. none of the element nodes are enriched. This is the default

domain type. The second domain type, Enriched domain Ω^{enr} requires that all elemental nodes are enriched. For the final domain type, Transitional Domain Ω^{bld}, the element must contain enriched nodes, as well unenriched nodes. The position of an element in relation to the interface will dictate into which domain a particular element will fall.

A description of each of the domain types is given in more detail below:

6.2.1.1 Ω - Standard Domain.

The domain Ω is discretized into finite elements and nodes

$$\Omega = \bigcup_{e \in \mathcal{E}} \Omega^e \tag{6.2}$$

$$\mathcal{J} = \bigcup_{I \in \mathcal{N}} \boldsymbol{x}_I \tag{6.3}$$

where Ω is sub divided into elements (Ω^e), and nodes ($\boldsymbol{x}_I$). $\mathcal{E}$ and $\mathcal{N}$ represent the sets of elements and nodes respectively. The General Finite Element basis is given as:

$$\mathcal{B} = \bigcup_{I \in \mathcal{N}} N_I(\boldsymbol{x}) \tag{6.4}$$

where $N_I(\boldsymbol{x})$ is the shape function of node I and $\mathcal{B}$ spans the space of piecewise continuous polynomials. The General Finite Element approximation of a scalar variable $\mathrm{w}(\boldsymbol{x})$ is written as:

$$\mathrm{w}^h(\boldsymbol{x}) = \sum_{I \in \mathcal{N}} N_I(\boldsymbol{x}) \mathrm{w}_I \tag{6.5}$$

The shape functions form a partition of unity

$$\sum_{I \in \mathcal{N}} N_I(\boldsymbol{x}) = 1 \tag{6.6}$$

Therefore it follows from (6.6) that:

$$\sum_{I \in \mathcal{N}} N_I(\boldsymbol{x}) \phi(\boldsymbol{x}) = \phi^h(\boldsymbol{x}) \tag{6.7}$$

thus proving that any function $\phi(\boldsymbol{x})$ can be reproduced in the domain by the local function $N_I(\boldsymbol{x})\phi(\boldsymbol{x})$. This is an important consequence of using the PUM framework for enrichment.

6.2.1.2 Ω^{enr} - Enriched Domain.

Whilst it is possible to enrich the entire domain, this is rarely needed and often would prove to be an inefficient use of computational resources. It is more common to restrict enrichment to the sections of the domain Ω where it is required. The union of all subdomains in which the solution is to be enriched is denoted by Ω^{enr}:

$$\Omega^{enr} = \bigcup_{e \in \mathcal{E}^{enr}} \Omega^e \tag{6.8}$$

where $\mathcal{E}^{enr} \subset \mathcal{E}$, and represents the elements that require enriching. If any node I lies in Ω^{enr} then it becomes an enriched node and included in the subset $\mathcal{N}^{enr}$, where $\mathcal{N}^{enr} \subset \mathcal{N}$.

If $\phi(\boldsymbol{x})$ is an enrichment function, the enriched Finite Element basis, for a local Partition of Unity method, is given by

$$\mathcal{B}^{enr} = \mathcal{B} \oplus \bigcup_{I \in \mathcal{N}^{enr}} N_I(\boldsymbol{x})\phi(\boldsymbol{x}) \tag{6.9}$$

If the basis of the scalar variable $\mathrm{w}(\boldsymbol{x})$, see (6.5) is enriched with $\phi(\boldsymbol{x})$, the approximation becomes:

$$\mathrm{w}^h(\boldsymbol{x}) = \sum_{I \in \mathcal{N}} N_I(\boldsymbol{x})\mathrm{w}_I + \sum_{I \in \mathcal{N}^{enr}} N_I(\boldsymbol{x})\phi(\boldsymbol{x})a_I \tag{6.10}$$

where a_I is the scalar enrichment variable. By fixing the enrichment scalar variable a_I to zero, see (6.10), the General Finite Element interpolation (6.5) is recovered. The enrichment shape function N_I helps to maintain sparsity of the global system matrix by contributing to $\mathrm{w}(\boldsymbol{x})$ only where N_I is non-zero, i.e. within the support of the node I.

In some situations it may be necessary to enrich using vector enrichment, therefore the resulting approximation would read:

$$\mathbf{w}^h(\boldsymbol{x}) = \sum_{I \in \mathcal{N}} N_I(\boldsymbol{x})\mathbf{w}_I + \sum_{\alpha=1}^{n_E} \sum_{I \in \mathcal{N}^{enr}} N_I(\boldsymbol{x})\phi_\alpha(\boldsymbol{x})\boldsymbol{a}_{\alpha I} \tag{6.11}$$

where n_E is the order of the enrichment vector.

6.2.1.3 Ω^{bld} - Transitional Domain (Blending).

The third and final domain, the transitional domain, acts as a buffer zone between domains with fully enriched elements and domains with standard elements. This

area is considered to be partially enriched, and therefore the elements do not form a partition of unity.

Blending is important and necessary to the approximation properties and convergence of the enrichment scheme. However it has been shown in Sukumar *et al* [73] that blending effects the convergence rate of the solution, yet error analysis of methods which ignore blending have also been shown to be deficient, see Chessa *et al* [19]. Therefore it is important that blending is implemented and that the choice of blending is carefully considered as it can often be crucial to the success of the solution. Both Chessa and Sukumar suggested procedures, such as enhanced strain elements, to improve the blending algorithm and some success was achieved, however in this thesis only *standard* blending is implemented.

Remark 6.1: In Brief.
Note that if $a_{\alpha I} = 1$ *and* $\mathrm{w}_I = 0$ *then:*

$$\mathrm{w}^h(\boldsymbol{x}) = \sum_{\alpha=1}^{n_E} \sum_{I \in \mathcal{N}^{enr}} \begin{cases} N_I(\boldsymbol{x})\phi_\alpha(\boldsymbol{x}) = \phi_\alpha^h(\boldsymbol{x}) & \forall \boldsymbol{x} \in \Omega^{enr} \\ N_I(\boldsymbol{x})\phi_\alpha(\boldsymbol{x}) \neq \phi_\alpha^h(\boldsymbol{x}) & \forall \boldsymbol{x} \in \Omega^{bld} \\ N_I(\boldsymbol{x})\phi_\alpha(\boldsymbol{x}) = 0 & \forall \boldsymbol{x} \in \Omega \end{cases} \tag{6.12}$$

6.3 Implementation of the X-FEM for fluid flows.

X-FEM is typically a complex procedure to implement, and the numerical approach to the problem requires careful consideration. In this section the entire process will be laid bare for the readers understanding and examples will be used, wherever possible, to ensure maximum clarity.

The remainder of this chapter will be laid out in the following manner. Initially there will be a brief introduction into the requirements for a successful implementation, followed by a simple 1D example which will give an overall impression on the methods involved. Subsequently the algorithms necessary to implement X-FEM and enrichment in both 1D and 2D numerical simulations will be developed and discussed in detail with examples to illustrate why it is important to the overall success of the solution.

As mentioned in the previous sections, the basis of X-FEM for fluids is the General Finite Element Method as developed in Chapters 3, 4, and 5, therefore in order

to successfully implement X-FEM we need to build upon techniques discussed in the previous chapters, and add new methods specialized for enrichment. The transformation from General Finite Element Method to X-FEM can be achieved in three steps:

- augment the General Finite Element discretization with a Partition of Unity method, see (6.11), and select an appropriate enrichment function.
- implement an algorithm to locate and position the interface
- implement an algorithm to accurately model the interface.

At this point it seems appropriate to include a simple 1D example to demonstrate X-FEM in its most simple form.

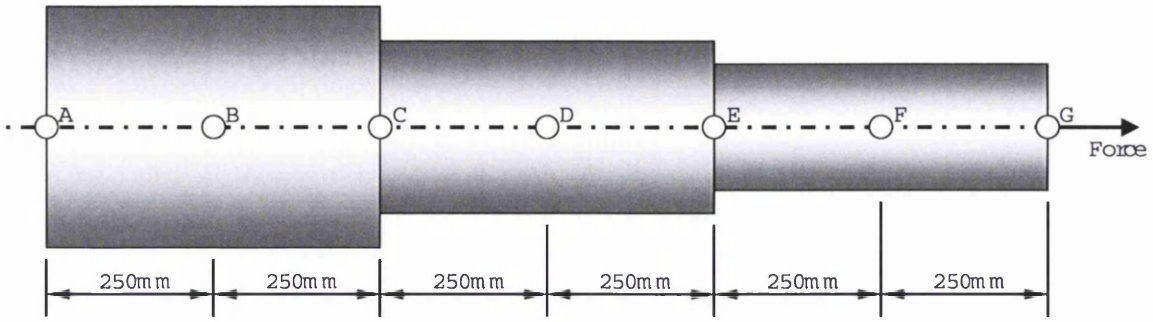
6.4 Numerical Example - X-FEM Simple 1D Example.

Now is a good time to briefly review a simple enrichment example, and test the ability of the enrichment method to model discontinuities:

Problem 1. Three rods, of similar length but of differing materials and cross sections, are joined end on end, to form a bar of length (1500mm). The bar, which contains 2 discontinuities - positioned at 500mm and 1000mm from node A, is restrained at node A and is subjected to an axial force at node G. Calculate the displacement at node C, E & G.

A comparison will be made between solutions that situate the discontinuity on a node (which should exactly mirror the analytical solution) and those that do not:

- analytical solution (using the stress-strain relationship)
- 3 element FEM (discontinuities on nodes)
- 4 element FEM (discontinuities mid-element)
- 4 element FEM with enrichment using Gauss Quadrature (discontinuities mid-element)



$E_{AC} = 70 \times 10^3 N/mm$ $E_{CE} = 200 \times 10^3 N/mm$ $E_{EG} = 100 \times 10^3 N/mm$

$A_{AC} = 1963.5mm^2$ $A_{CE} = 1134.1mm^2$ $A_{EG} = 1000mm^2$

$\sigma_{AC} = 40MPa$

The force Frequired to give the prescribed stress value σ_{AC}.

$$F = \sigma_{AC} \times A_{AC} = 7.854 \times 10^4 N$$

$$\Rightarrow \quad \sigma_{AC} = 40.00MPa$$
$$\sigma_{CE} = 69.25MPa$$
$$\sigma_{EG} = 78.54MPa$$

The stress values in the three distinct bar sections AC,CE & EF, highlight the stress discontinuities in the bar. These discontinuities will need to be properly considered if an accurate solution is required.

Analytical Solution (i).

The stress values are used to calculate the strain, and displacements in each rod:

$$\epsilon = \frac{dU}{dx} = \frac{\sigma}{E} \tag{6.13}$$

$$\Rightarrow U = \frac{\sigma}{E} \times L \tag{6.14}$$

where ϵ is the engineering strain, E is Young's modulus, L is original length of the rod, and U is displacement or change in length of the rod due to the axial force F .

It has been demonstrated that the problem is discontinuous, with regards to stress, and (6.13) shows that U must also be discontinuous.

The exact nodal displacements for this simple problem are:

$$U_A = 0.000mm$$
$$U_C = 0.286mm$$
$$U_E = 0.459mm$$
$$U_G = 0.852mm$$

6.4.1 FEM Solution.

The governing equation for axial loading of a 1D bar element can be written as:

$$EA\frac{d^2U}{dx^2} = 0 \qquad \text{where} \tag{6.15a}$$
$$U|_{x=0} = 0 \tag{6.15b}$$
$$\frac{dU}{dx}|_{x=L} = F \tag{6.15c}$$

using the Galerkin formulation to obtain the weak form, where w is the test function

$$\int_0^L w\left[EA\frac{d^2U}{dx^2}\right]dx = 0 \tag{6.16}$$

integrating by parts and substituting in Equ (6.15c)

$$\int_0^L \frac{dw}{dx}EA\frac{dU}{dx}dx = F \tag{6.17}$$

and the General Finite Element discretization applied to displacement U and the test function w can be written as:

$$U^h = \sum_i N_i U_i \qquad w^h = \sum_i N_i w_i \tag{6.18}$$

therefore the axial stiffness for a 1D bar is:

$$k_e = E_e A_e \int_0^{L_e} B^T B dx \qquad \text{where} \qquad B^T = \begin{bmatrix} \frac{dN_1}{dx} \\ \frac{dN_2}{dx} \end{bmatrix} \tag{6.19}$$

where A_e, k_e and $N_1(x)\,\&\,N_2(x)$ are the element cross sectional area, element stiffness and 1D linear shape functions respectively, where:

$$N_1(x) = \frac{L-x}{L} \qquad N_2(x) = \frac{x}{L} \tag{6.20}$$

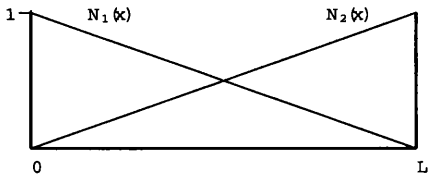


Figure 6.3: Graphical representation of N_1 & N_2

Therefore the element stiffness k_e can be rewritten:

$$k_e = \frac{E_e A_e}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{6.21}$$

Three element FEM solution (ii).

The three elements are: AC, CE, EG. Inserting the elemental values of Young's modulus, cross sectional area and length, for each section of the bar, into (6.21) produces three different local stiffness matrices.

$$k_{AC} = \frac{70 \times 10^3 \times 1963.5}{500} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$k_{CE} = \frac{200 \times 10^3 \times 1134.1}{500} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$k_{EG} = \frac{100 \times 10^3 \times 1000.0}{500} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

These can be assembled together to form the global stiffness matrix - the global stiffness matrix is assembled from the components of the three local stiffness matrices following the format laid out in (6.22).

$$\begin{bmatrix} k_{AC}^{11} & k_{AC}^{12} & 0 & 0 \\ k_{AC}^{21} & k_{AC}^{22} + k_{CE}^{11} & k_{CE}^{12} & 0 \\ 0 & k_{CE}^{21} & k_{CE}^{22} + k_{EG}^{11} & k_{EG}^{12} \\ 0 & 0 & k_{EG}^{21} & k_{EG}^{22} \end{bmatrix} \tag{6.22}$$

resulting in the following stiffness matrix:

$$K_{Global} = \begin{bmatrix} 274890 & -274890 & 0 & 0 \\ -274890 & 728530 & -453640 & 0 \\ 0 & -453640 & 653640 & -200000 \\ 0 & 0 & -200000 & 200000 \end{bmatrix} \tag{6.23}$$

The loading vector is also assembled in a similar manner.

$$F_{Global} = \begin{bmatrix} R_A \\ 0 \\ 0 \\ 78540 \end{bmatrix} \tag{6.24}$$

where R_A is the reaction at node A. (6.23) & (6.24) are assembled with the vector of unknown displacements to create a system of equations which can be solved.

$$\begin{bmatrix} 274890 & -274890 & 0 & 0 \\ -274890 & 728530 & -453640 & 0 \\ 0 & -453640 & 653640 & -200000 \\ 0 & 0 & -200000 & 200000 \end{bmatrix} \begin{bmatrix} U_A \\ U_C \\ U_E \\ U_F \end{bmatrix} = \begin{bmatrix} R_A \\ 0 \\ 0 \\ 78540 \end{bmatrix} \tag{6.25}$$

The nodal displacements obtained from the FEM solution for the three element problem are:

$$\begin{aligned} U_A &= 0.000mm \\ U_C &= 0.286mm \\ U_E &= 0.459mm \\ U_G &= 0.852mm \end{aligned}$$

Four element FEM solution (iii).

Once again solving the problem described in Section 6.4; this time there are four elements AB, BD, DF, FG, two of which contain discontinuities at their midpoint. The values of Young's modulus and cross sectional area for elements with a discontinuity within its boundary will be assumed to be constant, e.g. for mid element discontinuity $E_{CE} = \frac{1}{2}E_{CD} + \frac{1}{2}E_{DE}$.

There is no need to explain or even show the process leading to the four element solution, as it has been covered previously by the three element solution. However the results are as follows:

$$\begin{aligned} U_A &= 0.000mm \\ U_B &= 0.143mm \quad \Rightarrow \quad U_C = 0.251mm \\ U_D &= 0.358mm \quad \Rightarrow \quad U_E = 0.478mm \\ U_F &= 0.598mm \\ U_G &= 0.795mm \end{aligned}$$

6.4.2 FEM Solution with Enrichment.

The formulation is identical to the Finite Element solution described above. The addition of enrichment to the solution algorithm requires increased degrees of freedom per node, and a slightly more complex axial stiffness matrix.

The general Finite Element interpolation (6.18) is augmented by enrichment functions to become:

$$\begin{aligned} U^h(U_i, a_i, x) &= \sum_i N_i(x)(U_i + \phi(x)a_i) \\ &= \sum_i N_i(x)U_i + \sum_i N_i(x)\phi(x)a_i \end{aligned} \tag{6.26}$$

where a_i is the enrichment variable, and $\phi(x)$ the enrichment shape function. The axial stiffness formulation (6.19) is altered to reflect (6.26):

$$\begin{aligned} k_e^{UU} &= E_e A_e \int_0^{L_e} B^{UT} B^U dx & k_e^{Ua} &= E_e A_e \int_0^{L_e} B^{UT} B^a dx \\ k_e^{aU} &= E_e A_e \int_0^{L_e} B^{aT} B^U dx & k_e^{aa} &= E_e A_e \int_0^{L_e} B^{aT} B^a dx \end{aligned} \tag{6.27}$$

$$k_e = \begin{bmatrix} k_e^{UU} & k_e^{Ua} \\ k_e^{aU} & k_e^{aa} \end{bmatrix} \tag{6.28}$$

The choice of the enrichment shape function $\phi(x)$ - as part of $N_i(x)\phi(x)a_i$, will define the vector B^{aT}, and $N_i(x)U_i$ will continue to define B^{UT}. The new vectors are to reflect that B^T has become two separate entities due to the inclusion of enrichment.

For this problem we require a shape function which is discontinuous in order to accurately portray the stress discontinuity within the bar. Whilst other shape functions can be used just as successfully, we have chosen:

$$\phi(x) = \frac{x}{x_o} \qquad 0 \le x \le x_o \tag{6.29}$$

$$\phi(x) = 1 - \frac{x - x_o}{L - x_o} \qquad x_o \le x \le L \tag{6.30}$$

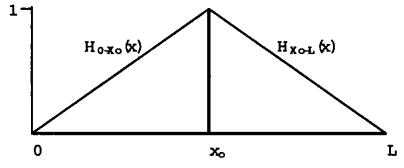


Figure 6.4: Graphical representation of $\phi(x)$

and in response

$$B^{UT} = \begin{bmatrix} \frac{dN_1}{dx} \\ \frac{dN_2}{dx} \end{bmatrix} \quad B^{aT}_{0\to x_o} = \begin{bmatrix} \frac{dN_1}{dx}\phi_{0\to x_o}(x) + N_1\frac{d\phi_{0\to x_o}(x)}{dx} \\ \frac{dN_2}{dx}\phi_{0\to x_o}(x) + N_2\frac{d\phi_{0\to x_o}(x)}{dx} \end{bmatrix} \quad 0 \le x \le x_o \quad (6.31a)$$

$$B^{UT} = \begin{bmatrix} \frac{dN_1}{dx} \\ \frac{dN_2}{dx} \end{bmatrix} \quad B^{aT}_{x_o\to L} = \begin{bmatrix} \frac{dN_1}{dx}\phi_{x_o\to L}(x) + N_1\frac{d\phi_{x_o\to L}(x)}{dx} \\ \frac{dN_2}{dx}\phi_{x_o\to L}(x) + N_2\frac{d\phi_{x_o\to L}(x)}{dx} \end{bmatrix} \quad x_o \le x \le 1 \quad (6.31b)$$

Equ (6.31b) & (6.31a) are substituted into (6.27). An expanded version of k_e^{Ua} is printed below to make the method more transparent.

$$k_e^{Ua} = E_{0\to x_o}A_{0\to x_o}\int_0^{x_o} B^{UT}B^a_{0\to x_o}dx + E_{x_o\to L}A_{x_o\to L}\int_{x_o}^{L} B^{UT}B^a_{x_o\to L}dx \quad (6.32)$$

The complete element axial stiffness matrix is assembled (with the vector of unknowns along side) and shown next to a non enriched stiffness matrix for comparison,

$$k_e^{\text{enrich}} = \begin{bmatrix} k_e^{UU} & k_e^{Ua} \\ k_e^{aU} & k_e^{aa} \end{bmatrix} \begin{bmatrix} U_i \\ U_{i+1} \\ a_i \\ a_{i+1} \end{bmatrix} \qquad k_e^{\text{non-enrich}} = \begin{bmatrix} k_e^{UU} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_i \\ U_{i+1} \\ 0 \\ 0 \end{bmatrix} \quad (6.33)$$

Four element FEM solution with enrichment (discontinuities mid-element) (iv).

Rather than integrating analytically to obtain the element axial stiffness, which would be a laborious process, we will integrate numerically using Gauss Quadrature. A brief review of the method of Gauss Quadrature and how it is applied in this case:

$$\int_I^b f(x)dx = \alpha_1 f(r_1) + \alpha_2 f(r_2) + ... + \alpha_n f(r_n) \quad (6.34)$$

where for 1D

$$\alpha_i = w_i\frac{(b-a)}{2} \qquad w_i = \text{weighting at Gauss Point}$$

$$r_i = \text{global position of Gauss Point}$$

For the case of enrichment (6.34) can be rewritten

$$\begin{aligned} \int_0^L f(x)dx &= \int_0^{x_o} f_{0\to x_o}(x)dx + \int_{x_o}^{L} f_{x_o\to L}(x)dx \\ &= \beta_1 f_{0\to x_o}(r_1) + \beta_2 f_{0\to x_o}(r_2) + ... + \beta_n f_{0\to x_o}(r_n) \\ &+ \gamma_1 f_{x_o\to L}(r_1) + \gamma_2 f_{x_o\to L}(r_2) + ... + \gamma_n f_{x_o\to L}(r_n) \end{aligned} \quad (6.35)$$

The minimum number of integration points per discontinuous element is two, as there must be at least one Gauss point before and after the discontinuity for the numerical integration to be accurate. Using a total of four Gauss points per discontinuous element, i.e. 2 Gauss points on each side of the discontinuity, the weights and positions, for a element of length 500mm, are as follows [4]:

$$0 \rightarrow x_o \qquad x_o \rightarrow L \tag{6.36}$$

$$w_1 = 1 \qquad x_1 = 52.831 \qquad x_1 = 302.831 \tag{6.37}$$

$$w_2 = 1 \qquad x_2 = 197.169 \qquad x_2 = 447.169 \tag{6.38}$$

assuming for a {-1, +1} isoparametric element that $r_1 = \frac{-1}{\sqrt{3}}$ & $r_2 = \frac{+1}{\sqrt{3}}$. These values, and those for integrating with more or less Gauss Points can be found in most standard FE texts in tabular form.

The complete assembled axial stiffness matrix and loading vector

$$\begin{bmatrix} 549780 & -549780 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -549780 & 914045 & 0 & 89375 & -364265 & 0 & 89375 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 89375 & 0 & 364265 & -89375 & 0 & 364265 & 0 & 0 & 0 \\ 0 & -364265 & 0 & -89375 & 691085 & -326820 & -216195 & -126820 & 0 & 0 \\ 0 & 0 & 0 & 0 & -326820 & 726820 & 126820 & 126820 & -400000 & 0 \\ 0 & 89375 & 0 & 364265 & -216195 & 126820 & 691085 & 326820 & 0 & 0 \\ 0 & 0 & 0 & 0 & -126820 & 126820 & 326820 & 326820 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -400000 & 0 & 0 & 400000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} U_A \\ U_B \\ a_A \\ a_B \\ U_D \\ U_F \\ a_D \\ a_F \\ U_G \\ a_G \end{bmatrix} = \begin{bmatrix} R_A \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 78540 \\ 0 \end{bmatrix} \tag{6.39}$$

System (6.39) is solved by removing both a_A & a_G. The solution

$$U_A = 0.000mm \qquad a_A = 0.000mm$$

$$U_B = 0.143mm \qquad a_B = 0.060mm$$

$$U_D = 0.372mm \qquad a_D = -0.003mm$$

$$U_F = 0.655mm \qquad a_F = -0.106mm$$

$$U_G = 0.852mm \qquad a_G = 0.000mm$$

We use the Finite Element interpolation (6.26) to calculate the values for displacement mid element, i.e. U_C & U_E. Figure 6.5 can be used as a reference.

[4] It should be noted that due to the discontinuity the numerical integration has been split into 2 separate parts (see (6.35)), i.e. integrating over two subelements. We treat the bar as if it is 500m in length, however we integrate over two 250mm lengths

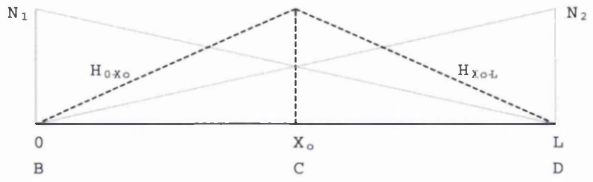


Figure 6.5: Standard (N) and Enrichment (ϕ) Shape functions

At node C using (6.26) for a mid element discontinuity[5] $x_o = \frac{L}{2}$

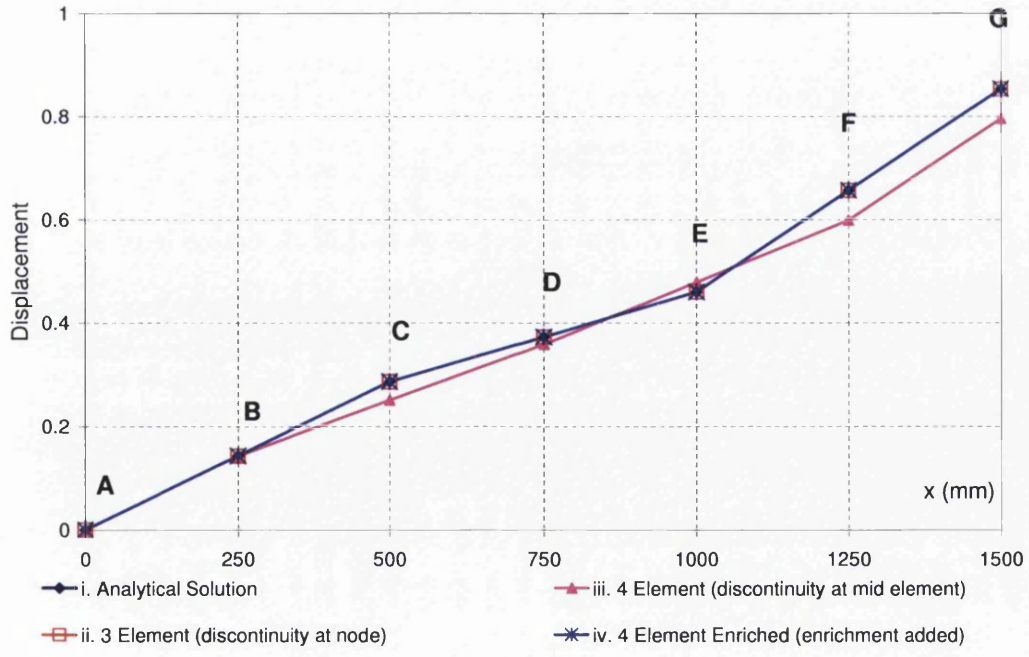
$$U_C = \sum_{i=1}^{2} N_i(x)u_i + \sum_{i=1}^{2} N_i(x)\phi(x)a_i$$
$$= N_1(x_o)U_B + N_2(x_o)U_D + N_1(x_o)\phi(x_o)a_B + N_2(x_o)\phi(x_o)a_D$$
$$= 0.5 \times 0.143mm + 0.5 \times 0.372mm + 0.5 \times 1 \times 0.060mm + 0.5 \times 1 \times -0.003mm$$

$$U_C = 0.286mm$$

Following the same process for U_D

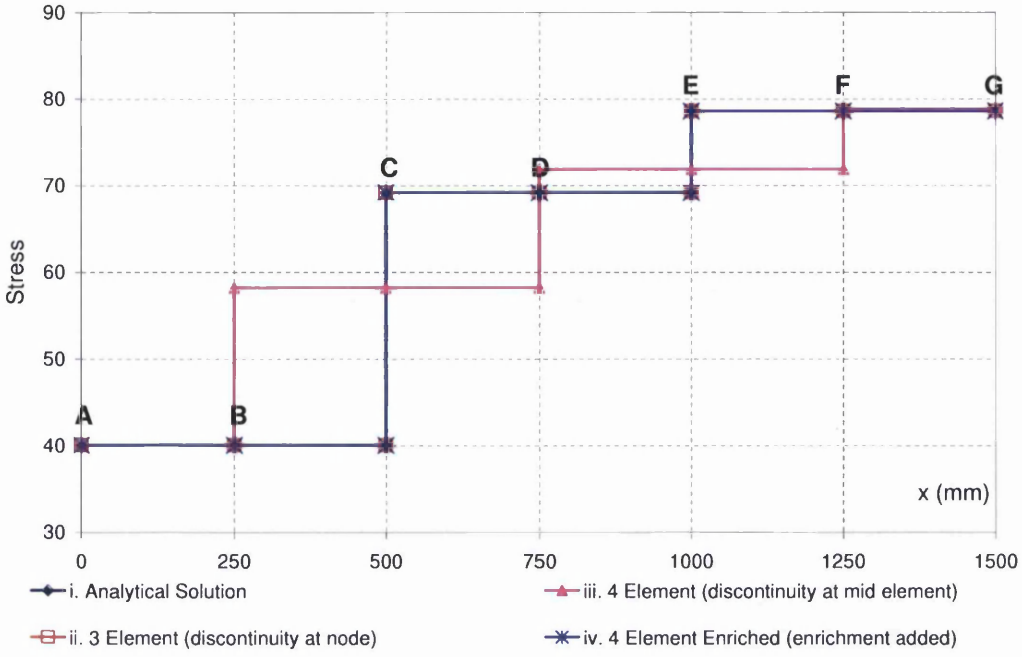
$$U_D = 0.459mm$$

6.4.3 Comparison of Solutions.



(a) Plot of nodal displacements for **i, ii, iii, iv**

[5]This calculation could be performed for any value of x_o and produce the same result.



(b) Plot of stresses for **i, ii, iii, iv**

Figure 6.6: The displacement and stress results for the 1D bar example

Figure 6.6(b) clearly demonstrates that the 1D bar problem is a discontinuous problem. The bar is made up of three different materials and this leads to discontinuities which are apparent in the stress distribution of Figure 6.6(b). The observations for this example are given below:

- As expected the only non exact solution is the General Finite Element formulation with discontinuities at mid element, i.e. solution **iii.**. The General Finite Element solution **iii.** was unable to take account of the discontinuities i.e. inaccuracies regarding the Young's modulus and cross sectional area for the two discontinuous elements, and this lead to errors in the final displacements.

- The solution **iv.** highlights one of the benefits of enrichment, i.e. enrichment produces the correct global nodal values, regardless whether the values of enrichment data are accounted for or not in the final result. If accurate interior displacements are required, at any point within the mesh, it is necessary to account for the enrichment result in the displacement calculations.

6.4.4 Disadvantages of Enrichment.

Whilst the enrichment method has been shown to produce accurate solutions, albeit only in 1D, there are a few notable disadvantages which became apparent at this early stage:

- the enriched FEM requires more accurate integration
- interpolating results to find values at mid element (or at any other point along the element) is awkward.
- increases the number of degrees of freedom.

The first disadvantage is unavoidable, and is a direct result of the need for a more accurate solution. The second is normally not important as it is a logical process and can be performed automatically. The final disadvantage is more serious. Doubling the number of degrees of freedom for a small problem is relatively simple, however if the number of degrees of freedom is already large, then doubling this value will considerably increase the solution time. It may even be considered more economical to increase the mesh density rather than implement enrichment. However there are ways to partially circumvent this problem, and these are explained below:

1. *only include the enrichment capability for elements which require enrichment.*

   This method is the most appropriate as it limits the number of degrees of freedom, however it is very awkward to implement and leads to difficulties compiling the global stiffness matrix if the discontinuity is moving.

2. *Static Condensation*

   Static condensation is employed to reduce the number of element degrees of freedom and thus, in effect, to partially solve the local system of equations, i.e. stiffness matrix $\boldsymbol{K}$ *and the residual vector* $\boldsymbol{R}$, prior to assembly of the global system of equations.

   It is possible to partly solve the system of equations at element level by statically condensing the larger enrichment matrices to the size of an unenriched matrix. The stiffness matrix shown below was calculated using the same mesh as for solution **iv.**, however this was statically condensed at element level and assembled.

$$\begin{bmatrix} 549780 & -549780 & 0 & 0 & 0 \\ -549780 & 892116 & -342336 & 0 & 0 \\ 0 & -342336 & 619944 & -277608 & 0 \\ 0 & 0 & -277608 & 677608 & -400000 \\ 0 & 0 & 0 & -400000 & 400000 \end{bmatrix} \begin{bmatrix} U_A \\ U_B \\ U_D \\ U_F \\ U_G \end{bmatrix} = \begin{bmatrix} R_A \\ 0 \\ 0 \\ 0 \\ 78540 \end{bmatrix} \tag{6.40}$$

This solves to give

$$U_A = 0.000mm$$
$$U_B = 0.143mm$$
$$U_D = 0.372mm$$
$$U_F = 0.655mm$$
$$U_G = 0.852mm$$

Which is exactly the same displacement result as obtained from solution **iv.**.

- generally only few elements require enrichment, therefore the advantage gained far outweighs the extra computational expense of solving at element level. If the percentage of enriched nodes increases then this method becomes inefficient.
- there would be the same number of degrees of freedom for every time step.
- finding the enrichment values would require extra computation post-solution

Both methods have there advantages and disadvantages, however Option 1 minimizes the number of degrees of freedom and does not require extra solution algorithm at element level, therefore we will use this method for modelling discontinuous problems.

6.4.5 Conclusion - 1D Enrichment Example.

In Section 6.3 the steps for a successful outcome when modelling problems that contained interfaces were defined. It was noted that the solution was reliant upon

proper implementation of the Finite Element enrichment discretization, on the solution algorithms ability to correctly locate the position of the discontinuity and accurately model the discontinuity.

In the case of the simple enrichment example above, i.e. displacement in a one dimensional discontinuous bar - Section 6.4, the criteria were relatively simple to achieve:

- The Finite Element enrichment discretization was performed correctly and an appropriate enrichment function was used.
- The position of the discontinuities were known, stationary and the discontinuity only cut each element once.
- The discontinuous elements were modelled accurately because the Gauss quadrature method was able integrate each element exactly.

Therefore a successful solution was achieved because the X-FEM criteria were fulfilled correctly.

6.5 X-FEM Algorithms.

In the case of a two dimensional problem the enrichment process is considerably more involved than in 1D problems, as simple methods used in 1D problems become significantly more complex. In this section the methods used to ensure that the above criteria are fulfilled are discussed in detail. In addition to this certain algorithms specific to programming and their implementation into 1DFLOW and FEAP will be explained.

The section will be divided into two subsections, firstly Interface Location followed by Interface Modelling. Interface Location will encompass all necessary methods to accurately position the interface at every point on the mesh, whilst Interface Modelling entails accurately integrating the enriched elements. These methods will be developed into a general all purpose algorithms.

In order to demonstrate the algorithms numerical examples are used as case studies. The numerical examples chosen are 1D Advection Diffusion for 1D i.e. propagating rectangular wave using Heaviside enrichment, and Particulate flow simulations, i.e. fluid flow past a stationary particle, with Stokes flow enrichment.

6.6 Interface Location.

The process of locating an interface can be divided in to two sub-processes, i) locating the movement every timestep, and ii) selecting the elements that require enriching and blending, and they are described in detail below.

6.6.1 Motion of the Interface.

Calculating the motion of the interface is dependent on the type of problem being considered. In certain scenarios the movement of the interface cannot be predetermined, as each movement is dependent on global equilibrium[6], such as in the study of crack propagation or multi-particle flow. In other cases the position can be predetermined, i.e. problems with stationary interfaces or those which follow a fixed path.

Both enrichment numerical examples are of the second category, i.e. the position of the interface is predetermined[7], therefore the method of calculating the exact position at every timestep is straightforward. If a point on the interface is denoted $\bar{\mathbf{x}}$ and the velocity is given by $a_{\bar{\mathbf{x}}}$ then the new position of $\bar{\mathbf{x}}$ $U_{\bar{\mathbf{x}}}$ is given as:

$$U_{\bar{\mathbf{x}}} = a_{\bar{\mathbf{x}}}\Delta t + \bar{\mathbf{x}} \tag{6.41}$$

For the propagating rectangular wave the global position of the actual interfaces are defined, whereas for particulate flow the interface is defined by the centre of the particle.

6.6.2 Search Algorithm.

It is important to know the location of the interface at all times, and know which elements require full enrichment i.e. Ω^{enr}, partial enrichment i.e. Ω^{bld} and those which are not enriched i.e. Ω^{std}.

For a stationary interface, such as a particle in Figure 6.7, it is only necessary to obtain this information once, as its position will not vary during the simulation and the background mesh is fixed. However in the case of a moving interface this information needs to be collated every time step, whilst the mesh remains fixed

[6]Which alters with every movement of the interface

[7]It is assumed that both examples model the movement of incompressible solids therefore the position of the interface in a local framework is unchanging, however globally the position can vary

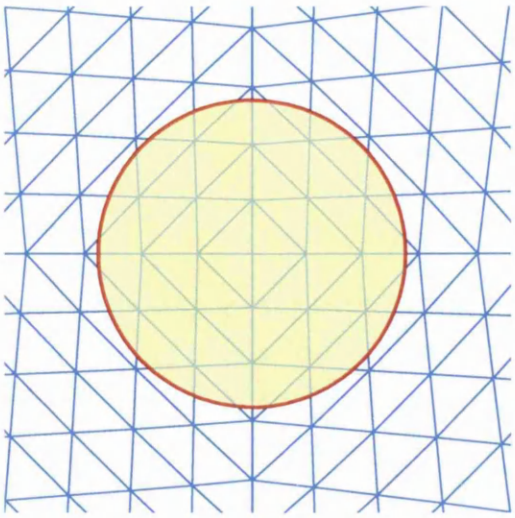


Figure 6.7: Particle superimposed on to a mesh

the interface may move freely, and the information from a previous time step may no longer be applicable. Collating the positional information every time is a laborious process and could conceivably be the most time consuming routine in the entire solution, so it is imperative to make the search as efficient as possible.

The method chosen to search for the elemental data uses a system of domains to break down the mesh into smaller regions. It is no longer necessary to search the entire mesh to find specific elemental data, as the same information can be found by searching over a small percentage of the total number of elements. The main advantage of this method is a time advantage, which is achieved when there are many time steps used, as initial costs are high during the period when a library of all elemental information is constructed. The main disadvantage is the need to store a large amount of data for the duration of the computation, however this is unavoidable if quick search methods are required.

Search by Domains.

This search algorithm divides the mesh into regions of interest, i.e. *domains*. Each *domain* is made up of smaller *domains* in a pyramid type structure. In order to find an element at a certain coordinate the search algorithm progressively searches a smaller and smaller area, focusing the search with every step until only a small portion of the entire mesh needs to be searched using a standard search routine.

Description of Method

The entire meshed region is classified as *Level 1.* The *Level 1* domain is divided into 4 equally sized *Level 2* domains, and these in turn are all divided into 16 *Level 3* domains. This process will continue until there are small number of elements e.g. $\approx$100, in each of the domains (*Level n*). In order to improve the search each *domain* at all levels contains the address of its *parent*, its *neighbours* and also its 4 *children.* See Figure 6.8 & Figure 6.9.

Level	No Domains
1	$2^{0\times 2}$
2	$2^{1\times 2}$
3	$2^{2\times 2}$
..	.....
n	$2^{(n-1)\times 2}$

(6.42)

In the diagram, see Figure 6.8 the domain hierarchy is explained in pictorial format using a flowchart[8].

An example of the division of a simple 2D unstructured mesh using domains is presented in Figure 6.9.

[8]In truncated form due to it size

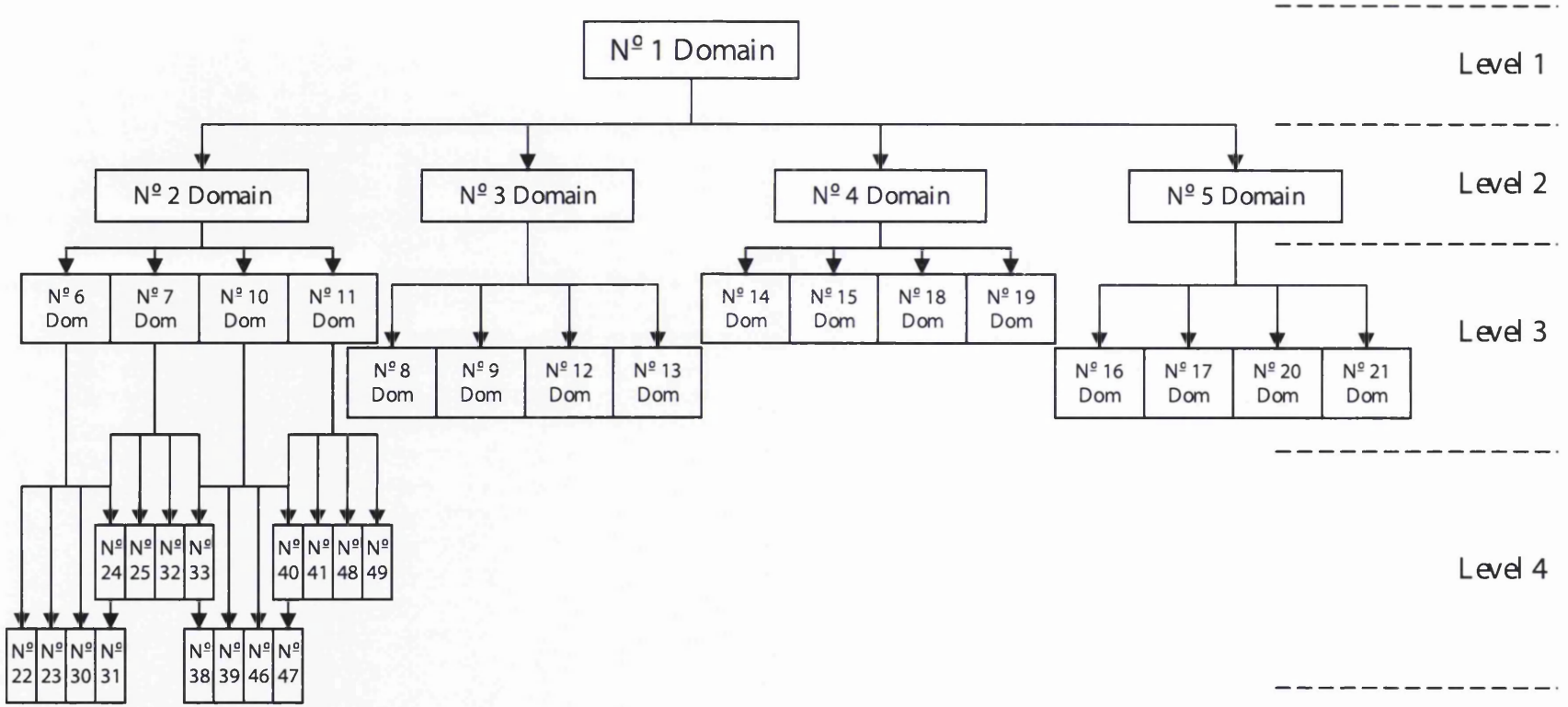
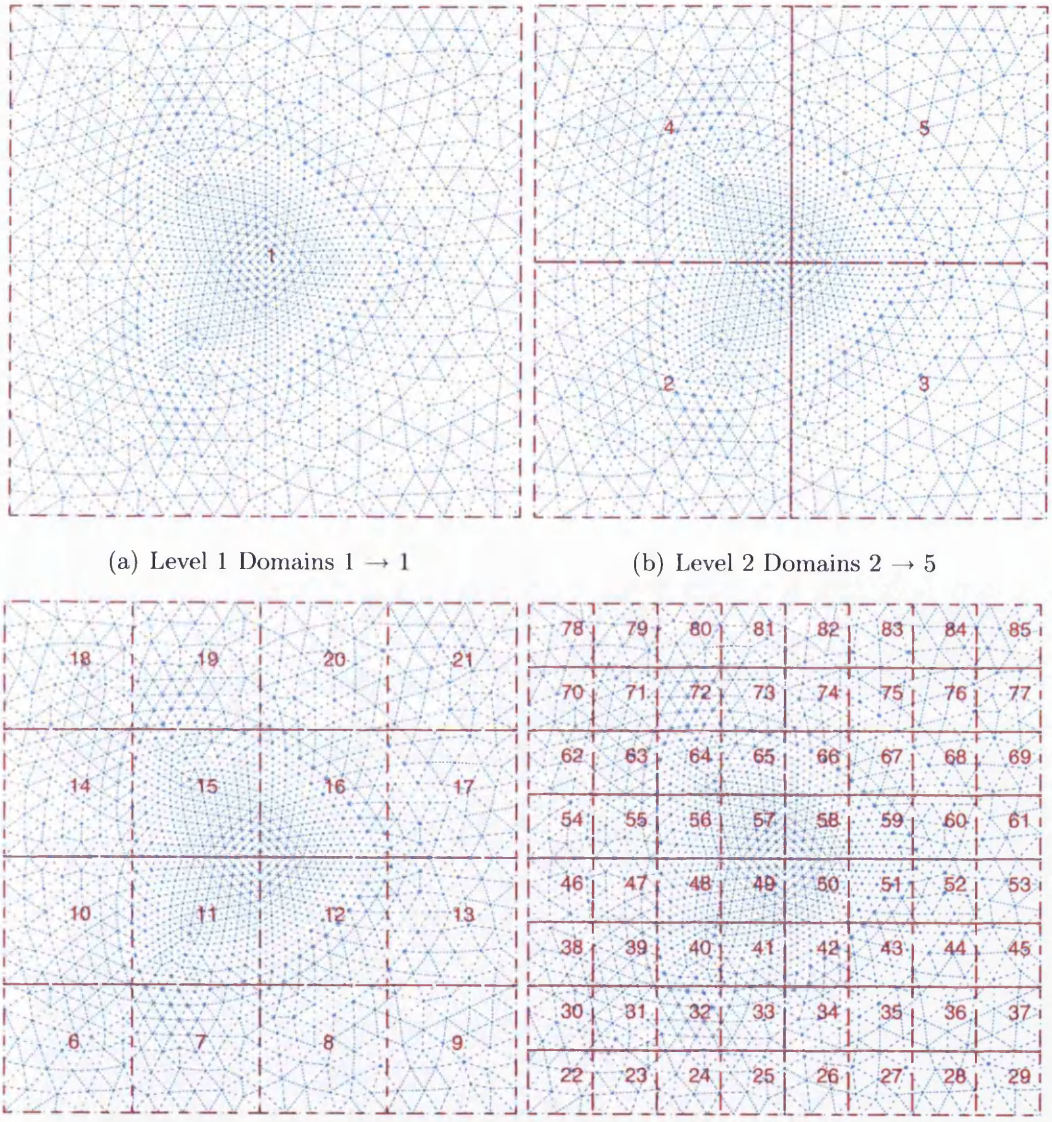


Figure 6.8: Flow Chart of Domain Hierarchy



(a) Level 1 Domains 1 → 1

(b) Level 2 Domains 2 → 5

(c) Level 3 Domains 6 → 21

(d) Level 4 Domains 22 → 85

Figure 6.9: Actual division of mesh by domains

Example - Search Algorithm.
The Domain search method is compared to that of a standard sequential search. Efficiency is quantified by comparing the number of steps required for each search to finish successfully.

NEAREST NODE SEARCH

Sequential search	**Domain search**
25600 nodes	5 levels
	level 5 - total 256 domains
	$\approx$ 100 nodes in each level 5 domain
	Maximum 16 steps per level to determine
	in which domain the coordinates lie
	$16 \times 5 + 100$nodes
25600 steps	**180** steps

It is shown by this simple example that the time saving is large after the initial process of building a library of elemental data is completed.

6.7 Interface Modelling.

6.7.1 Local Mesh refinement and Numerical Integration.

The task of meshing the numerical domain is performed in the pre-processing stage, however it is sometimes necessary to carry out low level meshing to implement a more accurate numerical integration procedure, during the solution process. The main advantages of this method is the ability to model the interface accurately and irrespective of the mesh size/interface length ratio. This results in accurate solutions from relatively coarse meshes.

This section will be used to explain the implementation of mesh refinement and corresponding numerical integration for selected elements. The 2D particulate flow example using Stokes flow enrichment will be used to demonstrate this method.

In this chapter the General Finite Element discretization, see (6.5), was extended to accommodate enrichment, see (6.11), to become X-FEM; the numerical integration procedure used successfully for the General Finite Element formulation is not appropriate when applied to X-FEM. The current numerical integra-

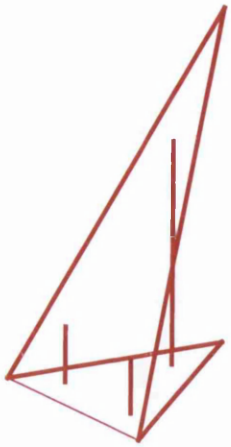
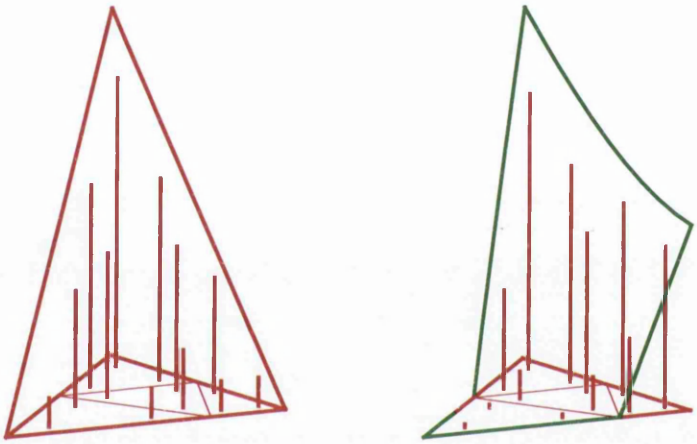


Figure 6.10: Standard 2D linear shape function - 3 Internal Gauss points



(a) Standard 2D linear shape function

(b) Enrichment function

Figure 6.11: Shape functions for Enrichment - 4 Subelements, 3 Internal Gauss points per subelement

tion scheme, i.e. Gauss quadrature[9] with 3 internal Gauss points for 2D triangular elements, see Figure 6.10 is only suitable for integrating linear shape functions. With the added enrichment the numerical integration scheme would be required to model the interface and integrate nonlinear functions over the element, as the product $N_I(\boldsymbol{x})\phi(\boldsymbol{x})$ is nonlinear.

The commonly used method, and the one implemented here, is to subdivide the element into subelements which do not cross the interface and perform nu-

[9]Gauss quadrature is a commonly used method for numerical integration, and is widely written about, see Bathe [6].

merical integration over these subelements, see Moes *et al* [60], and Belytschko *et al* [10]. The purpose of the local mesh refinement is to improve the accuracy of numerical integration for enriched elements. The edges of the subelements define the path of the interface and the increased number of integration points interpolate the nonlinear shape function more accurately, see Figure 6.11.

In the next section the complete process of mesh refinement and numerical integration will be explained.

6.7.2 Mesh Refinement.

In order to perform mesh refinement we must first specify which elements require it i.e. elements that are intersected by the interface, see Figure 6.12:

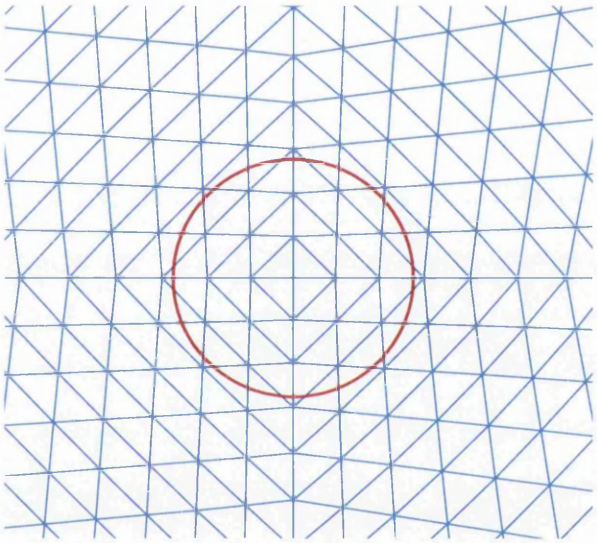


Figure 6.12: Elements intersected by particle

In the case of Particulate fluid flow, the interface is assumed to be particle, which to simplify the analysis is assumed to be circular and is defined by the equation of a circle:

$$R^2 = (x - x_0)^2 + (y - y_0)^2 \tag{6.43}$$

where x & y are the cartesian coordinates in 2D space, x_0 & y_0 are the coordinates of the centre of the particle and R is the radius of the particle. If the centre of the particle is at the origin then:

$$\left.\begin{array}{l} R^2 - x^2 - y^2 > 0 \\ R^2 - x^2 - y^2 < 0 \end{array}\right\} \begin{array}{l} \text{Exterior of the particle} \\ \text{Interior of the particle} \end{array} \tag{6.44}$$

The *Domains* search algorithm is used to find the nodes closest to the particle surface (interior/exterior), and this information is used to identify the elements intersected, see Figure 6.13

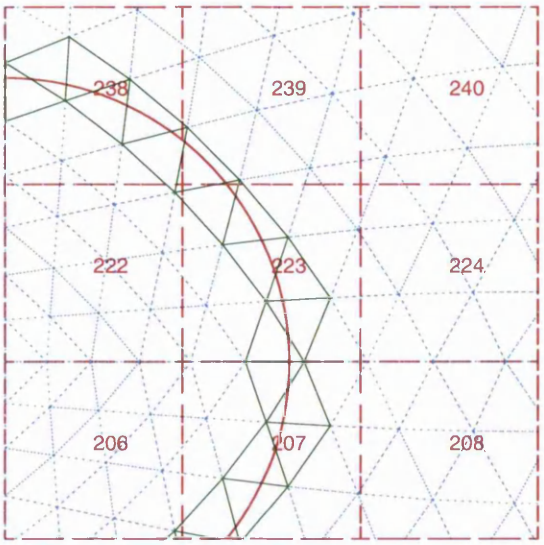


Figure 6.13: Elements intersected by particle

Each element is considered individually. The point of intersection of the surface of the particle with the two edges of the element, see Figure 6.14, is calculated[10].

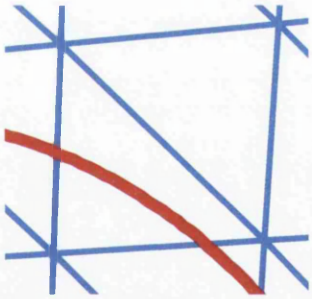


Figure 6.14: Intersected element by an interface

With this information the process of locating the interface is complete. The next solution process is the division of the intersected elements into smaller subelements and the description of this methods follows below.

[10]Equating the equation of the line of the element edge and equation describing the interface leaves a quadratic which can be solved simply

6.7.2.1 Subelements.

The division of an element into subelements is performed in order to split the element into smaller integration areas, thereby improving accuracy. It also allows for the interface to be modelled globally, over the domain, and locally, within an element.

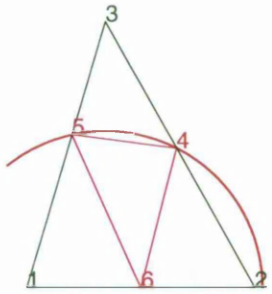


Figure 6.15: Element divided into 4 subelements

To subdivide the element into smaller subelements and match the subelement with the interface requires various steps. Initially a isoparametric element in ($\xi - \eta$ space is subdivided into subelements and the result is mapped onto each individual element in $x - y$ space. In the next stage the interface is defined within the element by molding the nodes of the subelements to match the interface. Finally the integration points are mapped from $\xi - \eta$ space into each subelement individually in $x - y$ space, then transformed back into $\xi - \eta$ space one more time to calculate their weighting for Gauss quadrature.

6.7.2.2 Isoparametric Element and Mapping.

The background to isoparametric elements and mapping was shown in Section 3.2.2.2, however to fulfill the requirements of enrichment and improve numerical integration we will be performing mappings for subelements and it is necessary to introduce the naming convention for element and subelements nodes:

$\xi - \eta$ element coordinates	ξ_1^e, η_1^e, ξ_2^e, η_2^e, ξ_3^e, η_3^e
$\xi - \eta$ subelement coordinates	ξ_1^{se}, η_1^{se}, ξ_2^{se}, η_2^{se}, ξ_3^{se}, η_3^{se}
$x - y$ element coordinates	x_1^e, y_1^e, x_2^e, y_2^e, x_3^e, y_3^e
$x - y$ subelement coordinates	x_1^{se}, y_1^{se}, x_2^{se}, y_2^{se}, x_3^{se}, y_3^{se}

Therefore the transformation from $\xi - \eta \rightarrow x - y$ for a subelement node, with given coordinates in $\xi - \eta$ space is written:

$$\begin{bmatrix} x_1^e & x_2^e & x_3^e \\ y_1^e & y_2^e & y_3^e \end{bmatrix} \begin{bmatrix} N_1^{se} \\ N_2^{se} \\ N_3^{se} \end{bmatrix} = \begin{bmatrix} x_i^{se} \\ y_i^{se} \end{bmatrix} \tag{6.45}$$

where

$$N_1^{se} = 1 - \xi_i^{se} - \eta_i^{se} \quad N_2^{se} = \xi_i^{se} \quad N_3^{se} = \eta_i^{se} \tag{6.46}$$

If we take a isoparametric element in $\xi - \eta$ space, Figure 6.16

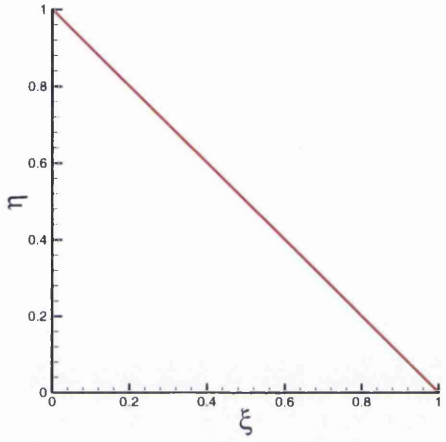
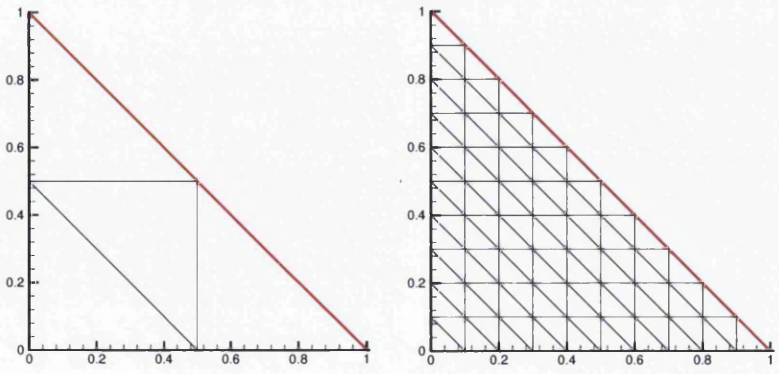


Figure 6.16: Isoparametric element in $\xi - \eta$ space

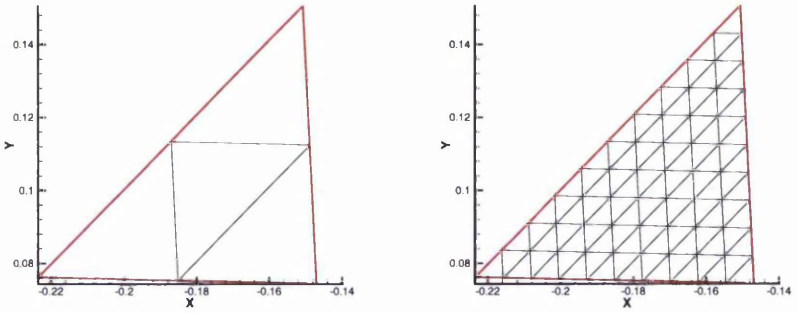
This element can be subdivided into any number of subelements. To demonstrate the process a dense mesh (100 subelements) and a coarse mesh (4 subelements) are shown below:



(a) 4 subelements, 6 subnodes (b) 100 subelements, 66 subnodes

Figure 6.17: Subdivision of Isoparametric element

and thus the elements shown in Figure 6.17 $\xi - \eta \rightarrow x - y$ become:



(a) 4 subelements, 6 subnodes (b) 100 subelements, 66 subnodes

Figure 6.18: Transformed $\xi - \eta \rightarrow x - y$

In the next stage of the process the shape of the interface is defined by repositioning subnodes so that the edges of the subelements delineate the interface. This process is carried out prior the calculation of the position of the integration points because certain subelements will change and therefore alter the internal layout of the subelements.

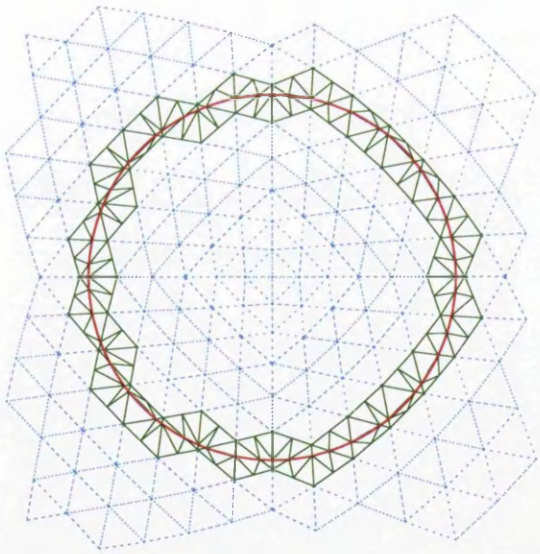
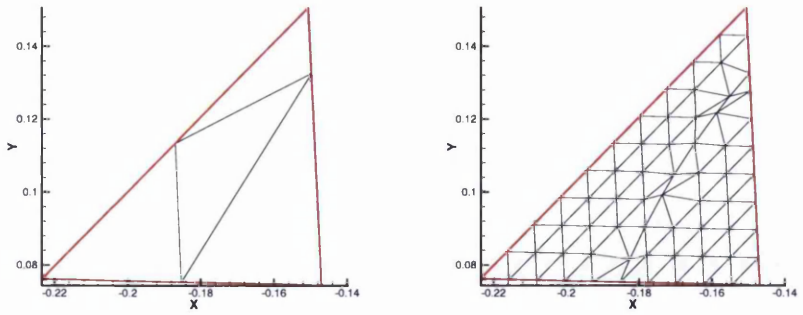


Figure 6.19: Intersected elements are subdivided for integration

The repositioning of the subnodes to define the interface is a simple process. A search is performed to ascertain which subnodes lie closest to the interface, and these are repositioned to lie on the interface. This method is best understood when it is displayed and below Figure 6.18 is redrawn to account for the interface:



(a) 4 subelements, 6 subnodes (b) 100 subelements, 66 subnodes

Figure 6.20: Repositioned subnodes to account for interface

The result of this process for the whole interface is shown graphically in Figure 6.21 and shows the elements intersected by the particle surface subdivided into 4 and 100 subelements, respectively.

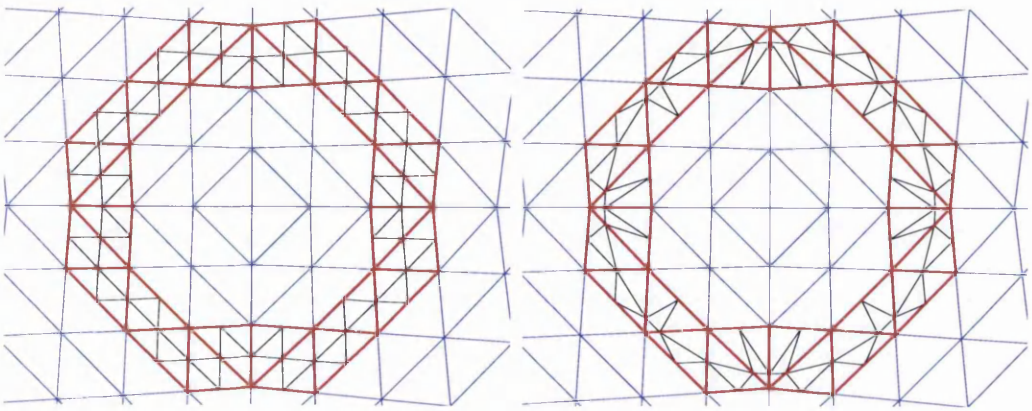
The mapping of the subelements to $x-y$ space generates the coordinates of every subnode which will be used in the next process, i.e. calculating the position of the element integration points.

6.7.2.3 Element Integration Points.

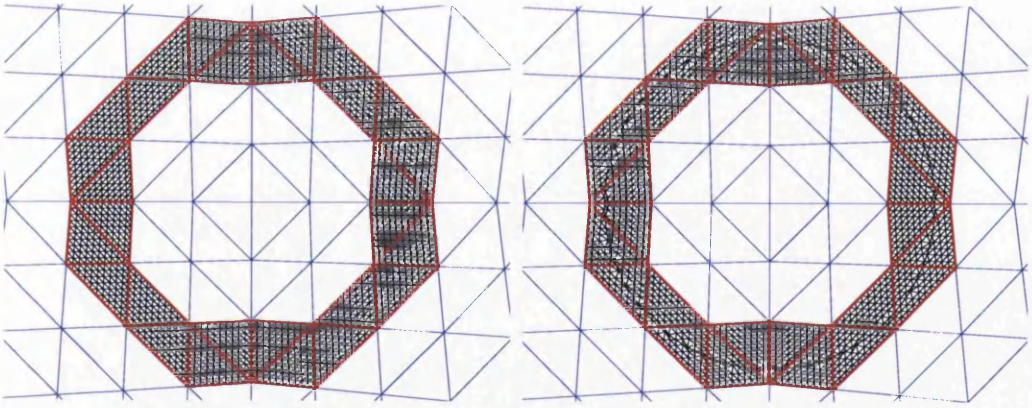
Each subelement will be initially treated as separate entity to calculate the position of the integration points for numerical integration. For the purposes of numerical integration Gauss quadrature with three internal Gauss points will continue to be used. The integration points in $\xi-\eta$ space for the isoparametric element defined in Figure 6.16 are given as:

Integration Point	ξ_{gp}	η_{gp}
1	$0.6666\dot{6}$	$0.1666\dot{6}$
2	$0.1666\dot{6}$	$0.6666\dot{6}$
3	$0.1666\dot{6}$	$0.1666\dot{6}$

Positioning each subelement Gauss point within the element in $x-y$ space and calculating their weighting is complicated and requires two transformations:



(a) (b)



(c) (d)

Figure 6.21: (a)(c) Elements crossed by the interface are subdivided in preparation; (b)(d) The subnodes are repositioned to describe the surface of the interface; (a)(b) 4 subelements per element; (c)(d) 100 subelements per element

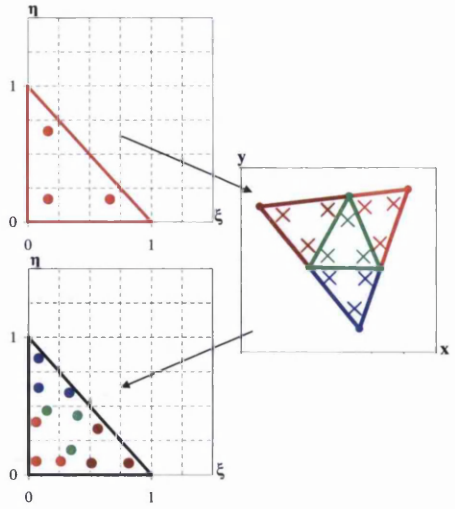


Figure 6.22: Mapping from $\xi - \eta$ space to $x - y$ and back again

1. $\xi - \eta \rightarrow x - y$ isoparametric Gauss points $\rightarrow$ subelement Gauss points

$$\begin{bmatrix} x_1^{se} & x_2^{se} & x_3^{se} \\ y_1^{se} & y_2^{se} & y_3^{se} \end{bmatrix} \begin{bmatrix} N_1^{gp} \\ N_2^{gp} \\ N_3^{gp} \end{bmatrix} = \begin{bmatrix} x_i^{gp} \\ y_i^{gp} \end{bmatrix} \tag{6.47}$$

where

$$N_1^{gp} = 1 - \xi_i^{gp} - \eta_i^{gp} \quad N_2^{gp} = \xi_i^{gp} \quad N_3^{gp} = \eta_i^{gp} \tag{6.48}$$

2. $x - y \rightarrow \xi - \eta$ all subelement Gauss points in the element $\rightarrow$ element isoparametric Gauss points

$$\begin{bmatrix} x_1^{e} & x_2^{e} & x_3^{e} \\ y_1^{e} & y_2^{e} & y_3^{e} \end{bmatrix}^{-1} \begin{bmatrix} x_i^{gp} \\ y_i^{gp} \end{bmatrix} = \begin{bmatrix} N_1^{gp} \\ N_2^{gp} \\ N_3^{gp} \end{bmatrix} \tag{6.49}$$

where

$$N_1^{gp} = 1 - \xi^{gp} - \eta^{gp} \quad N_2^{gp} = \xi^{gp} \quad N_3^{gp} = \eta^{gp} \tag{6.50}$$

The main disadvantage of enrichment is the need to accurately position and model the interface, as the process is computationally expensive. Therefore mesh refinement is best restricted to a minimum number of elements, i.e. where the improvement in solution outweigh the costs involved.

In the following section the choice of enrichment is explained.

6.8 Enrichment Functions.

An enrichment function can take the form of either a scalar or vector field and is dependent on the problem and the interface modelled. A scalar field increases the number of unknowns the process is more general and is easily augmented with further enrichment should such needs arise, whilst vector field enrichment ties component enrichment fields together, thereby reducing the overall number of unknowns.

Scalar function enrichment has already been demonstrated in the 1D axially loaded bar example and in the case of the two numerical examples, scalar enrichment is also utilized. Enrichment using a scalar field is applied to the propagating rectangular wave, and Particulate flow.

The enriched Finite Element approximation is rewritten:

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{I\in\mathcal{N}} N_I(\boldsymbol{x})\boldsymbol{u}_I + \sum_{\alpha=1}^{n_E}\sum_{I\in\mathcal{N}^{enr}} N_I(\boldsymbol{x})\phi_\alpha(\boldsymbol{x})\boldsymbol{a}_{\alpha I} \tag{6.51}$$

$$= \sum_{I\in\mathcal{N}} N_I(\boldsymbol{x})\boldsymbol{u}_I + \sum_{I\in\mathcal{N}^{enr}} N_I(\boldsymbol{x})\phi_1(\boldsymbol{x})\boldsymbol{a}_{1I} + \ldots + \sum_{I\in\mathcal{N}^{enr}} N_I(\boldsymbol{x})\phi_n(\boldsymbol{x})\boldsymbol{a}_{nI} \tag{6.52}$$

The manner in which this is applied to FEM code is briefly shown below:

$$\underbrace{\begin{bmatrix} \boldsymbol{k}_{uu} & \boldsymbol{k}_{ua_1} & \ldots & \boldsymbol{k}_{ua_n} \\ \boldsymbol{k}_{a_1u} & \boldsymbol{k}_{a_1a_1} & \ldots & \boldsymbol{k}_{a_1a_n} \\ \boldsymbol{k}_{a_nu} & \boldsymbol{k}_{a_na_1} & \ldots & \boldsymbol{k}_{a_na_n} \end{bmatrix}}_{\boldsymbol{K}^e} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_n \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{r}_u \\ \boldsymbol{r}_{a_1} \\ \vdots \\ \boldsymbol{r}_{a_n} \end{bmatrix}}_{\boldsymbol{R}^e} \tag{6.53}$$

where the $\boldsymbol{k}$ & $\boldsymbol{r}$ are a 9×9 matrix and 9×1 vector respectively for the 2D problem, and represent constructions using the following Finite Element discretization:

$$\boldsymbol{k}_{uu} \longrightarrow \quad \boldsymbol{w}^h = \sum_{I\in\mathcal{N}} N_I(\boldsymbol{x})\boldsymbol{w} \qquad \boldsymbol{u}^h = \sum_{I\in\mathcal{N}} N_I(\boldsymbol{x})\boldsymbol{u}_I \tag{6.54}$$

$$\boldsymbol{k}_{ua_1} \longrightarrow \quad \boldsymbol{w}^h = \sum_{I\in\mathcal{N}} N_I(\boldsymbol{x})\boldsymbol{w} \qquad \boldsymbol{u}^h = \sum_{I\in\mathcal{N}^{enr}} N_I(\boldsymbol{x})\phi_1(x)\boldsymbol{a}_{1I} \tag{6.55}$$

$$\boldsymbol{k}_{a_1u} \longrightarrow \quad \boldsymbol{w}^h = \sum_{I\in\mathcal{N}^{enr}} N_I(\boldsymbol{x})\phi_1(x)\boldsymbol{w} \qquad \boldsymbol{u}^h = \sum_{I\in\mathcal{N}} N_I(\boldsymbol{x})\boldsymbol{u}_I \tag{6.56}$$

$$\boldsymbol{k}_{a_1a_1} \longrightarrow \quad \boldsymbol{w}^h = \sum_{I\in\mathcal{N}^{enr}} N_I(\boldsymbol{x})\phi_1(x)\boldsymbol{w} \qquad \boldsymbol{u}^h = \sum_{I\in\mathcal{N}^{enr}} N_I(\boldsymbol{x})\phi_1(x)\boldsymbol{a}_{1I} \tag{6.57}$$

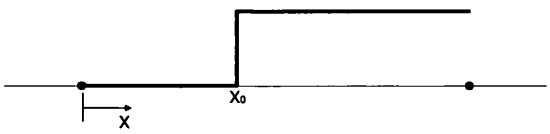
- k_{uu} - standard (velocity/pressure) weighting function and standard (velocity/pressure) solution variable discretizations.
- k_{ua_1} - standard (velocity/pressure) weighting function and enrichment (velocity/pressure) solution variable discretizations.
- k_{a_1u} - enrichment (velocity/pressure) weighting function and standard (velocity/pressure) solution variable discretizations.
- $k_{a_1a_1}$ - enrichment (velocity/pressure) weighting function and enrichment (velocity/pressure) solution variable discretizations.

6.8.1 1D Enrichment Functions.

For propagation of the 1D rectangular wave two enrichment functions were considered:

1. Heaviside

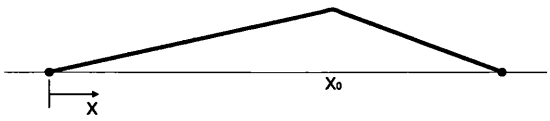
$$\phi_1(x) = H(x) \tag{6.58}$$



where $H(x)$ is the step function

$$H(x) = \begin{cases} 0 & \text{for} \quad x < x_0 \\ +1 & \text{for} \quad x > x_0 \end{cases} \tag{6.59}$$

2. Test function



where

$$\phi(x) = \begin{cases} \frac{x}{x_0} & \text{for} \quad x < x_0 \\ 1 - \frac{x - x_0}{L - x_0} & \text{for} \quad x > x_0 \end{cases} \tag{6.60}$$

6.8.2 2D Enrichment Functions.

The enrichment functions are given by the vector and scalar fields derived for Stokes flow, see 2.5.3.1. As mentioned previously these functions are only applicable to regions near the particle surface. As the enrichment is only applied to element intersected by the particle surface we can assume this constraint is satisfied.

The enrichment functions, split in Velocity Enrichment and Pressure Enrichment, were derived in Chapter 2 and are rewritten here for convenience.

Velocity Enrichment.

If the point under consideration is internal to the particle then:

$$\phi_1(\boldsymbol{x}) = 0 \qquad \phi_2(\boldsymbol{x}) = 0 \tag{6.61}$$

$$\phi_{1,x}(\boldsymbol{x}) = \phi_{1,y} = 0 \qquad \phi_{2,x}(\boldsymbol{x}) = \phi_{2,y} = 0 \tag{6.62}$$

otherwise

$$\phi_{1u} = \frac{(R^2 - r^2)\cos^2\theta + r^2 \ln\left[\frac{r}{R}\right] + \frac{1}{2}(r^2 - R^2)}{r^2} \tag{6.63a}$$

$$\phi_{1v} = \frac{(R^2 - r^2)\sin\theta\cos\theta}{r^2} \tag{6.63b}$$

$$\phi_{2u} = \frac{(r^2 - R^2)\sin\theta\cos\theta}{r^2} \tag{6.64a}$$

$$\phi_{2v} = \frac{(R^2 - r^2)\cos^2\theta - r^2 \ln\left[\frac{r}{R}\right] + \frac{1}{2}(r^2 - R^2)}{r^2} \tag{6.64b}$$

The velocity enrichment functions are shown graphically for the element below:

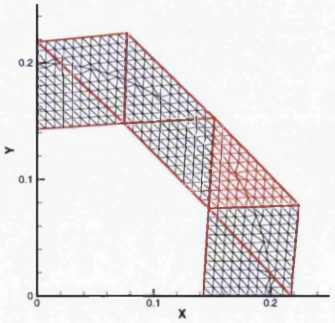
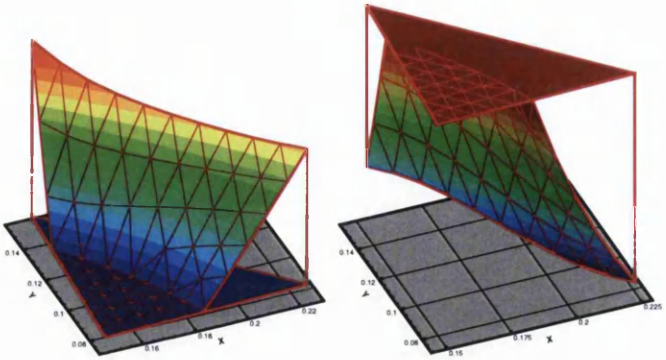
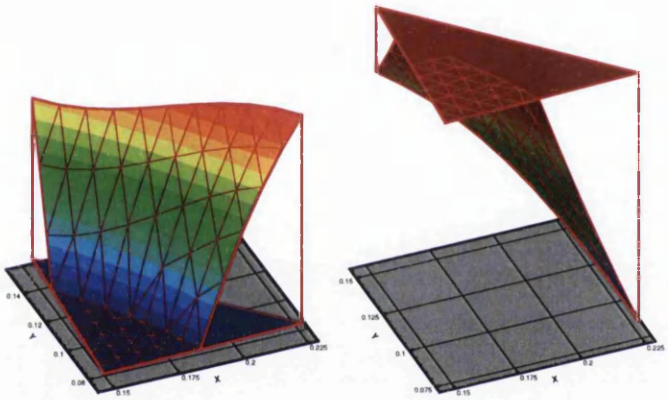


Figure 6.23: Location of plots



(a) ϕ_{1u} (b) ϕ_{1v}

Figure 6.24: Enrichment shape functions ϕ_1



(a) ϕ_{2u} (b) ϕ_{2v}

Figure 6.25: Enrichment shape functions ϕ_2

6.8.3 Pressure Enrichment.

$$\phi_{1p} = -\frac{2\cos\theta}{r} \tag{6.65}$$

$$\phi_{2p} = \frac{2\sin\theta}{r} \tag{6.66}$$

Some authors advocate the use of pressure enrichment in the formulation. Our initial tests have not shown any advantage in using the above enrichment. Therefore pressure enrichment will not be employed in this work.

6.8.4 Boundary Conditions.

In order to study the ability of enrichment to model interfaces, two different boundary condition approaches were considered.

1. In the rectangular propagating wave example the interface was treated as a two phase fluid problem i.e. the boundary conditions were left unchanged from those used for an unenriched solution. The aim was to study the ability of enrichment to produce an accurate solution without the aid of fixed nodes to enforce the exact solution.

2. The Particulate fluid flow example was treated as a rigid body analysis, and the nodes in the interior of the body were constrained to exactly satisfy the motion of the body. This was achieved by constructing a velocity field in the enrichment region that was equal to the translation and rotation of the body. The other enrichment fields vanish inside the body and surface so the total solution in the interior matches the body's motion. For a stationary particle this is a straightforward process as there are no translations or rotations to consider and the internal and surface nodes are set at $\boldsymbol{u}(\boldsymbol{x}) = \mathbf{0}$. See Table (6.1) for particle boundary conditions.

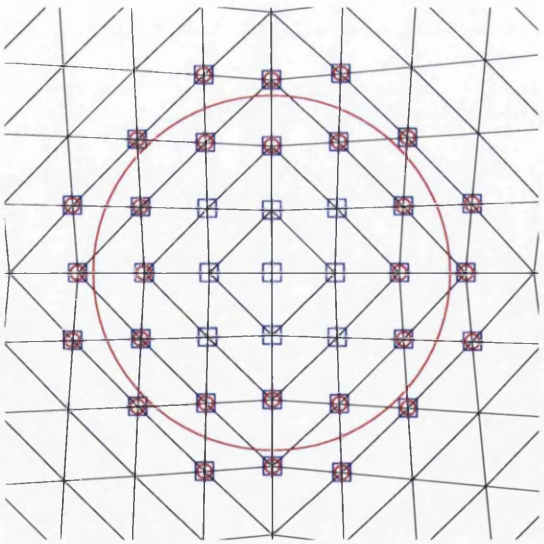


Figure 6.26: Boundary Conditions of enrichment nodes near to the particle surface. ◯ nodes enriched with Stokes flow solution. □ nodes with FE degrees of freedom set to rigid body motion

Nodal DoF	Internal nodes b.c	Intersected nodes b.c	External nodes b.c
u	1	1	0
v	1	1	0
p	0	0	0
a_{1x}	1	0	1
a_{1y}	1	0	1
p_{1a}	1	1	1
a_{2x}	1	0	1
a_{2y}	1	0	1
p_{2a}	1	1	1

Table 6.1: Boundary conditions for *Particle in Stokes flow* example. The boundary condition for each solution variable 1 = fixed, 0 = free. Internal node - the node is in the interior of particle and is not a support of an element intersected by the interface, Intersected node - is a support of an element intersected by the interface, External node - is not affected by the interface.

Remark 6.2: Moving interface.
A moving interface requires constantly changing boundary conditions, therefore it is important that a robust framework is in place to change boundary conditions during the solution process, including storage of permanent boundary conditions, and re-initialization of solver at every timestep.

6.9 Conclusion

The process of implementing X-FEM has been covered in this chapter. Integration of extra terms in a PUM framework into the General Finite Element formulation to form X-FEM, provides a framework which allows problems which incorporate interfaces to be modelled accurately whilst maintaining a computationally cost effective numerical process. In order to implement X-FEM required new methods and approaches previously not required by the GFEM, such as:

- Mesh refinement for improved numerical integration
- Search algorithms
- Variable boundary conditions
- Increased degrees of freedom
- Additional shape functions, i.e. Enrichment
- Differing element solution procedure for each Sub domain e.g. Enriched, Blending, and standard.

X-FEM is computationally expensive, and requires considerable storage for the increased number of operations performed by the solution algorithm. However the improvement in solutions and the accurate modelling of the interface will be shown to outweigh the disadvantages. In the next chapter two numerical examples will be used to demonstrate the ability of X-FEM.

Chapter 7

X-FEM - Numerical Results.

X-FEM is implemented according to the method explained in Chapter 6 using the stabilized FE formulation and software programs developed for this work. The X-FEM solutions are benchmarked against the stabilized FEM and/or the exact solution for a direct comparison and the success of X-FEM and the implementation will be judged from these solution results.

To ensure that X-FEM is a properly functioning method two numerical examples are presented in this chapter. The examples have been chosen because they represent the different aspects of X-FEM, i.e. 1D & 2D and stationary & evolving in time, are i) Propagation of a Rectangular Wave, and ii) Uniform flow past a stationary particle. The first example is a repeat of a previous example from Chapter 5 and the second is an example of Stokes flow.

The data compiled previously for this example clearly demonstrates the improved solution achieved from stabilized FEM, i.e. the results are less affected by spurious oscillations, than those obtained using the classical Galerkin method. For this reason only stabilized X-FEM is considered here.

7.1 Propagation of a Rectangular Wave.

The Propagation of a Rectangular wave was used previously in Chapter 5 to validate the effectiveness of timestepping algorithms i.e. Generalized Midpoint rule and the Generalized-α method. In this chapter the example will be used to assess the interface capturing ability of X-FEM in 1D.

Both Aliabadi and Tezduyar [4] and Chessa and Beltschko [17] have considered two-phase fluid problems similar to the propagating rectangular wave example, albeit in two dimensions. Chessa implemented X-FEM with some success and was

able to achieve good solution results on a relatively coarse mesh. Aliabadi implemented an interface sharpening/mass conservation (IS/MC)algorithm to achieve almost exact results for a *2D advection of a circle* problem. The solution results of Aliabadi will be discussed in greater depth in the Discussion section of this example.

The initial boundary value problem will be reiterated here for ease.

Model.

Only pure advection ($\mu = 0$ & $Pe^h = \infty$) is considered, as the ability of X-FEM to maintain a sharp interface is negated by an advection diffusion solution. The Heaviside enrichment function is used, though other enrichment function can and are used successfully to model similar problems.

Geometry.

$$\Omega =]0, L[\tag{7.1}$$

Boundary Conditions.

$$u(0,t) = 0 \qquad u(L,t) = 0 \tag{7.2}$$

Initial Conditions.

$$u(x,0) = 1 \qquad c \leqslant x \leqslant d \qquad (c = 0.1L, d = 0.2L) \tag{7.3}$$

$$u(x,0) = 0 \qquad \text{elsewhere} \tag{7.4}$$

Mesh.

$$\Delta x = \frac{1}{1000} L \tag{7.5}$$

Exact Solution.

Pure Advection - For this example the diffusion parameter $\mu = 0$, see (4.5) for 1D advection diffusion equation.

$$u(x,t) = 1 \qquad c + a_x t \leqslant x \leqslant d + a_x t \tag{7.6}$$

$$u(x,t) = 0 \qquad \text{elsewhere} \tag{7.7}$$

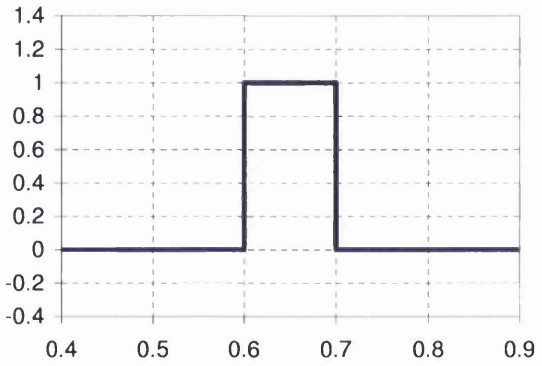


Figure 7.1: Exact solution for the Pure advection case

Numerical Results.

Figure 7.2 shows the plot of the rectangular wave at $t = 0.5s$, and includes the analytical, FEM and X-FEM solutions for $C^h = 0.1, 1.0$ for both Generalized Midpoint rule and the Generalized-α method.

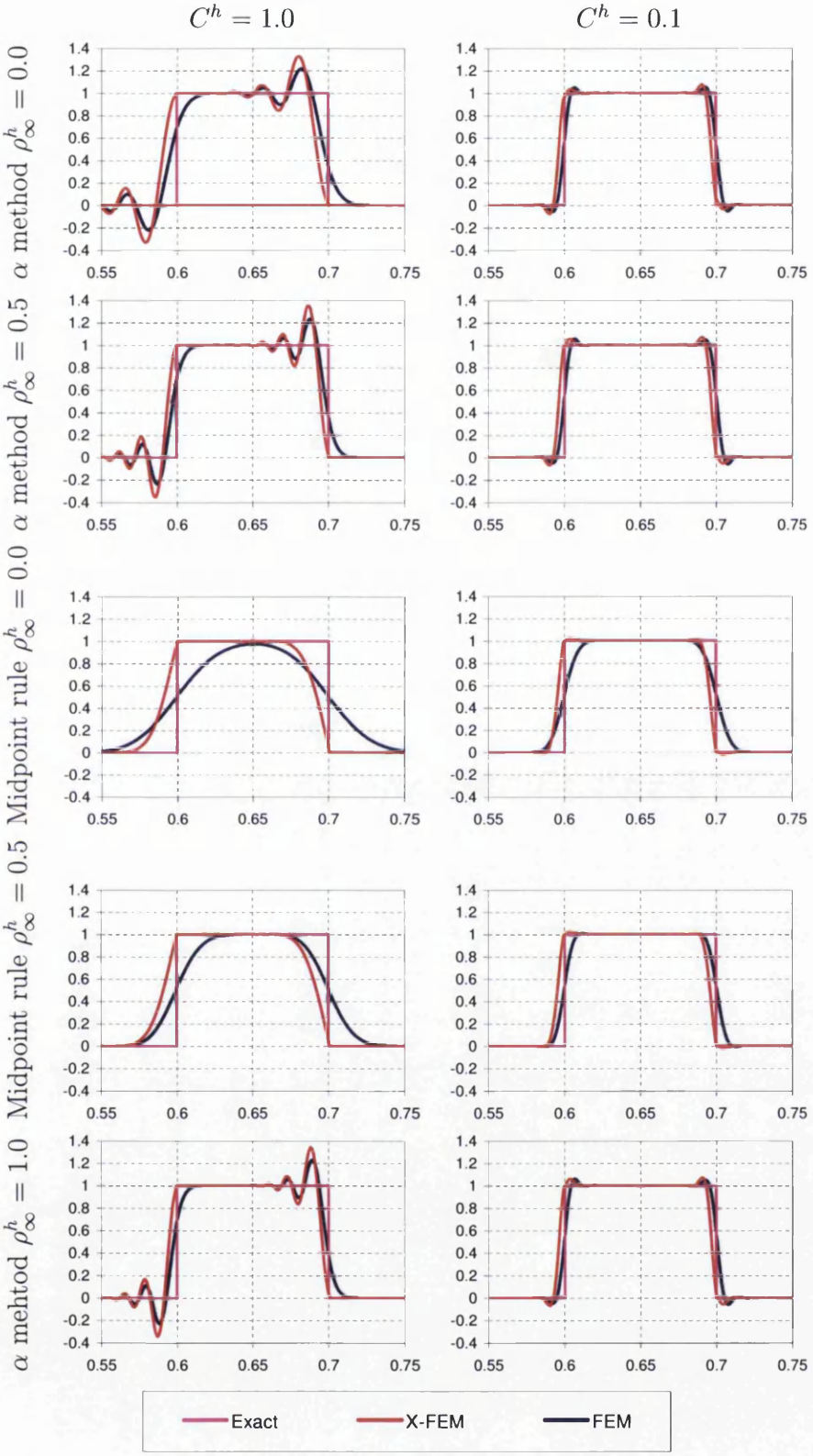


Figure 7.2: Comparison X-FEM solution with FEM and exact solutions for different timestepping algorithms and C^h

Discussion.

It is important to separate the effects due the timestepping algorithms and those due to X-FEM.

Timestepping Algorithm.

The conclusions drawn for the timestepping algorithms in Chapter 5 are also apparent in the X-FEM results:

- *Convergence.* The Generalized-α method converges sooner as $\Delta t \rightarrow 0$ than the Generalized Midpoint rule.
- *Limits of method.* The Generalized Midpoint rule for values of $\rho^h < 1.0$ suffers from too much numerical damping. When $\rho^h \rightarrow 1.0$ both the Midpoint rule and the Generalized-α method solutions suffer from spurious oscillations when larger timesteps are used.

X-FEM.

The success of X-FEM is dependent on the correct implementation of both the interface modelling and location algorithms; should one fail then the whole X-FEM solution fails. In this example the results demonstrate of the potential of X-FEM, however they also show the problems associated with X-FEM.

- *Interface Capturing.* Both $C^h = 1.0$ and $C^h = 0.1$ X-FEM solutions give better results than those obtained by the FEM. This seen in the improved resolution at downstream stream *corner* of each interface and the *steeper slope* of the interface. This result is apparent for all timestepping algorithms.

  For the timestepping algorithms that suffer from too much numerical damping, the enrichment alleviates the disadvantage of too much damping by forcing a more accurate depiction of the interface onto the solution. This is especially apparent for $C^h = 1.0$.

  For the timestepping algorithms which exhibit oscillations for large timesteps, the addition of enrichment increases the oscillations amplitude. The oscillations originate from incorrect spatial resolution of the interface, and the addition of enrichment aggravates the situation by enforcing the interface

at every timestep therefore maintaining the cause of the oscillations, leading to increased amplitude. In contrast FEM does not suffer this because the interface is soon *smoothed* by the time integration scheme, reducing the cause of the oscillations, which lessen.

The advantage gained from using the improved convergence properties of the Generalized-α method are canceled by this phenomenon; it is easily circumvented by a well posed boundary value problem.

- *Interface Location.* It can be assumed that the interface location algorithm functioned correctly for $C^h = 1.0$ and $C^h = 0.1$ solutions, as both exhibit the effects of enrichment at the required locations. Figure 7.3 demonstrates the deterioration of the solution if the locating algorithm wrongly positions the interface at the start of a simulation:

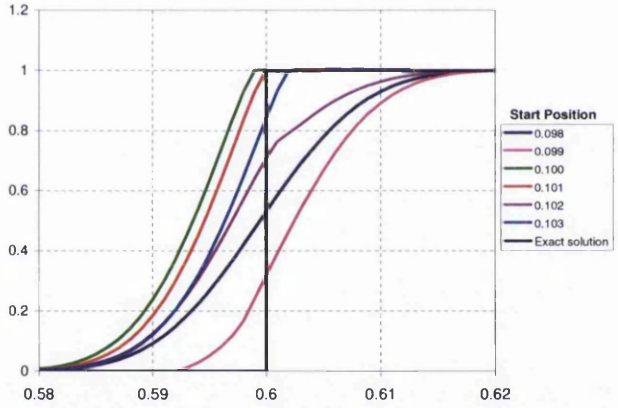


Figure 7.3: Effects of an error in locating the interface. Solution at $t = 0.5s$ - actual start position 0.1.

Therefore we can conclude that correct positioning of the interface has a significant effect on the overall outcome of a simulation.

- *Improvements.* Observations in papers studying two phase fluid flows, see Aliabadi and Tezduyar[4], suggest that X-FEM struggles in maintaining global mass conservation for long term time integration of two phase fluids. The effect is non negligible and non physical mass transfer that can lead to significant errors; the extent of the non physical mass transfer will be dependent on the nature of the formulation, as well as the application.

In order to maintain mass conservation Aliabadi implements Interface Sharpening algorithms for the post advection solution, in effect post-processing the solution results.

On face value this approach appears to be flawed for problems where the exact solution is not known, however it is appreciated that this method can provide significantly improved solution results for difficult problems. Therefore it is the authors opinion that the methods such as sharpening required continued investigation into the appropriate use of such methods. Aliabadi also places the interface over 2-3 elements, in the form of a parabola, in effect removing the discontinuous *interface* from the problem. This technique creates a requirement for a sufficiently refined mesh to avoid the interface extending out further than necessary.

These methods are not implemented in this work, and therefore the Aliabadi's results are not comparable. The motivation of this example was to document the functionality of X-FEM to model interfaces, unaided by further methods which would disguise the solution properties of the method.

Conclusion.

The results from this example are demonstrative of the capability of X-FEM. We have shown that X-FEM offers a improved solution over the standard FEM for a two phase fluid. However X-FEM requires a well posed boundary value problem if it is to avoid certain pitfalls described in the Discussion above.

7.2 Flow past a stationary particle.

In this we will repeating an example described in Wagner *et al* [86, 85], i.e. a Stokes flow type problem - Flow past a stationary particle.

In order for this example to successfully demonstrate X-FEM it is important that the solutions are directly comparable to an exact solution. Sadly Stokes flow for the problem of *laminar flow past a 2D cylinder*, as discussed in Chapter 2, has no analytical solution at infinity. However for a finite domain, with known boundary conditions, an analytical solution exists[1].

Two separate simulations are performed in this section, using identical domains, with differing external boundary conditions. The results for each simulation will be obtained for X-FEM and FEM to compare and contrast the observations and to demonstrate the effectiveness of X-FEM.

The first example will verify the X-FEM against a known analytical solution, using the enrichment function as the external boundary conditions and as enrichment. In this situation the solution is enriched with the exact solution. The second will demonstrate the more commonly found problem, i.e. uniform flow, however the enrichment and external boundary conditions are not the same function. This example will test the ability of *not exact* enrichment to model the solution accurately.

Model

The examples will model laminar flow past a stationary particle in a finite domain with fixed boundary conditions using the Stokes equations[2] to enrich the solution. The X-FEM solutions are produced for two different sets of external boundary conditions, i.e. *enrichment velocity* boundary conditions and *unit velocity* boundary conditions.

The enrichment functions utilized for both examples are $\boldsymbol{\phi}_\alpha$, see (6.63a) & (6.64a). The inflow is assumed to be laminar and is denoted as $\boldsymbol{u}$, and density and viscosity of the fluid given as ρ and μ respectively (the following values were used $\rho = 1$, $\mu = 0.01$). Timestepping method will be the Generalized-α method for $p^h_\infty = 0.9$.

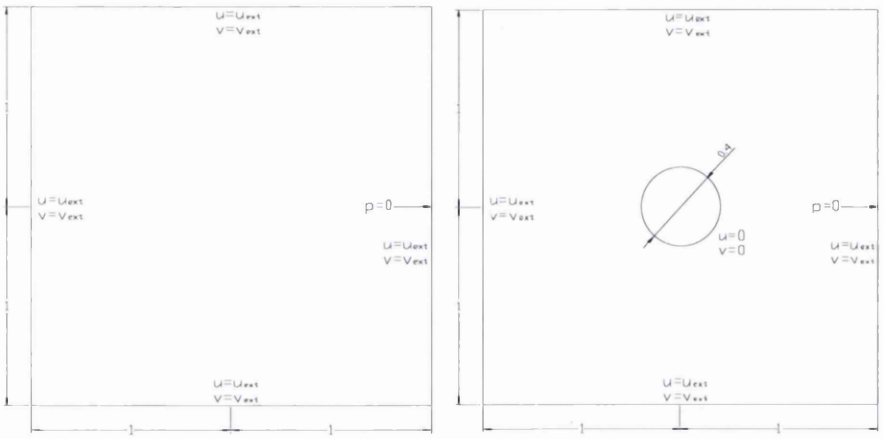
[1] Only applicable at finite distance from the cylinder

[2] i.e. incompressible viscous Navier Stokes equations ignoring time dependent and convective terms

Geometry

$$\Omega_x =]-1,1[\qquad \Omega_y =]-1,1[\tag{7.8}$$

The particle radius R is 0.2 and is located at the origin of the domain, whilst the distance from the particle centre is denoted by r.



(a) XFEM domain (b) FEM domain

Figure 7.4: Domain geometry and problem setup for X-FEM and FEM BVP's - u, v, and p are the velocity in the x and y directions and pressure respectively

Boundary Conditions.

The Navier-Stokes problem must be completed with suitable initial and boundary conditions to form a well-posed initial boundary problem. The initial and boundary conditions that form the initial boundary value problem for this problem are explained below.

Velocity Boundary Conditions

A velocity profile is assigned to the inflow/outflow/lateral boundaries, where the fluid velocity $\boldsymbol{u}$ is given as:

$$\textit{Exact enrichment example} \qquad \boldsymbol{u} = \phi_1 \tag{7.9}$$

$$\textit{Unit velocity example} \qquad \boldsymbol{u}_x = 1 \tag{7.10}$$

For the comparable general FEM problem there is a no-slip and no penetration condition on the particle surface.

Pressure Boundary Conditions.

A Pressure enrichment function was developed alongside the velocity enrichment; the pressure boundary condition requirements for the Navier-Stokes equations specifies that no initial condition need be set for the pressure unknown. The placement of the fixed pressure node is also arbitrary, therefore for these examples pressure has been fixed in the centre of the particle, in conjunction with Figure 7.4.

Mesh.

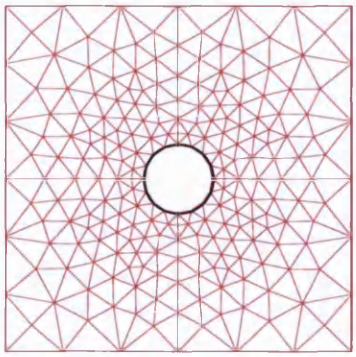
All meshes were produced for this example using ELFEN software. A total three pairs of meshes were generated (X-FEM - or nonconforming and FEM - or conforming), with increasing spatial refinement. X-FEM mesh 516, (1024), [1726] elements and 287, (553), [916] nodes, and FEM mesh 416, (1512), [2896] elements and 228, (808), [1522] nodes.

The FEM meshes are denser in comparison to the X-FEM meshes in order to properly define the particle's surface.

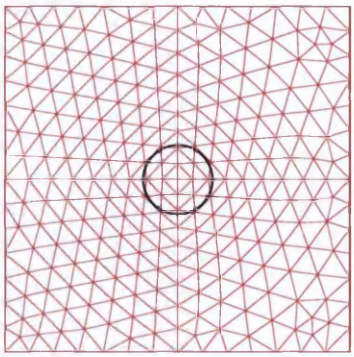
Conforming FEM Mesh

X-FEM Mesh

Pair 1

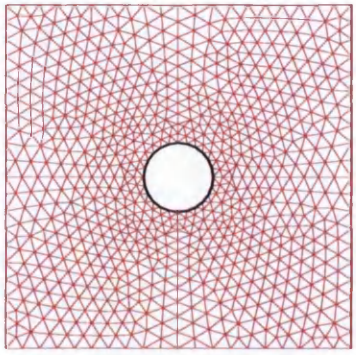


(a) 416 elements 228 nodes

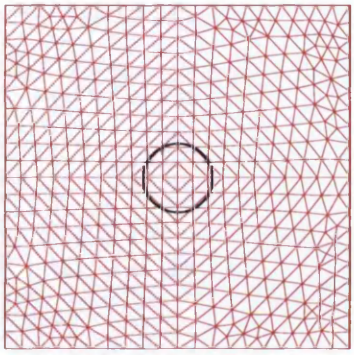


(b) 516 elements 287 nodes

Pair 2

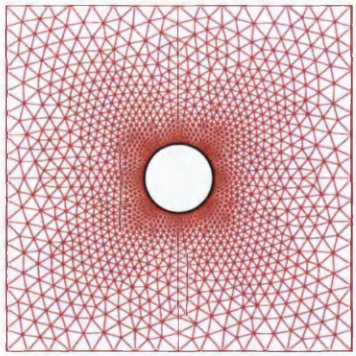


(c) 1512 elements 808 nodes

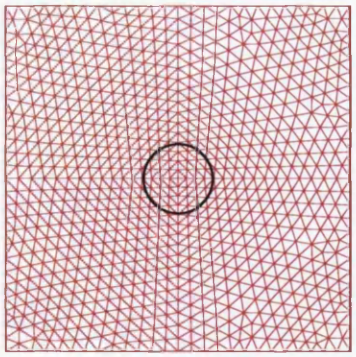


(d) 1024 elements 553 nodes

Pair 3



(e) 2896 elements 1522 nodes



(f) 1726 elements 916 nodes

Figure 7.5: X-FEM and FEM Meshes - X-FEM mesh 516, (1024), [1726] elements and 287, (553), [916] nodes, and FEM mesh 416, (1512), [2896] elements and 228, (808), [1522] nodes

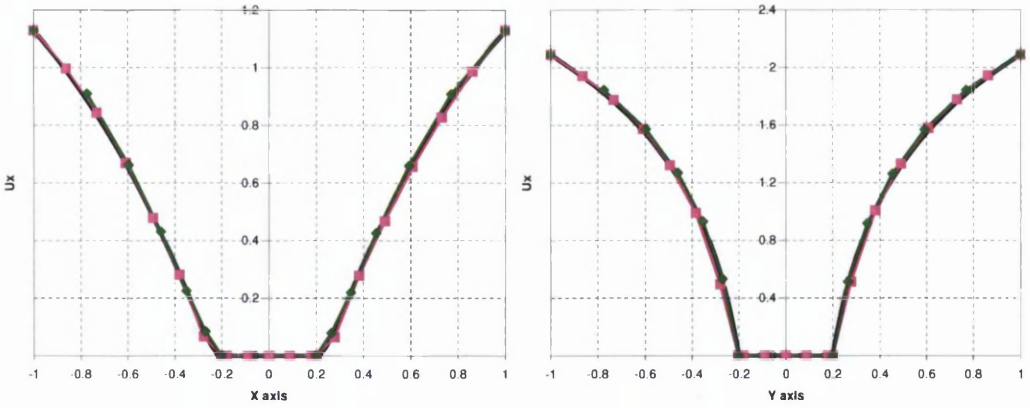
Numerical Results - Exact enrichment example.

In this example the enrichment function is the exact solution to the problem, therefore we can expect a solution which matches the exact solution.

Figure 7.6 shows the three pairs of meshes were generated in order to compare X-FEM with FEM for different levels of mesh coarseness. For Pairs 1 & 3 there is a direct comparison between the exact solution and the solutions obtained from X-FEM and FEM, see Figure 7.6.

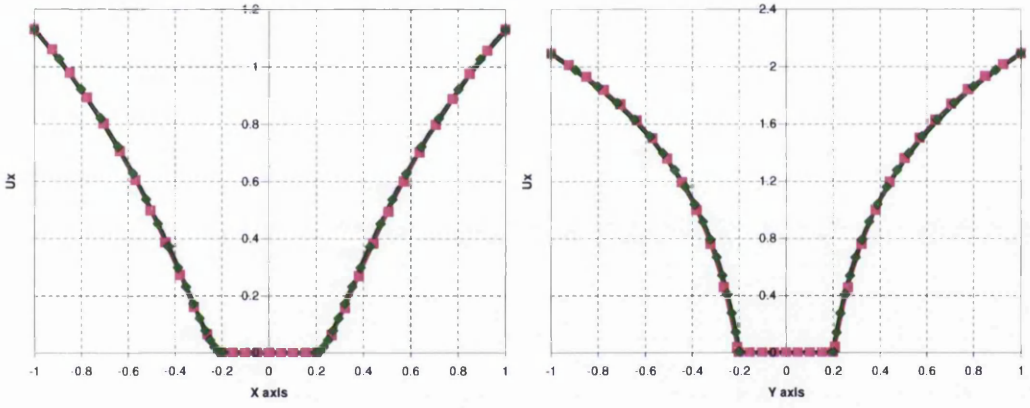
Only the charts for u are produced as these are the most demonstrative of the accuracy of X-FEM; v & p results are produced for the Pair 2 case in the form of contour plots, see Figure 7.9.

Figure 7.8→Figure 7.7 show for Pair 2 the comparison is more indepth. In addition to the X-FEM and FEM solutions there are 2 solutions obtained using FEM on a non-conforming mesh. In these solutions the FEM algorithm was used in conjunction with the X-FEM mesh; the particle was modelled by setting the nodes interior and exterior to the particle surface to zero velocity, i.e. were given fixed boundary conditions. The intention is to highlight the results that would be achieved on a nonconforming mesh if X-FEM was not implemented.



(a) Pair 1 - u velocity from $y = 0$ plane

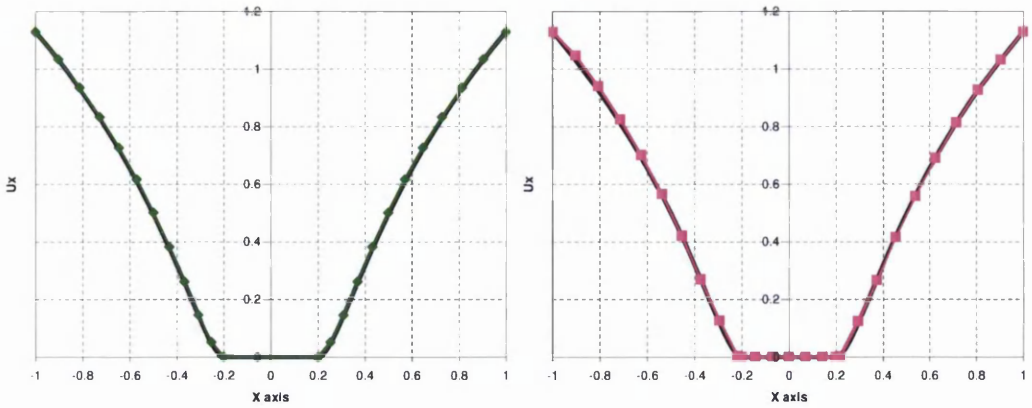
(b) Pair 1 - u velocity from $x = 0$ plane



(c) Pair 3 - u velocity from $y = 0$ plane

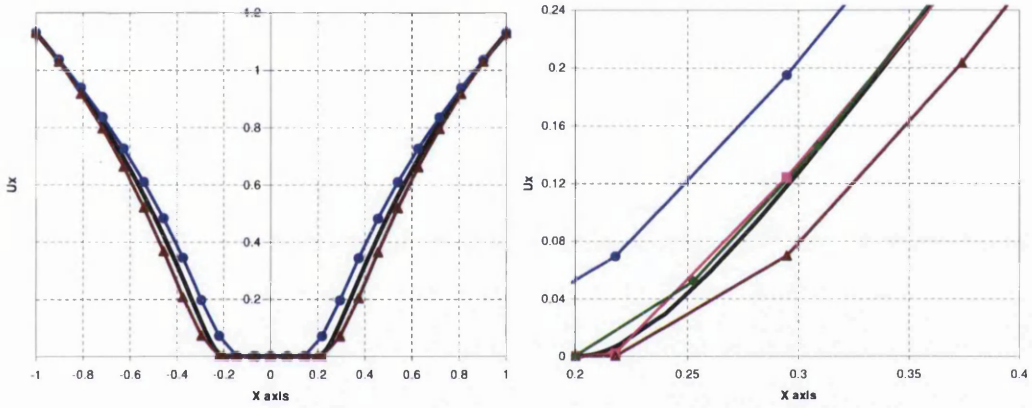
(d) Pair 3 - u velocity from $x = 0$ plane

Figure 7.6: Pair 1 & 3 - Comparison of u - exact solution with FEM and X-FEM solutions



(a) Conforming FEM solution

(b) X-FEM solution



(c) Non conforming FEM solution

(d) Zoom of charts (a)(b) & (c)

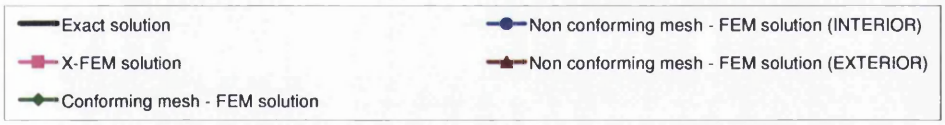
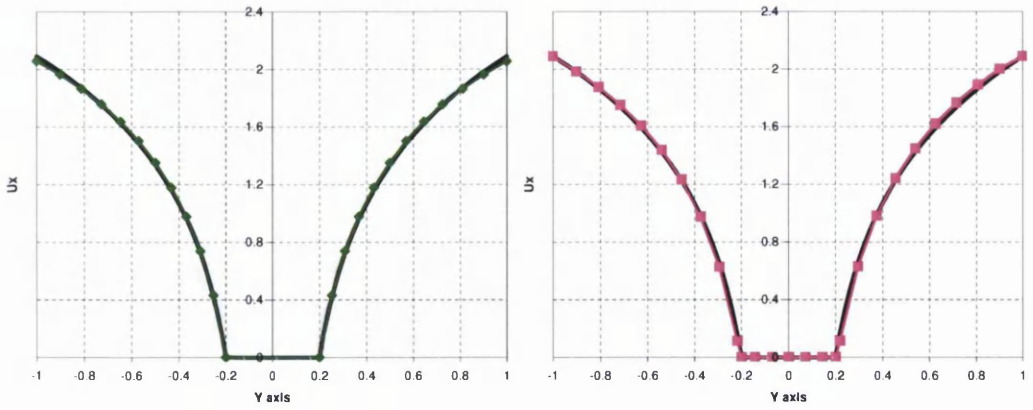
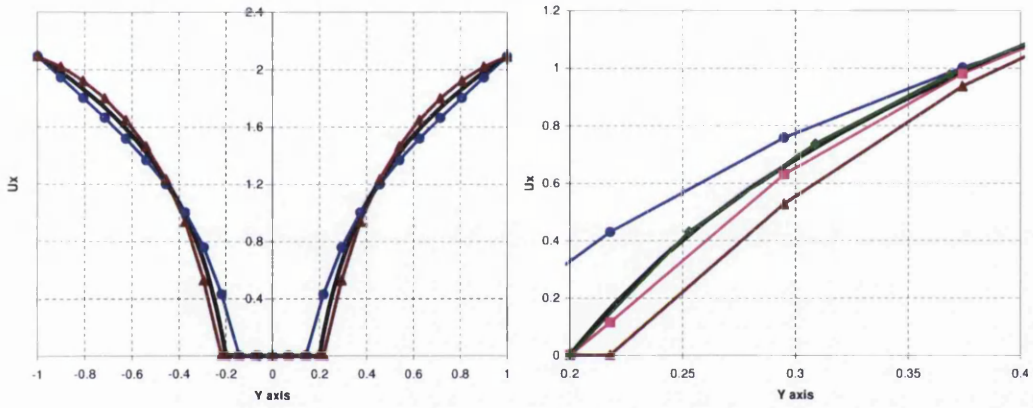


Figure 7.7: Pair 2 - u velocity from $y = 0$ plane, for X-FEM (using different mesh solutions) and FEM. (c) represents the FEM driven solution on the X-FEM mesh - for fixed nodes interior to the particle surface and for fixed nodes just exterior to the particle surface



(a) Conforming FEM solution

(b) X-FEM solution



(c) Non conforming FEM solution

(d) Zoom of charts (a)(b) & (c)

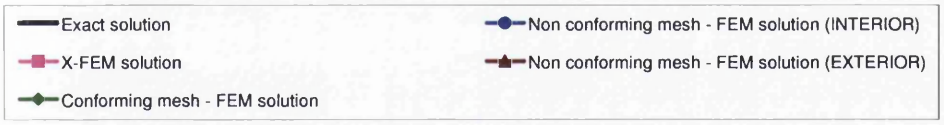


Figure 7.8: Pair 2 - u velocity from $x = 0$ plane, for X-FEM (using different mesh solutions) and FEM. (c) represents the FEM driven solution on the X-FEM mesh - for fixed nodes interior to the particle surface and for fixed nodes just exterior to the particle surface

Conforming FEM Mesh X-FEM Mesh

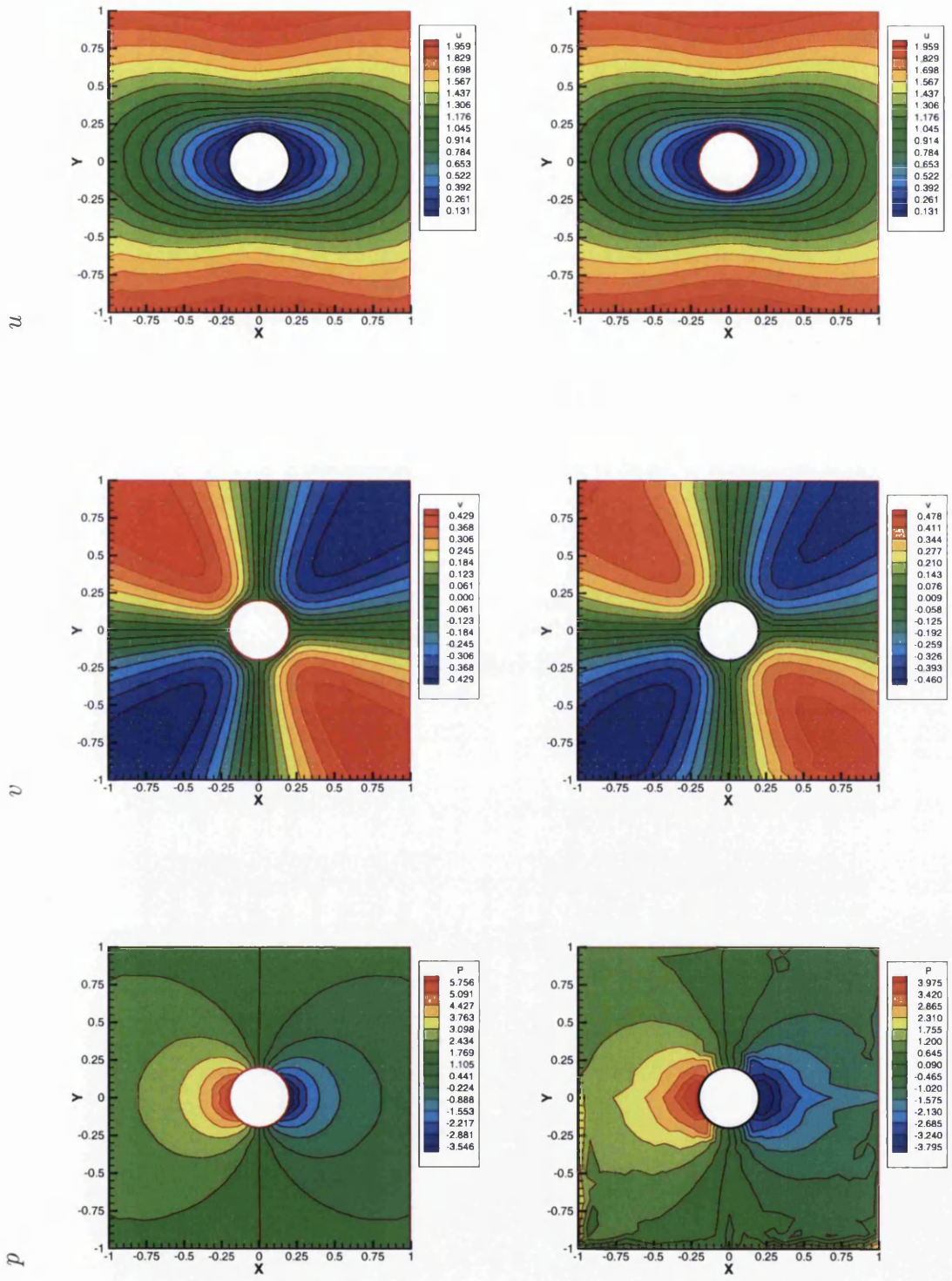


Figure 7.9: Pair 2 - Comparison of velocity and pressure contour plots

Numerical Results - Unit velocity boundary conditions.

In this example the enrichment function is not the exact solution to the problem being modelled.

Only the Pair 2 meshes will be considered in this example.

Figure 7.10 u, v & p results are produced for the Pair 2 case in the form of contour plots.

Figure 7.11 plots u, v velocity profile offset at y=0.21.

Conforming FEM Mesh X-FEM Mesh

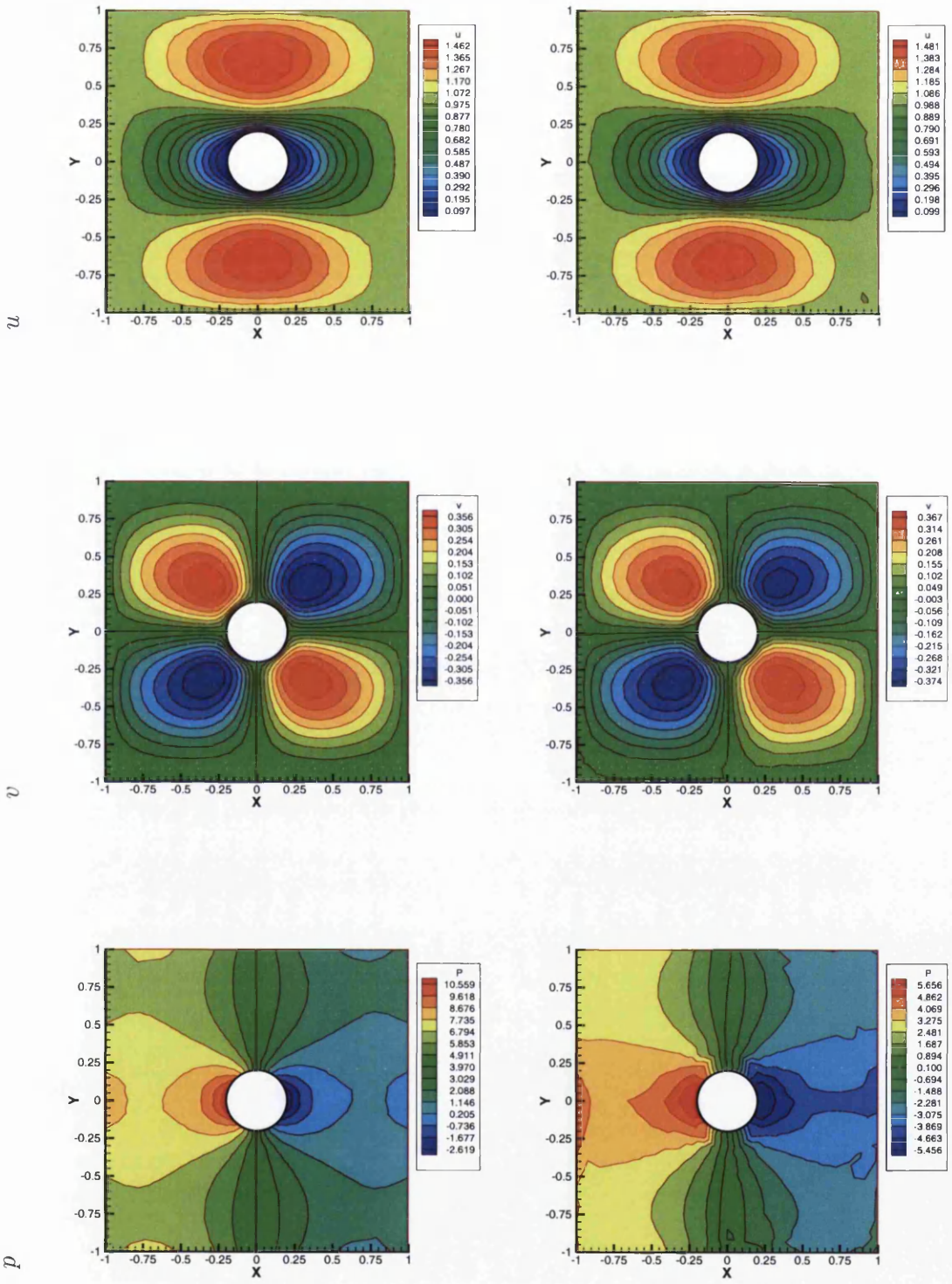


Figure 7.10: Pair 2 - Comparison of velocity and pressure contour plots

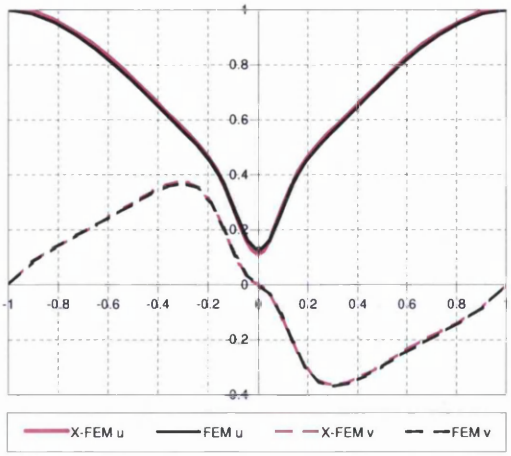


Figure 7.11: u & v at y=0.21

Discussion.

Due to the stationary nature of the problem the difficulties faced in continuously locating the interface, as seen previously in the *1D propagating rectangular wave* example, see Section 7.1, have been avoided in this example. Also in this example the treatment of the interior and exterior nodes close to the interface is different to the method used in the 1D example. The 1D example modelled two phase fluid flow and all nodes were free, whilst for the *Stokes flow past a stationary particle* examples the interface is assumed to be a boundary of a solid particle and therefore all enriched nodes are given fixed boundary conditions. The solution close to the particle surface is therefore entirely dependent on the enrichment function, and any failing can be correctly attributed to the incorrect implementation of the X-FEM method.

Enrichment velocity boundary conditions.

The first Stokes flow example is a direct comparison between the X-FEM, FEM and the exact solutions.

- Figure 7.6 show the velocity profiles for the coarse mesh (a) & (b) and a fine mesh (c) & (d). Comparatively there is little to differentiate between the X-FEM, FEM and exact solutions.

- Figure 7.7 & 7.8 shows the results for a medium coarse mesh, separated into individual plots for a more indepth analysis. In addition to the X-FEM and FEM solution plots (for u along the x and y axes) are plots (plot (c) for both figures) for an FEM driven solution using the X-FEM mesh,

i.e. effectively an X-FEM solution without the application of enrichment. The purpose of this plot is to demonstrate the effect enrichment has on the X-FEM solution. In plot (d) for both figures is a close up at the point of enrichment, i.e. particle surface, with the all velocity profiles plotted. The X-FEM result is shown clearly not to be exact however the deviation is minimal and considerably better in comparison to the FEM nonconforming solution.

- Figures 7.9 are contour plots for u, v & p for X-FEM and FEM. These plots give an overall indication of the performance of X-FEM and notably the result for p are not as accurate as the u plot. The pressure ranges for the X-FEM and FEM p plots are slightly different however this issues requires further investigation. This is not a reflection on the performance of the schemes.

- Though not apparent in this results the author noted that the accuracy of the solution was dependent on the proximity of the external fixed node to the particle surface. The closer node the more accurate the X-FEM approximation. This was not so readily apparent for the finer mesh as close proximity was always assured.

Numerical Results - Unit velocity example.
The first Stokes flow example demonstrated that the FEM solution was a good approximation of the exact solution. The second Stokes flow example is a comparison of the X-FEM and FEM solutions for a $\boldsymbol{u}_x = 1.0$ external boundary condition.

- Figure 7.9 are contour plots for u, v & p for X-FEM and FEM. All three contour X-FEM plots i.e. u, v & p, deviate slightly from the FEM solutions.

- Figure 7.11 shows the u & v velocity profiles for $y = 0.21$ axis. The results are comparable with little deviation from the FEM solution.

General Observations.
The motivation for these examples was given by Wagner *et al* [86], and the results obtained here in this thesis are directly comparable to Wagner's data. But it should be noted that Wagner has omitted to include pressure plots in his paper.

7.3 Conclusions.

X-FEM has been proven to be a powerful tool for numerical simulations. All the examples provided a good demonstration of the potential of X-FEM, and the solutions can be classified as a success. The clear advantage of X-FEM is its ability to be applied at any point within the domain without the need to remesh the domain. This is an extremely attractive quality when compared to the classical adaptive simulations of problems with moving interfaces.

Wagner *et al* [86] determined that it was possible to achieve acceptable 2D X-FEM results on meshes with almost element size interfaces. The observations in this work support this statement for fluid rigid body interaction only; for multiphase fluids this would lead to difficulties in maintaining the interface, and subsequently large solution errors would occur due to non physical mass transfer. The reason behind this discrepancy was touched upon previously, when it was noted that Interface Modelling methods fail to satisfy conservations laws for long time integration problems. For fluid rigid body interaction problems the fixed boundary conditions maintain the interface, and the enrichment is used to enforce the solution but not the interface, however in multiphase fluid flows mass conservation becomes a problem because the enrichment functions are required to enforce the solution and the interface.

Chapter 8

Conclusions.

In this thesis an implicit Semi-Discrete Stabilized eXtended Finite Element formulation has been successfully developed and implemented for laminar Newtonian incompressible fluid flows. In doing so we have contributed to the research into the field of incompressible fluid flows, multiphase flow and fluid-rigid body interaction.

8.1 General Conclusions.

8.1.1 Governing Equations

In Chapter 2 we developed the full Navier-Stokes equations, however these equations are complex and are difficult to solve in their present state, therefore the equations were simplified by introducing incompressibility condition etc. thereby reducing the overall nonlinearity of the equations. We will also only considered laminar problems as fluid problems such as turbulent flow are extremely difficult to model. The resulting equations i.e. the laminar incompressible Navier-Stokes equations, were used throughout to govern the fluid behaviour and form the basis of all numerical methods discussed in this thesis.

8.1.2 Newton-Raphson Procedure.

Due to the highly nonlinear nature of the problem we proposed the Newton-Raphson iterative solution procedure. However in order to implement the Newton-Raphson procedure we first had to linearize the governing equations, yielding a consistent stiffness matrix, see Slijepcevic [70]. This approach produced a robust and efficient method which demonstrated quadratic convergence for all but one

numerical example. Unsurprisingly the procedure failed for a coarse mesh for a solution with a large timestep.

8.1.3 Numerical Method - Finite Element Method.

The complexity of the problem prevented considerations of an analytical approach, therefore a numerical approach was advocated in the form of the Finite Element Method. In Chapter 3 we introduced the Finite Element Method and the Galerkin Weighted Residual method to formulate the weak form of the governing equations. It was soon determined that the Galerkin method required computationally inconvenient velocity/pressure interpolations, see Alexandrou *et al* [3], to satisfy the Babuška & Brezzi condition and in an effort to circumvent the Babuška & Brezzi condition and correct other failings of the Galerkin method a *stabilized* formulation was implemented. The stabilization was added in a Petrov-Galerkin framework and was shown to be successful in controlling non physical oscillations in two numerical examples *[1d advection diffusion, lid driven cavity].*

8.1.4 Time Integration.

In the following chapter we focused on unsteady fluid flows, and the integration of the temporal domain into the FE formulation of the previous chapter. A semi-discrete method, with a solid mechanics background, see Chung and Hulbert [21] and proposed for computational fluid dynamics by Jansen *et al* [50], was chosen to perform time integration. The scheme known as the Generalized-α method was implemented; the Generalized Midpoint rule was also implemented to act as a comparative scheme. Three numerical examples *[1d advection diffusion for a propagating rectangular wave, flow past a circular cylinder, flow past a square cylinder with near wall effects]* were used to validate the time integration schemes, with the Generalized-α method demonstrating improved convergence properties over Generalized Midpoint rule. We also noted that $\rho_\infty^h = 1.0$ caused difficulties which could be avoided without sacrificing the improved solution properties of the Generalized-α method if $\rho_\infty^h = 0.9$.

8.1.5 Numerical Method - eXtended Finite Element Method.

In the final chapters we extended the stabilized FEM formulation to include the eXtended Finite Element Method (X-FEM) as a method for modelling interfacial fluid flows. X-FEM is an extension of the General Finite Element Method with added terms in a Partition of Unity Method (PUM) framework, and is in the group of interface modelling methods known as Interface Capturing methods, i.e. uses a fixed mesh, which does not conform to the interfaces present.

The implementation of X-FEM proved to be a complex and challenging process, requiring additional methods to be incorporated into the existing solution procedure, including enrichment, interface location and modelling algorithms. These methods were described in some depth. Two problem types were modelled to validate X-FEM *[1D two phase fluid flow, rigid particle in Stokes flow]*. Each example represented a different focus, i.e. multiphase fluid flow and fluid-rigid body interaction, and required different problem setup, and different enrichment functions.

8.1.5.1 Enrichment.

The enrichment functions were problem dependent. For the *[1D advection diffusion two phase fluid]* example a Heaviside function was used to great effect, with considerable improvement over the FEM solution; it was noted that matching results could be obtained using other similar enrichment functions. The 2D enrichment functions for the *[rigid particle in Stokes flow]* example were obtained from an analytical solution, for a finite domain, of Stokes equations i.e. subset of the full Navier-Stokes equations developed Chapter 2, see Wagner [86]. The solutions were comparable to those obtain from similar FEM simulation, demonstrating the applicability of X-FEM for fluid-rigid body interaction problems.

The X-FEM results demonstrated that X-FEM was a method that had the advantages of the General Finite Element Method i.e. robustness, efficiency etc, yet was capable of modelling interfaces accurately and independently of the mesh.

8.2 Recommendations for future work.

The success of X-FEM should be followed up with further study into areas touched upon in this thesis i.e. level set methods, see Sukumar [73], pressure enrichment,

improved blending methods, fluid-rigid body interaction modelling moving particles, and multi-particle flow with lubrication, see Wagner *et al* [86]. These methods would help develop the understanding of the potential of X-FEM, and they would also allow us to perform obvious continuation to the examples demonstrated in Chapter 7, i.e. multi particle flow simulations.

Appendices

Appendix A

Leibnitz's Theorem.

The Leibnitz Theorem gives a formula for the differentiation of a definite integral whose limits are functions of the differential variable.

Consider a simple one dimensional example:

$$\Gamma(\alpha) = \int_{a(\alpha)}^{b(\alpha)} \gamma(x,\alpha)dx \tag{A.1}$$

A combination of the chain rule and the fundamental theorem of calculus yields the Liebnitz Theorem. The one dimensional case can be written as:

$$\frac{d\Gamma}{d\alpha} = \frac{d}{d\alpha}\int_{a(\alpha)}^{b(\alpha)} \gamma(x,\alpha)dx = \int_{a(\alpha)}^{b(\alpha)} \frac{\partial\gamma}{\partial\alpha}dx + \gamma(b(\alpha),\alpha)\frac{db}{d\alpha} - \gamma(a(\alpha),\alpha)\frac{da}{d\alpha} \tag{A.2}$$

The theorem can also be developed for three dimensional fluid flows. In this case the Leibnitz Theorem separates the time derivatives of integral quantities into two distinct terms, i.e. changes that are happening inside the boundaries and changes which occur due to fluxes through the boundaries.

Consider the below equation:

$$\Gamma(t) = \int_{V(t)} \gamma dV \tag{A.3}$$

The value of the time derivative of the variable Γ is of interest; the limits of the integration are time dependent. From the definition of a derivative:

$$\frac{d\Gamma}{dt} = \frac{d}{dt}\int_{V(t)} \gamma dV = \lim_{\Delta t\to 0}\left\{\frac{1}{\Delta t}\Big[\int_{V(t+\Delta t)} \gamma(t+\Delta t)dV - \int_{V(t)} \gamma(t)dV\Big]\right\} \tag{A.4}$$

Equ (A.4) can be rewritten by adding and subtracting the term $\int_{V(t)} \gamma(t+$

$\Delta t)dV$ inside the limit.

$$\frac{d\Gamma}{dt} = \frac{d}{dt}\int_{V(t)} \gamma dV =$$

$$\lim_{\Delta t \to 0}\left\{\frac{1}{\Delta t}\Big[\int_{V(t+\Delta t)} \gamma(t+\Delta t)dV - \int_{V(t)} \gamma(t+\Delta t)dV\Big]\right\} \tag{A.5a}$$

$$+ \lim_{\Delta t \to 0}\left\{\frac{1}{\Delta t}\Big[\int_{V(t)} \gamma(t+\Delta t)dV - \int_{V(t)} \gamma(t)dV\Big]\right\} \tag{A.5b}$$

Equ (A.5a) is the equivalent of fixing the integrand and varying the volume, and (A.5b) the equivalent of fixing the volume and varying the integrand. Also it will be noted that (A.5b) is the derivative of γ with respect to time, and (A.5) becomes:

$$\frac{d\Gamma}{dt} = \frac{d}{dt}\int_{V(t)} \gamma dV = \lim_{\Delta t \to 0}\left\{\frac{1}{\Delta t}\Big[\int_{V(t+\Delta t)-V(t)} \gamma(t+\Delta t)dV\Big]\right\} + \int_{V(t)} \frac{\partial \gamma}{\partial t} dV \tag{A.6}$$

The term (A.5a) represents the integral of the difference in volume from t to $t + \Delta t$. This integral can be transformed to a surface integral by observing that:

$$\partial V = dist \times \partial S \tag{A.7}$$

where *dist* is the orthogonal distance between the two volumes at $V(t)$ and $V(t + \Delta t)$, $\mathbf{n}$ normal to the surface and $\mathbf{w}$ is a velocity vector of points on the moving surface $S(t)$:

$$dist = \mathbf{w} \cdot \mathbf{n}\Delta t \tag{A.8}$$

Completing the limit process and replacing dV by $\mathbf{w} \cdot \mathbf{n}\Delta t\, dS$:

$$\frac{d\Gamma}{dt} = \underbrace{\frac{d}{dt}\int_{V(t)} \gamma dV}_{\text{total rate of change}} = \underbrace{\int_{V(t)} \frac{\partial \gamma}{\partial t} dV}_{\substack{\text{rate of change} \\ \text{inside volume}}} + \underbrace{\int_{S(t)} \gamma(t)\mathbf{w} \cdot \mathbf{n} dS}_{\substack{\text{net flow through} \\ \text{bounding surface}}} \tag{A.9}$$

By applying the Divergence Theorem (2.8) to (A.9) the three dimensional scalar version of the Leibnitz Theorem reads.

$$\boxed{\frac{d}{dt}\int_V \gamma dV = \int_V \frac{\partial \gamma}{\partial t} dV + \int_V \nabla \cdot (\gamma \mathbf{w}) dV} \tag{A.10}$$

Brief Comparison The simple one dimension Leibnitz Theorem (A.3) and three dimensional version (A.9) are comparable. In the 1D case there was no surface integral to evaluate, we simply consider the function at its endpoints. The terms $\frac{db}{d\alpha}$ & $\frac{da}{d\alpha}$ are the velocities of the bounding surface and are equivalent to $\mathbf{w}$, whereas the terms $\gamma(b(\alpha))$ & $\gamma(a(\alpha))$ are equivalent to the γ evaluated on the surface.

Appendix B

Stokes flow solution for a finite domain

We use standard procedure for solving linear homogeneous equations, to obtain the solution of the differential equation (4.3), rewritten for ease below:

$$\underbrace{(a_x u_{,x})}_{convection} - \underbrace{(\mu u_{,x})_{,x}}_{diffusion} - q_x = 0 \qquad \forall x \in \Omega \tag{B.1}$$

Equ (B.1), can be rewritten in the form of a simple quadratic equation, where u is replaced by r:

$$a_x r - \mu r^2 = 0 \quad \Rightarrow \quad r_1 = 0 \quad r_2 = \frac{a_x}{\mu} \tag{B.2}$$

Therefore the solution of $u(x)$ can be written as:

$$u(x) = C_1 e^{r_1 x} + C_2 e^{r_2 x} \tag{B.3}$$

where C_1 and C_2 are constants. If the solutions of (B.2) are substituted into (B.3) we arrive at:

$$u(x) = C_1 e^{0x} + C_2 e^{\frac{a_x}{\mu} x} \tag{B.4}$$

The boundary conditions are given as:

$$u(0) = u_0 \quad \rightarrow \quad u_0 = C_1 + C_2 \tag{B.5a}$$

$$u(L) = u_L \quad \rightarrow \quad u_L = C_1 + C_2 e^{2Pe} \tag{B.5b}$$

$$u(x) = u \quad \rightarrow \quad u = C_1 + C_2 e^{2\frac{Pe}{L} x} \tag{B.5c}$$

Rearranging (B.5a):

$$C_1 = u_0 - C_2 \tag{B.6}$$

which can be substituted separately into (B.5b) and (B.5c) resulting in the following equations for C_2:

$$C_2 = \frac{u_L - u_0}{e^{2Pe} - 1} \qquad \text{and} \qquad C_2 = \frac{u - u_0}{e^{2\frac{Pe}{L}x} - 1} \tag{B.7}$$

which can be equated to give the following equation:

$$\frac{u - u_0}{u_L - u_0} = \frac{e^{2\frac{Pe}{L}x} - 1}{e^{2Pe} - 1} \tag{B.8}$$

Bibliography

[1] M.B. Abbott and D.R. Brasco. *Computational Fluid Dynamics.* Longman Scientific & Technical, Harlow, 1989.

[2] R.A. Adams. *Sobolev Spaces. 65 of Pure and Applied Mathematics.* Academic Press, New York, 1975.

[3] A.N. Alexandrou, T.M. McGilvreay, and G. Burgos. Steady herschel-bulkley fluid flow in three-dimensional expansions. *Journal of Non-Newtonian Fluid Mechanics*, 100:77–96, 2001.

[4] S. Aliabadi and T.E. Tezduyar. Stabilized-finite-element/interface-capturing technique for parallel computation of unsteady flows with interfaces. *Computer Methods in Applied Mechanics and Engineering*, 190:243–261, 2000.

[5] I. Babuška and J.M. Melenk. The partition of unity method. *International Journal for Numerical Methods in Engineering*, 40:727–758, 1997.

[6] K-J. Bathe. *Finite Element Procedures.* Prentice-Hall, New Jersey, USA, 1996.

[7] M. Behr, D. Hastreiter, S. Mittal, and T.E. Tezduyar. Incompressible flow past a circular cylinder: dependence of the computed flow fields on the location of the lateral boundaries. *Computer Methods in Applied Mechanics and Engineering*, 123:309–316, 1995.

[8] M. Behr, J. Liou, R. Sith, and T.E. Tezduyar. Vorticity-stream function formulation of unsteady incompressible flow past a cylinder: Sensitivity of the computed flow field to the location of the outflow boundary. *International Journal for Numerical Methods in Fluids*, 12:323–342, 1991.

[9] T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45:601–620, 1999.

[10] T. Belytschko, N. Mos, S. Usui, and C. Parimi. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering*, 50:993–1013, 2001.

[11] M.S. Bloor. The transition to turbulence in the wake of a circular cylinder. *Journal of Fluid Mechanics*, 19:290, 1964.

[12] G. Bosch and W. Rodi. Simulation of vortex shedding past a square cylinder near a wall. *International Journal of Heat and Fluid Flow*, 17:267–275, 1996.

[13] M. Breuer, J. Bernsdorf, T. Zeiser, and F. Durst. Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-boltzmann and finite volume. *International Journal of Heat and Fluid Flow*, 21:186–196, 2000.

[14] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *RAIRO Anal. Numer.*, 8(R2):129–151, 1974.

[15] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods.* Springer, New York, 1991.

[16] A.N. Brooks and T.J.R. Hughes. Streamline-upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[17] J. Chessa and T. Belytschko. An enriched finite element method and level sets for axisymmetric two phase flow with surface tension. *International Journal for Numerical Methods in Engineering*, 58:2041–2064, 2003.

[18] J. Chessa, P. Smolinski, and T. Belytschko. The extended finite element method (x-fem) for solidification problems. *International Journal for Numerical Methods in Engineering*, 53:1959–1977, 2002.

[19] J. Chessa, H. Wang, and T. Belytschko. On the construction of blending elements for local partition of unity enriched finite elements. *International Journal for Numerical Methods in Engineering*, 57:1015–1038, 2003.

[20] A.J. Chorin and J.E. Marsden. *A Mathematical Introduction to Fluid Mechanics.*

[21] J. Chung and G.M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method. *Computer Methods in Applied Mechanics and Engineering*, 60:371–375, 1993.

[22] R. Codina. Comparison of some finite element methods for solving the convection diffusion reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 156:185–210, 1998.

[23] M.S. Cramer. *Navier Stokes equations.* http://www.Navier-Stokes.net, Cambridge, 2002.

[24] I.G. Currie. *Fundamentals of Fluid Mechanics.* McGraw-Hill, 1974.

[25] R.W. Davis and E.F. Moore. A numerical study of vortex shedding from rectangles. *Journal of Fluid Mechanics*, 116:475–506, 1982.

[26] R.W. Davis, E.F. Moore, and L.P. Purtell. A numerical-experimental study of confined flow around rectangular cylinders. *Phys. Fluids*, 27:307–323, 1984.

[27] W. Dettmer and D. Peric. An analysis of the time integration algorithms for the finite element solutions of incompressible navier-stokes equations based on a stabilized formulation. *Computer Methods in Applied Mechanics and Engineering*, 192:1177–1226, 2003.

[28] J. Donea and A. Huerta. *Finite Element Methods of Flow Problems.* John Wiley, Chichester, 2003.

[29] J.-J. Droux and T.J.R. Hughes. A boundary integral modification of the galerkin/least squares formulation for the stokes problem. *Computer Methods in Applied Mechanics and Engineering*, 113:173–182, 1994.

[30] D.F. Durao, P.S.T. Gouveia, and J.C.F Periera. Velocity characteristics of the flow around a square cross section cylinder placed near a channel wall. *Exp. Fluids*, 11:298–304, 1991.

[31] J.H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics.* Springer, New York.

[32] B.A. Finlayson. *The Method of Weighted Residuals and Variational Principles.* Academic Press, 1972.

[33] R. Franke, W. Rodi, and B. Schonung. Numerical calculation of laminar vortex shedding past cylinders. *J. Wind End. Ind. Aerodyn.*, 35:237–257, 1990.

[34] A.C. Galeo and E. Gomes Dutra do Carmo. A consistent approximate upwind petrov-galerkin method for convection dominated problems. *Computer Methods in Applied Mechanics and Engineering*, 68:83–95, 1988.

[35] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes equations, theory and algorithms.* Springer-Verlag, Berlin, 1986.

[36] P.M. Gresho and R.L. Sani. *Incompressible Flows and the Finite Element Method.* John Wiley, Chichester, 1998.

[37] S.K. Hannani, M. Stanislas, and P Dupont. Incompressible navier-stokes computations with supg and gls formulations - a comparison study. *Computer Methods in Applied Mechanics and Engineering*, 124:153–170, 1995.

[38] J.C. Heinrich and O.C. Zienkiewicz. The finite element method and "upwinding" techniques in the numerical solution of convection dominated flow problems. *American Society of Mechanical Engineers*, , in T.J.R. Hughes, ed., Finite Element Methods for Convection dominated flows, AMD - vol 34:105–136, 1979.

[39] J.G. Hirsch. *Numerical computation of internal and external flows. Vol. 2: Computational Methods for inviscid and viscous flows*, volume Wiley series in numerical Methods in Engineering. John Wiley, Chichester, 1990.

[40] C.W. Hirt and B.D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *J. Comput. Phys.*, 39:201–225, 1981.

[41] A. Huerta and W.K. Liu. Viscous flow with large free surface motion. *Computer Methods in Applied Mechanics and Engineering*, 69:277–324, 1988.

[42] T.J.R. Hughes. *The Finite Element Method.* Prentice-Hall, New Jersey, 1987.

[43] T.J.R. Hughes and A. Brooks. A multi-dimensional upwind scheme with no cross wind diffusion, in finite elements for convection dominated flows. *American Society of Mechanical Engineers*, AMD 34:19–35, 1979.

[44] T.J.R. Hughes and A.N. Brooks. A theoretician framework for petrov-galerkin methods with discontinuous weight functions, in finite elements in fluids. *Finite Elements in Fluids*, 4:47–65, 1982.

[45] T.J.R. Hughes, L.P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. circumventing the ***b***condition: a stable petrov-galerkin formulation of the stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59:85–99, 1986.

[46] T.J.R. Hughes, L.P. Franca, and G.M. Hulbert. A new finite element formulation for computational fluid dynamics: Viii. the galerkin/least squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering*, 73:173–189, 1989.

[47] T.J.R. Hughes, M. Mallet, and A. Mizukami. A new finite element formulation for computational fluid dynamics: II. beyond supg. *Computer Methods in Applied Mechanics and Engineering*, 54:341–355, 1986.

[48] B.M Irons. Engineering application of numerical integration in stiffness method. *Journal of American Institute of Aeronautics and Astronautics*, 14:2035–2037, 1966.

[49] K.E. Jansen, S.S. Collis, and F. Shakib. A better consistency for low-order stabilized finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 174:153–170, 1999.

[50] K.E. Jansen, C.H. Whiting, and G.M. Hulbert. A generalized-α method for integrating the filtered navierstokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:305–319, 2000.

[51] C. Johnson and J. Saranen. Streamline diffusion methods for the incompressible euler and navier-stokes equations. *Math. Comput.*, 47:1–18, 1984.

[52] H. Lamb. *Hydrodynamics.* Cambridge University Press, 1993.

[53] L.D. Landau and E.M. Lifschitz. *Fluid Mechanics.* Pergamon Press, 1959.

[54] C. Lei, L. Cheng, and K. Kavanagh. A finite difference solution of the shear flow over a circular cylinder. *Ocean Engineering*, 27:271–290, 2000.

[55] G. Li and J.A.C. Humphrey. Numerical modelling of confined flow past cylinder of square cross section at various orientations. *International Journal for Numerical Methods in Fluids*, 20:1215–1236, 1995.

[56] J.M. MacFadden. Computation modelling of the steady and unsteady advection diffusion based on the finite element method. Master's thesis, Swansea University, Swansea, 2000.

[57] R. J. Martinuzzi, S. C. C. Bailey, and G. A. Kopp. Influence of wall proximty on vortex shedding from a square cylinder. *Experiments on Fluids*, 34:585–596, 2003.

[58] R.E. Meyer. *Introduction to Mathematical Fluid Dynamics.* John Wiley, Dover, 1982.

[59] S.C. Mikhlin. *Variational Methods in Mathematical Physics.* Macmillan Publishing Company, 1964.

[60] N. Möes, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46:131–150, 1999.

[61] A. Okijima. Strouhal numbers of flow around rectangular cylinders. *Journal of Fluid Mechanics*, 123:379–398, 1982.

[62] J.M. Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables.*, volume New York. Society for Industrial and Applied Mathematics (SIAM)., 2000.

[63] C.W. Oseen. Beitrage zur hydrodynamik i. *Ann. der. Phys*, 46(2):231–252, 1915.

[64] R. Peyret and T.D. Taylor. *Computational Methods for Fluid flow.* Springer, New York, 1983.

[65] A. Roshko. PhD thesis, 1952.

[66] A. Roshko. On the development of turbulent wakes from vortex streets. *NACA report*, 1191, 1954.

[67] J. Serrin. *Mathematical Principles of Classical Fluid Mechanics.* Springer-Verlag, 1959.

[68] F. Shakib and T.J.R. Hughes. A new finite element formulation for computational fluid dynamics: IX. fourier analysis of space-time galerkin/least-squares algorithms. *Computer Methods in Applied Mechanics and Engineering*, 87:35–58, 1991.

[69] F. Shakib, T.J.R. Hughes, and Z. Johan. A new finite element formulation for computational fluid dynamics: X. the compressible euler and navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 89:141–219, 1991.

[70] S. Slijepcevic. *Computational modelling of non-Newtonian fluids based on the stabilized Finite Element Method.* PhD thesis, Swansea University, 2002.

[71] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer, New York, 1980.

[72] M. Stolarska, D.L. Chopp, N. Moés, and T. Belytschko. Modelling crack growth by level sets in the extended finite element method. *International Journal for Numerical Methods in Engineering*, 51:943–960, 2001.

[73] N. Sukumar, D.L. Chopp, N. Moes, and T. Belyschko. Modelling hole and inclusions by level sets in the extended finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:6183–6200, 2000.

[74] N. Sukumar, N. Moes, B. Moran, and T. Belyschko. extended finite element method for three-dimensional crack modelling. *International Journal for Numerical Methods in Engineering*, 48:1549–1570, 2000.

[75] I.C. Taig. *Structural Analysis by the matrix Displacement Method.* English Electric Aviation Report, 1961.

[76] C. Taylor and P. Hood. A numerical solution of the navier-stokes equations using the fem technique. *Comput. & Fluids I*, pages 73–100, 1973.

[77] T.E. Tezduyar. Stabilized finite element formulations for incompressible flow computations. *Adv. Appl. Mech*, 28:1–44, 1992.

[78] T.E. Tezduyar, M. Behr, and J. Liou. New strategy for finite element computations involving moving boundaries and interfaces. the deforming-spatial-domain/space-time procedure. i. the concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*, 94:339–351, 1992.

[79] T.E. Tezduyar, M. Behr, and J. Liou. A new strategy for finite element flow computations involving moving boundaries and interfaces-the deforming-spatial-domain/space-time procedures: I. the concept and preliminary tests. *Computer Methods in Applied Mechanics and Engineering*, 94:339–353, 1992.

[80] T.E. Tezduyar, M. Behr, and J. Liou. A new strategy for finite element flow computations involving moving boundaries and interfaces-the deforming-spatial-domain/space-time procedures: II. computation of free surface flows, two liquid flows and flows with drifting cylinders. *Computer Methods in Applied Mechanics and Engineering*, 94:359–371, 1992.

[81] T.E. Tezduyar and D.K. Ganjoo. Petrov-galerkin formulations with weighting functions dependent upon spatial and temporal discretization: applications to transient convection diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 59:49–71, 1986.

[82] T.E. Tezduyar, S. Mittal, S.E. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal-order interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, 95:221–242, 1992.

[83] V. Thomée. *Galerkin Finite Element Methods for Parabolic Problems.* Springer, Berlin, Heidelberg, 1997.

[84] S. Turki, H. Abbassi, and S.B. Nasrallah. Effect of the blockage ratio on the flow in a channel with a built-in square cylinder. *Computational Mechanics*, 33:22–29, 2003.

[85] G.J. Wagner, S. Ghosal, and W.K. Liu. Particulate flow simulations using lubrication theory solution enrichment. *International Journal for Numerical Methods in Engineering*, 56:1261–1289, 2003.

[86] G.J. Wagner, N. Mos, W.K. Liu, and T. Belytschko. The extended finite element method for rigid particles in stokes flow. *International Journal for Numerical Methods in Engineering*, 51:293–313, 2001.

[87] C.H.K. Williamson. Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics*, 28:477–539, 1996.

[88] O.C. Zienkiewicz, R.H. Gallagher, and P. Hood. Newtonian and non newtonian viscous incompressible flow. temperature induced flows. *Finite Element*

Solution, in J. R. Whiteman. ed. The mathematics of Finite Elements and Applications II, MAFELAP 1975 Academic Press, 1975.

[89] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method: The Basis.* Butterworth-Heinemann, Oxford, 2000.