APPLICATION OF FULLY IMPLICIT COUPLED METHOD FOR 2D INCOMPRESSIBLE FLOWS ON UNSTRUCTURED GRIDS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ŞEYDA ZENGİN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTATIONAL MECHANICS

SEPTEMBER 2012

Approval of the thesis:

APPLICATION OF FULLY IMPLICIT COUPLED METHOD FOR 2D INCOMPRESSIBLE FLOWS ON UNSTRUCTURED GRIDS

submitted by **ŞEYDA ZENGİN** in partial fulfilment of the requirements for the degree of **Master of Science in Department of Engineering Science**, **Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Director, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Murat Dicleli _____
Head of Department, **Engineering Sciences**

Assoc. Prof. Dr. Işık Hakan Tarman _____
Supervisor, **Department of Engineering Sciences, METU**

Examining Committee Members:

Prof. Dr. Turgut Tokdemir _____
Department of Engineering Sciences, METU

Assoc. Prof. Dr. Işık Hakan Tarman _____
Department of Engineering Sciences, METU

Prof. Dr. Münevver Tezer _____
Department of Mathematics, METU

Prof. Dr. Gülin Ayşe Birlik _____
Department of Engineering Sciences, METU

Assoc. Prof. Dr. Utku Kanoğlu _____
Department of Engineering Sciences, METU

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: ŞEYDA ZENGİN

Signature :

ABSTRACT

APPLICATION OF FULLY IMPLICIT COUPLED METHOD FOR 2D INCOMPRESSIBLE FLOWS ON UNSTRUCTURED GRIDS

Zengin, Şeyda

M.Sc., Department of Engineering Sciences

Supervisor: Assoc.Prof. Dr. Işık Hakan Tarman

September 2012, 59 pages

In the subject of Computational Fluid Dynamics (CFD), there seems to be small number of important progress in the pressure-based methods for several decades. Recent studies on the implicit coupled algorithms for pressure-based methods have brought a new insight. This method seems to provide a huge reduction in the solution times over segregated methods.

Fully implicit coupled algorithm for pressure-based methods is very new subject with only few papers in literature. One of the most important work in this area is referenced as [1] in this thesis. Another source of information about the method comes from a commercially available code FLUENT which includes the algorithm as an option for pressure-based solver. However the algorithm in FLUENT does not seem to be a fully implicit with a little information in its manual.

In this thesis, a fully implicit coupled pressure-based solver is developed mainly based on the available literature. The developed code is succesfully tested against some test cases.

Keywords: CFD, Pressure-based coupled solver, Fully implicit solver, Incompressible flow.

ÖZ

İKİ BOYUTLU SIKIŞTIRILAMAZ AKIŞLARDA TÜM ÖRTÜK AKUPLE YÖNTEMİN YAPISIZ AĞ KULLANILARAK UYGULAMASI

Zengin, Şeyda
Yüksek Lisans, Mühendislik Bilimleri Bölümü
Tez Yöneticisi: Assist. Prof. Dr. Işık Hakan Tarman

Eylül 2012, 59 Sayfa

Hesaplamalı Akışkanlar Dinamiği (HAD) alanında, basınç temelli yöntemler üzerinde son yirmi ile otuz senedir önemli gelişmeler gözlenmemiştir. Basınç temelli yöntemler için tam örtük akuple çözücüler üzerine yeni çalışmaların gerçekleştirilmesi bu çalışma alanına yeni bir bakış açısı getirmiştir. Bu metod sıralı ayrık çözücü metodlara göre çözüm zamanını önemli ölçüde kısaltma avantajı sağlamaktadır.

Basınç temelli yöntemler için tam örtük akuple çözücüler, literatürde çok yeni ve bir kaç çalışmanın yer aldığı bir alandır. En önemli çalışmalardan birini de bu tezde referans [1] olarak verilen yayın sağlamaktadır. Diğer bir kaynak ise ticari bir yazılım olan FLUENT sağlamaktadır. Fakat FLUENT kullandığı algoritmayı detaylı belirtmemekle birlikte, bu yazılımdaki yöntem literatürde sözü edilen başarımı sağlayamamaktadır.

Bu tez çalışmasında açık literatürde bulunan kaynaklar kullanılarak tam örtük akupla basınç temelli yönteme dayanan bir yazılım geliştirilerek bazı test durumları başarıyla sınanmıştır.

Anahtar Kelimeler: HAD, Basınç temelli akuple çözücü, Tüm örtük çözücü, Sıkıştırılamaz akış.

Dedicated to my family

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost, I would like to thank my supervisor Assoc. Prof. Dr. Işık Hakan Tarman for his guidance support, encouragement, indulge and patience throughout the study.

I also would like to thank;

Bora Kalpaklı for his technical support and endless moral support that he gave.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 Aim of the Study

The aim of this thesis is to develop a Computational Fluid Dynamics (CFD) code which is based on a fully implicit pressure-based coupled method for the solution of steady two dimensional laminar incompressible flow problems on unstructured grids. The implicit pressure-based coupled algorithm is developed by assembling the momentum and continuity equations for velocity and mass and then using the Rhie–Chow interpolation to derive a pressure equation. The coefficients of the momentum and pressure equations are assembled into one diagonally dominant matrix. The equations are solved simultaneously for the convergence of the equations with an ILU preconditioned GMRES method with false transient time stepping. The algorithm used in this thesis is based on a recent paper referenced in [1]. The code is tested on some test cases and compared with a commercial CFD solver, FLUENT.

1.2 Review of Relevant Works

Computational Fluid Dynamics uses numerical methods and algorithms for analyzing and solving the problems that involve fluid flow. Large number of work on CFD have given rise to the developments of various algorithms and these work have established reliable and robust simulation tools of fluid flow processes.

In the growing usage of CFD during the past few years, several issues are addressed. The concerns releated to the accuracy are met by the development of the High Resolution (HR) schemes [27, 28, 29]. One of the main concerns in CFD world is the computational cost of the algorithms and solvers. Better solution algorithms [30, 31, 32], solvers [33, 34], and multigrid techniques [35, 36, 37, 38] have lowered the computational cost and brought the feasible solutions to real life fluid flow problems. Over the past decades the work on pressure-

based algorithms [4, 6, 7, 12, 30, 31, 32, 36], extended the techniques to solve flow problems in various Reynolds and Mach number regimes using both structured and unstructured grid methods. The density-based algorithms have provided the means for getting more robust and higher performance.

The coupled solvers [6, 7] delivers lower run times and better copes with the convergence problems experienced by segregated solvers when used with dense computational meshes [8]. In the segregated approach, each equation is solved separately using the results obtained in the previously steps in solving the equations. On the other hand, the coupled approach discretizes and solves the conservation equations as one system.

The pressure-based algorithms, which are also used in this thesis, originated first in the work of Harlow and Welch [9] and Chorin [10]. The widely known segregated SIMPLE algorithm (semi-implicit method for pressure linked equations) for incompressible flows was developed based on a new trend immediately after [11] followed by SIMPLEST [40], SIMPLEC [30], SIMPLEM [41], PISO [39], PRIME [42], and SIMPLEX [32] algorithms, to cite a few. The velocity field is obtained by numerically solving the momentum equations and the pressure field is extracted by solving a pressure correction equation which is obtained by manipulating continuity and momentum equations in the pressure-based approach.

In this thesis, collocated grid approach is used to handle the velocity and pressure data in which pressure and velocity values are computed at the same grid point. Simple averaging of cell velocities in the collocated grid approach leads to unphysical checker-boarding pattern of the pressure values. This problem is a result of storing the pressure and velocity values in the same location on the solution grid. In this study, Rhie-Chow Interpolation of the face velocities is used to avoid pressure checker-boarding [16].

The resulting linear system of equations from unstructured finite volume discretization is a diagonally dominant non-symmetric sparse matrix system. This system is solved by using an Incomplete Lower Upper (ILU) pre-conditioned Generalized Minimal Residual Method (GMRES) solver.

In this thesis, cell centred Finite Volume (FV) method is used for discretization. The FV method has been extended by a number of authors into forms which allow the discretization of the conservation equations over unstructured meshes. Chow [12, 13] employed the FV

method, using polygonal element types, triangles through to octagons, to solve simple flow and heat transfer problems. Thomadakis [14] used a staggered mesh approach, solving the velocity components at the element centres and the pressures at the element vertices, to simulate low Reynolds number flow problems. Pan et al. [15] used an unstructured mesh consisting of triangular elements to solve a variety of laminar flow problems.

1.3 Outline of the Thesis

The study is documented in five main chapters and an appendix. An outline of the thesis is briefly given below:

Chapter 1: Introduction chapter that is including aim of the study and review of the relevant work.
Chapter 2: Theoretical background on which the thesis is based on.
Chapter 3: Solution procedure; methods, approaches used in the study.
Chapter 4: Test cases
Chapter 5: A brief summary, and conclusions.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 The Navier-Stokes Equations

The Navier-Stokes equations are a set of nonlinear partial differential equations that govern the evolution of the velocity, temperature and the density of the moving fluid. This set of equations mainly describes the flow of fluids. The Navier-Stokes equations include conservation of mass as continuity equation, time dependent conservation of momentum with three time-dependent equations, and a conservation of energy equation [2], [3]:

Continuity:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} + \frac{\partial (\rho w)}{\partial z} = 0 \tag{1}$$

X Momentum:

$$\frac{\partial (\rho u)}{\partial t} + \frac{\partial (\rho u^2)}{\partial x} + \frac{\partial (\rho uv)}{\partial y} + \frac{\partial (\rho uw)}{\partial z} = -\frac{\partial p}{\partial x} + \left[\frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} \right] + \rho g_x \tag{2}$$

Y Momentum:

$$\frac{\partial (\rho v)}{\partial t} + \frac{\partial (\rho uv)}{\partial x} + \frac{\partial (\rho v^2)}{\partial y} + \frac{\partial (\rho vw)}{\partial z} = -\frac{\partial p}{\partial y} + \left[\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} \right] + \rho g_y \tag{3}$$

Z Momentum:

$$\frac{\partial (\rho w)}{\partial t} + \frac{\partial (\rho uw)}{\partial x} + \frac{\partial (\rho vw)}{\partial y} + \frac{\partial (\rho w^2)}{\partial z} = -\frac{\partial p}{\partial z} + \left[\frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} \right] + \rho g_z \tag{4}$$

Energy:

$$\frac{\partial (\rho E_T)}{\partial t} + \frac{\partial (\rho u E_T)}{\partial x} + \frac{\partial (\rho v E_T)}{\partial y} + \frac{\partial (\rho w E_T)}{\partial z} = \tag{5}$$

$$-\frac{\partial(up)}{\partial x}-\frac{\partial(vp)}{\partial y}-\frac{\partial(wp)}{\partial z}+\left[\frac{\partial\left(\kappa\partial \mathrm{T}/\partial x\right)}{\partial x}+\frac{\partial\left(\kappa\partial \mathrm{T}/\partial y\right)}{\partial y}+\frac{\partial\left(\kappa\partial \mathrm{T}/\partial z\right)}{\partial z}\right]+$$
$$+\left[\frac{\partial}{\partial x}(u\tau_{xx}+\mathrm{v}\tau_{xy}+\mathrm{w}\tau_{xz})+\frac{\partial}{\partial y}(u\tau_{yx}+\mathrm{v}\tau_{yy}+\mathrm{w}\tau_{yz})+\frac{\partial}{\partial z}(u\tau_{zx}+\mathrm{v}\tau_{zy}+\mathrm{w}\tau_{zz})\right]$$

where τ , E_T and κ are stress, total energy and thermal conductivities respectively.

The independent variables in the Navier-Stokes equations are; x, y, and z spatial coordinates of some domain, and time t. The six dependent variables are listed as; the pressure p, density ρ, temperature T (which is contained in the energy equation) and three components of the velocity vector; the u component is in the x direction, the v component is in the y direction, and the w component is in the z direction. The ideal gas law, p V = n R T (as an equation of state), the six equations can be used to determine the six dependent variables [2], [3].

The thesis deals with steady, laminar incompressible Newtonian fluid flow which means equations are time independent and density values are constant. The time derivative terms become zero in Eqns. (6) through (10) for a steady flow. The Navier-Stokes equations then take the form;

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{6}$$

$$\frac{\partial(u^2)}{\partial x}+\frac{\partial(uv)}{\partial y}+\frac{\partial(uw)}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\frac{\mu}{\rho}\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right]+g_x \tag{7}$$

$$\frac{\partial(uv)}{\partial x}+\frac{\partial(v^2)}{\partial y}+\frac{\partial(vw)}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial y}+\frac{\mu}{\rho}\left[\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right]+g_y \tag{8}$$

$$\frac{\partial(uw)}{\partial x}+\frac{\partial(vw)}{\partial y}+\frac{\partial(w^2)}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial z}+\frac{\mu}{\rho}\left[\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right]+g_z \tag{9}$$

$$\frac{\partial(uE_T)}{\partial x}+\frac{\partial(vE_T)}{\partial y}+\frac{\partial(wE_T)}{\partial z}= \tag{10}$$

$$+\frac{1}{\rho}\left[\frac{\partial\left(\kappa\partial \mathrm{T}/\partial x\right)}{\partial x}+\frac{\partial\left(\kappa\partial \mathrm{T}/\partial y\right)}{\partial y}+\frac{\partial\left(\kappa\partial \mathrm{T}/\partial z\right)}{\partial z}\right]+\frac{2\mu}{\rho}\left[\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2+\left(\frac{\partial w}{\partial z}\right)^2\right]$$

$$+\frac{\mu}{\rho}\left[\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)^2+\left(\frac{\partial w}{\partial y}+\frac{\partial v}{\partial z}\right)^2+\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)^2\right]$$

A Newtonian fluid has a linear relation between the applied shear stress and velocity gradient which is defined in Eqn. (11) for an incompressible fluid;

$$\tau_{xy}=\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) \tag{11}$$

where μ the viscosity of the fluid, τ_{xy} can be interpreted as viscous flux of x-momentum in y-direction. The other shear stress components can be written similarly. Viscosity may depend on pressure and temperature for Newtonian fluids.

2.2 Finite Volume Method

The finite volume method is a numerical method which calculates values of the dependent variables averaged across the volume for solving partial differential equations. Volume integrals of the partial differential equation, which contain divergence terms, are converted to surface integrals, using the divergence theorem.

The finite volume method is generally preferred in CFD solvers for successes of solving the problems over unstructured grids those have geometrical features of arbitrary complexity and steep orography. The comparison between the finite volume method and the finite difference method is mainly resulted in favour of the finite volume method which can be used over both structured and unstructured meshes. Finite volume methods are also successful at calculations for moving mesh applications such as for tracking the shocks. It carries the conservation of physical laws in the continuous problem to the discretized problem. This is especially important in the numerical solution of diffusion-convection equations. Finite volume method stores the conserved variables values within the volume element instead of at nodes or surfaces and this allows applying boundary conditions noninvasively directly on the control

volume surfaces [4], [5]. On the other hand, it does not allow higher-order approaches, unlike finite element method that has so-called p-version. The mathematical analysis, such as stability and convergence, is generally difficult in comparison.

2.2.1 Finite Volume Discretization

The conservation equations governing steady, laminar incompressible Newtonian fluid flow are given [1]:

$$\nabla \cdot (\rho \mathbf{v}) = 0 \tag{12}$$

$$\nabla \cdot (\rho \mathbf{v}\mathbf{v}) = \nabla \cdot (\mu \nabla \mathbf{v}) - \nabla \cdot (p\mathbf{I}) \tag{13}$$

where $\mathbf{v}$ is the velocity vector $\mathbf{v} = [u, v, w]^T$, p is the pressure and $\mathbf{I}$ is the identity matrix in Eqns. (12) and (13).

These equations can be expressed in the general conservative form as in Eqn. (14).

$$\nabla \cdot (\rho \mathbf{v}\phi) = \nabla \cdot (\Gamma \nabla \phi) + \mathrm{Q} \tag{14}$$

where, ϕ is the transported quantity Γ is the diffusion coefficient and Q is the general source term in the equation.

In the Finite Volume discretization, the quantity $\phi = \phi(\mathbf{x})$ is approximated to have linear variation within the control volume

$$\phi(\mathbf{x}) \approx \phi_P + (\mathbf{x} - \mathbf{x}_P) \cdot (\nabla \phi)_P \tag{15}$$

around the nodal point P with coordinate $\mathbf{x}_P$, which amounts to second-order accuracy.

Integrating the general transport equation over the control volume displayed in Figure 1 (a) and transforming the volume integrals of the diffusion and convection terms into surface

integrals through the use of the divergence theorem, the semi-discretized form of the governing conservation equation is obtained as

$$\oint_{\partial\Omega}(\rho\mathbf{v}\phi)\cdot d\mathbf{S}=\oint_{\partial\Omega}(\Gamma\nabla\phi)\cdot d\mathbf{S}+\iint_{\Omega}Q\,d\Omega. \tag{16}$$

Thus, the finite volume method takes evaluation of volume and surface integrals over the control volume Ω and its surface $\partial\Omega$. The assumed linear variation of ϕ leads to the following volume integral:

$$\begin{aligned}\int_{\Omega}\phi(\mathbf{x})\,d\Omega &\approx \int_{\Omega}\left[\phi_P+(\mathbf{x}-\mathbf{x}_P)\cdot(\nabla\phi)_P\right]d\Omega\\ &=\phi_P\Omega_P+\left[\int_{\Omega}(\mathbf{x}-\mathbf{x}_P)\,d\Omega\right]\cdot(\nabla\phi)_P\\ &=\phi_P\Omega_P\end{aligned} \tag{17}$$

where Ω_P is the volume of the control volume and $\mathbf{x}_P$ is taken as the centroid of the volume thus the integral in the second line vanishes.

Similarly, assuming that $\partial\Omega$ is composed of flat faces and Φ is a vector quantity assumed to have linear variation over the faces, the surface integral thus becomes as defined in Eq. (18):

$$\int_{\partial\Omega}\mathbf{\Phi}\cdot d\mathbf{S}\approx\sum_{f=nb(P)}\mathbf{\Phi}_f\cdot\mathbf{S}_f+\left[\int_{\Omega}(\mathbf{x}-\mathbf{x}_f)\,d\mathbf{S}\right]:(\nabla\mathbf{\Phi})_f=\sum_{f=nb(P)}\mathbf{\Phi}_f\cdot\mathbf{S}_f \tag{18}$$

where $f=nb(P)$ refers to values at the faces obtained by interpolating between grid point P and its neighbour, and the face area vector S_f points outward having the magnitude of the face area. The second integral vanishes due to the consideration that $\mathbf{x}_f$ lies at the center of the face *f*.

In this light, the evaluation of the integrals in Eq. (10) using the second order integration scheme yields

$$\sum_{f=nb(P)}\left(\rho\mathbf{v}\phi-\Gamma\nabla\phi\right)_f\cdot\mathbf{S}_f=Q_p\Omega_p \tag{19}$$

Finally, the equation is expressed in algebraic form by representing the variables at the control volume faces in terms of nodal values. The resulting equation is written as

$$a_P^{\phi}\phi_P + \sum_{F=NB(P)} a_F^{\phi}\phi_F = b_P^{\phi} \tag{20}$$

where, $F = NB(P)$ refers to the neighbours of the grid point P, *a's* are the coefficients in the discretized equation for ϕ and *b* is the source term.

The above equation could equivalently be written as

$$\phi_P + \sum_{F=NB(P)} \frac{a_F^{\phi}}{a_P^{\phi}}\phi_F = \frac{b_P^{\phi}}{a_P^{\phi}} \quad \text{or} \quad \phi_P + \sum_{F=NB(P)} A_F^{\phi}\phi_F = B_P^{\phi} \tag{21}$$

while for the continuity equation the following discrete form is used:

$$\sum_{f=nb(P)} \dot{m}_f = 0 \quad \text{With} \quad \dot{m}_f = \rho \mathbf{v}_f \cdot \mathbf{S}_f . \tag{22}$$

For the momentum equation, the pressure gradient term is explicitly displayed as

$$\mathbf{V}_P + \sum_{F=NB(P)} A_F^V \mathbf{v}_F = \mathbf{B}_P^V - \mathbf{D}_P \nabla p_P \tag{23}$$

where

$$\mathbf{D}_p = \begin{bmatrix} \dfrac{\Omega_P}{a_P^u} & 0 \\ 0 & \dfrac{\Omega_P}{a_P^v} \end{bmatrix} \tag{24}$$

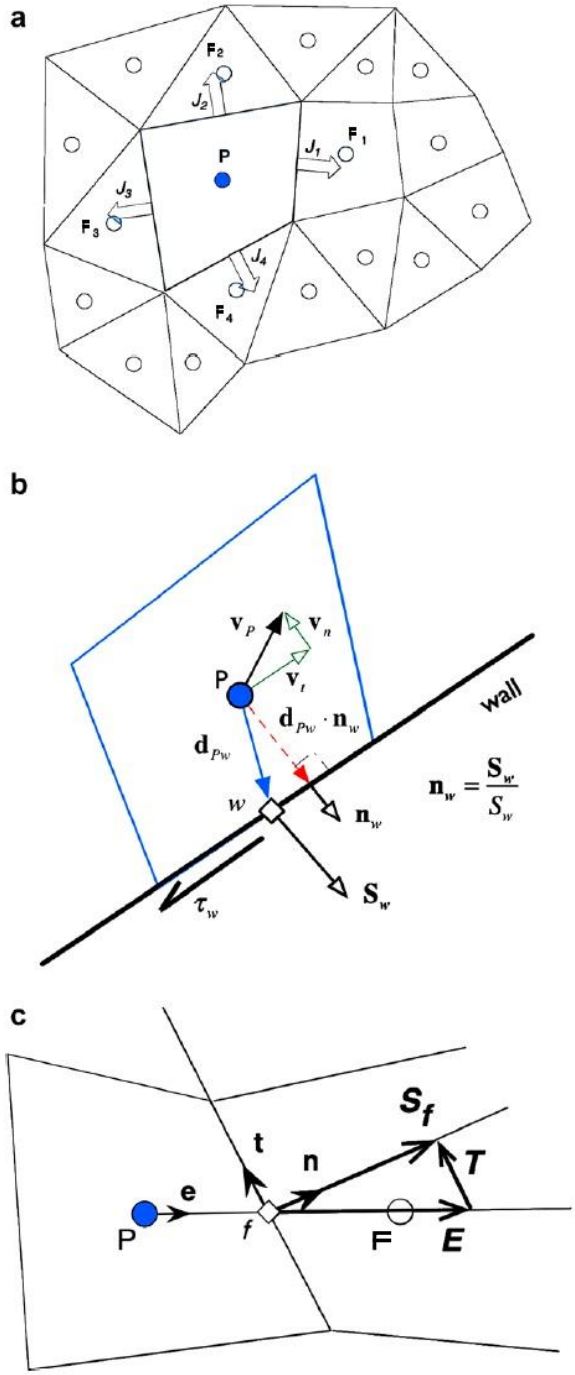

Figure 1: (a) Control volume with the nodal point P at the centroid and neighbouring points F, (b) normal velocity $\mathbf{v}_n$ and tangential velocity $\mathbf{v}_t$ components at a wall, and (c) decomposition of the surface into two components one aligned **E** with the grid and one normal **T** to the surface vector [5].

2.3 Discretization Methods for a Transported Scalar

Finite Volume discretization technique is briefly explained in this chapter. The technique is applied to an unstructured mesh by discretization of the general conservation equation. The method used in this thesis can also be applied to any equation describing the transport of a scalar quantity. Source, diffusion and convection terms in the conservation equation are detailed in the chapter. Rhie and Chow Interpolation method is used to calculate face velocities in the convection term discretization [16].

2.3.1 Discretization on an Unstructured Mesh

The conservation equation is discretized with an assumption of fully non skew mesh. However for non-simple geometry which cannot be meshed with structured meshes, there is always some degree skewness. Due to this skewness a cross diffusion correction is always needed in disretization of the diffusion terms, which will be explained later in the next section and in Section 3.1.

A non skew mesh (Fig. 2) is simply defined as;

- The line connecting the centroids of the adjacent elements is parallel to the normal of the face between the elements, termed orthogonality.
- The line connecting the centroids of the adjacent elements intersects the face between the elements at the face centroid, termed conjunctionality [17].

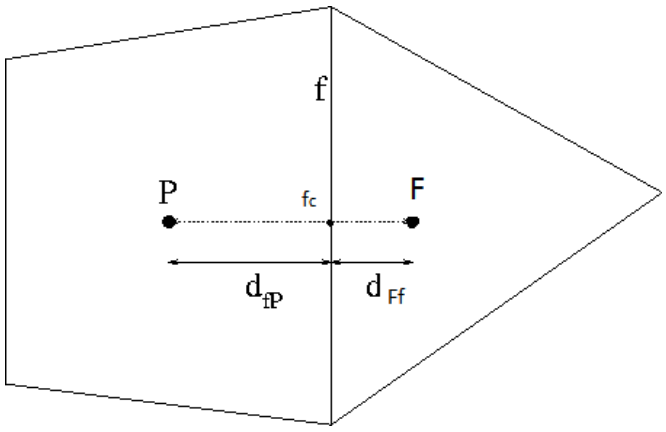

Figure 2: Non-Skew Mesh

The conservation equation, for a scalar quantity ϕ is written in the form [17];

$$\frac{\partial(\rho\phi)}{\partial t} + div(\rho \mathbf{v}\phi) = div(\Gamma_\phi grad(\phi)) + S_\phi \quad (25)$$

Transient + Convection = Diffusion + Source

and with the suitable approximations, Eqn. (25) can be expressed as a linear matrix equation in the form;

$$\mathbf{A}\Phi = \mathbf{b} \quad (26)$$

where Φ is the vector of the unknown values of ϕ at the nodal points. The technique is applied for discretization of the various conservation equations and it is based on the cell centred Finite - Volume formulation. The first step in solving the problem over the defined domain in this method is dividing the domain into a set of non overlapping control volumes. A unique single node is defined at the centre of the each volume elements.

The equation, which is in linear form, contains unknown values of the scalar quantity at the node in the control volume also in the neighbouring control volumes. These linear equations can be identified in a matrix equation of the form as in Eqn. (26) [17].

The conservation equation only contains convection and diffusion terms in this thesis as a result of the steady state and zero source conditions.

2.3.2 Diffusion Term

The discretization of the diffusion term starts with the use of the divergence theorem to convert the volume integral into the surface integral as shown in Eqn. (27) [17]

$$\int_V div(\mathbf{F})\, dV = \int_S \mathbf{F} \cdot \mathbf{n}\, dS. \quad (27)$$

Inserting the expression $\Gamma_\phi grad(\phi)$ in place of $\mathbf{F}$, Eqn. (27) can be written as in Eqn. (28),

$$\int_V div(\Gamma_\phi \, grad(\phi))\, dV = \int_S \Gamma_\phi \, grad(\phi) \cdot \mathbf{n}\, dS \tag{28}$$

where $\mathbf{n}$ is the unit outward normal to the surface.

Since polyhedral control volumes have a set of faces, the surface integral in Eqn. (28) can be written as a sum of surface integrals over each face bounding the control volume

$$\sum_f \int_f \Gamma \phi \frac{\partial \phi}{\partial n} dS. \tag{29}$$

If the grid is orthogonal as in Fig. 2, the derivative of ϕ in the normal direction to the face is approximated as Eqn. (30),

$$\left.\frac{\partial \phi}{\partial n}\right|_f \simeq \frac{\phi_F - \phi_P}{d_{FP}} \tag{30}$$

where d_{FP} is the distance between the centres of neighbouring elements F and P.

The discretized form of Eq. (29) becomes

$$\sum_f \Gamma_\phi S_f \frac{\phi_N - \phi_P}{d_{NP}} \tag{31}$$

where S_f is the area of face f.

The coefficient Γ_ϕ can be calculated on the face, by the means of an arithmetic mean:

$$\begin{aligned} (\Gamma_\phi)_f &= \alpha_f (\Gamma_\phi)_P + (1-\alpha_f)(\Gamma_\phi)_F, \\ \alpha_f &= \frac{d_{Ff}}{d_{Ff} + d_{Pf}}. \end{aligned} \tag{32}$$

The method has a drawback when $(\Gamma_\phi)_F$ is equal to zero. In Equation (32) the diffusion flux at the interface of the control volumes F and P includes an approximated value for Γ_ϕ

between the nodes which will not normally be expected to be 0. If $(\Gamma_\phi)_F$ is relatively much less than $(\Gamma_\phi)_P$ there will be relatively little resistance to the flux of ϕ between P and the interface in comparison to that between F and the face. In this case, it is expected that $(\Gamma_\phi)_f$ would depend on $(\Gamma_\phi)_F$ and inversely on α_f whereas equation (32) would lead to $(\Gamma_\phi)_f = \alpha_f(\Gamma_\phi)_P$. These drawbacks can be eliminated by a better model using the harmonic mean between the nodes for the variation of Γ_ϕ as follows:

$$(\Gamma_\phi)_f = \frac{(\Gamma_\phi)_F(\Gamma_\phi)_P}{\alpha_f(\Gamma_\phi)_P + (1-\alpha_f)(\Gamma_\phi)_F}. \tag{33}$$

This formula gives $(\Gamma_\phi)_f = 0$ if either $(\Gamma_\phi)_F$ or $(\Gamma_\phi)_P$ is zero and for $(\Gamma_\phi)_P \gg (\Gamma_\phi)_F$ $(\Gamma_\phi)_f \approx (\Gamma_\phi)_F / \alpha_f$ as required.

In the case of a skew mesh like in Figure 1, the face normal vector S_f comes with two components:

$$\begin{aligned} \mathrm{S}_f &= \mathrm{E} + \mathrm{T} \\ &= \frac{\mathrm{S}_f \cdot \mathrm{S}_f}{\mathrm{S}_f \cdot \mathrm{d}_{PF}} \mathrm{d}_{PF} + \left(\mathrm{S}_f - \frac{\mathrm{S}_f \cdot \mathrm{S}_f}{\mathrm{S}_f \cdot \mathrm{d}_{PF}} \mathrm{d}_{PF} \right) \end{aligned} \tag{34}$$

where $\mathbf{d}_{PF}$ is the vector from point P to F as shown in Figure 1, S_f is the surface normal vector which has a length equal to the area of the surface. Thus, a correction term should be added to the approximation in Eqn. (30) which is called cross diffusion term,

$$\frac{\partial \phi}{\partial n} \simeq \frac{\mathrm{S}_f \cdot \mathrm{S}_f}{\mathrm{S}_f \cdot \mathrm{d}_{PF}} (\phi_N - \phi_P) + \overline{\nabla \phi_f} \cdot \left(\mathrm{S}_f - \frac{\mathrm{S}_f \cdot \mathrm{S}_f}{\mathrm{S}_f \cdot \mathrm{d}_{PF}} \mathrm{d}_{PF} \right) \tag{35}$$

where $\overline{\nabla \phi_f}$ is the average of gradients at the cell centres P and F:

$$\overline{\nabla \phi_f} = \alpha_f (\nabla \phi)_P + (1-\alpha_f)(\nabla \phi)_F \tag{36}$$

with

$$(\nabla\phi)_P = \frac{1}{\Omega_P}\sum_f S_f\,\phi_f. \tag{37}$$

2.3.2.1 Convection Term

Divergence theorem is used to transform the volume integral of the convection term into surface integral as in diffusion term [17]

$$\int_V div(\rho\mathbf{u}\phi)\,dV = \int_S \rho(\mathbf{u}\cdot\mathbf{n})\phi\,dS. \tag{38}$$

The surface integral is split into a set of surface integrals over each of the faces bounding the control volume. The integrand values are estimated on the face to give:

$$\sum_f \rho_f(\mathbf{u}\cdot\mathbf{n})_f F_f\,\phi_f. \tag{39}$$

In Eqn. (39), the value of ρ_f is given by the value in the upwind element:

$$\begin{aligned} \rho_f &= \rho_P && if\ (\mathbf{u}\cdot\mathbf{n})_f > 0, \\ \rho_f &= \rho_F && if\ (\mathbf{u}\cdot\mathbf{n})_f < 0. \end{aligned} \tag{40}$$

The normal component of the velocity $(\mathbf{u}\cdot\mathbf{n})_f$ at the face is evaluated by the Rhie - Chow interpolation method [16] which will be explained in the next section. There is only one step left to complete the discretization of the convection term; the calculation of the face value of ϕ. One of the widely used methods is the arithmetic averaging;

$$\phi_f = \alpha_f\phi_P + (1-\alpha_f)\phi_F. \tag{41}$$

The discretized form of the convection term becomes (after the arithmetic averaging is introduced)

$$\sum_f \rho_f (\mathbf{u}\cdot\mathbf{n})_f N_f \left[\alpha_f \phi_P + (1-\alpha_f)\phi_F\right]. \tag{42}$$

This simple averaging of the transported scalar ϕ on the control volume faces leads to unbounded solutions and non-physical wiggles near steep solution gradients. In order to demonstrate this, an example solution to a simple one dimensional convection equation (43) is given in Figure 3 [21].

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{43}$$

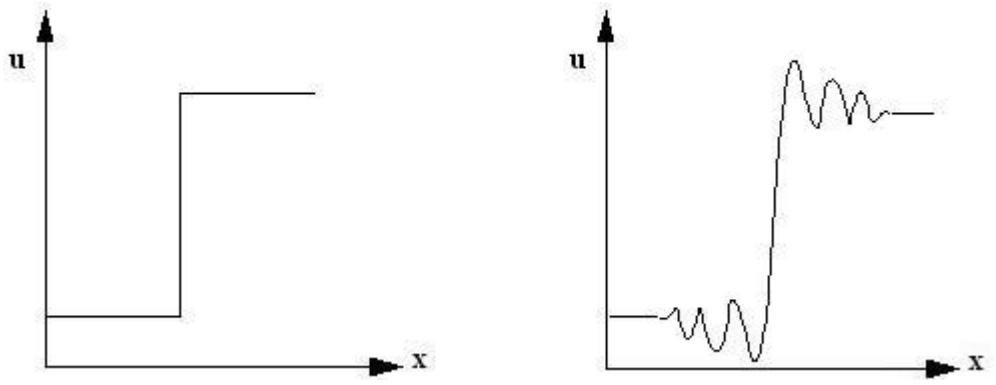

Figure 3: Exact solution to the simple convection equation (Eqn. (43)) (Left), and the solution found by using Eqn. (41) (Right).

In order to overcome this problem an up-winding method similar to Eqn. (40) is used

$$\phi_f = \begin{cases} \phi_P, & (\mathbf{u}\cdot\mathbf{n})_f > 0 \\ \phi_F, & (\mathbf{u}\cdot\mathbf{n})_f < 0. \end{cases} \tag{44}$$

The solution found using upwind interpolation as in Eqn. (44) is shown in Figure 4. The discontinuity in exact solution is smeared with simple upwind interpolation. Since, in this study, only incompressible flow is considered, this smearing property of simple upwind interpolation is not a problem due to the lack of high gradients in the solution profiles.

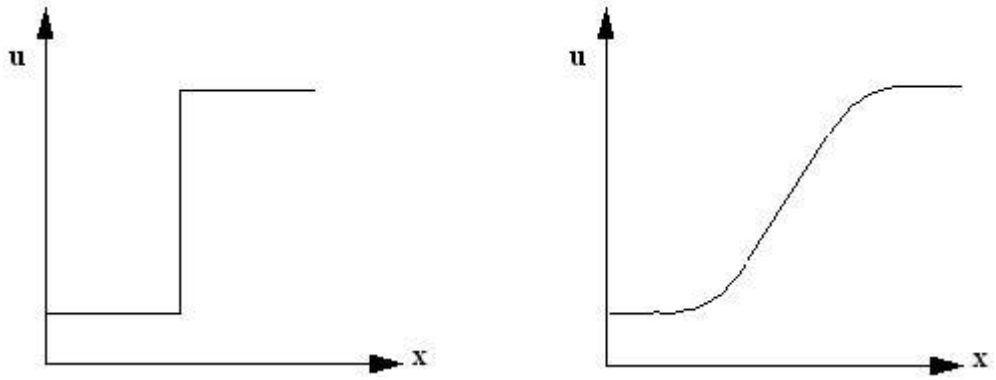

Figure 4: Exact solution to the simple convection equation (Eqn. (43)) (Left), and the solution found using simple upwind method as in Eqn. (44) (Right).

2.3.3 Rhie-Chow Interpolation Method

The usage of weighted linear interpolation may lead to the possibility of checker boarding in the velocity and pressure fields on collocated grids [18]. The Rhie-Chow interpolation method [16] overcomes this problem by adding a correction term to the arithmetic interpolation of the velocities. This correction term is written in terms of pressure gradients. The derivation of Rhie-Chow interpolation is given below.

After the discretization of the momentum equation, the resulting system of equations can be written as in Eqn. (45). Derivation of Eqn. (45) is detailed in the next chapter. The pressure gradient is not discretized intentionally:

$$a_P \mathbf{u}_P + (\nabla p)_P = (\sum a_{nb} \mathbf{u}_{nb})_P + \mathbf{S}_P \tag{45}$$

where a_p and a_{nb} are the coefficients of the unknown velocity values at the centres of the cell P and its neighbour cell nb respectively. The linear interpolation can also be used to obtain a discretized equation on the face as in Eqn. (46)

$$a_f \mathbf{u}_f + (\nabla p)_f = (\sum a_{nb} \mathbf{u}_{nb})_f + \mathbf{S}_f . \tag{46}$$

Rhie-Chow interpolation method uses Eqn. (46) to approximate it in the form of Eqn. (47). It is assumed that the right hand side of Eqn. (47) may be approximated by using a weighted linear interpolation of the corresponding terms in Eqn. (46). Thus

$$a_f \mathbf{u}_f + (\nabla p)_f = \overline{(\sum a_{nb}\mathbf{u}_{nb})_f + \mathbf{S}_f} = \overline{a_f \mathbf{u}_f} + \overline{(\nabla p)_f} \tag{47}$$

where overbar denotes weighted interpolation. Assuming that $a_f \approx \overline{a_f}$ then

$$\mathbf{u}_f = \overline{\mathbf{u}_f} + \overline{d_f}(\overline{\nabla p_f} - \nabla p_f) \tag{48}$$

where the interpolated values on the face f are written as

$$\begin{aligned}
&\overline{\mathbf{u}_f} = \alpha \mathbf{u}_P + (1-\alpha)\mathbf{u}_F \\
&\overline{\nabla p_f} = \alpha \nabla p_P + (1-\alpha)\nabla p_F \\
&\nabla p_f = F_f \mathbf{n}(p_F - p_P) \\
&a_f = \alpha a_P + (1-\alpha) a_F \\
&\overline{d_f} = a_f^{-1}
\end{aligned} \tag{49}$$

and α is used as the weighting factor. The simplest choice for α is 0.5 which results in a simplification in solving the resulting linear system of equations.

2.4 General Approaches in Flow Solvers

The Flow Solvers in the CFD world are based on two numerical approaches: pressure-based and density-based.

The significant difference between the two approaches is that the density-based approach is mainly developed for high-speed compressible flows on the other hand the pressured-based approach is first developed for low-speed incompressible flows. The increased attention on numerical methods leads to studies for extension to wide range of flow conditions by reformulations.

A brief summary of the pressure-based approach may include the following:

- The velocity field is obtained from the momentum equations.
- The pressure field is obtained using a pressure correction procedure or using a modified continuity equation as in the coupled solver in this thesis.

On the other hand the density-based approach may include:

- The velocity field is obtained from the momentum equations.
- The continuity equation is used to obtain the density field.
- An energy equation should be solved to obtain the temperature distribution.
- Pressure is extracted from the equation of state (EOS) formulation using the density and the internal energy values obtained by solving the continuity and the energy equations.

The control-volume based techniques are mostly preferred for the integral form of the continuity and momentum equations. The technique is also used for the equation of energy, equations for the turbulent flow and for other scalars (under appropriate conditions) [20], [21].

2.4.1 Pressure-Based Solver

The pressure-based algorithms consist of deriving the pressure equations from the continuity and the momentum equations. The constraint of mass conservation (continuity) of the velocity field is satisfied by solving a pressure (or pressure correction) equation. The entire set of governing equations is solved and reiterated until the solution satisfies a convergence criteria due to the non-linearity and the coupling characteristics of the equations [19, 20, 21].

The pressure-based solver algorithms are further classified as being segregated or coupled algorithms.

2.4.1.1 Pressure-Based Segregated Algorithm

The individual governing equations for the dependent variables are solved one after another in the pressure-based segregated algorithm. The governing equations are non-linear and coupled, thus, the solution loop must be carried out iteratively in order to obtain a converged numerical solution. The segregated solution method in the pressure-based flow solvers involves a

predictor–corrector iteration in which fluid velocities are first calculated (predicted) from fully nonlinear momentum equations and the pressure is computed by solving the pressure correction equation [20, 21, 22],

As an example, in order to show how the segregated algorithm is implemented in the CFD solver software FLUENT [21], the iteration steps are illustrated in Figure 5 and outlined below:

Step 1: The fluid properties are updated. (e,g, density, viscosity, specific heat)

Step 2: The momentum equations are solved by using the recently updated values of the face mass fluxes and pressure.

Step 3: The pressure correction equation is solved by using the recently obtained mass-flux and velocity field values.

Step 4: The pressure correction obtained from Step 3 is used to correct the pressure, the face mass fluxes, and the velocity field.

Step 5: The equations for additional scalars are solved such as turbulent quantities, energy, species, and radiation intensity using the current values of the solution variables if they are included in the problem.

Step 6: The interactions among different phases results in the source terms that should be updated. (e.g., source term for the carrier phase due to discrete particles).

Step 7: Check the convergence of the equations. If the convergence criteria are not met, continue to Step 1.

2.4.1.2 Pressure-Based Coupled Algorithm

The momentum equations and pressure based continuity equation involved in a coupled system of equations is solved by pressure-coupled algorithm. As in the previous case of the pressure-based segregated algorithm, the corresponding FLUENT CFD program steps are shown in Figure 5.

The difference between the pressure-based segregated and coupled algorithms can be seen in the steps 2 and 3 of the coupled algorithm. In the case of the segregated algorithm, these steps are replaced by a single step of solving the momentum and pressure-based continuity equations simultaneously. The remaining equations are solved in the same fashion as in the segregated algorithm.

The main advantage of the coupled algorithm when compared to the segregated algorithm is the rate of convergence (since the momentum and continuity equations are solved in a closely coupled manner) and requiring fewer places for the storage [20, 21, 22].

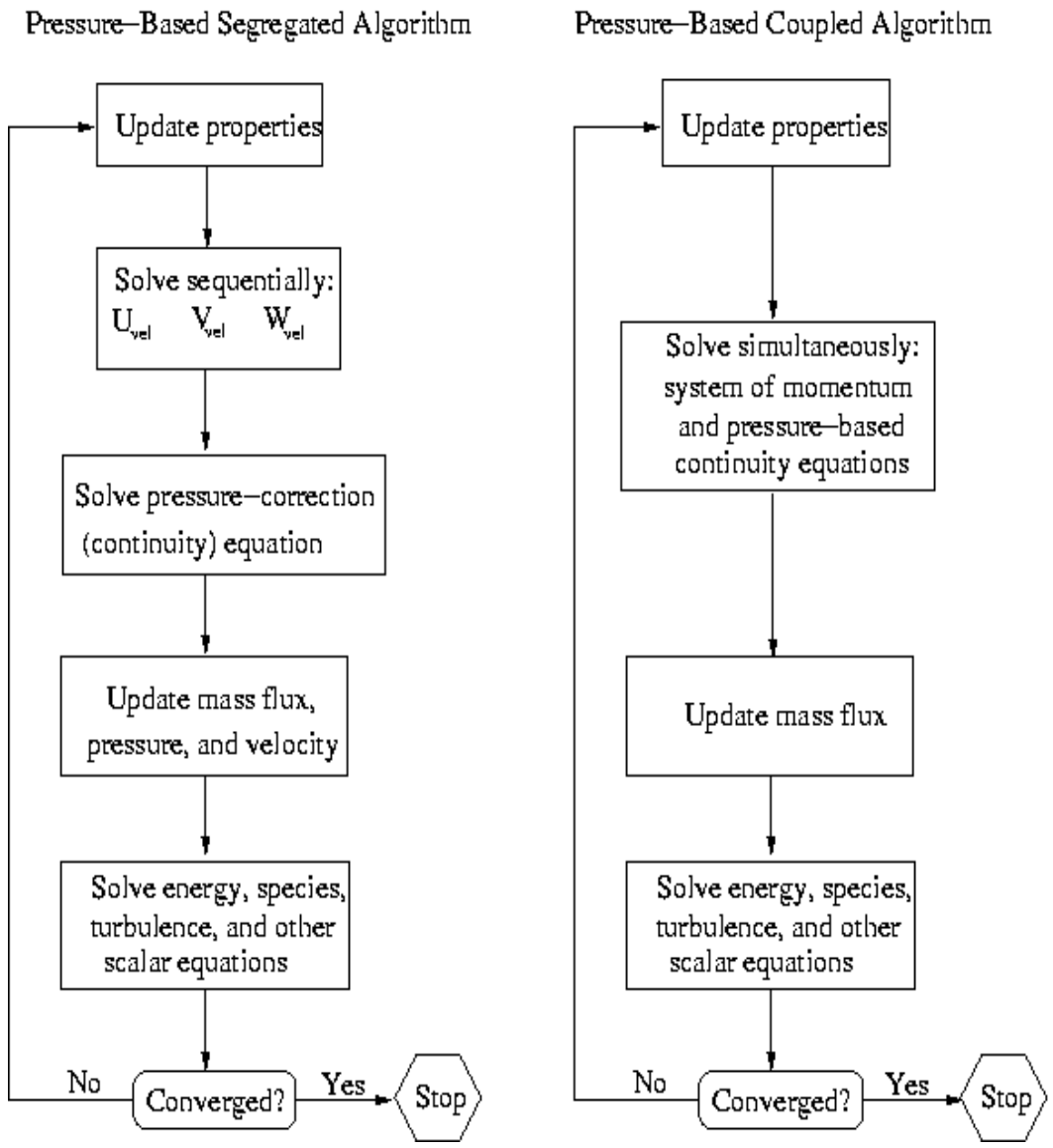

Figure 5: Overview of the Pressure-Based Solution Methods

2.4.2 Density-Based Solver

The density-based solver involves solving the governing equations simultaneously (the continuity, the momentum and in appropriate cases, the energy and species transport coupling together). Additional scalars are preferred to be solved subsequently due to the governing equations being nonlinear (and coupled). The density-based method requires the linearization of the discrete, non-linear equations resulting in a set of equations for the dependent variables in all control volumes. The linear system is then solved to obtain an updated solution for the flow field.

In density-based algorithm, several iterations of the solution loop must be performed before a converged solution is obtained as illustrated in Figure 6 and outlined in steps below:

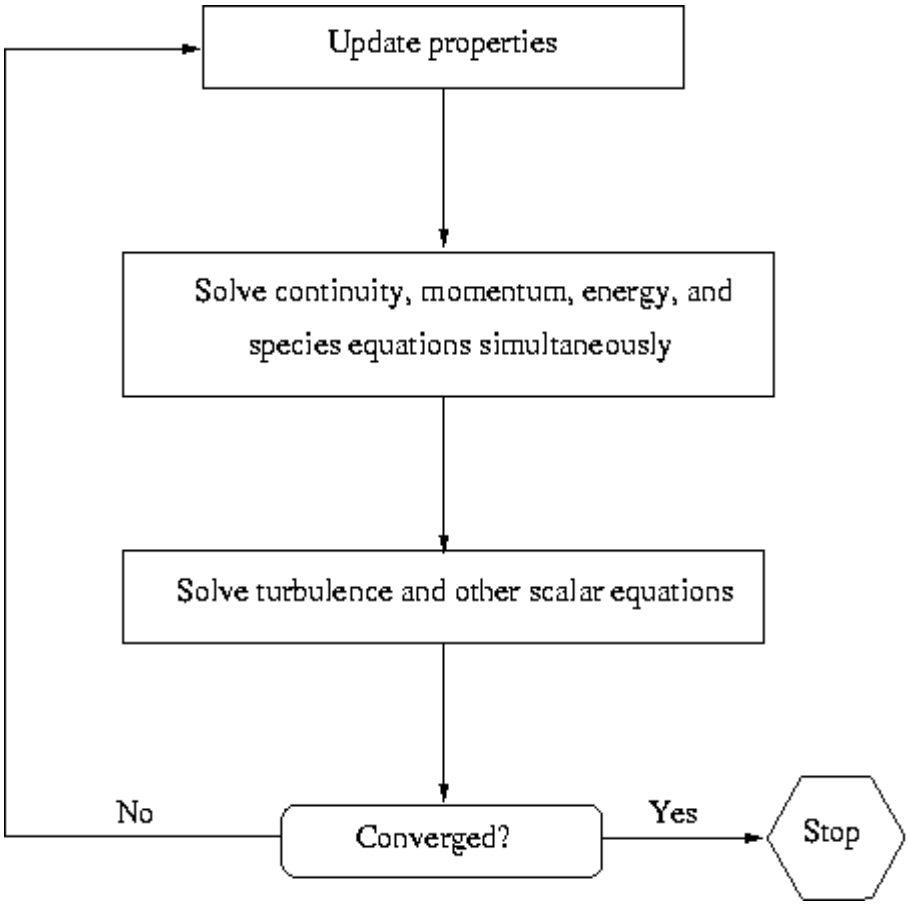

Figure 6: Overview of the Density-Based Solution Methods

Step 1: The fluid properties are updated (e, g, density, viscosity, specific heat). For the first iteration at the beginning of the solution procedure, the fluid properties will be updated based on the initial solution.

Step 2: The momentum, continuity and also in appropriate cases, the energy and species equations are solved simultaneously.

Step 3: The equations for additional scalars, such as the turbulent quantities, energy, species, and radiation intensity, are solved using the current values of the solution variables if they are involved in the problem.

Step 4: With a discrete phase trajectory calculation method, the source terms should be updated while interphase coupling is to be involved.

Step 5: Check for the convergence of the solution, and if the convergence criteria are not met, continue to Step 1.

2.4.3 Implicit and Explicit Iterative Methods

Although the steady-state forms of the governing equations are solved in this thesis, an iterative method with a relaxation should be used to get converged solution due to non-linear nature of the governing equations. Successive iterations may need to be performed similar to solution of time dependent problems which will be explained in Section 3.1.5. Relaxation is used to control the change in the solution variable ϕ. Using a relaxation factor α, the simplest relaxation formulation can be written as,

$$\phi = \phi_{old} + \alpha \Delta\phi \tag{50}$$

where $\Delta\phi$ is the computed change in ϕ.

Since the iterative methods with relaxation that are used to solve steady non-linear conservation equations resembles the solution procedure for time dependent problems, the concepts of implicit and explicit solvers are still valid. For example the simple relaxation formula in (50) is called an explicit relaxation [19] if $\Delta\phi$ is calculated using the values from last iteration (not from the current unknown values). In this case, the computed change $\Delta\phi$ is simply added to the solution from the previous iteration.

In the case of a implicit iteration process, $\Delta\phi$ is written in terms of the current unknown values $\Delta\phi = F(\phi)$ and (50) becomes a nonlinear system of equations,

$$\phi - \phi_{old} = F(\phi). \tag{51}$$

This system of equations is again solved iteratively due to non-linearities at each time level (between each outer iterations for steady problems).

CHAPTER 3

PRESSURE-BASED COUPLED FULLY IMPLICIT SOLVER

3.1 The Coupled Algorithm

Explicit treatments of the pressure gradient in the momentum equations and also the velocity field in the continuity equation have a negative effect on convergence. The main object of the coupled algorithm is to treat both terms in an implicit manner. This could be achieved by coupling the momentum equations and the pressure equation through a set of coefficients which represents the mutual influence of the continuity and momentum equations on the pressure and the velocity fields, as detailed in following sections.

3.1.1 Discretization of Momentum Equation

The momentum conservation equations for steady, laminar, incompressible Newtonian fluid flow are given by,

$$\nabla \cdot (\rho \mathbf{v}\mathbf{v}) = \nabla \cdot (\mu \nabla \mathbf{v}) - \nabla \cdot (p\mathbf{I}). \tag{52}$$

Using the divergence theorem and integrating momentum equation over the control volume, the diffusion and the convection terms can be transformed into surface integrals over the control volume surface and can be written in the form of Eqn. (53):

$$\oint_{\partial\Omega} (\rho \mathbf{v}\mathbf{v})_f \cdot d\mathbf{S} = \oint_{\partial\Omega} (\mu \nabla \mathbf{v})_f \cdot d\mathbf{S} - \oint_{\partial\Omega} p_f \, d\mathbf{S} \tag{53}$$

where Ω is the control volume, $\partial\Omega$ is the bounding surface of the control volume and $\mathbf{S}$ is the surface normal vector.

The surface integrals in Eqn. (53) can be written in the discretizated form for a control volume represented by P as in Eqn. (54):

$$\sum_{f=nb(P)} (\rho \mathbf{vv} - \mu \nabla \mathbf{v})_f \cdot \mathbf{S}_f + \sum_{f=nb(P)} p_f \mathbf{S}_f = \mathbf{b}_P \Omega_P \tag{54}$$

where index *f* defines the surface between P and the neighbouring control volumes.

The pressure gradient term in the semi-discretized momentum equation has been integrated over the faces of the control volume. At each face, the pressure is evaluated by averaging

$$p_f = \frac{p_P + p_F}{2} \tag{55}$$

where index F denotes the neighbouring control volume to P.

3.1.1.1 Discretization of Diffusion Term in Momentum Equation

The diffusion term $(\mu \nabla \mathbf{u})_f$ on the face *f* can be approximated as in Eqn. (56):

$$(\mu \nabla u)_f \cdot \mathbf{S}_f = \mu_f \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} (u_F - u_P) + \mu_f \overline{\nabla u_f} \cdot \left(\mathbf{S}_f - \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \mathbf{d}_{PF} \right) \tag{56}$$

where the first term on the right side is the main diffusion flux, the second term is the cross diffusion flux, u is the velocity component in the x direction, $\overline{\nabla u_f}$ is the velocity gradient, $\mathbf{d}_{PF}$ is the vector from cell centers of P to F and $\mathbf{S}_f$ is the surface normal.

If $\mathbf{d}_{PF}$ and $\mathbf{S}_f$ is in the same direction, then the grid is called orthogonal and the cross diffusion flux vanishes. The same approximation can be applied for the velocity in the y direction below in Eqn. (57):

$$\left(\mu\nabla v\right)_f \cdot \mathbf{S}_f = \mu_f \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}}\left(v_F - v_P\right) + \mu_f \overline{\nabla v_f} \cdot \left(\mathbf{S}_f - \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}}\mathbf{d}_{PF}\right) \tag{57}$$

where v is the velocity component in the y direction and $\overline{\nabla v_f}$ is the velocity gradient.

The velocity gradients $\overline{\nabla u_f}$ and $\overline{\nabla v_f}$ are the averages of the gradient values of the neighbouring cell centres, calculated as in Eqn. (58),

$$\begin{aligned} \overline{\nabla u_f} &= \frac{\left(\nabla u\right)_P + \left(\nabla u\right)_F}{2}, \\ \overline{\nabla v_f} &= \frac{\left(\nabla v\right)_P + \left(\nabla v\right)_F}{2}. \end{aligned} \tag{58}$$

The cell gradient values in Eqn. (58) at the cell centres are calculated using Green-Gauss method as described below:

$$\begin{aligned} \left(\nabla u\right)_P &= \frac{1}{\Omega_P}\sum_{f=nb(P)} u_f \mathbf{S}_f \ , \quad \left(\nabla v\right)_P = \frac{1}{\Omega_P}\sum_{f=nb(P)} v_f \mathbf{S}_f , \\ \left(\nabla v\right)_F &= \frac{1}{\Omega_F}\sum_{f=nb(F)} v_f \mathbf{S}_f \ , \quad \left(\nabla u\right)_F = \frac{1}{\Omega_F}\sum_{f=nb(F)} u_f \mathbf{S}_f , \end{aligned} \tag{59}$$

where Ω_P and Ω_F denote the volumes of the cell P and the cell F, respectively.

The face velocities in the equations above are calculated as the averages of the neighbouring cell centre values as follows:

$$\begin{aligned} u_f &= \frac{\left(u\right)_P + \left(u\right)_F}{2}, \\ v_f &= \frac{\left(v\right)_P + \left(v\right)_F}{2}. \end{aligned} \tag{60}$$

3.1.1.2 Discretization of Convective Term in Momentum Equation

The convective flux $(\rho \mathbf{vv})_f \cdot \mathbf{S}_f$ in Eqn. (54) can be written as in Eqn. (61)

$$(\rho \mathbf{vv})_f \cdot \mathbf{S}_f = \mathbf{v}_f \dot{m}_f \tag{61}$$

where $\dot{m}_f$ is the total mass flow over the face *f*

$$\dot{m}_f = (\rho \mathbf{v})_f \cdot \mathbf{S}_f . \tag{62}$$

The convective flux $\mathbf{v}_f \dot{m}_f$ in Eqn. (61) is calculated using a simple up-winding scheme operator $\|\dot{m}_f, 0\|$, defined by

$$\|\dot{m}_f, 0\| \mathbf{v}_f = \begin{cases} \mathbf{v}_P \dot{m}_f, & \dot{m}_f > 0 \\ \mathbf{v}_F \dot{m}_f, & \dot{m}_f < 0 \end{cases} \tag{63}$$

where $\mathbf{v}_P$ the velocity at the control volume P and $\mathbf{v}_F$ is velocity at the neighbouring control volume F. Mass flux $\dot{m}_f$ over the face *f* is calculated in the previous iteration or time step using Rhie-Chow interpolation method which will be explained in section 3.1.2.

3.1.1.3 Fully Discretized Form of Momentum Equation

Using the derived discrete form of the convective and diffusive terms of the momentum equation (Eqn. (54)), fully discretized momentum equation in the x direction for velocity component *u* is written as:

$$\sum_{f=nb(P)} \|\dot{m}_f, 0\| u_f - \sum_{\substack{F=NB(P) \\ f=nb(P)}} \mu_f \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} (u_F - u_P) + \sum_{\substack{F=NB(P) \\ f=nb(P)}} \frac{p_F + p_P}{2} \cdot \mathbf{S}_{f,x} = \sum_{\substack{F=NB(P) \\ f=nb(P)}} \mu_f \overline{\nabla u_f} \cdot \left(\mathbf{S}_f - \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \mathbf{d}_{PF} \right) \tag{64}$$

where $S_{f,x}$ is the component of face area normal vector $\mathbf{S}_f$ in x direction. The cross diffusion terms are treated as source term in the explicit form.

The fully discretized momentum equation in the y direction for the velocity component *v* can be written in the similar way:

$$\sum_{f=nb(P)} \left\| \dot{m}_f, 0 \right\| v_f - \sum_{\substack{F=NB(P) \\ f=nb(P)}} \mu_f \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} (v_F - v_P) + \sum_{\substack{F=NB(P) \\ f=nb(P)}} \frac{p_F + p_P}{2} \cdot \mathbf{S}_{f,y} = \sum_{\substack{F=NB(P) \\ f=nb(P)}} \overline{\nabla v_f} \cdot \left(\mathbf{S}_f - \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \mathbf{d}_{PF} \right). \tag{65}$$

Now, collecting the common coefficients of the unknowns u_P, u_F, p_P, p_F in Eqns. (64) and (65) above, the following system of equations are obtained in Eqn. (66):

$$\underbrace{u_P \sum_{\substack{F=NB(P) \\ f=nb(P)}} \left(\left\| \dot{m}_f, 0 \right\| + \mu_f \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \right)}_{a_P^{uu}} + \sum_{\substack{F=NB(P) \\ f=nb(P)}} \underbrace{\left(\left\| \dot{m}_f, 0 \right\| - \mu_f \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \right)}_{a_F^{uu}} u_F + \sum_{\substack{F=NB(P) \\ f=nb(P)}} \left(\underbrace{\tfrac{1}{2} S_{f,x}}_{a_F^{up}} p_F \right) + p_P \underbrace{\sum_{f=nb(P)} \left(\tfrac{1}{2} S_{f,x} \right)}_{a_P^{up}} = \underbrace{\sum_{\substack{F=NB(P) \\ f=nb(P)}} \mu_f \overline{\nabla u_f} \cdot \left(\mathbf{S}_f - \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \mathbf{d}_{PF} \right)}_{b_P^u}. \tag{66}$$

This can be rewritten as,

$$a_p^{uu} u_P + a_p^{up} p_P + \sum_{F=NB(P)} a_F^{uu} u_F + \sum_{F=NB(P)} a_F^{up} p_F = b_P^u \tag{67}$$

The same procedure is applied in the y direction by collecting the common coefficients of the unknowns v_P, v_F, p_P, p_F and the following system of equations are obtained in Eqn. (68):

$$v_P \underbrace{\sum_{\substack{F=NB(P)\\ f=nb(P)}} \left(\left\| \dot{m}_f, 0 \right\| + \mu_f \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \right)}_{a_P^{vv}} + \sum_{\substack{F=NB(P)\\ f=nb(P)}} \underbrace{\left(\left\| \dot{m}_f, 0 \right\| - \mu_f \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \right)}_{a_F^{vv}} v_F +$$

$$\sum_{\substack{F=NB(P)\\ f=nb(P)}} \underbrace{\left(\tfrac{1}{2} S_{f,x} \right)}_{a_F^{vp}} p_F + p_P \underbrace{\sum_{f=nb(P)} \left(\tfrac{1}{2} S_{f,y} \right)}_{a_P^{vp}} = \underbrace{\sum_{\substack{F=NB(P)\\ f=nb(P)}} \mu_f \overline{\nabla v_f} \cdot \left(\mathbf{S}_f - \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \mathbf{d}_{PF} \right)}_{b_P^u}. \tag{68}$$

This can be rewritten as

$$a_p^{vv} v_P + a_p^{vp} p_P + \sum_{F=NB(P)} a_F^{vv} v_F + \sum_{F=NB(P)} a_F^{vp} p_F = b_P^v. \tag{69}$$

The final form of the discretized momentum equations is expressed as

$$\begin{aligned} a_p^{uu} u_P + a_p^{up} p_P + \sum_{F=NB(P)} a_F^{uu} u_F + \sum_{F=NB(P)} a_F^{up} p_F &= b_P^u, \\ a_p^{vv} v_P + a_p^{vp} p_P + \sum_{F=NB(P)} a_F^{vv} v_F + \sum_{F=NB(P)} a_F^{vp} p_F &= b_P^v, \end{aligned} \tag{70}$$

where the coefficients are given by

$$\begin{aligned} a_F^{uu} &= a_F^{vv} = \sum_{f=nb(P)} \left\| \dot{m}_f, 0 \right\| - \mu_f \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}}, \\ a_P^{uu} &= \sum_{\substack{F=NB(P)\\ f=nb(P)}} \left\| \dot{m}_f, 0 \right\| + \mu_f \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}}, \\ a_P^{vv} &= a_P^{uu}, \\ a_F^{up} &= \sum_{f=nb(P)} \tfrac{1}{2} S_{f,x}, \\ a_F^{vp} &= \sum_{f=nb(P)} \tfrac{1}{2} S_{f,y}, \\ a_P^{up} &= \sum_{f=nb(P)} \tfrac{1}{2} S_{f,x}, \\ a_P^{up} &= \sum_{f=nb(P)} \tfrac{1}{2} S_{f,y}, \end{aligned}$$

$$b_P^u = \sum_{\substack{F=NB(P) \\ f=nb(P)}} \left[\mu_f \overline{\nabla u_f} \cdot \left(\mathbf{S}_f - \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \mathbf{d}_{PF} \right) \right],$$

$$b_P^v = \sum_{\substack{F=NB(P) \\ f=nb(P)}} \left[\mu_f \overline{\nabla v_f} \cdot \left(\mathbf{S}_f - \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \mathbf{d}_{PF} \right) \right]. \quad (71)$$

3.1.2 Discretization of Continuity Equation

The continuity equation for the steady, laminar, incompressible Newtonian fluid flow is given by,

$$\nabla \cdot \left(\rho \mathbf{v} \right) = 0. \quad (72)$$

Using the divergence theorem and integrating over the control volume, the continuity equation can be rewritten in the following conservation form:

$$\oint_{\partial\Omega} \left(\rho \mathbf{v} \right)_f \cdot d\mathbf{S} = 0. \quad (73)$$

The surface integral in Eqn. (74) can be semi-discretized to get the form of the continuity equation given by Eqn. (75):

$$\sum_{f=nb(P)} \rho_f \mathbf{v}_f \cdot \mathbf{S}_f = 0. \quad (75)$$

One way of writing the face velocities $\mathbf{v}_f$ in Eqn. (75) above is the simple averaging of the neighbouring cell centred values which is,

$$\mathbf{v}_f = \frac{\mathbf{v}_P + \mathbf{v}_F}{2}. \quad (76)$$

This simple averaging of cell velocities leads to unphysical checker-boarding pattern of the pressure values. This problem is a result of storing the pressure and velocity values in the

same location on the solution grid. In this study Rhie-Chow Interpolation of face velocity is used to avoid pressure checker-boarding [16].

For an interior face *f*, the interpolated velocity is written using Rhie-Chow interpolation as,

$$\mathbf{v}_f = \frac{\mathbf{v}_P + \mathbf{v}_F}{2} - \frac{\Omega_P + \Omega_F}{a_P + a_F}\left(\nabla p_f - \overline{\nabla p_f}\right) \tag{77}$$

where a_P is an average value of the coefficients of a_p^{uu} and a_p^{vv}, and a_P is an average value of the coefficients of a_F^{uu} and a_F^{vv} in Eqn. (78). The average values are

$$a_p = \left[a_p^{uu}, a_p^{vv}\right]^T \mathbf{n}_f \ , \quad a_F = \left[a_F^{uu}, a_F^{vv}\right]^T \mathbf{n}_f \tag{79}$$

where $\mathbf{n}_f$ is the face unit normal vector. Ω_P and Ω_F are the volumes of the neighbouring cells P and F, respectively.

The discrete form of the continuity equation and Rhie-Chow interpolation are combined to get Eqn. (80),

$$\sum_{f=nb(P)} \rho_f \left(\overline{\mathbf{v}_f} - \overline{D_f}\left(\nabla p_f - \overline{\nabla p_f}\right)\right)\cdot \mathbf{S}_f = 0 \tag{80}$$

where,

$$\overline{\mathbf{v}_f} = \frac{\mathbf{v}_P + \mathbf{v}_F}{2} \tag{81}$$

and

$$\overline{D_f} = \frac{\Omega_P + \Omega_F}{a_P + a_F}. \tag{82}$$

$\overline{\nabla p_f}$ is calculated by averaging pressure gradient values at the cell centres neighbouring face f in Eqn. (83),

$$\overline{\nabla p_f} = \frac{(\nabla p)_P + (\nabla p)_F}{2}. \tag{83}$$

The cell gradient values of pressure are calculated using Green-Gauss method as described in Eqn. (84),

$$(\nabla p)_P = \frac{1}{\Omega_P} \sum_{f=nb(P)} p_f \mathbf{S}_f, \quad (\nabla p)_F = \frac{1}{\Omega_F} \sum_{f=nb(F)} p_f \mathbf{S}_f. \tag{84}$$

The face pressure gradient term ∇p_f in Eqn. (80) is defined as,

$$\nabla p_f \cdot \mathbf{S}_f = \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} (p_F - p_P) + \overline{\nabla p_f} \cdot \left(\mathbf{S}_f - \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \mathbf{d}_{PF} \right) \tag{85}$$

where the first term on the right side is the main pressure diffusion term and the second term is the cross diffusion correction term for non-orthogonal grids.

Eqn. (80) is rewritten in the following form,

$$\begin{aligned} &\sum_{\substack{F=NB(P) \\ f=nb(P)}} \rho_f \left(-\overline{D_f} \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} (p_F - p_P) \right) + \sum_{\substack{F=NB(P) \\ f=nb(P)}} \rho_f \overline{\mathbf{v}_f} \cdot \mathbf{S}_f = \\ &- \sum_{\substack{F=NB(P) \\ f=nb(P)}} \rho_f \left(\overline{D_f} \overline{\nabla p_f} \right) \cdot \mathbf{S}_f + \sum_{\substack{F=NB(P) \\ f=nb(P)}} \rho_f \left(\overline{D_f} \overline{\nabla p_f} \right) \cdot \left(\mathbf{S}_f - \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \mathbf{d}_{PF} \right). \end{aligned} \tag{86}$$

Collecting the common coefficients of the terms of p_P, p_F, u_P, u_F, v_P, v_F, Eqn. (86) can be rewritten as in Eqn. (87),

$$a_P^{pp} p_P + a_P^{pu} u_P + a_P^{pv} v_P + \sum_{F=NB(P)} a_F^{pp} p_F + \sum_{F=NB(P)} a_F^{pu} u_F + \sum_{F=NB(P)} a_F^{pv} v_F = b_P^p \tag{87}$$

with coefficients as in Eqn. (88),

$$a_F^{pp} = \sum_{f=nb(P)} -\overline{D_f}\frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}}, \quad a_P^{pp} = -\sum_{\substack{F=NB(P)\\ f=nb(P)}} a_F^{pp}, \quad a_F^{pu} = \sum_{f=nb(P)} \tfrac{1}{2}\rho_f S_{f,x},$$

$$a_F^{pv} = \sum_{f=nb(P)} \tfrac{1}{2}\rho_f S_{f,y}, \quad a_P^{pu} = \sum_{\substack{F=NB(P)\\ f=nb(P)}} \tfrac{1}{2}\rho_f S_{f,x}, \quad a_P^{pv} = \sum_{\substack{F=NB(P)\\ f=nb(P)}} \tfrac{1}{2}\rho_f S_{f,y}, \tag{88}$$

$$b_P^p = -\sum_{\substack{F=NB(P)\\ f=nb(P)}} \rho_f \left(\overline{D_f}\,\overline{\nabla p_f}\right)\cdot \mathbf{S}_f + \sum_{\substack{F=NB(P)\\ f=nb(P)}} \rho_f \left(\overline{D_f}\,\overline{\nabla p_f}\right)\cdot\left(\mathbf{S}_f - \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}}\mathbf{d}_{PF}\right).$$

3.1.3 Coupled System of Equations

The discretized momentum and continuity equations in Eqns. (70) and (87) are combined to get a fully coupled system of equations for each control volume cell in the computational grid. This system of equations for a control volume cell is written as,

$$\begin{bmatrix} a_P^{uu} & a_P^{uv} & a_P^{up} \\ a_P^{vu} & a_P^{vv} & a_P^{vp} \\ a_P^{pu} & a_P^{pv} & a_P^{pp} \end{bmatrix}\begin{bmatrix} u_P \\ v_P \\ p_P \end{bmatrix} + \sum_{F=NP(P)} \begin{bmatrix} a_F^{uu} & a_F^{uv} & a_F^{up} \\ a_F^{vu} & a_F^{vv} & a_F^{vp} \\ a_F^{pu} & a_F^{pv} & a_F^{pp} \end{bmatrix}\begin{bmatrix} u_F \\ v_F \\ p_F \end{bmatrix} = \begin{bmatrix} b_P^u \\ b_P^v \\ b_P^p \end{bmatrix}. \tag{89}$$

These set of equations can be assembled to get the equations for all computational cells over the entire computational domain

$$\mathbf{A\Phi} = \mathbf{B}. \tag{90}$$

All variables (u, v, p) which are expressed in the set of equations above are now solved for simultaneously. The continuity equation is now written in terms of the pressure rather than the pressure correction.

The steps in the coupled algorithm can be listed as follows:

1. The latest available values of $(\dot{m}_f^{(n)}, u^{(n)}, v^{(n)}, p^{(n)})$ are the starting values.
2. The next step is assembling and solving the momentum and continuity equation for the new values of (u, v, p).
3. Assemble $\dot{m}_f$ using the modified Rhie–Chow interpolation. [25]
4. Solve all the other scalar equations in the order.
5. These steps will be repeated starting from step 2 until convergence.

3.1.4 Boundary Conditions

The coupling between the governing equations and the boundary conditions plays the most important role towards the success of the proposed algorithm in this thesis. The type of the boundary conditions is determinative for the limitations that control volume may face. The details of the commonly implied boundary conditions at wall, inlet and outlet of the domain are described in the following sections [1].

3.1.4.1 The no-slip boundary condition

The no-slip boundary condition at a stationary wall should be defined on the domain in order to solve the problem.

The wall shear stress can be calculated by Eqn. (91):

$$\tau_w = \mu \frac{\mathrm{v}_t}{\mathbf{d}_{Pw} \cdot \mathbf{n}_w}. \tag{91}$$

The tangential velocity v_t in the equation is the magnitude of the velocity vector $\mathbf{v}_t$ at the interior grid point shown in Fig. 1(b), and the other velocity vector $\mathbf{v}_n$ is normal to the wall. These vectors are designated by $\mathbf{v}_p\,(=\mathrm{u}_p\mathbf{i}+\mathrm{v}_p\mathbf{j}\,)$ on the cell P shown in Fig. 1(b). Here, $\mathbf{d}_{Pw}$ is the distance vector between the internal and boundary grid point, $\mathbf{n}_w$ is the outward unit vector normal to the wall with $\mathbf{n}_w = n_{w,x}\mathbf{i} + n_{w,y}\mathbf{j} = \mathbf{S}_w / S_w$, where $S_w = \left\| \mathbf{S}_w \right\|$ and $(\mathbf{d}_{Pw} \cdot \mathbf{n}_w)$ is the normal distance to the wall.

The shear force F_s can be calculated by using

$$\mathrm{F}_s = -\tau_w S_w. \tag{92}$$

The tangential velocity vector $\mathbf{v}_t$ is given by

$$\mathbf{v}_t = \mathbf{v}_P - (\mathbf{v}_P \cdot \mathbf{n}_w)\mathbf{n}_w. \tag{93}$$

The two dimensional form of the shear force can be written by combining Eqns. (91) and (93) as follows:

$$\mathbf{F}_s = \begin{bmatrix} F_{s,x} \\ F_{s,y} \end{bmatrix} = -\frac{\mu S_w}{\mathbf{d}_{pw} \cdot \mathbf{n}_w} \begin{bmatrix} u_P (1 - n_{w,x}^2) - v_P \, n_{w,x} \, n_{w,y} \\ v_P (1 - n_{w,y}^2) - u_P \, n_{w,x} \, n_{w,y} \end{bmatrix}. \tag{94}$$

The coefficients now can be defined in a form which includes the wall shear stress:

$$\begin{aligned} a_P^{uu} &= a_P^{uu} + \frac{\mu S_w}{\mathbf{d}_{pw} \cdot \mathbf{n}_w} (1 - n_{w,x}^2), \\ a_P^{vv} &= a_P^{vv} + \frac{\mu S_w}{\mathbf{d}_{pw} \cdot \mathbf{n}_w} (1 - n_{w,y}^2), \\ a_P^{uv} &= a_P^{uv} - \frac{\mu S_w}{\mathbf{d}_{pw} \cdot \mathbf{n}_w} n_{w,x} \, n_{w,y}, \\ a_P^{vu} &= a_P^{vu} - \frac{\mu S_w}{\mathbf{d}_{pw} \cdot \mathbf{n}_w} n_{w,x} \, n_{w,y}. \end{aligned} \tag{95}$$

Mass flow rate is zero at the wall boundary which means no modification can be applied for the pressure equations and its coefficients remain unchanged. The pressure at the wall can be extrapolated from the pressure at the main grid point by using a zero order profile to yield Eqn. (96):

$$p_w = p_P \tag{96}$$

and the reflection of the participation to momentum equation can be written in the form:

$$\begin{aligned} a_P^{up} &= a_P^{up} + S_{w,x} \\ a_P^{vp} &= a_P^{vp} + S_{w,y} \end{aligned} \tag{97}$$

where $\mathbf{S}_w = S_{w,x}\mathbf{i} + S_{w,y}\mathbf{j}$.

3.1.4.2 Inlet boundary conditions

The pressure and the velocity can be specified at the inlet when solving for an incompressible flow field.

3.1.4.2.1 Specified static pressure

The calculation of the flux on a boundary face gives the function which is identical with the control volume and the boundary face itself. On the other hand when the flux is calculated at interior control volume face, the flux is a function of the two control volumes sharing the same interior face. The corresponding boundary flux can be calculated and moved to the source term if the value of the dependent variable is specified. If the static pressure is specified at the inlet which means that the pressure is known but velocity and velocity direction are not known, then the velocity can be calculated by interpolation while the velocity direction has to be specified.

The modified coefficients of the momentum equations at the inlet boundary are defined in Eqn. (98):

$$
\begin{aligned}
a_P^{uu} &= a_P^{uu} + \| \dot{m}_i, 0 \| + \mu \frac{\mathbf{S}_i \cdot \mathbf{S}_i}{\mathbf{d}_{pi} \cdot \mathbf{S}_i}, \\
a_P^{vv} &= a_P^{vv} + \| \dot{m}_i, 0 \| + \mu \frac{\mathbf{S}_i \cdot \mathbf{S}_i}{\mathbf{d}_{pi} \cdot \mathbf{S}_i}, \\
b_P^{u} &= b_P^{u} + (\mu \nabla u \cdot \mathrm{T})_i + \left[\| \dot{m}_i, 0 \| + \mu \frac{\mathbf{S}_i \cdot \mathbf{S}_i}{\mathbf{d}_{pi} \cdot \mathbf{S}_i} \right] u_i - p_i S_{i,x}, \\
b_P^{v} &= b_P^{v} + (\mu \nabla v \cdot \mathrm{T})_i + \left[\| \dot{m}_i, 0 \| + \mu \frac{\mathbf{S}_i \cdot \mathbf{S}_i}{\mathbf{d}_{pi} \cdot \mathbf{S}_i} \right] v_i - p_i S_{i,y}.
\end{aligned}
\tag{98}
$$

The coefficients are written in the form in Eqn. (98) by the help of splitting the surface vector **S** into two components **E** and **T** (i.e. **S** = **E**+**T**), with **E** being aligned with the distance vector and **T** normal to the **S** vector (Fig. 1(c)) [1].

The modified coefficients of the pressure equation are given in Eqn. (99). Now velocity is extrapolated from the closest control volume and the pressure gradient term is obtained using the known inlet pressure value considered explicitly for the pressure equation [1]:

$$
\begin{aligned}
a_P^{pu} &= a_P^{up} + \rho S_{i,x}, \\
a_P^{pv} &= a_P^{vp} + \rho S_{i,y}, \\
a_P^{pp} &= a_P^{pp} + \rho_i \frac{(\overline{D_i}\,\mathbf{S}_i)\cdot\mathbf{S}_i}{\mathbf{S}_i\cdot\mathbf{d}_{pi}}, \\
b_P^{p} &= \rho_i \overline{D_i}\,\overline{\nabla p_i}\cdot\mathbf{T}_i + \rho_i \frac{(\overline{D_i}\,\mathbf{S}_i)\cdot\mathbf{S}_i}{\mathbf{S}_i\cdot\mathbf{d}_{pi}} p_i - \rho_i \overline{D_i}\,\overline{\nabla p_i}\cdot\mathbf{S}_i.
\end{aligned}
\tag{99}
$$

Where the averaged values $\overline{D_i}$, $\overline{\nabla p_i}$ are equal to cell center values of the boundary cells.

3.1.4.2.2 Specified velocity

The convection term can be used explicitly because the velocity is specified at the inlet. A source term appears due to the involvement of the stress term and it effects the coefficients of the interior control volume and the boundary itself. The pressure is extrapolated from the interior as it is mentioned for the case of the wall conditions for the pressure gradient term in the momentum equations. Now, the coefficients for the momentum equations at the inlet is defined as in Eqn. (100) as a result of these considerations [1].

$$
\begin{aligned}
a_P^{uu} &= a_P^{uu} + \| \dot{m}_i, 0 \| + \mu \frac{\mathbf{S}_i\cdot\mathbf{S}_i}{\mathbf{d}_{pi}\cdot\mathbf{S}_i}, \\
a_P^{vv} &= a_P^{vv} + \| \dot{m}_i, 0 \| + \mu \frac{\mathbf{S}_i\cdot\mathbf{S}_i}{\mathbf{d}_{pi}\cdot\mathbf{S}_i}, \\
a_P^{up} &= a_P^{up} + S_{i,x}, \\
a_P^{vp} &= a_P^{vp} + S_{i,y}, \\
a_P^{pu} &= a_P^{pu} + \rho_f S_f^x, \\
a_P^{pv} &= a_P^{pv} + \rho_f S_f^y, \\
b_P^{u} &= b_P^{u} + (\mu\nabla u\cdot\mathbf{T})_i + \left[\| \dot{m}_i, 0 \| + \mu \frac{\mathbf{S}_i\cdot\mathbf{S}_i}{\mathbf{d}_{pi}\cdot\mathbf{S}_i} \right] u_i, \\
b_P^{v} &= b_P^{v} + (\mu\nabla v\cdot\mathbf{T})_i + \left[\| \dot{m}_i, 0 \| + \mu \frac{\mathbf{S}_i\cdot\mathbf{S}_i}{\mathbf{d}_{pi}\cdot\mathbf{S}_i} \right] v_i.
\end{aligned}
\tag{100}
$$

3.1.4.3 Outlet boundary condition

The outlet boundary boundary condition can be defined as a specified value for static pressure. The coefficients are similar to which are detailed for inlet boundary conditions with Eqns. (98) and (99) [1].

3.1.5 False Transient Time Stepping for Under-relaxation of Equations

False time step relaxation method modifies the finite-volume equations by adding an additional, pseudo-transient term which is defined as the false time step Δt . This false time step value can make the additional term small (light/loose relaxation) when it is big and a small value makes the additional term large (heavy/tight relaxation). This method changes the momentum equation to a new form in Eqn. (101):

$$\left(a_p^{uu}+\frac{(\rho_P V_P)}{\Delta t}\right)u_P+\underline{\underline{a_p^{uv}v_P}}+\underline{a_p^{up}p_P}+\sum_{F=NB(P)}a_F^{uv}u_F+\underline{\underline{\sum_{F=NB(P)}a_F^{uv}v_F}}+\underline{\sum_{F=NB(P)}a_F^{uv}p_F}=$$
$$b_P^u+\frac{(\rho_P V_P v_P)^0}{\Delta t},$$
$$\left(a_p^{vv}+\frac{(\rho_P V_P)}{\Delta t}\right)v_P+\underline{\underline{a_p^{vu}u_P}}+\underline{a_p^{vp}p_P}+\sum_{F=NB(P)}a_F^{uv}v_F+\underline{\underline{\sum_{F=NB(P)}a_F^{vu}u_F}}+\underline{\sum_{F=NB(P)}a_F^{vp}p_F}=$$
$$b_P^v+\frac{(\rho_P V_P u_P)^0}{\Delta t},\qquad(101)$$

where V is the volume of the specified cell. It should be noted that the single underlined terms in Eqn. (101) represent the pressure gradient in its implicit form; while the double underlined terms account for the velocity component interactions with their values being zero except at wall boundaries.

CHAPTER 4

TEST CASES

4.1 Flow Over a Circular Bump

4.1.1 First Circular Bump Test Case

In this test case, flow over a circular bump, having 3 meters of length in x direction and a maximum height of 0,3 meters in y direction, in a 2D channel with the dimensions of $x = 15m$ and $y = 3m$ is considered. The fluid properties of air are used with density and viscosity values of 1.225 kg/m^3 and 1.7894 kg/ms, respectively. The geometry and mesh structure is given in Figure 7.

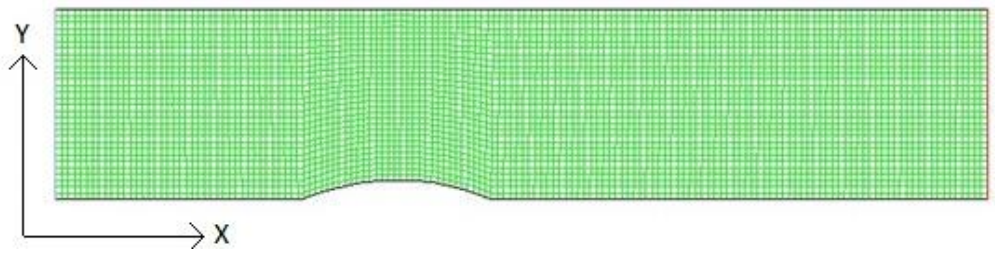

Figure 7: Mesh structure for first circular bump test case

Air enters the channel at 10 m/s and exit is given as pressure outlet. The results are compared with the coupled solver of FLUENT code.

The convergence is controlled with the following formulation,

$$RES(\phi) = \max_{i=1}^{N} \left(\frac{\left| \phi_i^{n+1} - \phi_i^n \right|}{\phi_i^n} \right) \tag{102}$$

where i denotes the cell number and n is the iteration number. In this test case with a false transient time step of 1.0, residuals reach to a value of 10^{-6} at below 100 iterations and continue to decrease. FLUENT solver couldn't reach to the value below 10^{-4} and stalled at these values. The results are compared in Figure 8, Figure 9 and Figure 10. The two results are in good agreement.

In each figure below, the top plot is from the code developed in this thesis and the other is the result from FLUENT coupled solver.

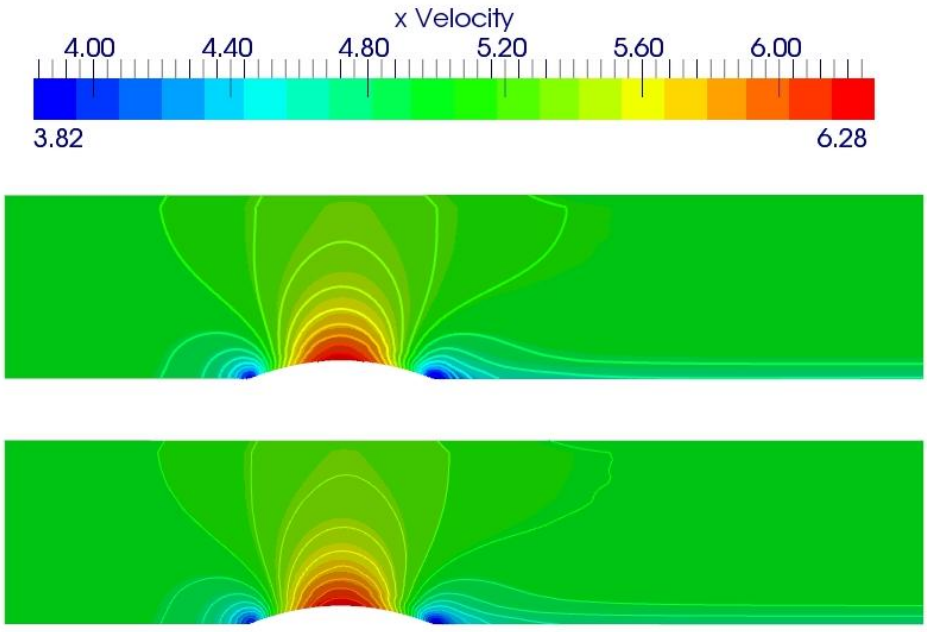

Figure 8: The contours plot of x velocity.

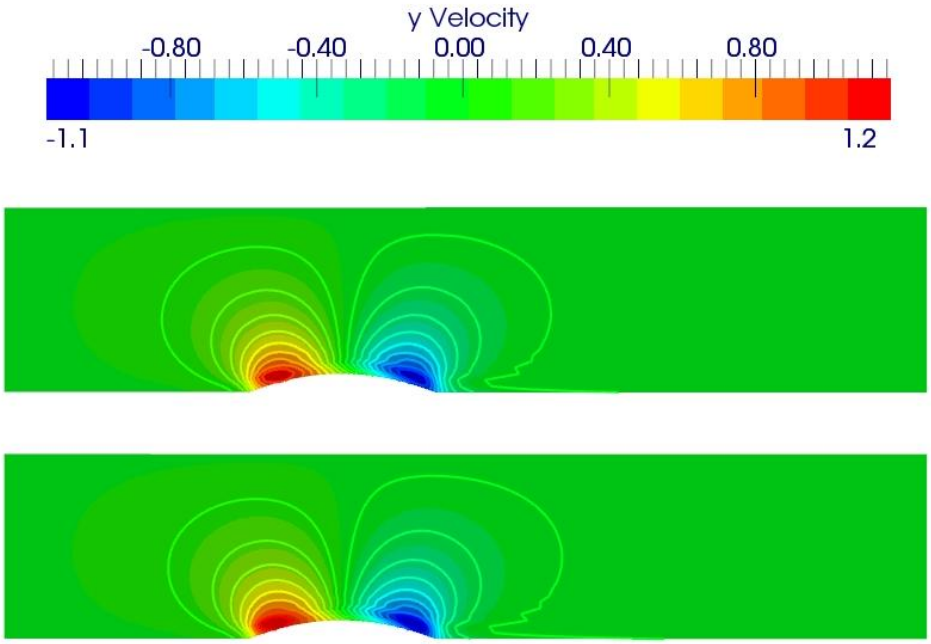

Figure 9: The contours plot of y velocity.

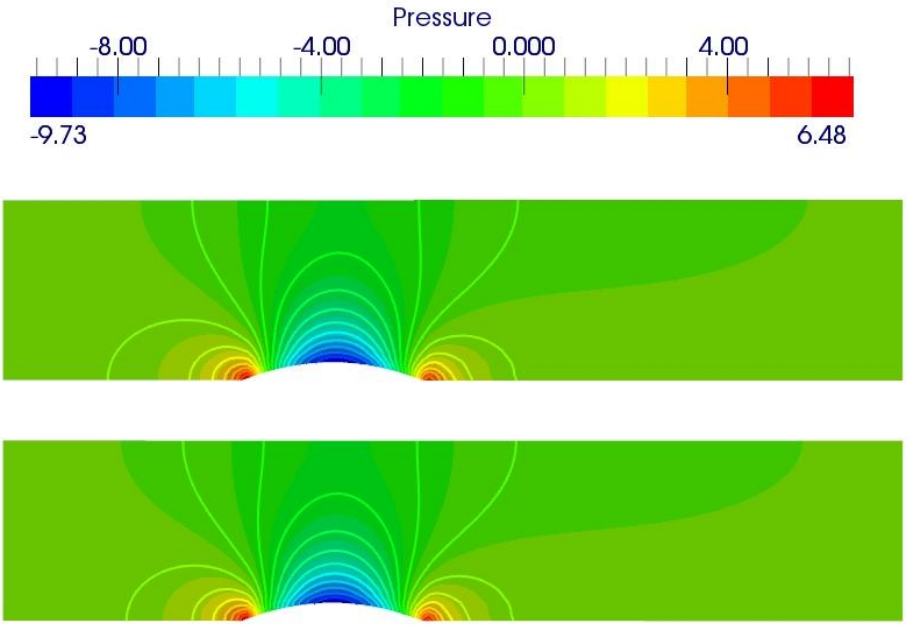

Figure 10: The contours plot of pressure.

4.1.2 Second Circular Bump Test Case

In this test case, flow over a circular bump, having 3 meters of length in x direction and a maximum height of 0.7 meters in y direction, in a 2D channel with the dimensions of $x = 15m$ and $y = 3m$ is considered. The fluid properties of air are used with density and viscosity values of 1.225 kg/m^3 and 1.7894 kg/ms, respectively. The geometry and mesh structure is given in Figure 11. Air enters the channel at 10 m/s and exit is given as pressure outlet. The results are compared with the coupled solver of FLUENT code. The convergence is controlled as in Eqn. (102).

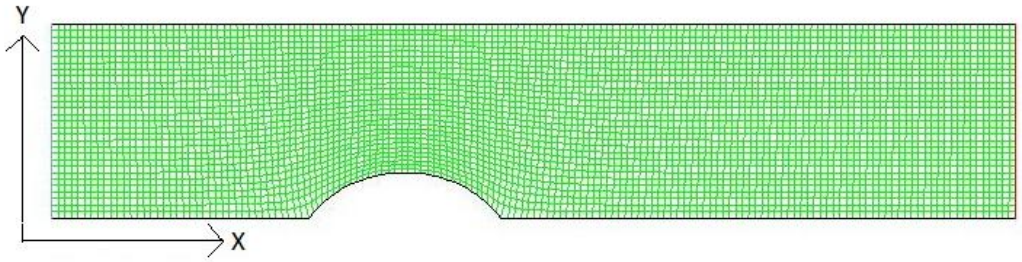

Figure 11: Mesh structure for the second circular bump test case.

In each figure below, the top plot is from the code developed in this thesis and the other is the result from FLUENT coupled solver.

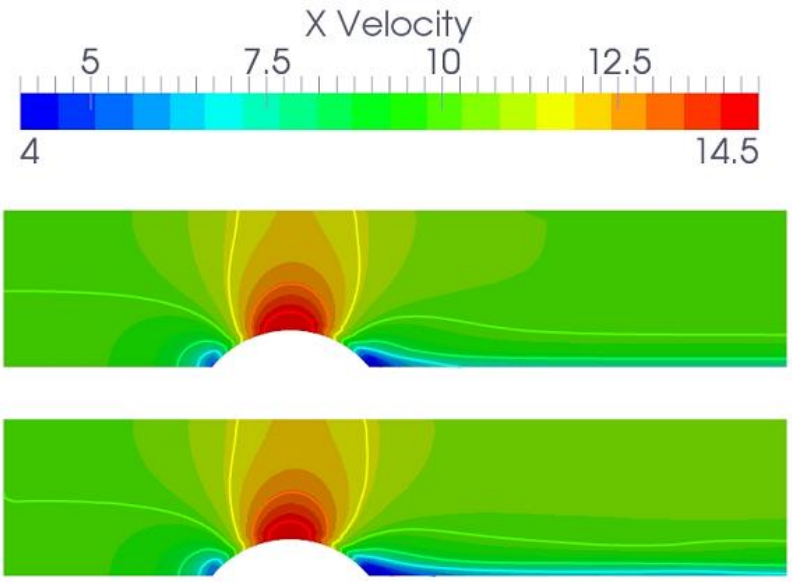

Figure 12: The contours plot of x velocity.

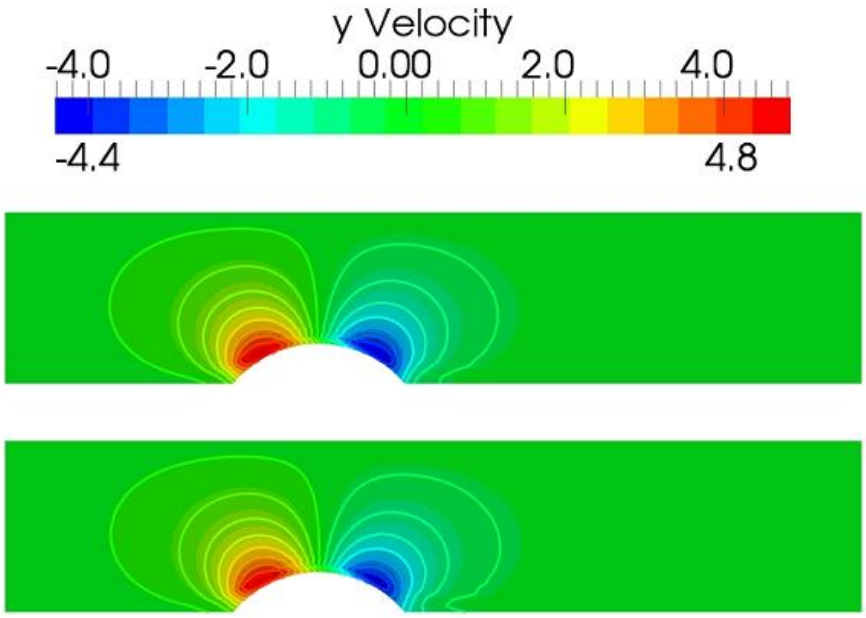

Figure 13: The contours plot of y velocity.

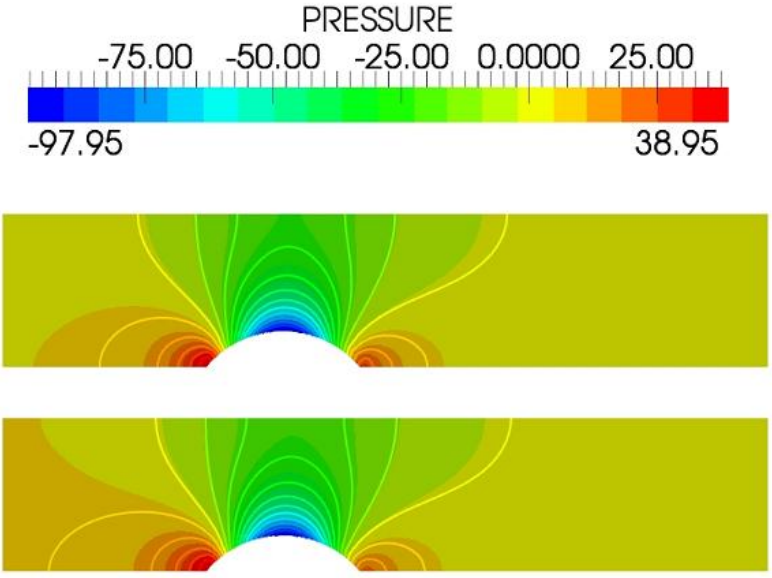

Figure 14: The contours plot of pressure.

In this test case with a false transient time step of 1.0, residuals reach to a value of 10^{-6} at below 400 iterations and continue to decrease. FLUENT solver couldn't reach values below 10^{-5} and stalled at these values. The results are compared in Figure 12, Figure 13 and Figure 14. The two results are in good agreement.

4.2 Step Test Case

In this test case, flow over a down step is considered. The geometry and the mesh structure of this problem are given in Figure 15. The solution grid consists fully of triangle elements. The convergence of the case is tested as given in Eqn. (102).

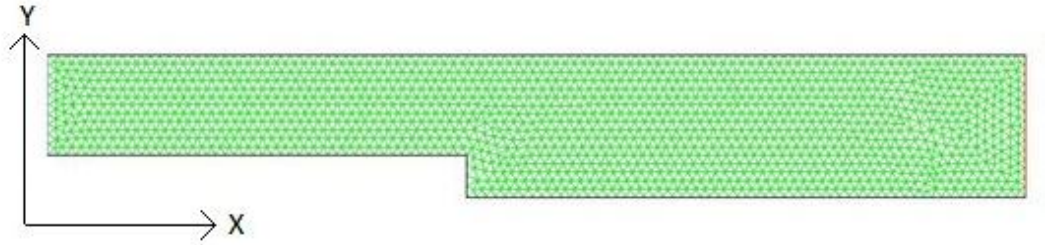

Figure 15: Grid structure for step test case.

Air enters the channel with speed of 1 m/s and exits from pressure outlet boundary. This problem converges below 60 iterations with false time step of 0.7. Ccontinuity residuals from FLUENT code do not get below the value of 10^{-4}. Results are given in Figure 16, Figure 17 and Figure 18. They are in very good agreement.

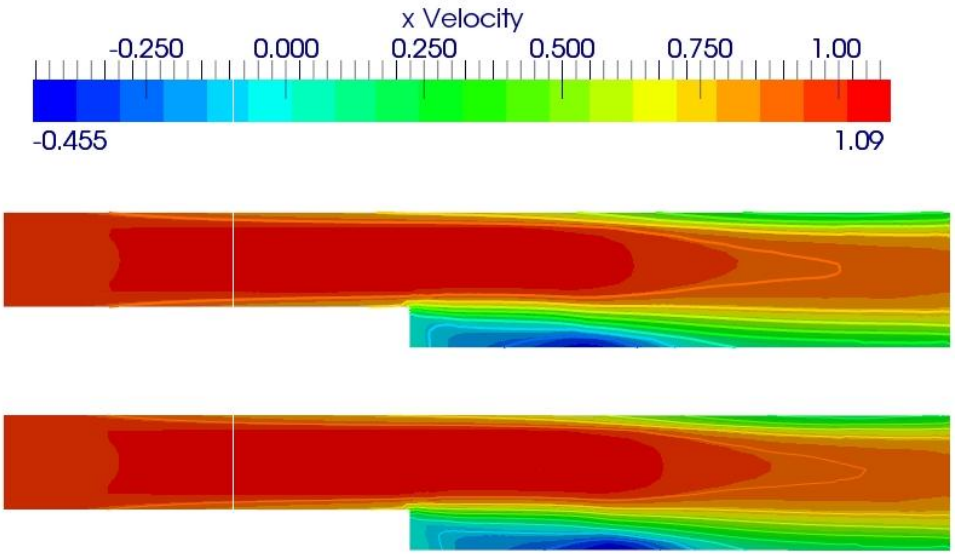

Figure 16: The contours plot of x velocity.

In each figure below, the top plot is from the code developed in this thesis and the other is the result from FLUENT coupled solver.

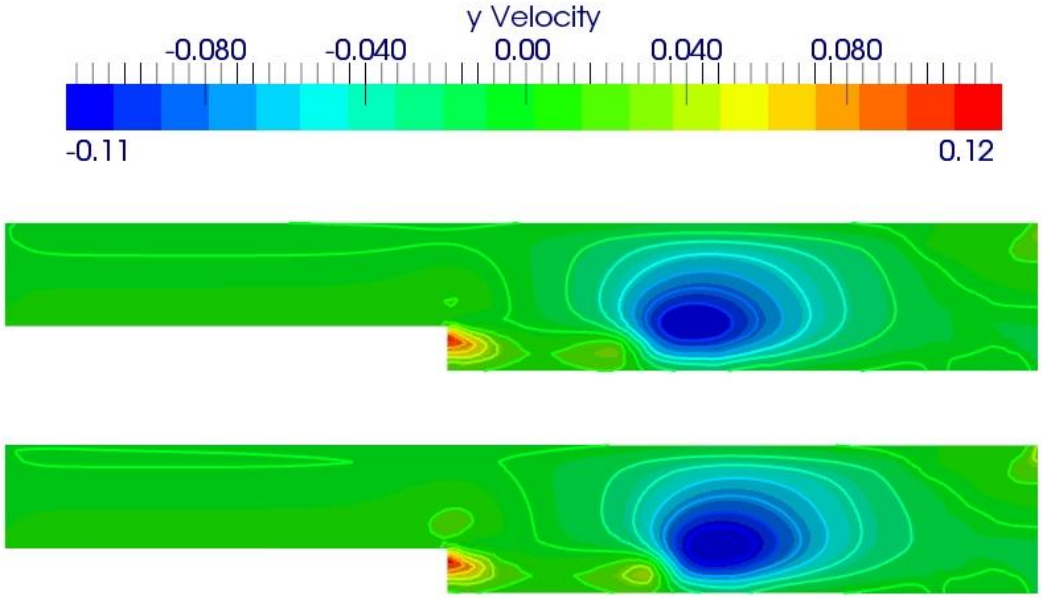

Figure 17: The contours plot of y velocity.

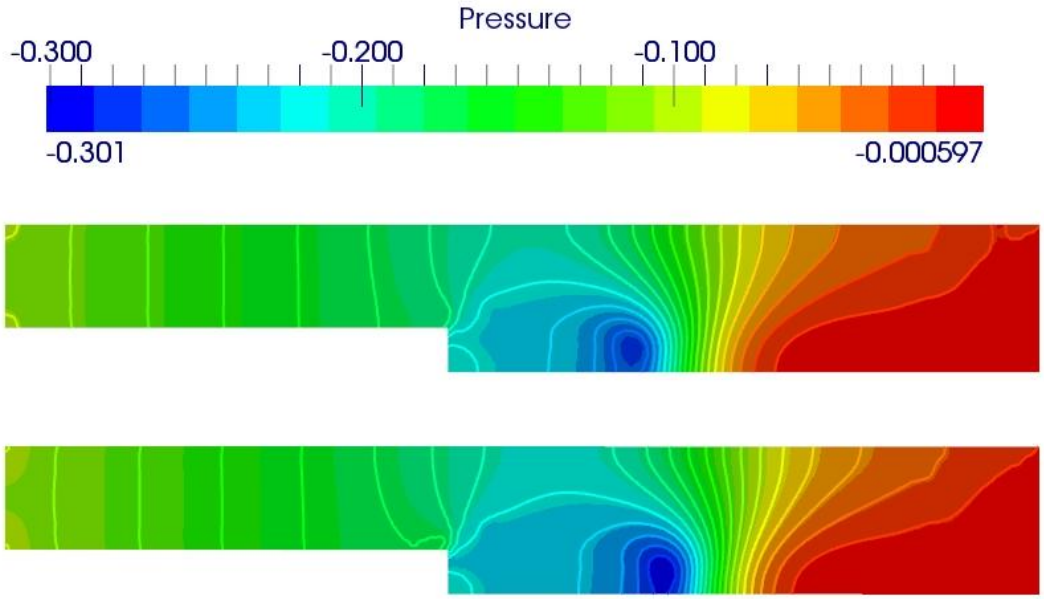

Figure 18: The contours plot of pressure.

4.3 Cavity Test Case

In this test case, air enters with a speed of 2 m/s into 2D channel with the dimensions of $x = 20$ meters and $y = 1$ meter, and in the middle of the channel there is a cavity of 2 meters long and 2 meters in depth. Outlet of the channel is modelled as pressure outlet again. The geometry and mesh of the test case are given in Figure 19.

The convergence of the case is tested as given in Eqn. (102). All the residuals reached below 10^{-6} again after 100 iterations. FLUENT coupled solver stalled around 10^{-5}. The results are given in Figure 20, Figure 21 and Figure 22.

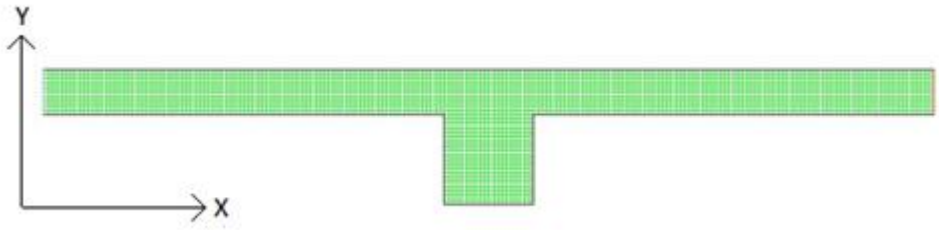

Figure 19: Mesh structure for cavity test case.

In each figure below, the top plot is from the code developed in this thesis and the other is the result from FLUENT coupled solver.

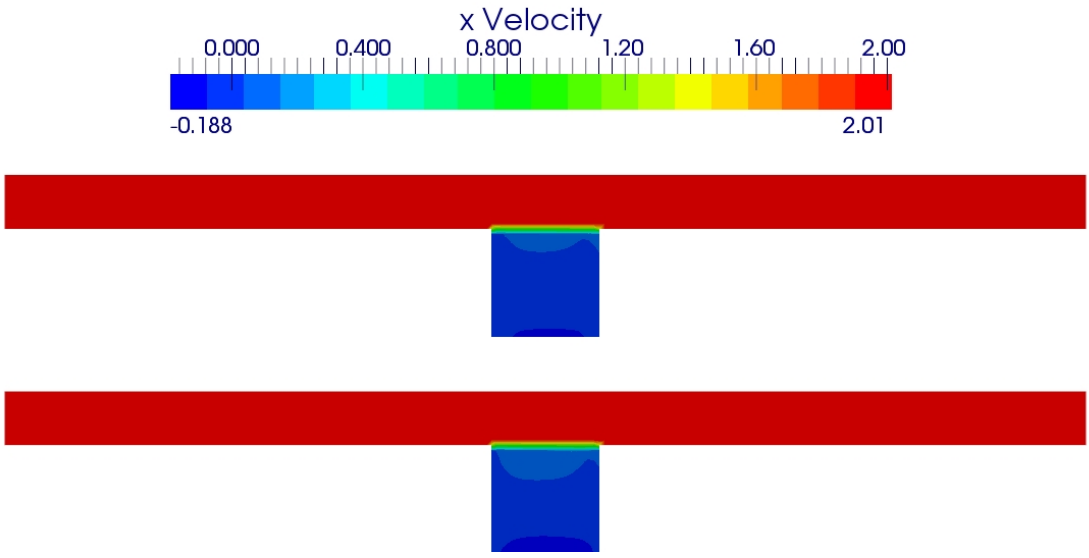

Figure 20: The contours plot of x velocity.

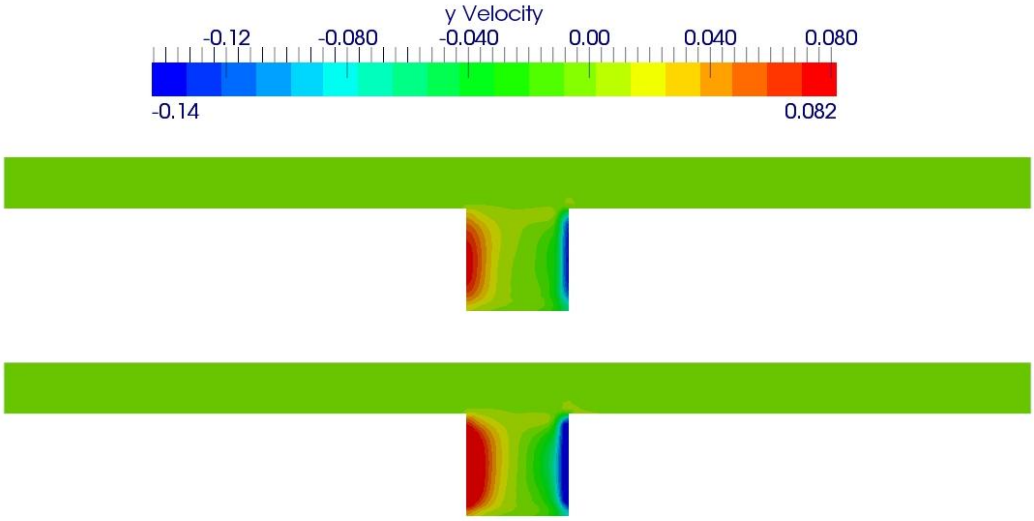

Figure 21: The contours plot of y velocity.

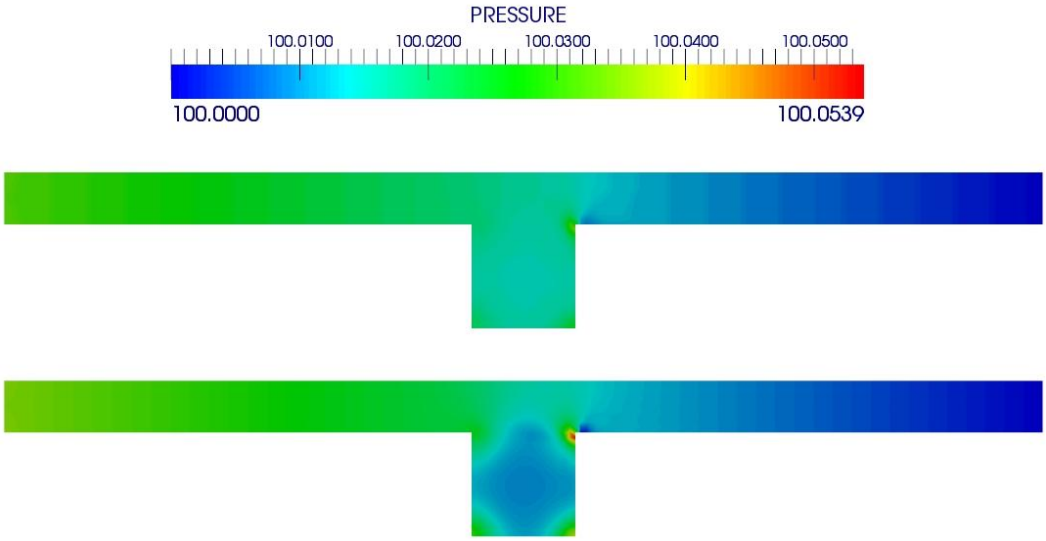

Figure 22: The contours plot of pressure.

4.4 Driven Cavity Test Case

In this test case, the fluid motion in the cavity is driven by the shear forces due to the moving wall boundary condition at the top of the cavity. The top face of the 1×1 meter cavity is moving at V m/s speed given as in Figure 23 where the fluid density is taken as 1 kg/m^3 and viscosity 0.001 Pa s. The reference data for this case is from Ghia [26] which provides tabular solution data for comparison.

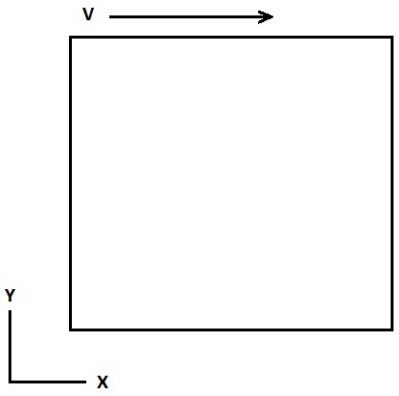

Figure 23: Geometry for driven cavity test case.

4.4.1 High Reynolds Driven Cavity Test Case

The top face of the 1×1 meter cavity is moving at 1 m/s speed and Reynolds number is 1000.

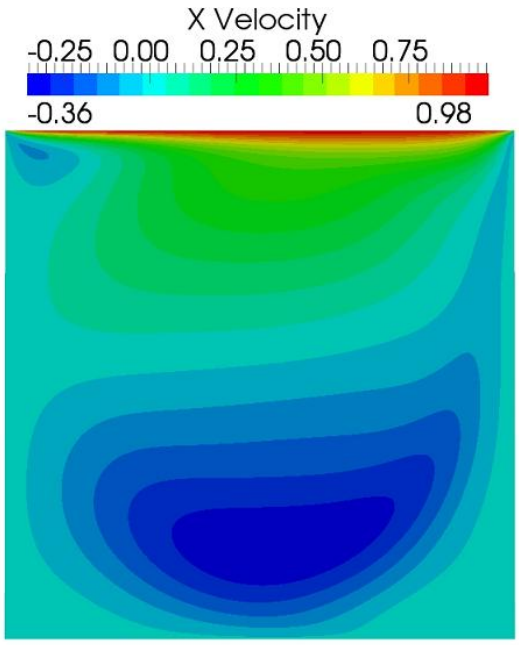

Figure 24: The contours plot of x velocity.

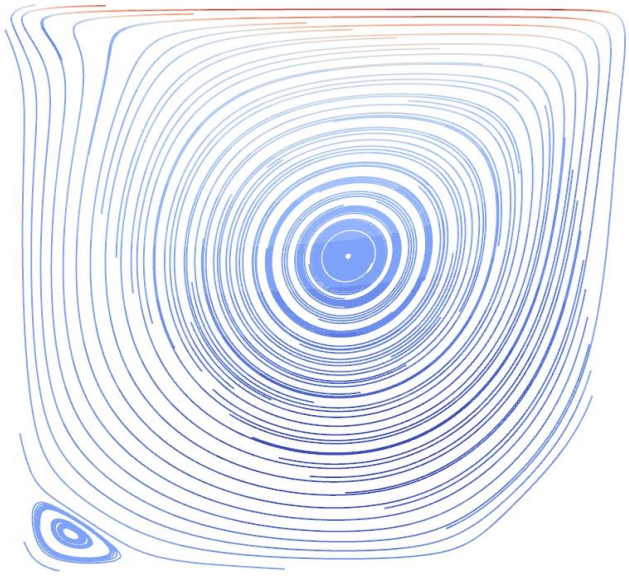

Figure 25: The streamline of the velocity

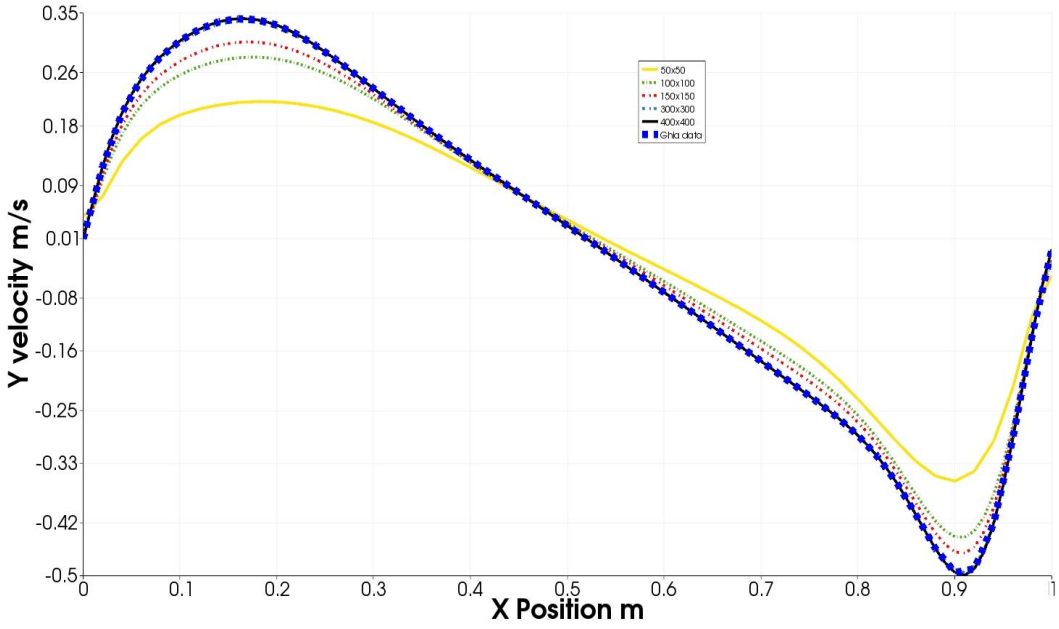

Figure 26: The comprasion of x velocity with Ghia [26].

The convergence of the case is tested as given as in Eqn. (102). All the residuals reached below 10^{-6} again after 100 iterations.

The test case is solved with 50 × 50, 100 × 100, 150 × 150, 300 × 300 and 400 × 400 grid resolutions. The result with the 400 × 400 grid resolution is matched exactly with the solution given in [26] as shown in Figure 26. The contours plot of x velocity is given in Figure 24 for 400 × 400 grid resolution case.

4.4.2 Low Reynolds Driven Cavity Test Case

The top face of the 1x1 meter cavity is moving at 0,1 m/s speed and Reynolds number is 100.

The convergence of the case is tested as in Eq.(102). All the residuals reached below 10^{-6} again after 100 iterations. The result with the 400 × 400 grid resolution is matched exactly with the solution in [26] as shown in Figure 28. The contours plot of x velocity is given in Figure 27 for 400 × 400 grid resolution of the test case.

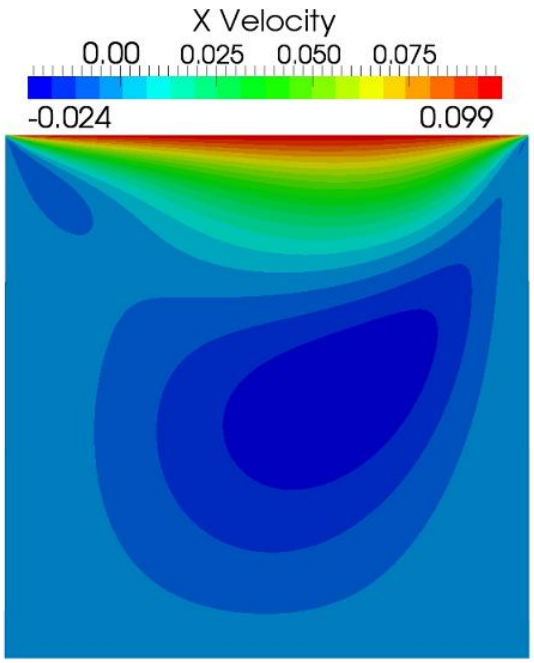

Figure 27: The contours plot of x velocity.

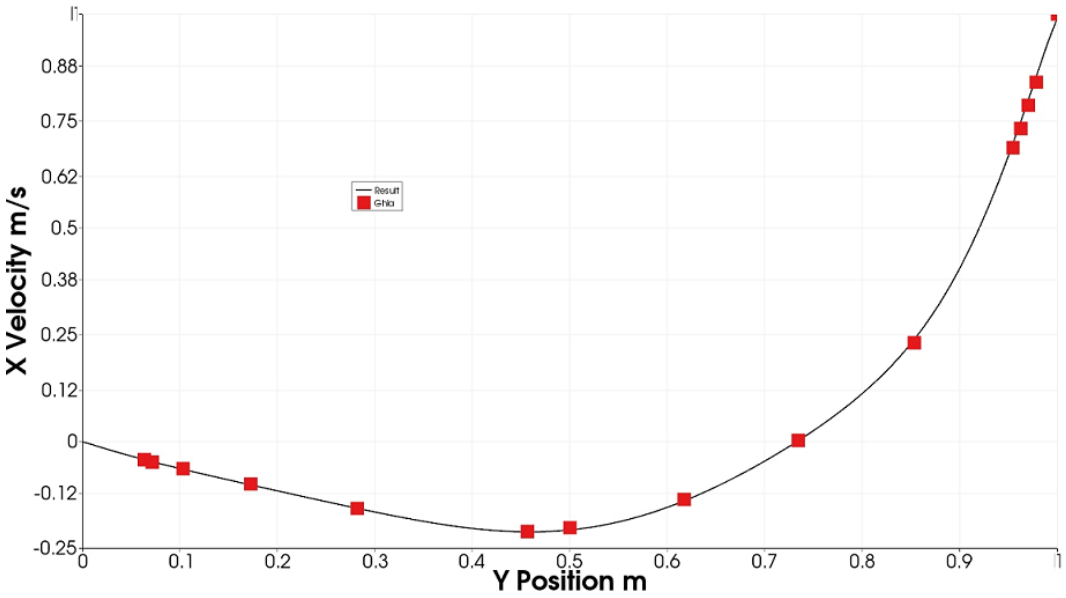

Figure 28: The comparison of x velocity with Ghia [26].

CHAPTER 5

CONCLUSIONS

The fully implicit coupled pressure-based method is one of the most important improvement in the pressure-based algorithms over the past 20-30 years. This coupled solver delivers lower run times in the range between 10 and 100 times. One of the important work on this method and the main source of this thesis is the work by Darwish, Sraf & Moukalled in 2009 [1]. This thesis is mainly based on the implementation of this algorithm developed in [1]. There are some differences between the equations presented in [1] and those derived in this thesis. The correct forms of the disretized equations are presented in this thesis.

The power of the scheme comes from the full implicitness of the derived system of equations. In the case of orthogonal grids, the only term on the right hand side of the equations is a pressure gradient term for correction of velocity difference equations. In the case of non-orthogonal grids, there are some cross diffusion correction terms on the right hand side of the equations which are not easy to discretize.

5.1 Recommendations and Future Work

This fully implicit algorithm even without a multigrid solver is very efficient and converges very quickly. However, better linear solvers with multi-grid capabilities will even improve the current convergence behavior.

Another possible improvement (generalization) on the scheme would be adding compressible flow solving capabilities. Currently there is no supersonic compressible implementation of this scheme and compressible pressure based coupled fully implicit solver would be very important as future work in CFD field.

The extension of this method to three dimensional geometries is straightforward and can easily be implemented in the current code framework developed for this thesis.

APPENDIX 1

ASSESMENT OF THE ALGORITHM IN CMPS CODE INFRASTRUCTURE

The methods explained in this thesis are coded on the object oriented code structure of the CMPS software developed by Bora Kalpaklı. The code structure provides an easy to use unstructured coding environment especially for CFD studies. The code base provides the user with ready data structures and solution variables like gradients and allow the change in the current algorithm without damaging the previous version.

In order to add a new implicit solver to the code, user should start with adding the new algorithm by filling the following lines in the solver part of the code. User should put the available functions in the "for" loop proper to the studied algorithm. Then user will change the code in the provided functions (in fact the original code in these functions will not change but the new code will be chosen at run time).

```
case PRESSURE_BASED_COUPLED:
{
        // Create a new equtaion solver
        pEqSolver = new CEqSolver(pDomain, pGui);
        // Update solution of Ax=b with the current solution variables
        pEqSolver->Update_x(0);

        for ( itNumber = 0; itNumber < maximumIteration; itNumber++ )
        {
                // Cunstruct matrix A and vector b in Ax=b
                pEqSolver->ConstructMatrix2D();
                // Solve the equation Ax=b with one of the availabale
                // solvers
                //pEqSolver->GaussSeidel();
                pEqSolver->GMRES();
                UpdateCellPropertiesPressureBased(pDomain);
                pEqSolver->Update_x(0);
                CalculateVelocityAndPressureGradient2D(pDomain);
                CalculateFaceMassFlow(pDomain);
        }

        delete pEqSolver;
}
```

```
break;
```

The algorithm given above follows the following steps:

1. Construct matrix **A** and source vector **B** with the currently available solution variables.
2. Solve the $\mathbf{AX} = \mathbf{B}$ for **X** using one of the methods available (Gauss-Seidel or GMRES).
3. Calculate required gradients of velocity and pressure values then calculate mass fluxes.
4. Return to 1 until convergence criteria is satisfied.

To fill the required functions, CMPS again provides some macros and methods. For example to loop over all the cells and their faces in the fluid domains, the part of code below is sufficient. This loop is used to calculate coefficients in the matrix **A** and vector **B**.

```
// Calculate momentum equation coefficients ------------------

 ZONE_LOOP(zone, pDomain) // Loop on zones of the solution domain
 {
     if ( pDomain->zones[zone].BCType == FLUID )
     {
       CELL_LOOP(cell, zone, pDomain)
       {
        cell_P = &pDomain->cells[cell];

        // Loop on cell faces
        for ( face = 0; face < cell_P->numOfFaces; face++ )
        {
           pFace = cell_P->faces[face];
           cell_F = cell_F = pFace->c1;

           // Calculate coefficients (a_F_uu,..) for neighbor cells and
           a_F_uu = MAX(-pFace->mass_flow, 0) + ......;
           .......

           // Update A matrix. Such as,
           A[cell][cell_f->index] = a_F_uu;

           // Update coefficients (a_P_uu, ..) for the cell

        }
        // Update A matrix. Such as,
        A[cell][cell + 2] = a_P_up;
        b[cell + 1] = b_P_v;
       }
     }
   }
 }
```

To get variables such as velocity components,

```
u = cell_P->cellPropertiesPD[phase].velocity[0];
v = cell_P->cellPropertiesPD[phase].velocity[1];
```

Velocity gradient,

```
du[0] = cell_P->cellPropertiesPD[phase].gradXVelocity[0];
du[1] = cell_P->cellPropertiesPD[phase].gradXVelocity[1];
(also cell_P->cellPropertiesPD[phase].gradPressure)
```

Geometric variables,

$$\frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}}$$ `= cell_P->Sf_by_D_pf[face];`

```
a_f = pFace->area;
```

REFERENCES

[1] Darwish, M., Sraj, I., and Moukalled, F., A coupled finite volume solver for the solution of incompressible flows on unstructured grids, *Journal of Computational Physics*, 228(2009) 180-201.

[2] National Aeronautics and Space Administration (NASA), Glenn Research Center, http://www.grc.nasa.gov/WWW/k-12/airplane/nseqs.html, 25/09/2012

[3] Bird, R.B., Stewart, W.E., and Lightfoot, E.N., *Transport of Phenomena*, John Wiley and Sons Inc., ISBN 0 471 07392 X.

[4] Bart, T., Numerical Methods for Conservation Laws on Structured and Unstructured Meshes, *NASA Ames Research Center, VKI Lecture Series*, March 2003.

[5] Versteeg, H. D., and Malalasekra, W., *An Introduction to Computational Fluid Dynamics, The Finite Volume Method*, Longman Group Ltd., 1995.

[6] G.B. Deng, J. Piquet, X. Vasseur, M. Visonneau, A new fully coupled method for computing turbulent flows, *Computers and Fluids*, 30 (2001) 445–472.

[7] R. Webster, An algebraic multigrid solver for Navier–Stokes problems, *International Journal for Numerical Methods in Fluids*, 18 (1994) 761–780.

[8] R.F. Hanby, D.J. Silvester, A comparison of coupled and segregated iterative solution techniques for incompressible swirling flow, *International Journal for Numerical Methods in Fluids*, 22 (1996) 353–373.

[9] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Physics of Fluids*, 8 (1965) 2182–2189.

[10] A.J. Chorin, Numerical solution of the Navier–Stokes equation, *Mathematics of Computation*, 22 (1971) 745–762.

[11] S.V. Patankar, D.B. Spadling, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, *International Journal of Heat and Mass Transfer*, 15 (1972) 1787–1806.

[12] P. Chow, M. Cross and K. Pericleous, `A Natural Extension of the Conventional Finite Volume Method into Polygonal Unstructured Meshes for CFD Applications', *Applied Mathematical Modelling*, 20, 170-183 (1996).

[13] P. M-Y. Chow, `Control Volume Unstructured Mesh Procedure for Convection-Diffusion Solidification Process', *Ph.D. Thesis, University of Greenwich, London* (1993).

[14] M. Thomadakis and M. Leschziner, `Numerical Simulation of Viscous Incompressible Flows Using a Pressure-Correction Method and Unstructured Grids', in *ECCOMAS '94 Conference* (1994).

[15] D. Pan, C.-H. Lu and J.-C. Cheng, `Incompressible Flow Solution on Unstructured Triangular Meshes', *Numerical Heat Transfer*, Part B, 26, 207-224 (1994).

[16] C. M. Rhie and W. L. Chow, `Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation', *AIAA Journal*, 21(11), 1525-1532 (1983).

[17] N. Drakos, PHYSICA User Guide, Centre for *Numerical Modelling and Process Analysis*, School of Computing and Mathematical Sciences University of Greenwich, London, UK, http://www.greenwich.ac.uk/~physica/phy2.10/user/uguide.htm, 25/09/2012

[18] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York (1980).

[19] A. J. Chorin, Numerical solution of navier-stokes equations. *Mathematics of Computation*, 22:745-762, 1968.

[20] D. Kuzmin, A Guide to Numerical Methods for Transport Equations, Friedrich-Alexander-University, Nürnberg.
http://www.mathematik.uni-dortmund.de/~kuzmin/Transport.pdf , 25/09/2012

[21] ANSYS FLUENT 12.0/12.1 Documentation, User Guide.

[22] Z. J. Chen, A. J. Przekwas, A coupled pressure-based computational method for incompressible/compressible flows, *Journal of Computational Physics*, 229 (2010) 9150–9165

[23] R.V Southwell, *Relaxation Methods in Engineering Science*, Oxford University Press, Oxford (1940).

[24] Singiresu S. Rao, *Applied Numerical Methods for Engineers and Scientists*, Prentice Hall Inc., 2002

[25] ANSYS FLUENT 18.4.2 Documentation, User Guide.

[26] U. Ghia, K.N. Ghia, and C.T. Shin, High-Re solution for incompressible flow using the Navier-Stokes equations and a multigrid method, *Journal of Computational Physics*, 48 (1982), 387–411

[27] Darwish, M.S., A new High-Resolution Scheme Based on the Normalized Variable Formulation, *Numerical Heat Transfer, Part B: Fundamentals*, vol. 24, pp. 353-371, 1993.

[28] Moukalled, F. and Darwish, M., New Family of Adaptive Very High Resolution Schemes, *Numerical Heat Transfer, Part B: Fundamentals*, vol. 34, pp. 215-239, 1998.

[29] Leonard, B.P., Locally Modified Quick Scheme for Highly Convective 2-D and 3-D Flows, Taylor, C. and Morgan, K. (eds.), *Numerical Methods in Laminar and Turbulent Flows*, Pineridge Press, Swansea, U.K., vol. 15, pp. 35-47,1987.

[30] Van Doormaal, J. P. and Raithby, G. D., Enhancement of the SIMPLE Method for Predicting Incompressible Fluid Flows, *Numerical Heat Transfer, Part B: Fundamentals*, vol. 7, pp. 147-163,1984.

[31] Jang, D.S., Jetli, R. and Acharya, S., Comparison of the PISO, SIMPLER and SIMPLEC Algorithms for the Treatment of the Pressure-Velocity Coupling in Steady Flow Problems, *Numerical Heat Transfer*, vol. 10, pp. 209-228,1986.

[32] Van Doormaal, J. P. and Raithby, G. D., An Evaluation of the Segregated Approach for Predicting Incompressible Fluid Flows, *ASME* Paper 85-HT-9, presented at the National Heat Transfer Conference, Denver, Colorado, August 4-7,1985

[33] Kershaw, D., The Incomplete Cholesky-Conjugate Gradient Method for The Iterative Solution of Systems of Linear Equations, *Journal of Computational Physics*, vol. 26, pp. 43-65, 1978.

[34] Stone, H.L., Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations, *SIAM J. Numer. Anal.*, vol. 5, No. 3, pp. 530-558,1968.

[35] Brandt, A., Multi-Level Adaptive Solutions to Boundary-Value Problems, *Math. Comp.*, vol. 31, pp. 333-390, 1977.

[36] Rhie, C.M., A Pressure Based Navier-Stokes Solver Using the Multigrid Method, *AIAA* paper 86-0207,1986.

[37] Shyy, W. and Chen, M.H.,"Pressure-Based Multigrid Algorithm for Flow at All Speeds," AIAA Journal, vol. 30, no. 11, pp. 2660-2669,1992.

[38] Shyy, W. and Braaten, M.E., Adaptive Grid Computation for Inviscid Compressible Flows Using a Pressure Correction Method, *AIAA* Paper 88-3566-CP, 1988.

[39] Issa, R.I., Solution of the Implicit Discretized Fluid Flow Equations by Operator Splitting, *Mechanical Engineering Report*, FS/82/15, Imperial College, London, 1982.

[40] Spalding D. B. Mathematical Modelling of Fluid Mechanics, Heat Transfer and Mass Transfer Processes, *Mech. Eng. Dept, Rept.* HTS/80/1, Imperial College of Science, Technology and Medecine, London, 1980.

[41] Acharya, S. and Moukalled, F., Improvements to Incompressible Flow Calculation on a Non- Staggered Curvilinear Grid, *Numerical Heat Transfer, Part B: Fundamentals*, vol. 15, pp. 131-152, 1989.

[42] Maliska, C.R. and Raithby, G.D., Calculating 3-D fluid Flows Using non-orthogonal Grid, *Proc. Third Int. Conf. on Numerical Methods in Laminar and Turbulent Flows*, Seattle, pp. 656-666,1983.