DEVELOPMENT OF A PRESSURE-BASED SOLVER FOR BOTH INCOMPRESSIBLE AND COMPRESSIBLE FLOWS

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

BY

KEREM DENK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR

THE DEGREE OF MASTER OF SCIENCE

IN

MECHANICAL ENGINEERING

DECEMBER 2007

Approval of the thesis:

DEVELOPMENT OF A PRESSURE-BASED SOLVER FOR BOTH INCOMPRESSIBLE AND COMPRESSIBLE FLOWS

submitted by **KEREM DENK** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Gradute School of **Natural and Applied Sciences** _____

Prof. Dr. S. Kemal İder
Head of Department, **Mechanical Engineering** _____

Asst. Prof. Dr. Cüneyt Sert
Supervisor, **Mechanical Engineering Dept., METU** _____

Examining Committee Members:

Prof. Dr. Kahraman Albayrak
Mechanical Engineering Dept., METU _____

Asst. Prof. Dr. Cüneyt Sert
Mechanical Engineering Dept., METU _____

Prof. Dr. Haluk Aksel
Mechanical Engineering Dept., METU _____

Asst. Prof. Dr. İlker Tarı
Mechanical Engineering Dept., METU _____

Assoc. Prof. Dr. Serkan Özgen
Aerospace Engineering Dept., 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.

Kerem Denk

ABSTRACT

DEVELOPMENT OF A PRESSURE-BASED SOLVER FOR BOTH INCOMPRESSIBLE AND COMPRESSIBLE FLOWS

Denk, Kerem

M.S., Department of Mechanical Engineering

Supervisor: Asst. Prof. Dr. Cüneyt Sert

December 2007, 58 pages

The aim of this study is to develop a two-dimensional pressure-based Navier-Stokes solver for incompressible/compressible flows. Main variables are Cartesian velocity components, pressure and temperature while density is linked to pressure via equation of state. Modified SIMPLE algorithm is used to achieve pressure-velocity coupling. Finite Volume discretisation is performed on non-orthogonal and boundary-fitted grids. Collocated variable arrangement is preferred because of its advantage on staggered arrangement in non-orthogonal meshes. Face velocities are calculated using Rhie-Chow momentum interpolation scheme to avoid pressure checkerboarding effect. The solver is validated by solving a number of benchmark problems.

Keywords: Navier-Stokes, Non-orthogonal Grids, Collocated Arrangement, Finite Volume Method, Pressure Based Methods, SIMPLE method,

ÖZ

SIKIŞTIRILAMAZ VE SIKIŞTIRILABİLİR AKIŞLAR İÇİN BASINÇ TABANLI BİR ÇÖZÜCÜSÜNÜN GELİŞTİRİLMESİ

Denk, Kerem

Yüksek Lisans, Makine Mühendisliği Bölümü

Tez Yöneticisi: Asst. Prof. Dr. Cüneyt Sert

Aralık 2007, 58 sayfa

Bu çalışmanın amacı, sıkıştırılamaz/sıkıştırılabilir akışlar için iki boyutlu, basınç tabanlı bir Navier-Stokes çözücüsünün geliştirilmesidir. Çözücüdeki ana değişkenler Kartezyen hız bileşenleri, basınç ve sıcaklıkken, yoğunluk basınca duru denklemi yoluyla bağlıdır. Basınç-hız bağlantısı değistirlimiş SIMPLE algoritması ile elde edilmiştir. Diklik şartı olmayan ve sınır-uyumlu çözüm ağları üzerinde Sonlu Hacim ayrıklaştırma yöntemi uygulanmıştır. Karışık yerleşimli değişken düzeni yerine, elemanları diklik şartı olmayan çözüm ağlarına daha uygun olmasından dolayı, eş yerleşimli değişken düzeni tercih edilmiştir. Dama tahtası etkisini engellemek için Rhie-Chow momentum iç değerbiçim yöntemi kullanılmıştır. Çözücünün geçerliliğini kontrol etmek için bir kaç değerlendirme problemi çözülmüştür.

Anahtar Kelimeler: Navier-Stokes, Diklik Şartı Olmayan Çözüm Ağları, Eş Yerleşimli Değişken Düzeni, Sonlu Hacim Metodu, Basınç Tabanlı Metodlar, SIMPLE

Dedicated to my parents . . .

ACKNOWLEDGMENTS

Firstly, I would like to thank my supervisor, Dr. Cüneyt Sert, without whom thesis would have eventuated. His support has always been gratefully appreciated, and his knowledge and scholarship has proved invaluable.

I would like to thank Çağlar KIRAL, Gökhan ARAN and other colleagues at TAI for their patience and support.

I also thank Onur BAŞ for his most noteworthy support.

I am grateful to my beloved Jale, for her patient and spiritual support which kept me going on.

Finally, I want to express my feelings of love for my parents, who were always encouraging and affectionate.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ v

ACKNOWLEDGMENTS vii

TABLE OF CONTENTS viii

LIST OF FIGURES xi

LIST OF SYMBOLS xiii

CHAPTERS

1 INTRODUCTION 1

1.1 Background 1

1.2 Incompressible Flows 2

1.3 Compressible Flows 4

1.4 Incompressible/Compressible Flows 5

1.5 The Choice of Grid 6

1.5.1 Orthogonal Grids 6

1.5.2 Block Structured Grids with Overlapping Blocks 7

1.5.3 Boundary-Fitted Non-Orthogonal Grids 8

1.6 The Choice of Velocity Components 8

1.6.1 Grid-Oriented Velocity Components 8

1.6.2 Cartesian Velocity Components 9

1.7 The Choice of Variable Arrangement 10

1.7.1 Staggered Arrangement 10

1.7.2 Collocated Arangement 10
1.8 Present Thesis 11
2 NUMERICAL METHOD 12
2.1 Governing Equations 12
2.1.1 Continuity Equation 12
2.1.2 Momentum Equation 13
2.1.3 Energy Equation 13
2.2 Spatial Discretisation 13
2.2.1 Notation 14
2.2.2 Non-orthogonal Mesh Geometry 15
2.3 Differencing Schemes 16
2.3.1 First Order Upwind Differencing Scheme (UDS) 16
2.3.2 Second Order Central Differencing Scheme (CDS) 17
2.4 Discretisation of Advective Terms in Momentum Equations 17
2.5 Discretisation of Diffusive Terms in Momentum Equations 18
2.6 Approximation of Pressure and Source Terms in Momentum Equations . 20
2.7 Discrete Form of Momentum Equation 22
2.8 Discretisation of Continuity Equation 24
2.9 Interpolation of the Face Velocities 25
2.10 Solution of the Navier-Stokes Equations 26
2.10.1 SIMPLE Scheme 27
2.10.2 Pressure Correction Equation 29
2.10.3 Overall Solution Procedure 32
2.11 Implemenetation of Boundary Conditions 33
2.11.1 Wall Boundaries 34
2.11.2 Subsonic Inflow Boundaries 36
2.11.3 Supersonic Inflow Boundaries 36
2.11.4 Subsonic Outflow Boundaries 36
2.11.5 Supersonic Outflow Boundaries 37

2.11.6 Symmetry Boundaries 37
3 RESULTS AND DISCUSSION 39
3.1 General 39
3.2 Incompressible Test Cases 40
3.2.1 Test Case 1: Lid-Driven Skewed Cavity 40
3.2.2 Test Case 2: Laminar Flow Through a Gradual Expansion . . . 45
3.3 Compressible Test Cases 48
3.3.1 Test Case 1: 2D Laminar Flat Plate 48
3.3.2 Test Case 2: 2D Converging-Diverging Nozzle 50
4 CONCLUSIONS 55
REFERENCES 56

LIST OF FIGURES

FIGURES

1.1 Mach number values and their flow regimes 2
1.2 Orthogonal Grid [31] 7
1.3 Overlapping (composite) Grid [31] 7
1.4 2D, Structured, Non-orthogonal Grid [31] 8
1.5 Grid-Oriented Velocity Components 9
1.6 Cartesian Velocity Components 9
1.7 Staggered Variable Arrangement 10
1.8 Collocated Variable Arrangement 11

2.1 Compass notation 14
2.2 Conversion between compass and index notation 14
2.3 Two-dimensional cell geometry 15
2.4 Face area vectors of auxiliary cell for face e 19
2.5 Auxiliary cell for face e 19
2.6 Notation used in calculation of source terms 21
2.7 Notation used in calculation of cell face velocities 26
2.8 Inflow Boundary 34
2.9 Wall Boundary 35
2.10 Outflow Boundary 36
2.11 Symmetry Boundary 37

3.1 Geometry for the test case 2 40
3.2 u velocity profiles along line A-B for 45-degree skewed cavity solution at Re=1000 41
3.3 v velocity profiles along line C-D for 45-degree skewed cavity solution at Re=1000 41
3.4 Streamlines for 45-degree skewed cavity solution at Re=1000 42
3.5 u velocity profiles along line A-B for 30-degree skewed cavity solutions at Re=1000 43
3.6 v velocity profiles along line C-D for 30-degree skewed cavity solutions at Re=1000 43
3.7 Geometry of gradual expansion 45

3.8 41x41 grid for gradual expansion problem at Re=10 45
3.9 Pressure values at solid wall for Re=10 46
3.10 Pressure values at solid wall for Re=100 47
3.11 2D boundary layer over a flat plate 48
3.12 Grid used in boundary layer solution 48
3.13 Comparison of numerical solution with the Blasius' solution 49
3.14 Geometry of 2D Converging-Diverging Half Nozzle 50
3.15 61x21 mesh used in 2D Converging-Diverging Half Nozzle problem 50
3.16 Reversed Flow at the Upper Right Corner of the Nozzle - FLUENT Solution . . 52
3.17 Reversed Flow at the Upper Right Corner of the Nozzle - Present Solution 52
3.18 Static pressure at symmetry boundary 53
3.19 Static temperature at symmetry boundary 53
3.20 Mach Number at symmetry boundary 54
3.21 Density at symmetry boundary 54

LIST OF SYMBOLS

ROMAN SYMBOLS

F	Flux vector
$\dot{m}$	Mass flux
N_x	Maximum cell number along x direction
N_y	Maximum cell number along y direction
p	Pressure
Re	Reynolds number
$\vec{u}$	Velocity vector
u	X axis component of velocity vector
v	Y axis component of velocity vector
$\vec{n}$	Unit normal vector
$\vec{i}$	Unit vector along x axis
$\vec{j}$	Unit vector along y axis
$\mathbf{A}$	Area vector
S	Source term
a	Coefficient of discretised momentum equation
c	Discretised momentum equation source term
x	X coordinate of the node
y	Y coordinate of the node
$\vec{r}$	Diagonal vector of CV
b	Coefficient of pressure correction equation

GREEK SYMBOLS

Γ	Diffusivity
μ	Dynamic viscosity
Ω	Volume
λ	Linear interpolation factor
∇	Gradient
ρ	Density
ϕ	Scalar variable
α	Relaxation parameter

SUBSCRIPTS

ϕ	Related with scalar variable
p	Related with pressure
P	Cell center of CV
E	East neighbour of CV
W	West neighbour of CV
N	North neighbour of CV
S	South neighbour of CV
e	East face of CV
w	West face of CV
n	North face of CV
s	South face of CV
c	Constant part
u	Related with u velocity
v	Related with v velocity
ne	North-East corner of CV
se	South-East corner of CV
nw	North-West corner of CV
sw	South-East corner of CV

SUPERSCRIPTS

$*$	Guessed value
$'$	Corrector value
D	Related with diffusion
C	Related with advection

COMMONLY USED ACRONYMS

CFD	Computational Fluid Dynamics
CV	Control Volume
FV	Finite Volume
FD	Finite Difference

SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
PDE	Partial Differential Equation
CDS	Central Differencing Scheme
UDS	Upwind Differencing Scheme

CHAPTER 1

INTRODUCTION

1.1 Background

Computational Fluid Dynamics (CFD) plays a great role in the study of thermofluidic transport phenomena. Due to the difficulty and cost of experimental studies and the limited capability of analytical solutions, numerical methods have become crucial tools for the study of fluid mechanics and heat transfer problems. An important step in the use of any numerical method involves the discretisation of governing partial differential equations. In other words, the governing PDEs are written as a set of algebraic equations that can be solved by a computer.

To predict flows at different Mach number regimes, many flow solvers have been developed. Most solvers deal with incompressible or compressible flow regimes only. For instance, if incompressible solvers are used to predict the flow in compressible regime or vice versa, predicted values become unrealistic and solution becomes totally wrong. The mathematical character is the major difference between incompressible and compressible flow equations. Compressible steady flow equations are hyperbolic which means flow characteristics travel at finite propagation speeds. On the contrary, incompresible steady flow equations have a mixed parabolic/elliptic character. Figure 1.1 shows Mach number values and the corresponding flow regimes.

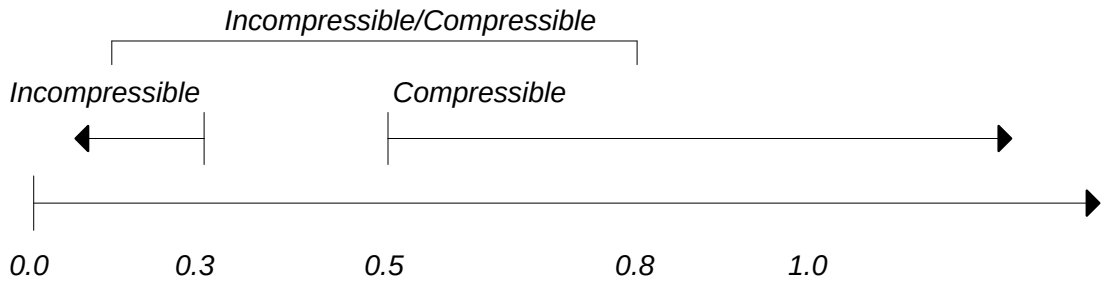

Figure 1.1: Mach number values and their flow regimes

On the other hand, many engineering applications include both incompressible and compressible flow regions. Hence, the development of a flow solver that can deal with both incompressible and compressible flows became a crucial need for adequate numerical simulations.

In this study, a solver is developed to predict both incompressible and compressible flows.

1.2 Incompressible Flows

In incompressible flow methods, it is assumed that the velocities of fluid particles are small relative to the speed of sound which makes the density changes following a particle negligible ($\frac{\partial \rho}{\partial t} = 0$). Viscosity is generally taken to be a constant instead of being a function of temperature. With these two assumptions, the energy equation becomes decoupled from the continuity and momentum equations. Therefore, there are three non-linear equations involving three primitive variables which are p, u, v. In addition to these, the speed of sound for incompressible flows is infinite which is the reason for the elliptic behaviour of incompressible Navier-Stokes equations.

In the literature, there are mainly three approaches to predict incompressible flows; vorticity-streamfunction based methods, artificial compressibility methods and pressure correction methods.

In vorticity-stream function based methods, dependent variables in the governing equations

of compressible flows are replaced with vorticty and streamfunction which are new unknowns. Stream function is used to solve for the velocity field. After calculation of u and v, p is calculated with the help of streamfunction. Although the vorticity-streamfunction is effective for two dimensional flows, in three dimensional space, the method is ineffective. In three dimensional space, the number of unknowns increase from four to six which are three components of vorticity and streamfunction. Hence, with the problems of implementing the boundary conditions and compressibility effects, this method loses its popularity.

In the artificial compressibility approach, first presented by Chorin [1], a time derivative of pressure is added to the continuity equation. The 'unsteady' pressure field is updated using a pseudo time step to satisfy the conservation of mass. After the steady-state is reached, $\frac{\partial p}{\partial t}$ becomes zero which means that the continuity is satisfied.

First multidimensional pressure correction method is developed by Harlow and Welsh [2]. They were insterested in transient flow and used explicit time integration. After a short time, it was understood that explicit time integration is inefficient in steady-state flows. Then, Patankar and Spalding [3] introduced an implicit pressure correction method. The correction of pressure field using the continuity equation is the main phenomena in the proposal of Patankar and Spalding. The method is also known as SIMPLE (Semi Implicit Method for Pressure-Linked Equations).

The most popular way of solving incompressible flows is the pressure correction approach. The SIMPLE algorithm has been used by Kobayashi and Pereira [5], Peric and Kesler [6], Lapworth [7], Peric [8], Jessee and Fiveland [9], Choi et al. [10], Jeng and Liou [11], Rhie and Chow [12], Melaaen [13],[14], Mathur and Murthy [15]. After the SIMPLE algortihm, a number of extensions (including SIMPLER by Patankar [4], SIMPLEC by van Doormal and Raithby[16], SIMPLEST by Spalding[17], PISO by Issa[18]) were developed for solving incompressible flows.

1.3 Compressible Flows

Two dimensional compressible flow equations involve four differential equations (two momentum, energy and continuity) and a scalar equation for the equation of state. The speed of sound is finite this time which means the pressure perturbations require more care in the formulation. The governing equations have hyperbolic behaviour in compressible steady flows.

A two step, predicor corrector method which uses two different directional sweeps to calculate the fluxes was proposed by McCormack. Various combinations of forward and backward differencing schemes are used in predictor and corrector steps. However, to achieve stability, artificial viscosity is needed in this method.

Jameson et al. [19] stated a finite volume formulation which uses artificial dissipation terms and centrally discretised fluxes to avoid oscillations near steep gradients. In addition to these, to improve the solution in time, Runge-Kutta scheme is used in this method. Because of good shock capturing capabilities, this method is very popular in compressible flow predictions.

Other methods can be categorized in one group called upwind schemes. In order to model the behaviour of the flow more precisiely, this scheme gives a directional tendency like upwind differencing or backward differencing. Steger and Warming [20] proposed a scheme which splits the flux vector into two parts. Flux vector, which has negative eigenvalues is backward differenced and the second one that has positive eigenvalues is forward differenced. With this method, only the physically acceptable data is used. The most popular scheme in upwind schemes today is Godunov's method [21] which provides very good shock capturing performance because it is the result of an exact solution of the Euler equations. However, it is cumbersome to compute. Hence Roe scheme [22] can also be used instead.

1.4 Incompressible/Compressible Flows

Flows between Mach number of 0.1 and 0.8 have both incompressible and also compressible character. It is obviously incorrect to use incompressible methods to deal with flows that have regions strong compressible character. Similarly, in low Mach number regimes, compressible solvers have some convergence difficulties.

Incompressible/compressible flow solvers are capable of dealing with flows in any Mach number i.e between 0 and ∞. While having good shock capturing capabilities at high Mach number regimes, it reduces to incompressible solver as Mach number approaches to zero.

A method for incompressible/compressible flows is pioneered by Harlow and Amsden [23]. They have proposed a modified Implicit Continuous Eulerian. Then, Karki and Patankar used the modified SIMPLE scheme (SIMPLE is actually developed for incompressible flows) to solve compressible flows. They also adopted this scheme to non-orthogonal grids. The results were satisfying in some 1D and 2D benchmark problems. After these works, van Doormal [16] modified the variants of SIMPLE and applied them to subsonic/supersonic flow around a flat plate. All these methods mentioned about before were developed for staggered grids. Time modifications to compressible SIMPLE algorithm is performed by Zhou and Davidson [25], and also by Demirdzic, Lilek and Peric [26] to deal with non-orthogonal grids with collocated variable arrangement. McGuirk and Paige [27] also modified the SIMPLE scheme in a new approach. They used the face mass fluxes as new dependent variables instead of velocities. This approach provided very satisfying shock capturing capabilities to compressible SIMPLE algorithm although in one dimensional basis. Kobayashi and Pereira [5] introduced a new approach to calculate the mass flow rates at cell faces called Essentially Non-Oscillatory schemes which has good shock capturing capabilities but also high computational cost. The approach of McGuirk and Paige was extended to collocated variable arrangement by Lien and Leschziner [28].

In this study, a pressure based flow solver is developed which is capable solving the Navier-Stokes

equations in both the incompressible and compressible flow regimes. For pressure-velocity coupling, SIMPLE scheme of Patankar and Spalding [3] is used which is actually developed for staggered variable arrangement and hence for structured grids is used. During the development process, some choices were made among some possible options. In the following, these choices are discussed in detail.

1.5 The Choice of Grid

In this study, structured grids are used. Before discussing the choice for the grid, it is useful to introduce types of structured grids briefly.

1.5.1 Orthogonal Grids

In orthogonal grids, two sets of lines are perpendicular to each other. Orthogonal grids are popular because with the new grid generation techniques, such as elliptic differential equations, even complex geometries can be mapped with orthogonal or nearly orthogonal grids. Another and most important simplification of orthogonal grids is vanishing of curvature terms in governing equations. On the other hand, on the boundaries of complex geometries, some error terms occur in case of nearly orthogonal grids. This is the main disadvantage of orthogonal grids.

In Figure 1.2, an example to orthogonal grids is shown.

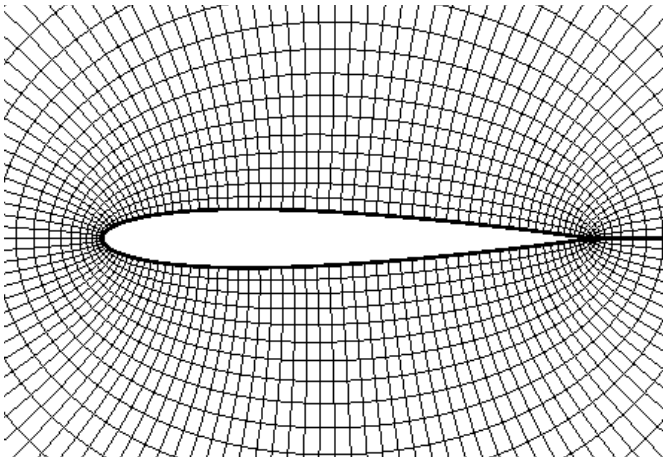

Figure 1.2: Orthogonal Grid [31]

1.5.2 Block Structured Grids with Overlapping Blocks

This type of grids are sometimes called *composite* (Figure 1.3) or *Chimera* grids. Boundary conditions for one block are obtained from other (overlapped) block in overlapping region. Easy usage in complex geometries is the main advantage of this approach. Moreover, block-structured grids with overlapping blocks can be used to follow moving bodies. One block is moving with the moving body when the other one covers the surroundings of the moving body. The disadvantage of this approach is that the coupling and programming of the grids is complicated. And also, at the interfaces, it is difficult to maintain the conservation.

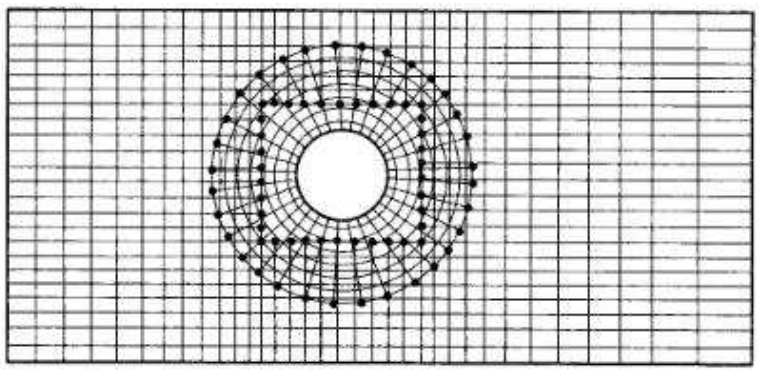

Figure 1.3: Overlapping (composite) Grid [31]

1.5.3 Boundary-Fitted Non-Orthogonal Grids

The most often used type of grids is boundary-fitted non-orthogonal grids because they can be easily adapted to any complex geometry. Because the grid lines follow the boundaries (grid lines are also boundary lines at boundaries), the implementation of boundary conditions also becomes easier than block-structured grids with overlapping blocks. The main disadvantage of this type of grids is the extra non-orthogonality terms such as cross derivatives in transformed equation which may cause unphysical solutions. Hence, it is desirable to achieve orthogonality at least at near boundary regions where the cross-derivatives are the largest and the gradients are the steepest [29].

In this study, boundary-fitted non-orthogonal grid formulation is chosen because of the difficulty and cost of implementation of orthogonal grids to complex geometries.

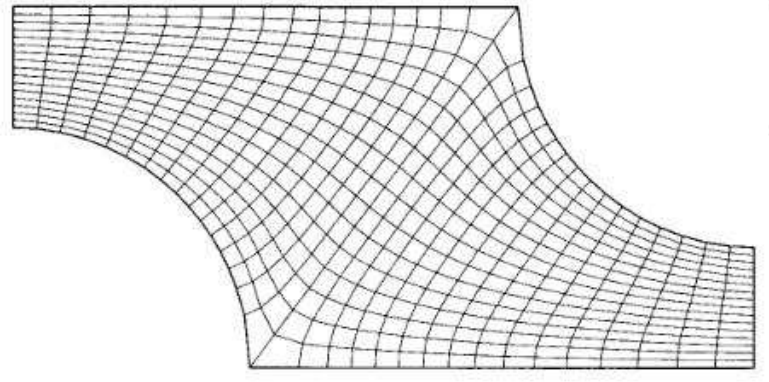

Figure 1.4: 2D, Structured, Non-orthogonal Grid [31]

1.6 The Choice of Velocity Components

1.6.1 Grid-Oriented Velocity Components

Grid-oriented velocity components change direction as the orientation of the grid chages. Because of this, non-conservative source terms occur in the momentum equations when grid-oriented velocity components are used. Also, curvilinear coordinates involve curvature terms

and higher-order coordinate derivatives, which cause large numerical errors. If curvilinear coordinate system is used, the difference in the grid direction from one point to another must be as small as possible. Figure 1.5 displays grid-oriented velocity components.

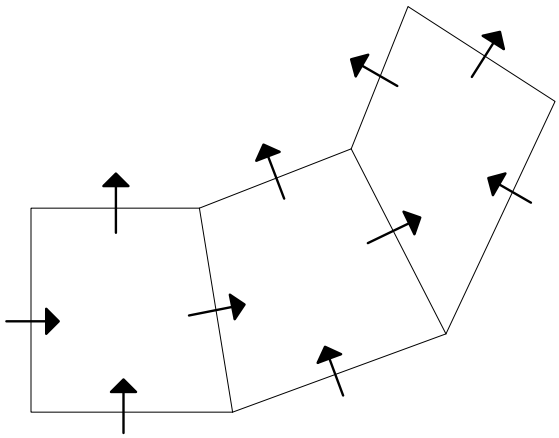

Figure 1.5: Grid-Oriented Velocity Components

1.6.2 Cartesian Velocity Components

Cartesian velocity components are always in the direction of Cartesian coordinates (Figure 1.6). If finite difference methods are used, one must apply the suitable forms of gradient and divergence terms for non-orthogonal grids. If finite volume methods are used, Cartesian coordinate system is advantageous that there is no need for coordiante transformation [31]. In this study, with the use of finite volume method, Cartesian velocity components are chosen.

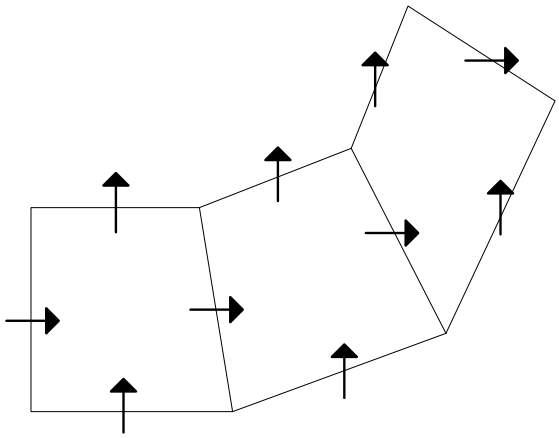

Figure 1.6: Cartesian Velocity Components

1.7 The Choice of Variable Arrangement

1.7.1 Staggered Arrangement

In staggered arrangement which is originally suggested by Harlow and Welsh [2], pressure is stored at the cell center while velocities are stored at the cell faces (Figure 1.7). With this type of placement of variables, the need for inerpolation of velocities to the cell faces is eliminated. If all variables are stored at cell center (called collocated variable arrangement which is discussed later) and central differencing scheme (linear) is used to interpolate the velocities to the cell faces, only pressures of the neighbouring cells playe role in this inerpolation. This problem is known in literature as pressure checkerboard effect. With the staggered arrangement, this problem is solved. However, staggered arrangement is more time consuming because each of the velocity components at each of the cell faces requires their own control volumes. Moreover, the computer code becomes more complex with this type of arrangement.

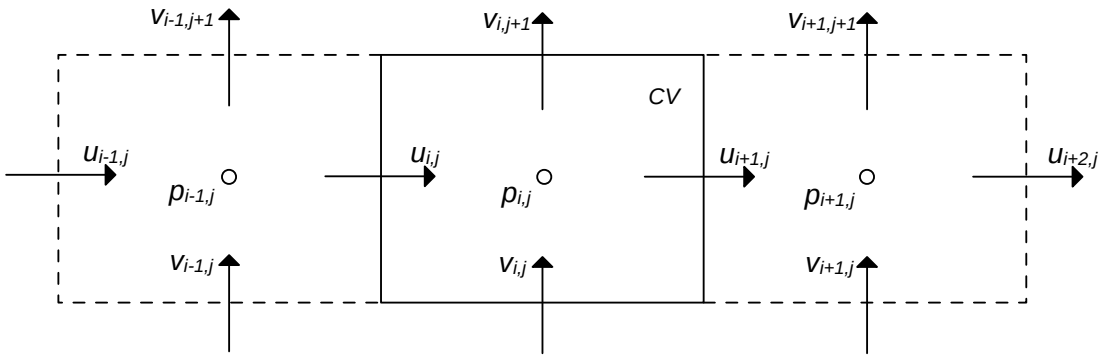

Figure 1.7: Staggered Variable Arrangement

1.7.2 Collocated Arangement

In collocated variable arrangement (Figure 1.8), all flow variables are located at the cell centers. As mentioned before, this causes non-physical oscillations (pressure checkerboarding effect) and difficulties in obtaining a converged solution. To overcome this problem, Rhie and Chow [12] proposed a novel momentum interpolation technique to evaluate the cell face veloicites. In this study, collocated arrangement has been chosen for the location of flow variables.

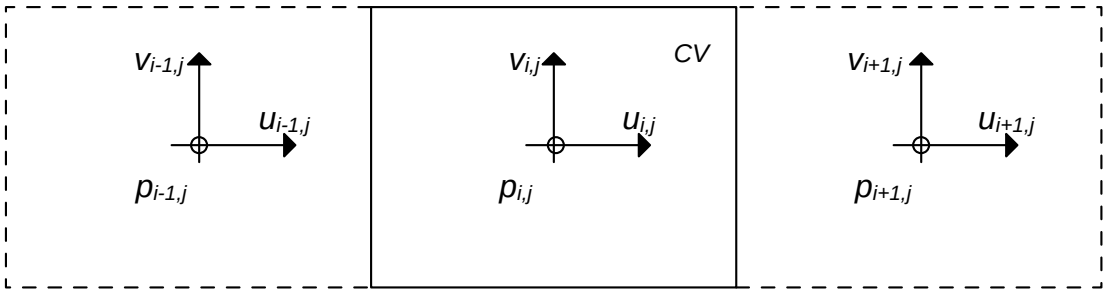

Figure 1.8: Collocated Variable Arrangement

1.8 Present Thesis

In this study, main variables are Cartesian velocity components, pressure and temperature while density is linked to pressure via equation of state. Modified SIMPLE algorithm is used to achieve pressure-velocity coupling. The discretisation is performed on non-orthogonal and boundary-fitted grids. Collocated variable arrangement is preferred because of its advantage on staggered arrangement in non-orthogonal meshes.

In the second chapter, numerical method used for the solution of the Navier-Stokes equations is described in detail. Chapter 3 presents the validation of the code. Solution of some incompressible and compressible problems and their comparisons with other solutions in the literature are given in this chapter. The last chapter concludes the thesis.

CHAPTER 2

NUMERICAL METHOD

2.1 Governing Equations

Generic steady-state transport equation of any scalar ϕ is given by

$$\nabla \cdot (\rho \vec{u} \phi) = \nabla(\Gamma \nabla \phi) + S_\phi \tag{2.1}$$

where Γ is diffusion coefficient, ρ is density and S is any source term for the scalar ϕ. If one writes gradient terms explicitly, generic transport equation becomes

$$\frac{\partial(\rho u \phi)}{\partial x} + \frac{\partial(\rho v \phi)}{\partial y} = \frac{\partial}{\partial x}(\Gamma \frac{\partial \phi}{\partial x}) + \frac{\partial}{\partial y}(\Gamma \frac{\partial \phi}{\partial y}) + S_\phi \tag{2.2}$$

For Newtonian fluids, the motion is described by a set of equations including continuity, momentum and energy equations. This set of equations is called *Navier-Stokes equations*.

2.1.1 Continuity Equation

If ϕ is replaced by one, Γ is replaced by zero and S is replaced by zero in the generic transport equation, *Continuity equation* is obtained

$$\nabla(\rho \vec{u}) = 0 \tag{2.3}$$

which symbolizes the relation between velocity components that forcing the conservation of mass.

2.1.2 Momentum Equation

If, ϕ is replaced by u and Γ is replaced by μ, then *u-Momentum equation* becomes

$$\nabla(\rho u\vec{u}) = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x}(\mu\frac{\partial u}{\partial x} + \lambda\nabla\vec{u}) + \frac{\partial}{\partial y}(\mu\frac{\partial v}{\partial x}) \tag{2.4}$$

v-Momentum equation can be written in the same manner

$$\nabla(\rho v\vec{u}) = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial y}(\mu\frac{\partial v}{\partial y} + \lambda\nabla\vec{u}) + \frac{\partial}{\partial x}(\mu\frac{\partial u}{\partial y}) \tag{2.5}$$

where μ is dynamic viscosity, p is pressure, $\lambda = \frac{-2}{3}\mu$. The term on the left hand side of the momentum equation is the advective term which symbolizes the change of the velocity field in space. On the right hand side of the Equation (2.4) and (2.5), the first term stands for pressure forces in the field. The second and third terms represent the shear stress due to viscosity of the flow.

2.1.3 Energy Equation

In generic transport equation, ϕ is replaced by T (temperature) and Γ is replaced by k (thermal conductivity constant) to obtain the energy equation.

$$\rho C_p(u\frac{\partial T}{\partial x} + z\frac{\partial T}{\partial y}) = \frac{\partial}{\partial x}(k\frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(k\frac{\partial T}{\partial y}) + S_e \tag{2.6}$$

S_e is the source term which includes pressure forces and shear stress:

$$S_e = (u\frac{\partial p}{\partial x} + v\frac{\partial p}{\partial y}) + \theta \tag{2.7}$$

θ is viscous dissipation and given by

$$\theta = 2((\frac{\partial u}{\partial x})^2 + (\frac{\partial v}{\partial y})^2) + (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})^2 - \frac{2}{3}(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y})^2 \tag{2.8}$$

2.2 Spatial Discretisation

Transforming PDEs to system of linear equations is called *discretisation*. In this study, finite volume technique is used for spatial discretisation of the governing equations over quadrilateral cells. With the help of this method, one can deal with complex geometries without writing

the governing equations in curvilinear coordinates. In finite volume methods, the domain is divided into a number of non-overlapping control volumes over which the conservation of ϕ is enforced in a discrete sense. The derivatives at control volume faces are calculated via finite difference approximations. A system of linear equations, which can be solved by any of the standard linear methods are generated.

2.2.1 Notation

For a two dimensional domain, cells of the structured mesh are numbered from 1 to N_x for x direction and from 1 to N_y for y direction. In addition, they are addressed with their i and j value e.g. $[i, j]$. In general, the compass notation is used if one wants to address the neighbors of the cell. The subscript P identifies the cell over which the discretisation is performed. E identifies the east neighbor cell while W is the west neighbor cell. In the same manner, N and S is the north and south neighbors, respectively. Diagram of neighboring nodes of the cell P and their index notations are given in Figure 2.1 and 2.2, respectively.

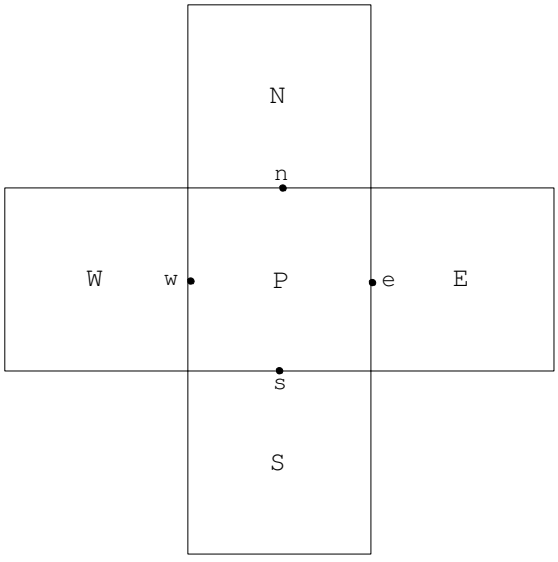

Figure 2.1: Compass notation

Compass Notation	*P*	*E*	*W*	*N*	*S*
Index Notation	*i , j*	*i +1, j*	*i -1, j*	*i , j+1*	*i , j-1*

Figure 2.2: Conversion between compass and index notation

To identify the values at the center of the cells, capital letters are used. To signify the values at cell faces, lower case letters are used. For the cells further away from the cell, the chain of location method is applied. For instance, for the northeast and southwest neighbors, subscript NE and SW are used, respectively.

2.2.2 Non-orthogonal Mesh Geometry

The grid is defined by the coordinates of the cell vertices. Cell center and cell face center coordinates are calculated using cell vertex coordinates. In addition, cell volume and the area vectors are also approximated from cell vertex points. In Figure 2.3, two dimensional cell geometry is shown.

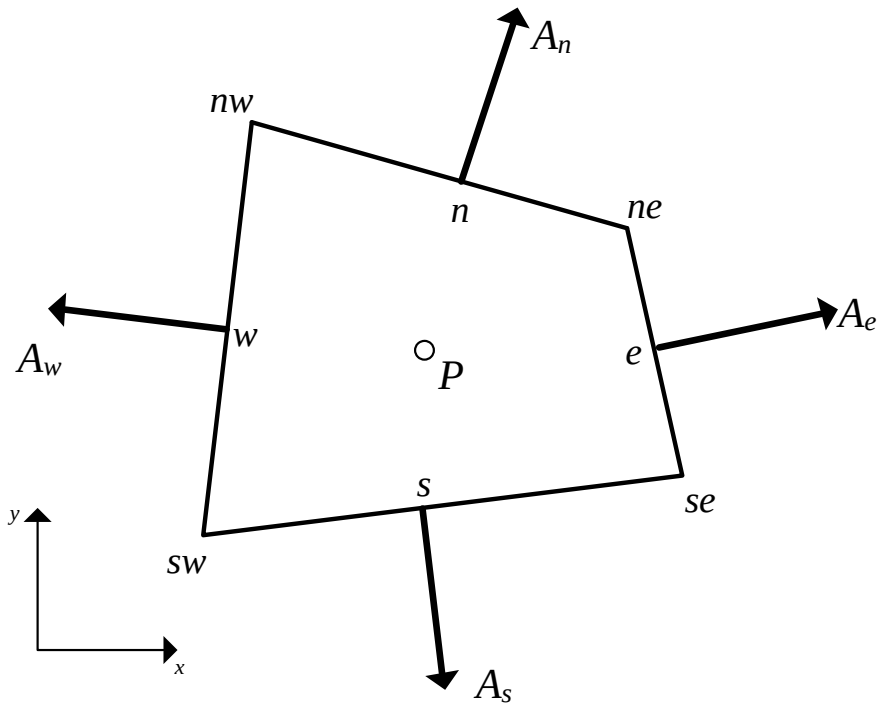

Figure 2.3: Two-dimensional cell geometry

The cell face vectors can be computed according to Figure 2.3 as [32]:

$$\begin{aligned}
\boldsymbol{A_e} &= (y_{ne} - y_{se})\vec{i} + (x_{se} - x_{ne})\vec{j} \\
\boldsymbol{A_w} &= (y_{sw} - y_{nw})\vec{i} + (x_{nw} - x_{sw})\vec{j} \\
\boldsymbol{A_n} &= (y_{nw} - y_{ne})\vec{i} + (x_{ne} - x_{nw})\vec{j} \\
\boldsymbol{A_s} &= (y_{se} - y_{sw})\vec{i} + (x_{se} - x_{se})\vec{j}
\end{aligned} \tag{2.9}$$

The cell area is equal to half of the magnitude of the vector product of two diagonals of the cell shown in Figure 2.3,

$$\begin{aligned}
\Omega &= \frac{1}{2}\left[(r_{ne} - r_{sw}) \times (r_{nw} - r_{se})\right] \\
&= \frac{1}{2}\left[(x_{ne} - x_{sw})(y_{nw} - y_{se}) - (y_{ne} - y_{sw})(x_{nw} - x_{se})\right]
\end{aligned} \tag{2.10}$$

2.3 Differencing Schemes

2.3.1 First Order Upwind Differencing Scheme (UDS)

First order schemes are originally developed to cope with the stability problem of discrete advection equation. Their implementation is very easy and provide smooth solutions. However, because of their excessively diffusive nature for high Re problems, the solution becomes very different from the exact solution [33].

In this study, only the first order upwind differencing scheme is used. In this scheme, the value of any scalar at a cell face is equal to the value at the center of the neighboring upstream face. For instance, for e face of the control volume,

$$\phi_e = \begin{cases} \phi_P & \text{if } \dot{m}_e < 0; \\ \phi_E & \text{if } \dot{m}_e > 0. \end{cases}$$

where $\dot{m}_e$ is mass flux at the cell face e. Details for the calculation of mass fluxes are given in the coming sections. Similarly, for the other faces of the cell, values for the scalar unknown are

calculated as:

$$\phi_n = \begin{cases} \phi_P & \text{if } \dot{m}_n < 0; \\ \phi_N & \text{if } \dot{m}_n > 0. \end{cases}$$

$$\phi_w = \begin{cases} \phi_W & \text{if } \dot{m}_w < 0; \\ \phi_P & \text{if } \dot{m}_w > 0. \end{cases}$$

$$\phi_s = \begin{cases} \phi_S & \text{if } \dot{m}_s < 0; \\ \phi_P & \text{if } \dot{m}_s > 0. \end{cases}$$

2.3.2 Second Order Central Differencing Scheme (CDS)

In second order central differencing scheme (also known as linear interpolation), the cell face value of any scalar is approximated by the linear interpolation of its values at the two neighbor cell centers. For instance, for the e face of control volume,

$$\phi_e = \phi_E \lambda_e + (1 - \lambda_e)\phi_P \tag{2.11}$$

If one assumes that the grid is uniform such that $\lambda_e = 0.5$, than Equation (2.11) becomes,

$$\phi_e = \frac{\phi_E + \phi_P}{2} \tag{2.12}$$

Equation (2.12) is second order accurate and may produce oscillatory solutions [31].

2.4 Discretisation of Advective Terms in Momentum Equations

For the non-orthogonal grid, the integral for advective flux in Equation (2.4)

$$F^c = \int_A \rho u \vec{u} dA \tag{2.13}$$

can be approximated using mid-point rule. For e face of the control volume, advective flux can be approximated as,

$$F_e^c \approx \dot{m}_e |\vec{u}|_e \tag{2.14}$$

The mass flux is calculated from the dot product of the face area normal and the velocity at the cell face. To perform this, all velocity components must be interpolated to the cell

faces, because, in non-orthogonal grids, all velocity components contribute to mass flux. This interpolation will be described later in the section "Interpolation of cell face velocities". The mass flux for the cell face e is given by,

$$\dot{m}_e = \rho(\vec{u}_e \cdot \boldsymbol{A_e}) \tag{2.15}$$

At the cell face e, the velocity vector is defined by,

$$\vec{u}_e = u_e\vec{i} + v_e\vec{j} \tag{2.16}$$

where u_e and v_e are Cartesian velocity components at cell face e, respectively. Using (2.15) and (2.16), the mass flux for the cell face e can be written as,

$$\dot{m}_e = \rho\left[u_e(y_{ne} - y_{se}) + v_e(x_{se} - x_{ne})\right] \tag{2.17}$$

With the same method, all mass fluxes on each faces can be given by,

$$\begin{aligned}
\dot{m}_n &= \rho\left[u_n(y_{nw} - y_{ne}) + v_n(x_{ne} - x_{nw})\right] \\
\dot{m}_w &= \rho\left[u_w(y_{sw} - y_{sw}) + v_w(x_{nw} - x_{sw})\right] \\
\dot{m}_s &= \rho\left[u_s(y_{se} - y_{sw}) + v_s(x_{sw} - x_{se})\right]
\end{aligned} \tag{2.18}$$

2.5 Discretisation of Diffusive Terms in Momentum Equations

In Equation (2.4), diffusion flux term in integral form is

$$F^D = \int_A \mu\nabla u \cdot dA \tag{2.19}$$

Using the mid-point rule, for e face of the control volume, this integral can be approximated as,

$$F_e^D \approx (\mu\nabla u)_e \cdot \boldsymbol{A_e} \tag{2.20}$$

In the mid-point approximation, the line connecting the point P and E (in Figure 2.4) is considered passing from the center of the face e.

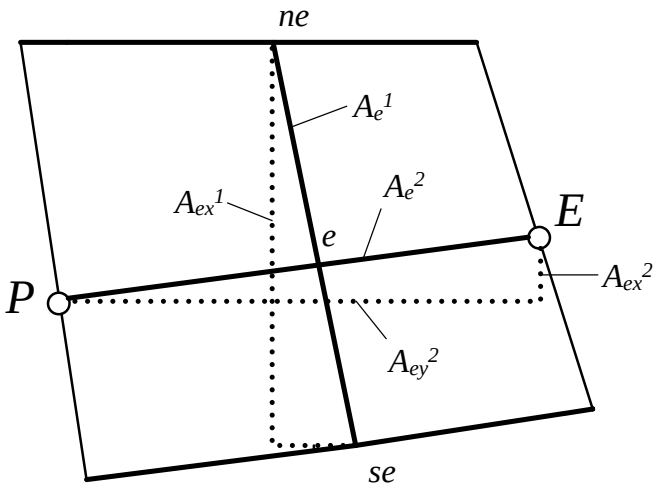

Figure 2.4: Face area vectors of auxiliary cell for face e

The derivatives of u are approximated using second order central differencing scheme for the e face of the cell. Using the auxiliary cell 1'2'3'4' which has a center e shown in Figure 2.5 and Gauss Divergence theorem, diffusive flux on cell face e is given by [34],

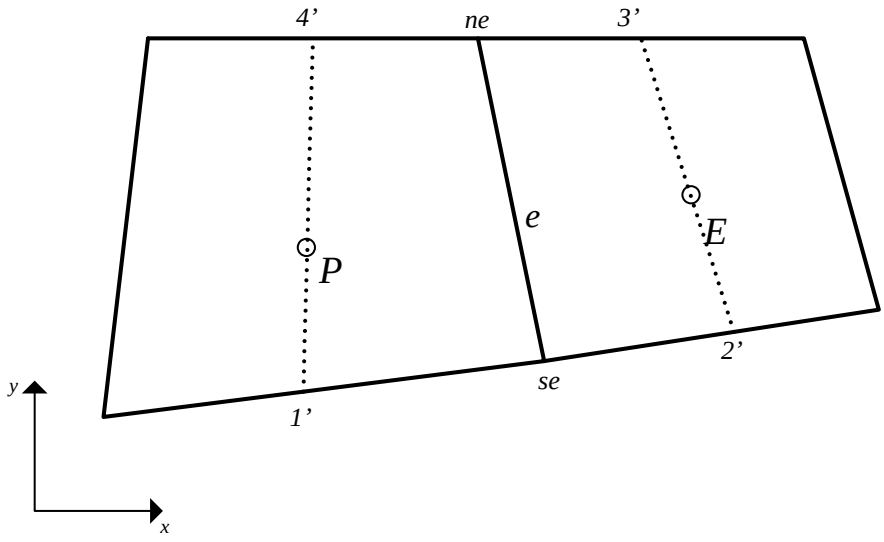

Figure 2.5: Auxiliary cell for face e

$$
\begin{aligned}
F_e^D &\approx \sum(\mu\frac{\partial u}{\partial x})_e A_{ex} + \sum(\mu\frac{\partial u}{\partial y})_e A_{ey} \\
&\approx \frac{\mu}{\Omega_e}(A_{ex}^1\left[A_{ex}^1(u_E-u_P)+A_{ex}^2(u_{ne}-u_{se})\right]+A_{ey}^1\left[A_{ey}^1(u_E-u_P)+A_{ey}^2(u_{ne}-u_{se})\right]) \\
&\approx \frac{\mu}{\Omega_e}\left[(A_{ex}^1A_{ex}^1+A_{ey}^1A_{ey}^1)(u_E-u_P)+\overline{(A_{ex}^1A_{ex}^2+A_{ey}^1A_{ey}^2)(u_{ne}-u_{se})}\right] \qquad (2.21)
\end{aligned}
$$

where

$\boldsymbol{A_e^1}, \boldsymbol{A_e^2}$: Area vectors at face e as shown in Figure 2.4

A_{ex}^1 : Projection of $\boldsymbol{A_e^1}$ along x axis $= (y_{ne} - y_{se})$

A_{ey}^1 : Projection of $\boldsymbol{A_e^1}$ along y axis $= (x_{se} - x_{ne})$

A_{ex}^2 : Projection of $\boldsymbol{A_e^2}$ along x axis $= (y_P - y_E)$

A_{ey}^2 : Projection of $\boldsymbol{A_e^2}$ along y axis $= (x_E - x_P)$

Ω_e : Volume of the control volume 1'2'3'4' of Figure 2.5

In Equation (2.21), the overbar denoted term is the non-orthogonal part of diffusion flux. It can be treated explicitly and added to the source term. And also, this part vanishes when the mesh is orthogonal because dot products are equal to zero. Similarly, diffusion fluxes through other faces can be computed in the same manner.

2.6 Approximation of Pressure and Source Terms in Momentum Equations

Pressure terms in momentum equations can be evaluated as,

$$S_P^{U_i} = -\int_A p\vec{i_i} \cdot dA = -\int_\Omega (\nabla p \cdot \vec{i_i}) \cdot d\Omega \approx -\left[(p_e - p_w)\boldsymbol{A_P^1} + (p_n - p_s)\boldsymbol{A_P^2}\right] \cdot \vec{i_i} \qquad (2.22)$$

For instance, if one writes this equation for u-momentum equation, with the help of Figure 2.6,

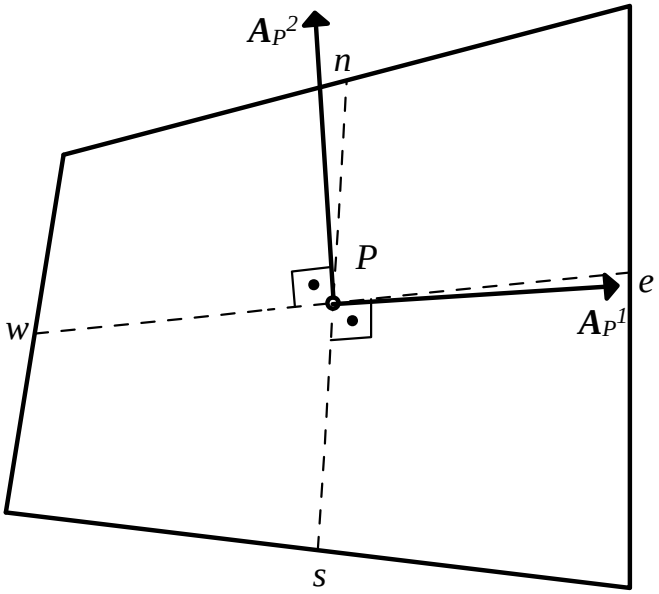

Figure 2.6: Notation used in calculation of source terms

$$S_P^u = -\int_A p\vec{i}\cdot dA = -\int_\Omega (\nabla p\cdot\vec{i})\cdot d\Omega \approx -\left[(p_e-p_w)\boldsymbol{A_P^1}+(p_n-p_s)\boldsymbol{A_P^2}\right]\cdot\vec{i} \tag{2.23}$$

where

$$\boldsymbol{A_P^1} = (y_n-y_s)\vec{i}+(x_s-x_n)\vec{j}$$

$$\boldsymbol{A_P^2} = (y_w-y_e)\vec{i}+(x_e-x_w)\vec{j} \tag{2.24}$$

$$\tag{2.25}$$

Then, Equation 2.23 becomes,

$$S_P^u \approx -\left[(p_e-p_w)\left[(y_n-y_s)\vec{i}+(x_s-x_n)\vec{j}\right]+(p_n-p_s)\left[(y_w-y_e)\vec{i}+(x_e-x_w)\vec{j}\right]\right]\cdot\vec{i} \tag{2.26}$$

After evaluating dot products in the Equation 2.26, pressure term for u-momentum equation is given by,

$$S_P^u \approx -\left[(p_e-p_w)(y_n-y_s)+(p_n-p_s)(y_w-y_e)\right] \tag{2.27}$$

Derivation of the pressure term S_P^v for v-momentum equation is obtained similarly as

$$S_P^v \approx -\left[(p_e-p_w)(x_s-x_n)+(p_n-p_s)(x_e-x_w)\right] \tag{2.28}$$

Discretisation of source terms is very simple. The value at the cell center is taken as an average value for a whole cell and integral of source is approximated as

$$\int_{\Omega} S d\Omega \approx S\Omega \tag{2.29}$$

In general, source term is a function of the dependent variable. For instance, for u-momentum and v-momentum equations, with velocity-dependent and constant parts of source term,

$$\begin{aligned} S^u &= S_u u_P + S_c \\ S^v &= S_v v_P + S_c \end{aligned} \tag{2.30}$$

2.7 Discrete Form of Momentum Equation

In order to write the discrete u-momentum equation, all the terms in this equation are discretised. Now, as presented above, one can write the discrete form of u-momentum equation,

$$F_e^C - F_w^C + F_n^C - F_s^C = F_e^D - F_w^D + F_n^D - F_s^D + S_P + (S_u u_P + S_c)\Omega \tag{2.31}$$

Then, using Equation (2.14) for advactive terms and Equation (2.21) for diffusive terms, Equation (2.31) becomes,

$$\begin{aligned} \dot{m}_e u_e - \dot{m}_w u_w + \dot{m}_n u_n - \dot{m}_n u_n =& \frac{\mu}{\Omega_e}(A_e)^2(u_E - u_P) - \frac{\mu}{\Omega_w}(A_w)^2(u_P - u_W) \\ &+ \frac{\mu}{\Omega_n}(A_n)^2(u_N - u_P) - \frac{\mu}{\Omega_s}(A_s)^2(u_P - u_S) \\ &+ S_P + ((S_u u_P + S_c)\Omega + F_{exp}^D) \end{aligned} \tag{2.32}$$

where, F_{exp}^D is the term representing linearly interpolated part of diffusive fluxes which are treated explicitly and added to the source term. In Equation (2.32), face velocities appearing in advective terms in the left hand side must be approximated. If central differencing scheme is used:

$$\begin{aligned} \dot{m}_e u_e - \dot{m}_w u_w + \dot{m}_n u_n - \dot{m}_s u_s =& \dot{m}_e \frac{u_E + u_P}{2} - \dot{m}_w \frac{u_W + u_P}{2} \\ &+ \dot{m}_n \frac{u_N + u_P}{2} - \dot{m}_s \frac{u_S + u_P}{2} \end{aligned} \tag{2.33}$$

Let us define some new terms to simplfy Equation (2.32),

$$
\begin{aligned}
d_e &= \frac{\mu}{\Omega_e}(A_e)^2 \\
d_w &= \frac{\mu}{\Omega_w}(A_w)^2 \\
d_n &= \frac{\mu}{\Omega_n}(A_n)^2 \\
d_s &= \frac{\mu}{\Omega_s}(A_s)^2
\end{aligned} \tag{2.34}
$$

Then Equation (2.32) becomes,

$$
\begin{aligned}
\dot{m}_e\frac{u_E+u_P}{2} - \dot{m}_w\frac{u_W+u_P}{2} + \dot{m}_n\frac{u_N+u_P}{2} - \dot{m}_s\frac{u_S+u_P}{2} =& d_e(u_E-u_P) - d_w(u_P-u_W) \\
&+ d_n(u_N-u_P) - d_s(u_P-u_S) \\
&+ S_P + ((S_u u_P + S_c)\Omega + F^D_{exp})
\end{aligned} \tag{2.35}
$$

After rearranging the Equation (2.35),

$$
\begin{aligned}
u_P\left[(d_e - \frac{\dot{m}_e}{2}) + (d_w + \frac{\dot{m}_w}{2}) + (d_n - \frac{\dot{m}_n}{2}) + (d_s + \frac{\dot{m}_s}{2}) + \dot{m}_e - \dot{m}_w + \dot{m}_n - \dot{m}_s\right] = \\
u_E(d_e - \frac{\dot{m}_e}{2}) + u_W(d_w + \frac{\dot{m}_w}{2}) + u_N(d_n - \frac{\dot{m}_n}{2}) + u_S(d_s + \frac{\dot{m}_s}{2}) \\
+S_P + ((S_u u_P + S_c)\Omega + F^D_{exp})
\end{aligned} \tag{2.36}
$$

Finally Equation (2.36) is factorised to give

$$
a_P = a_E u_E + a_W u_W + a_N u_N + a_S u_S + c \tag{2.37}
$$

where

$$
\begin{aligned}
a_E &= (d_e - \frac{\dot{m}_e}{2}) \\
a_W &= (d_w + \frac{\dot{m}_w}{2}) \\
a_N &= (d_n - \frac{\dot{m}_n}{2}) \\
a_S &= (d_s + \frac{\dot{m}_s}{2}) \\
a_P &= a_E + a_W + a_N + a_S + \dot{m}_e - \dot{m}_w + \dot{m}_n - \dot{m}_s \\
c &= S_P + ((S_u u_P + S_c)\Omega + F^D_{exp})
\end{aligned}
$$

If first order upwind differencing scheme is used to approximate the face velocities appearing in advective terms, using the MAX operator $\|\|$ of Patankar [4], the advective terms in Equation (2.31) become,

$$u_P\left[(d_e + \|\dot{m}_e, 0\|) + (d_w + \| - \dot{m}_w, 0\|) + (d_n + \|\dot{m}_n, 0\|) + (d_s + \| - \dot{m}_s, 0\|)\right] =$$

$$u_E(d_e + \|\dot{m}_e, 0\|) + u_W(d_w + \| - \dot{m}_w, 0\|) + u_N(d_n + \|\dot{m}_n, 0\|) + u_S(d_s + \| - \dot{m}_s, 0\|)$$

$$+S_P + ((S_u u_P + S_c)\Omega + F^D_{exp}) \quad (2.38)$$

Using the relationship about MAX operator

$$\dot{m} = \|\dot{m}, 0\| - \| \ -\dot{m}, 0\| \quad (2.39)$$

Then Equation (2.38) may be factorised as

$$a_P = a_E u_E + a_W u_W + a_N u_N + a_S u_S + c \quad (2.40)$$

where

$$a_E = (d_e + \| - \dot{m}_e, 0\|)$$

$$a_W = (d_w + \|\dot{m}_w, 0\|)$$

$$a_N = (d_n + \| - \dot{m}_n, 0\|)$$

$$a_S = (d_s + \|\dot{m}_s, 0\|)$$

$$a_P = a_E + a_W + a_N + a_S + \dot{m}_e - \dot{m}_w + \dot{m}_n - \dot{m}_s$$

$$c = S_P + ((S_u u_P + S_c)\Omega + F^D_{exp})$$

2.8 Discretisation of Continuity Equation

Steady state continuity equation is given by (from Equation (2.3))

$$\nabla(\rho\vec{u}) = 0 \quad (2.41)$$

To apply the mid-point rule, one first integrates Equation (2.41) over control volume to get

$$\int_\Omega \nabla(\rho\vec{u}) d\Omega \quad (2.42)$$

Then Gauss Divergence Theorem is applied to Equation (2.42),

$$\int_{\Omega} \nabla(\rho\vec{u})d\Omega \approx \int_{A} \rho\vec{u}dA \tag{2.43}$$

Now it is time to apply the mid-point rule to Equation (2.43),

$$\int_{\Omega} \nabla(\rho\vec{u})d\Omega \approx \int_{A} \rho\vec{u}dA \approx \sum_{f}(\rho\vec{u})_f \cdot \boldsymbol{A_f} \tag{2.44}$$

where f refers the faces of the cell, i.e. e, w, n, s. Therefore, continuity equation for a non-orthogonal meshes can be written as

$$(\rho\vec{u})_e \cdot \boldsymbol{A_e} - (\rho\vec{u})_e \cdot \boldsymbol{A_w} + (\rho\vec{u})_n \cdot \boldsymbol{A_n} - (\rho\vec{u})_s \cdot \boldsymbol{A_s} = 0 \tag{2.45}$$

or

$$\dot{m}_e - \dot{m}_w + \dot{m}_n - \dot{m}_s = 0 \tag{2.46}$$

2.9 Interpolation of the Face Velocities

As mentioned in section 1.5, to avoid the pressure chequer-boarding effect in the collocated variable arrangement, Rhie and Chow proposed an interpolation method. In this method, for the cell face velocities, one interpolates the sum of the velocities and pressures at the cell center instead interpolating velocities only. At the cell center P, momentum conservation equation is given by [33],

$$u_P - \frac{1}{a_P}\sum_{f} a_f p_f = \frac{S}{a_P} - \sum_{n=nb} a_n p_n \tag{2.47}$$

where u_P is the velocity vector at cell center, a_P is the diagonal element of the system of linear equations $\sum_{n=nb} a_n u_n$, $\sum_f a_f u_f$ is the sum of the pressure forces at the cell faces. Rhie and Chow states that, this momentum conservation equation can be written for the faces of the control volume. If one writes this equation for cell face e, then,

$$u_e - (\frac{1}{a_P}\sum_{f} a_f p_f)_e = (\frac{S}{a_P})_e - (\sum_{n=nb} a_n p_n)_e \tag{2.48}$$

For Equation (2.48), Rhie and Chow declared that the left hand side is equal to linear interpolation of left hand side for cells P and E. This gives,

$$u_e - (\frac{1}{a_P}\sum_{f} a_f p_f)_e = \overline{u_e} - \overline{(\frac{1}{a_P}\sum_{f} a_f p_f)_e} \tag{2.49}$$

where overbar denotes linear interpolation. Then,

$$u_e = \overline{u_e} + (\frac{1}{a_P}\sum_f a_f p_f)_e - \overline{(\frac{1}{a_P}\sum_f a_f p_f)_e} \tag{2.50}$$

For the e face of the cell, using notation in the Figure 2.7;

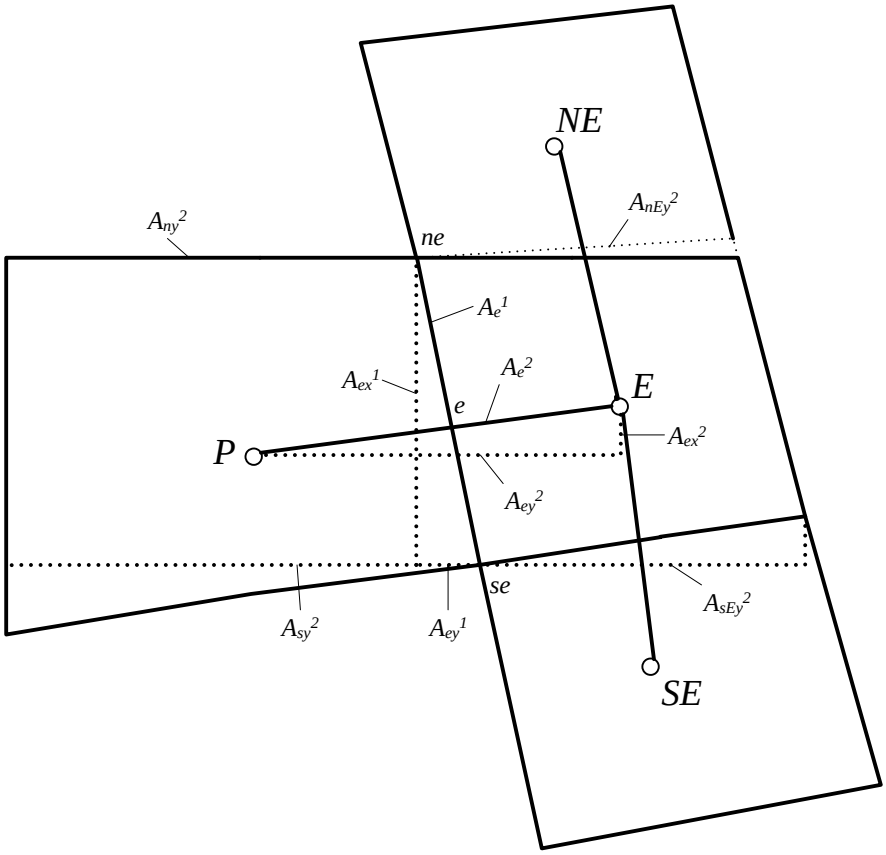

Figure 2.7: Notation used in calculation of cell face velocities

$$(\sum_f a_f p_f)_e = A_{ex}^1(p_E - p_P) + \frac{1}{4}(A_{ey}^2 p_N - A_{sy}^2 p_S + A_{nEy}^2 p_{NE} - A_{sEy}^2 p_{SE}) \tag{2.51}$$

2.10 Solution of the Navier-Stokes Equations

To solve the system of equations, one needs to couple the equations. Mostly, this is done iteratively, meaning result of one is used in the other. As mentioned in the Introduction chapter, there are many methods to solve Navier-Stokes equations. In this study, SIMPLE method which is originally proposed by Patankar and Spalding [3] and modified for non-staggered grids by Rhie and Chow [12] is used to couple the equations.

2.10.1 SIMPLE Scheme

As mentioned in the previous chapter, pressure-based finite volume approaches have been chosen in most of the solvers in the past. The reason is, pressure has enabled the compressible solvers to compensate the poor role of density in incompressible limits. To recover the role of pressure in the continuity equation, SIMPLE (Semi Implicit Method for Pressure-Linked Equations) scheme is proposed by Patankar and Spalding.

SIMPLE scheme is a predictor-corrector type method. In other words, one solves the momentum equations with the guessed pressure and velocity field which does not satisfy the continuity equation. Actually, continuity equation does not include any pressure terms. Because of this, SIMPLE method converts the continuity equation into a new type of equation which is called the pressure correction equation. One solves the pressure correction equation to obtain pressure correction terms. These calculated pressure correction terms are used to correct the (predicted) velocity and pressure fields. This predict-and-correct loop continues until converged solution is obtained that satisfies conservation of mass.

Using Equation (2.37), discrete form of u and v-momentum equations are given respectively by,

$$\begin{aligned} a_P^u u_P &= \sum_{nb} a_{nb} u_{nb} + S_P^u + c_P \\ a_P^v v_P &= \sum_{nb} a_{nb} v_{nb} + S_P^v + c_P \end{aligned} \tag{2.52}$$

At first, the discrete momentum equations are solved with guessed pressure and velocity fields

$$\begin{aligned} a_P^u u_P^* &= \sum_{nb} a_{nb} u_{nb}^* + S_P^u + c_P \\ a_P^v v_P^* &= \sum_{nb} a_{nb} v_{nb}^* + S_P^v + c_P \end{aligned} \tag{2.53}$$

where u^* and v^* is the initial estimates for velocity, nb refers neighboring cells of P, which are E, W, N and S. S_P terms are also calculated by using the guessed pressure field. However, velocity field which is obtained by solving Equation (2.53) will not satisfy the continuity equation. Since the aim is to correct these velocity fields to obtain the conservation of mass, following

equations can be written for velocity fields,

$$u = u^* + u'$$
$$v = v^* + v' \tag{2.54}$$

and for pressure field,

$$p = p^* + p' \tag{2.55}$$

where u', v' and p' are the corrections for u, v and p respectively. If one put Equation (2.54) and Equation (2.55) into Equation (2.52) gives,

$$\begin{aligned} a_P^u(u_P^* + u_P') &= \sum_{nb} a_{nb}(u_{nb}^* + u_{nb}') + S_P^u + c_P \\ a_P^v(v_P^* + v_P') &= \sum_{nb} a_{nb}(v_{nb}^* + v_{nb}') + S_P^v + c_P \end{aligned} \tag{2.56}$$

At this point, sum of the neighboring velocity terms can be approximated by,

$$\begin{aligned} \sum_{nb} a_{nb}(u_{nb}^* + u_{nb}') &\approx \sum_{nb} a_{nb}u_{nb}^* \\ \sum_{nb} a_{nb}(v_{nb}^* + v_{nb}') &\approx \sum_{nb} a_{nb}v_{nb}^* \end{aligned} \tag{2.57}$$

Then, Equation (2.56) becomes

$$\begin{aligned} a_P^u(u_P^* + u_P') &= \sum_{nb} a_{nb}u_{nb}^* + S_P^u + c_P \\ a_P^v(v_P^* + v_P') &= \sum_{nb} a_{nb}v_{nb}^* + S_P^v + c_P \end{aligned} \tag{2.58}$$

which is the main approximation of SIMPLE scheme and should be valid as velocity and pressure corrections approach to zero during the iterative solution process. After subtracting Equation (2.53) from Equation (2.58) gives

$$a_P^u u_P' = S_P^u$$
$$a_P^v v_P' = S_P^v \tag{2.59}$$

Finally, velocity corrections become,

$$\begin{aligned} u_P' &= \frac{-\left[(p_e - p_w)(y_n - y_s) + (p_n - p_s)(y_w - y_e)\right]}{a_P^u} \\ v_P' &= \frac{\left[(p_e - p_w)(x_n - x_s) - (p_n - p_s)(x_w - x_e)\right]}{a_P^v} \end{aligned} \tag{2.60}$$

Then using the discretised version of the pressure terms, Rhie and Chow state in their research that for near orthogonal meshes, the cross derivative terms at the faces are small and can be ignored. As a result, corrections for the face velocities can be given as

$$\begin{aligned} u'_e &= \frac{1}{a_e} A_e \frac{p'_E - p'_P}{2} \\ u'_w &= \frac{1}{a_w} A_w \frac{p'_P - p'_W}{2} \\ v'_n &= \frac{1}{a_n} A_n \frac{p'_N - p'_P}{2} \\ v'_s &= \frac{1}{a_s} A_s \frac{p'_P - p'_S}{2} \end{aligned} \tag{2.61}$$

where

$$\begin{aligned} a_e &= \frac{a_E + a_P}{2} \\ a_w &= \frac{a_W + a_P}{2} \\ a_n &= \frac{a_N + a_P}{2} \\ a_s &= \frac{a_S + a_P}{2} \end{aligned} \tag{2.62}$$

2.10.2 Pressure Correction Equation

The discretised continuity equation, from Equation (2.46),

$$\dot{m}_e - \dot{m}_w + \dot{m}_n - \dot{m}_s = 0 \tag{2.63}$$

For e face of the cell;

$$\dot{m}_e = (\rho^*_e + \rho'_e)(U^*_e + U'_e)A_e \tag{2.64}$$

where

$$U'_e = -\frac{A_e}{a_e}(P'_E - P'_P) \tag{2.65}$$

$$\rho'_e = (\frac{\partial \rho}{\partial p})_e P'_e \tag{2.66}$$

$\frac{\partial \rho}{\partial p}$ is the compressibility factor and given by

$$(\frac{\partial \rho}{\partial p})_e = Z_e = \frac{1}{c^2} = \frac{1}{\gamma RT} \tag{2.67}$$

where γ is specific heat ratio and R is universal gas constant.

Then

$$\dot{m}_e = (\rho_e^* U_e^* A_e + U_e^* Z_e p_e' A_e + \rho_e^* d_e (P_P' - P_E')) \tag{2.68}$$

where

$$d_e = \frac{A_e^2}{a_e} \tag{2.69}$$

If one writes Equation (2.68) for all faces of the cell Equation (2.63) becomes

$$\begin{aligned}
&(U_e^* Z_e p_e' A_e) - (U_w^* Z_w p_w' A_w) + (U_n^* Z_n p_n' A_n) - (U_s^* Z_s p_s' A_s) + \\
&\rho_e^* d_e (P_P' - P_E') - \rho_w^* d_w (P_W' - P_P') + \rho_n^* d_n (P_P' - P_N') - \rho_s^* d_s (P_S' - P_P') = \\
&\dot{m}_w - \dot{m}_e + \dot{m}_s - \dot{m}_n
\end{aligned} \tag{2.70}$$

Now, again for the east face of the cell, using first order upwinding scheme for the cell face density

$$\rho_e' = \begin{cases} \rho_P' = Z_P P_P' & \text{if } U_e A_e > 0; \\ \rho_E' = Z_E P_E' & \text{if } U_e A_e < 0. \end{cases}$$

$$\rho_e' U_e A_e = Z_P P_P' max(U_e A_e, 0) + Z_E P_E' max(-U_e A_e, 0) \tag{2.71}$$

For the cell centering node P, pressure correction equation becomes,

$$\begin{aligned}
&P_P'(Z_P max(U_e A_e, 0) - Z_P max(U_w A_w, 0) + Z_P max(U_n A_n, 0) - Z_P max(U_s A_s, 0)) + \\
&P_E'(Z_E max(-U_e A_e, 0)) - P_W'(Z_W max(U_w A_w, 0)) + P_N'(Z_N max(-U_n A_n, 0)) - P_S'(Z_S max(U_s A_s, 0)) + \\
&P_P'(\rho_e^* d_e + \rho_w^* d_w + \rho_n^* d_n + \rho_s^* d_s) + P_E'(-\rho_e^* d_e) + P_W'(-\rho_w^* d_w) + P_N'(-\rho_n^* d_n) + P_S'(-\rho_s^* d_s) = \\
&\dot{m}_w - \dot{m}_e + \dot{m}_s - \dot{m}_n
\end{aligned} \tag{2.72}$$

Equation (2.76) can be factorised to give

$$b_P P_P' + b_E p_E' + b_W P_W' + b_N P_N' + b_S P_S' = c \tag{2.73}$$

where

$$
\begin{aligned}
b_E =& \rho_e^* d_e - Z_E max(-U_e A_e, 0) \\
b_W =& \rho_w^* d_w + Z_W max(U_w A_w, 0) \\
b_N =& \rho_n^* d_n - Z_N max(-U_n A_n, 0) \\
b_S =& \rho_s^* d_s + Z_S max(U_s A_s, 0) \\
b_P =& (\rho_e^* d_e + Z_P max(U_e^* A_e, 0)) + (\rho_w^* d_w - Z_P max(U_w^* A_w, 0)) + \\
& (\rho_n^* d_n + Z_P max(U_n^* A_n, 0)) + (\rho_s^* d_s - Z_P max(U_s^* A_s, 0)) \\
c =& \dot{m}_w^* - \dot{m}_e^* + \dot{m}_s^* - \dot{m}_n^*
\end{aligned} \tag{2.74}
$$

This system of equations can be solved to yield p' which is the pressure correction to correct pressure and velocity fields. In Equation (2.68), source term c is the divergence (imbalance) of mass flux field which can be used as a criteria for the convergence. The computed pressure corrections are designed to destroy this imbalance. Hence, it is obvious that the corected mass fluxes satisfies the discrete continuity equation at each iteration of SIMPLE procedure. However, the cell center velocities do not satisfiy the discrete continuity equation neither before nor after the correction by p'. Because of this, mass fluxes are never calculated directly with cell center velocities, instead momentum interpolated velocities are used. So, in a collocated formulation, cell center velocities satisfy the discrete momentum equations, however not the dicsrete continuity equation.

To increase the convergence rate of the solution, velocity and pressure fields may be underrelaxed. There are two ways of underrelaxation. If it is desired, velocity and pressure fields can be underrelaxed

$$
\begin{aligned}
u &= u^* + \alpha_u u' \\
v &= v^* + \alpha_v v' \\
p &= p^* + \alpha_p p'
\end{aligned} \tag{2.75}
$$

where α_u , α_v and α_p are relaxation parameters for u velocity, v velocity and pressure respectively which are between 0 and 1. Other under-relaxation technique is widely used is in many solvers. u_P can be written as follows:

$$u_P^{new} = \frac{\sum_{nb} a_{nb} u_{nb} + S_P^u + c_P}{a_P^u} \tag{2.76}$$

However, allowing u_P to change as Equation (2.76) can generate instability. Because of this, the change in u_P must be only a fraction α_u of the difference between two iterations.

$$u_P = u_P^* + \alpha_u (u_P^{new} - u_P^*) \tag{2.77}$$

Replacing u_P^{new} with Equation (2.76) yields

$$u_P = u_P^* + \alpha_u (\frac{\sum_{nb} a_{nb} u_{nb} + S_P^u + c_P}{a_P^u} - u_P^*) \tag{2.78}$$

If one rearranges Equation (2.78) to obtain underrelaxed version of u-momentum equation

$$\frac{a_P^u}{\alpha_u} u_P = \sum_{nb} a_{nb} u_{nb} + S_P^u + c_P + \frac{1 - \alpha_u}{\alpha_u} a_P^u u_P^* \tag{2.79}$$

Similarly, underrelaxed v-momentum equation can be written as

$$\frac{a_P^v}{\alpha_v} v_P = \sum_{nb} a_{nb} v_{nb} + S_P^v + c_P + \frac{1 - \alpha_v}{\alpha_v} a_P^v v_P^* \tag{2.80}$$

These modified equations are solved in inner iterations. The terms involving α_u and α_v cancel out when the outer iterations convergence [31]. Unlike the momentum equations, pressure equation is diagonally dominant does not need the underrelaxation of its diagonal. Hence, pressure relaxation is always in the form of Equation (2.75). An finally it should be noted that if Equation (2.79) and Equation (2.80) are used as momentum equations, in the velocity interpolation and presure correction equations, non-relaxed a_P^u and a_P^v must be used.

2.10.3 Overall Solution Procedure

Finally, overall solution procedure can be given

1. Guess the pressure field p^*.

2. Solve the u and v-momentum equations with guessed pressure field p^* to get u^* and v^* at cell centers.

3. Calculate face velocities using Rhie and Chow's inerpolation technique.

4. Calculate mass fluxes at faces.

5. Solve the pressure correction equation to obtain p'.

6. Correct cell center velocities u_P^* and v_P^*.

7. Correct the pressure field.

8. Calculate temperature field.

9. Calculate density field using equation of state.

10. Calculate compressibility factor (Z).

11. If solution is converged, exit. Otherwise, go to step 2.

2.11 Implemenetation of Boundary Conditions

Until now, the boundaries of the domain are ignored in the solution process. However, boundary conditions have great influence on the solution. In this work, ghost cell approach is applied. In the ghost cell approach, it is assumed that there are invisible (ghost) cells outside the boundaries. When the scalar values at the boundaries are interpolated, scalar values at these ghost cells are used.

In this study, collocated variable arrangement is used. Because of this, implementation of boundary conditions is different from staggered arrangement. To specify the boundary value of any variable, one has to use the ghost cells. For instance, say the desired inlet value of u velocity is u_b. If one considers the boundary as any face between two cells, P is the first cell and W is the ghost cell on the west side of P cell (Figure 2.8). Using second order CDS,

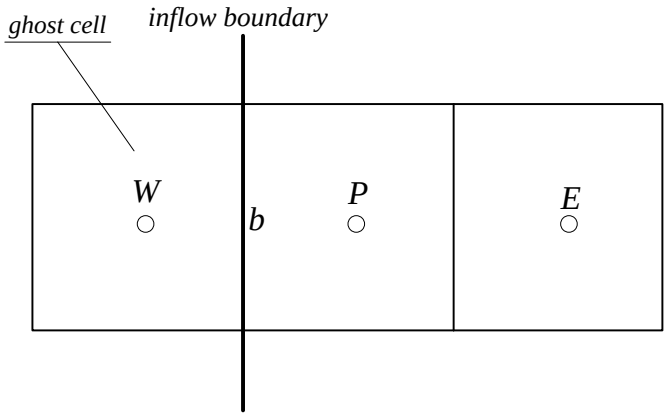

Figure 2.8: Inflow Boundary

$$u_b = \frac{u_P + u_W}{2} \tag{2.81}$$

However, in equation eqn (3), u_W is unknown. Writing Equation (2.81) for u_W yields

$$u_W = 2u_b - u_P \tag{2.82}$$

As can be seen, if ghost cell value is specified and Equation (2.82) is applied to solution, boundary condition for flow variables can be implemented. For v velocity and pressure, boundary values can be calculated similarly,

$$v_W = 2v_b - v_P$$

$$p_W = 2p_b - p_P \tag{2.83}$$

In following sections, four types of boundaries which are inflow, outflow, wall and symmetry are described briefly.

2.11.1 Wall Boundaries

The no-slip condition which arises from the fact that viscous fluids sticks to solid boundary is a suitable condition at wall boudaries. Because there is no flow through the wall, advective fluxes are zero.

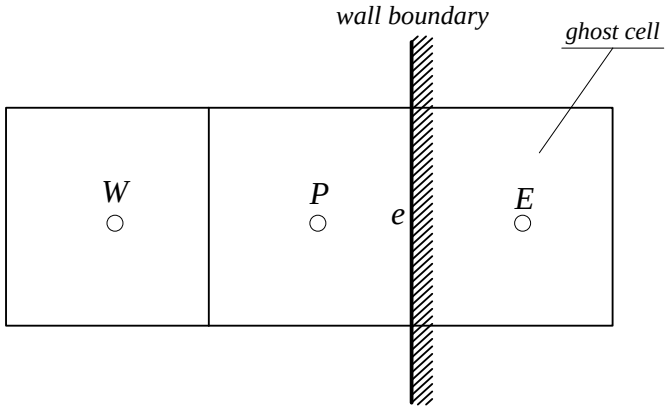

Figure 2.9: Wall Boundary

Dirichlet type boundary conditions are used at walls. Let's assume that the east boundary of the domain is wall. Using Figure 2.9, u velocity at the boundary is given by

$$u_e = \frac{u_P + u_E}{2} \tag{2.84}$$

and ghost cell value for u velocity is

$$u_E = 2u_e - u_P \tag{2.85}$$

If Equation (2.85) is substituted in Equation (2.37)

$$a_P = a_E(2u_e - u_P) + a_W u_W + a_N u_N + a_S u_S + c \tag{2.86}$$

If Equation (2.86) is rearranged to yield

$$(a_P + a_E) = a_W u_W + a_N u_N + a_S u_S + c + 2a_E u_e \tag{2.87}$$

Using this substitution, unknown u_E for the East cell is dropped from the u-momentum equation. New coefficients for the u-momentum equation are (denoted as *)

$$a_E^* = 0$$

$$a_W^* = a_W$$

$$a_N^* = a_N$$

$$a_S^* = a_S$$

$$a_P^* = a_P + a_E$$

$$c^* = c + 2a_E u_e$$

2.11.2 Subsonic Inflow Boundaries

At a subsonic inflow boundary, two flow variables must be specified, namely velocity and density. Pressure is extrapolated from interior of the domain. Since pressure is calculated and density is specified, temperature can easily be calculated using the ideal gas equation of state.

2.11.3 Supersonic Inflow Boundaries

At a supersonic inflow boundary, three flow variables must be specified, velocity, pressure and density. Again, because pressure and density are specified, temperature is extracted using the ideal gas equation of state.

2.11.4 Subsonic Outflow Boundaries

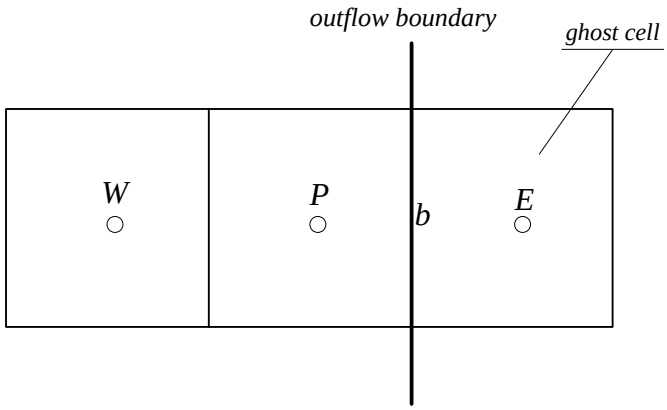

Figure 2.10: Outflow Boundary

At subsonic outflow boundaries, all the flow variables except pressure are extrapolated from the interior of the domain. Pressure value must be specified at the subsonic outflow boundaries. According to Figure (2.10)

$$u_b = u_P$$

$$v_b = v_P$$

$$\rho_b = \rho_P \tag{2.88}$$

2.11.5 Supersonic Outflow Boundaries

All the flow variables which are pressure, velocity and density are extrapolated from the interior of the domain at supersonic outflow boundaries. According to Figure (2.10)

$$u_b = u_P$$

$$\rho_b = \rho_P$$

$$p_b = p_P \tag{2.89}$$

It is very important to place outflow boundary at the suitable place. Otherwise, error may disturb upstream. Thus, outlet boundaries should be placed such that downstream conditions have no influence on he solution.

2.11.6 Symmetry Boundaries

At the symmetry boundaries, normal gradients of the velocity components which are parallel to symmetry boundary are zero. Thus, advective fluxes of all the quantities are also zero. Despite the normal velocity component is zero, its gradient is not zero. Hence, the normal stress is non-zero [31].

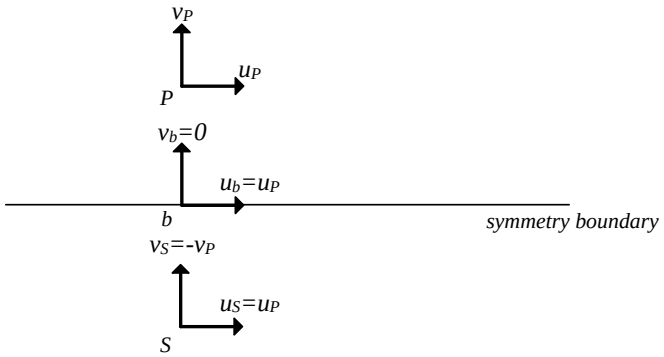

Figure 2.11: Symmetry Boundary

According to Figure 2.11, for the parallel component of velocity to the boundary. If Equation (2.86) is rearranged to yield

$$\begin{aligned} u_S &= u_P \\ u_s &= \frac{u_S + u_P}{2} = u_P \end{aligned} \tag{2.90}$$

and for the perpendicular component of velocity to the boundary

$$\begin{aligned} v_b &= 0 \\ v_S &= 2v_b - v_P = -v_P \end{aligned} \tag{2.91}$$

and for the pressure at the boundary can be written as

$$\begin{aligned} p_S &= p_P \\ p_s &= \frac{p_S + p_P}{2} = p_P \end{aligned} \tag{2.92}$$

CHAPTER 3

RESULTS AND DISCUSSION

3.1 General

To validate the code, two incompressible and two compressible test cases are chosen: For incompressible flows, lid-driven skewed cavity and laminar flow through a gradual expansion, for compressible flows 2D laminar flat plate and 2D converging-diverging nozzle.

Lid-driven square cavity flow is one of the most popular test cases for cartesian grids. It has been considered as a standart benchmark problem for long years because of its simple geometry and boundary conditions. There are no-slip boundary conditions on its walls and it is driven by the sliding top wall. In literature, Ghia et al[36], provide a numerical solution for this problem and so many flow solver codes have been validated with this solution. It is a good idea that to create a test case based on lid-driven square cavity flow. For example, side walls of the domain can be inclined to set up a new problem. For this study, 30 and 45 degree lid-driven skewed cavity problem which is already provided as a benchmark problem by Demirdzic, Lilek and Peric [37] are solved. The second incompressible test case is laminar flow through a gradual expansion. In this problem, wall pressure is compared with well known solutions in literature.

As a compressible test case, 2D laminar flat plate problem is chosen to verify the extra stress

terms coming from compresibility effect are properly modeled. And finally, 2D converging-diverging nozzle problem is solved.

Mass residual is actually source term in pressure correction equation. In each iteration, pressure values at each cell are added up and divided into the total number of computational cells to calculate the pressure residual. In all problems, iterations are performed until the mass and pressure residuals reduce eight orders of magnitude as a convergence criteria.

3.2 Incompressible Test Cases

3.2.1 Test Case 1: Lid-Driven Skewed Cavity

As mentioned before, lid-driven square cavity flow is very popular benchmark problem because of its simple geometry and boundary conditions. In Figure 3.1, the geometry for test case 1 is given. The problem is created via inclining the side walls of the square cavity by an angle α as shown in Figure 3.1.

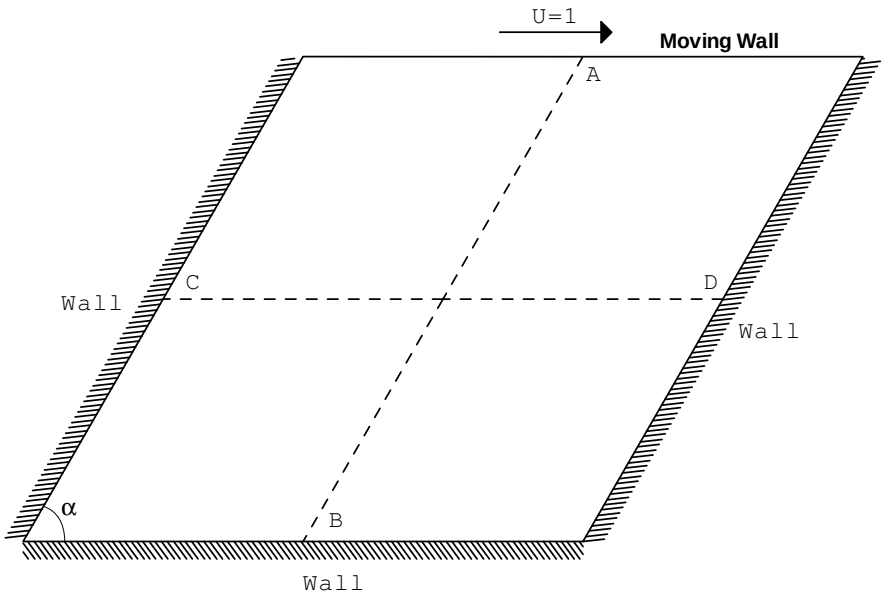

Figure 3.1: Geometry for the test case 2

In this study, to test the solver on skewed cavity flow, two different skew angles are used: $\alpha = 45$ and $\alpha = 30$. Demirdzic, Lilek and Peric [37] provides solution for 45 and 30 degree skewed cavity. Each case is solved for Reynolds number of 1000.

45-degree cavity flow problem is solved on 81x81 and 161x161 meshes for Re=1000. Figure 3.2 and Figure 3.3 show u velocity and v velocity profiles along lines A-B and C-D respectively.

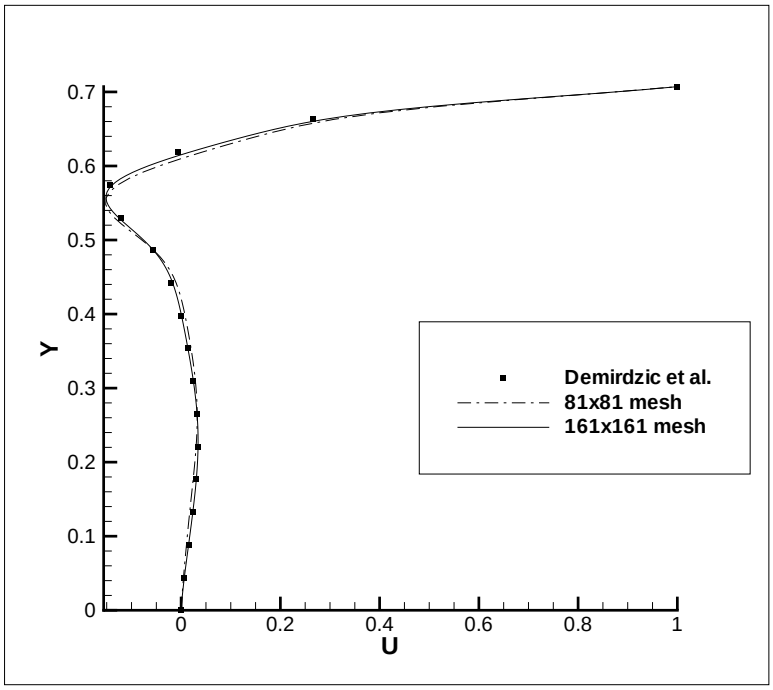

Figure 3.2: u velocity profiles along line A-B for 45-degree skewed cavity solution at Re=1000

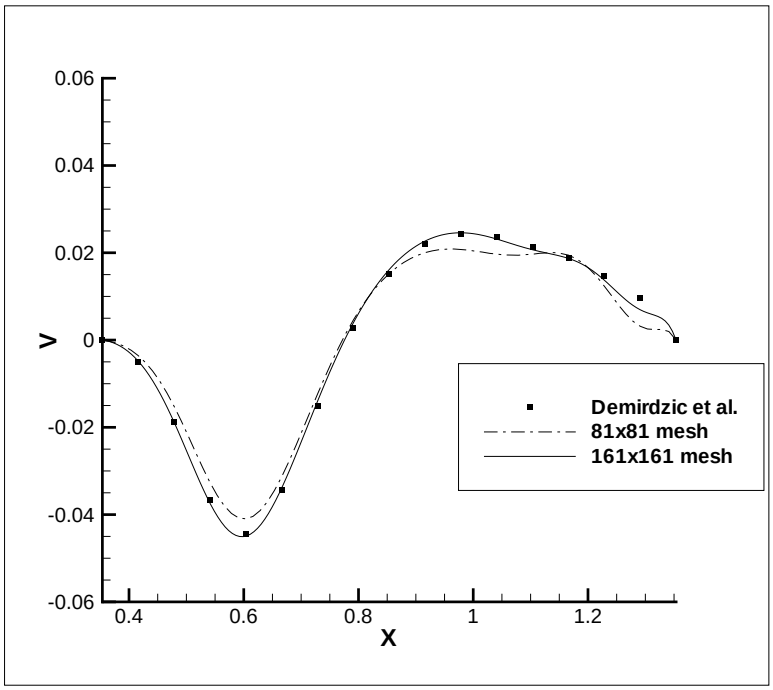

Figure 3.3: v velocity profiles along line C-D for 45-degree skewed cavity solution at Re=1000

As can be seen from Figure 3.2 and Figure 3.3, results are very satisfactory. Streamline-contour can also be seen in Figure 3.4.

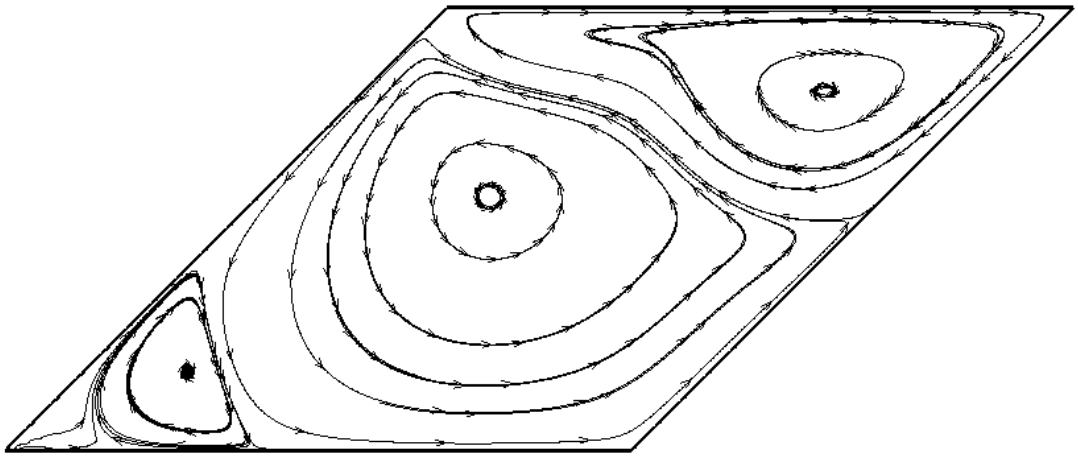

Figure 3.4: Streamlines for 45-degree skewed cavity solution at Re=1000

According to Ferziger and Peric [37], if the incline angle is smaller than 45-degree, it is natural to get unrealistic solutions because of the shape of the grid cells. To see the capability of the code to solve skewed cavity flows below 45-degree, 30-degree cavity is created. 30-degree cavity flow problem is solved for 81x81, 161x161 and 321x321 meshes at Reynolds number of 1000. Figure 3.2 and Figure 3.3 show u velocity and v velocity profiles along lines A-B and C-D respectively.

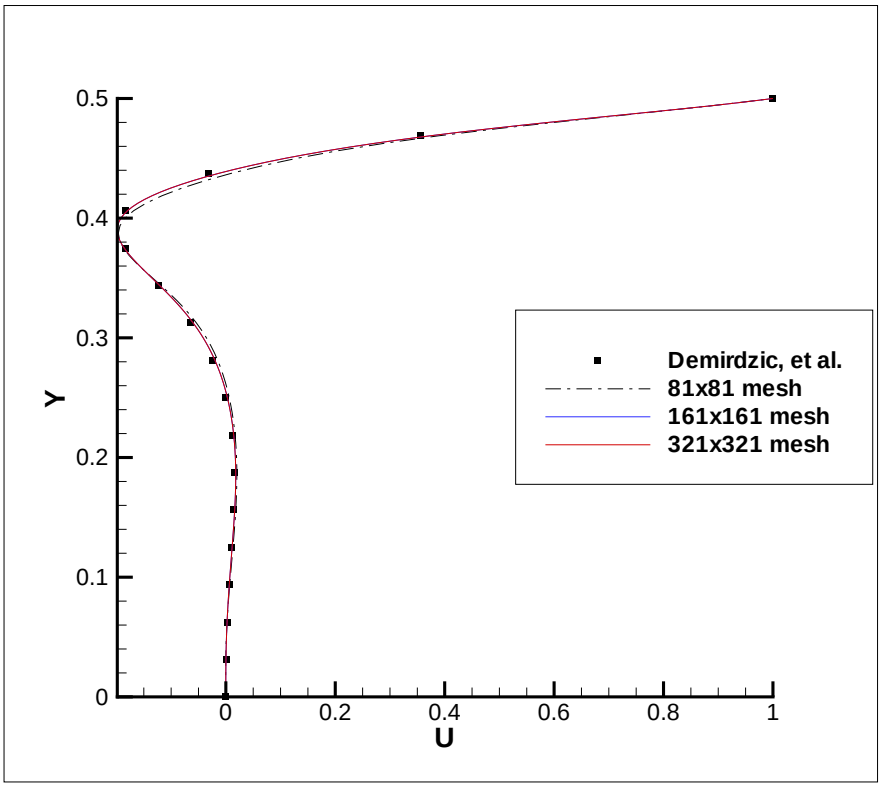

Figure 3.5: u velocity profiles along line A-B for 30-degree skewed cavity solutions at Re=1000

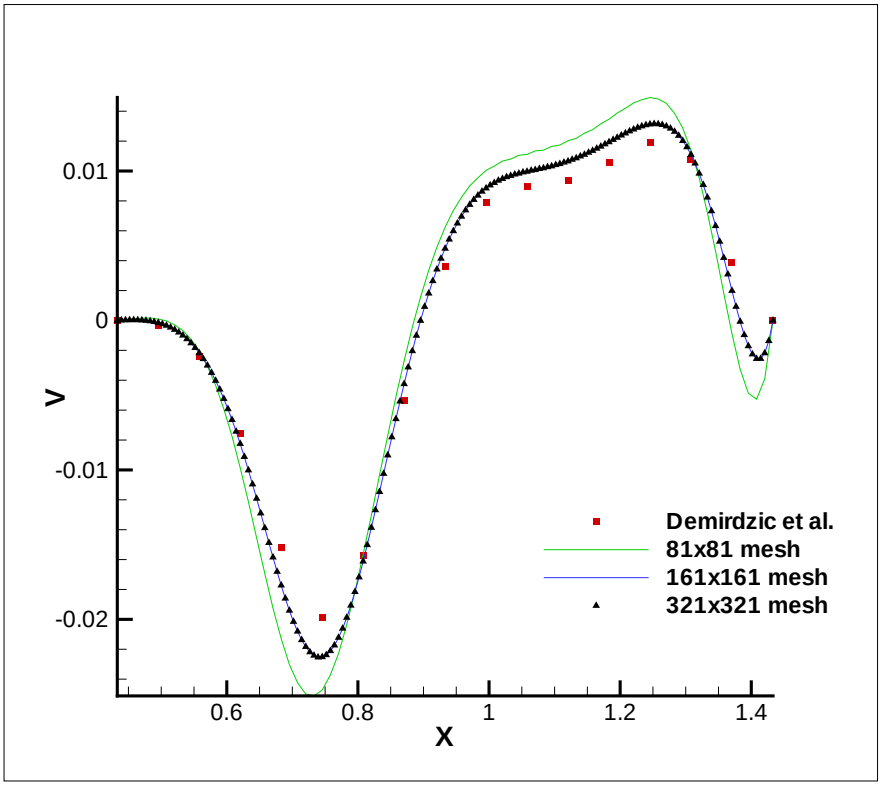

Figure 3.6: v velocity profiles along line C-D for 30-degree skewed cavity solutions at Re=1000

Although u velocity profile is very satisfactory, in v velocity profile, some regions are different from the Demirdzic's solution. It is easily be seen that, 161x161 mesh solution is the best solution because 321x321 solution is the same with 161x161 solution.

3.2.2 Test Case 2: Laminar Flow Through a Gradual Expansion

The geometry for this problem is proposed by Roache [40] and can be seen in Figure 3.7

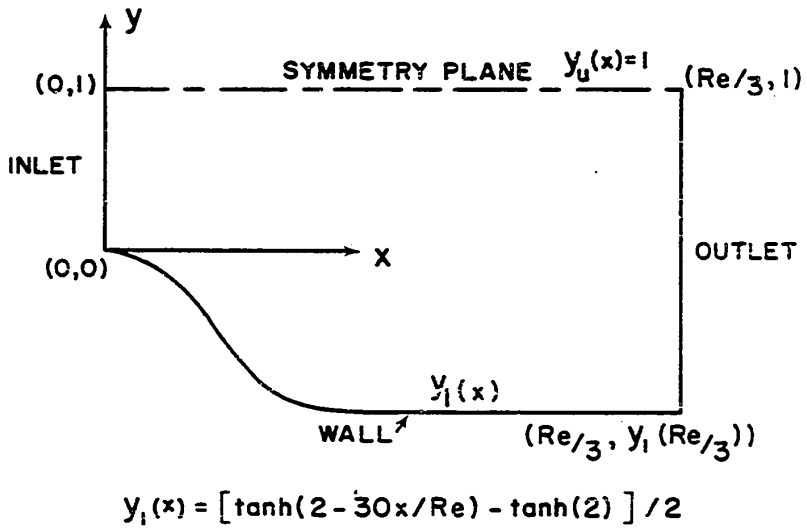

Figure 3.7: Geometry of gradual expansion

As can be seen from Figure 3.7, the geometry is function of Re. For example, in Figure 3.8, grid that is used for Re=10 solution can be seen.

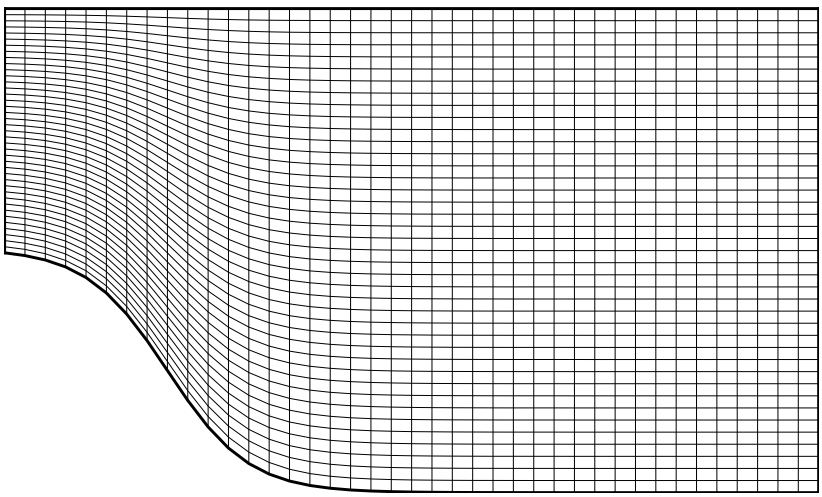

Figure 3.8: 41x41 grid for gradual expansion problem at Re=10

Characteristic length for calculating Re is the height of the inflow boundary. At the inflow boundary, the flow is assumed to be fully developed and the velocity profile is given by

$$u(0, y) = \frac{3}{2}y - \frac{3}{2}y^2 \tag{3.1}$$

Computed pressure values at the solid wall are compared with methods proposed in [41] and [42]. For Re=10 and Re=100, results are displayed in Figure 3.9 and Figure 3.10, respectively. For both Re=10 and Re=100, the results are seen to be in good agreement with the available results in literature.

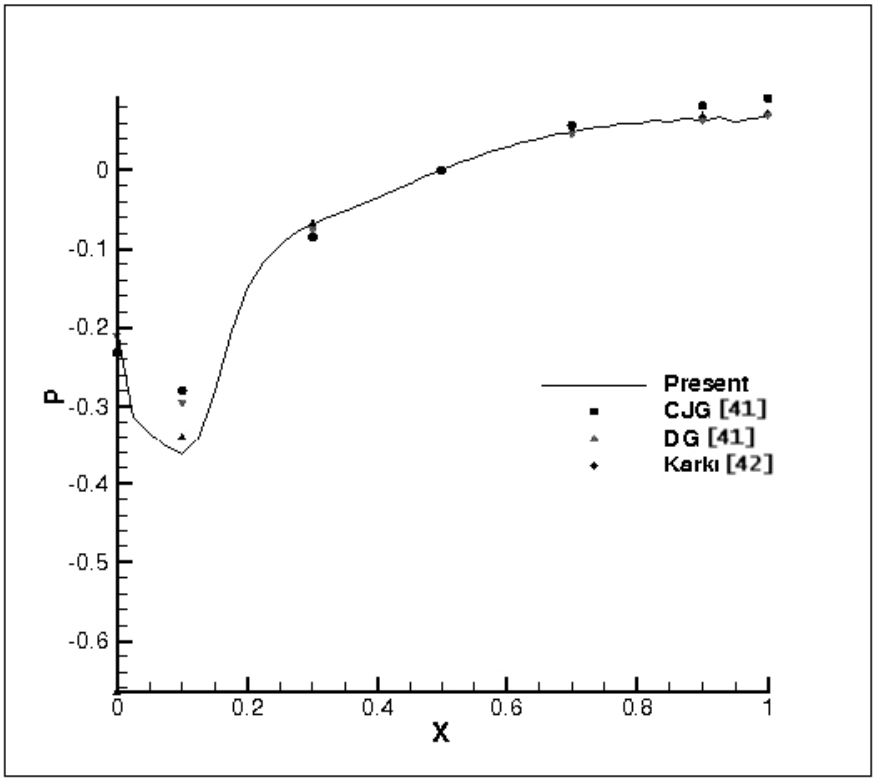

Figure 3.9: Pressure values at solid wall for Re=10

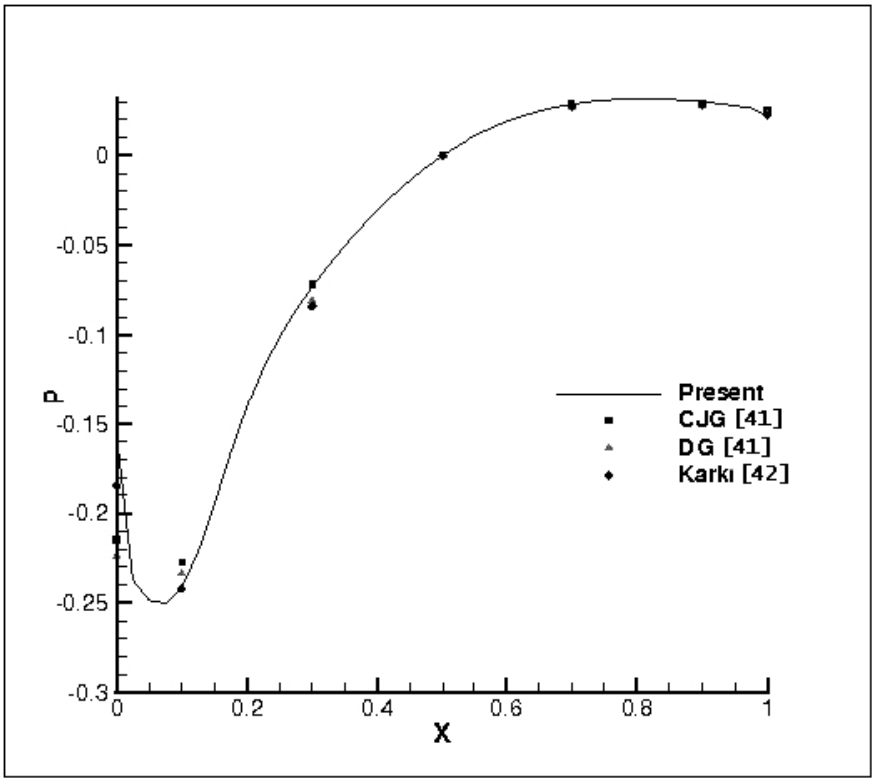

Figure 3.10: Pressure values at solid wall for Re=100

3.3 Compressible Test Cases

3.3.1 Test Case 1: 2D Laminar Flat Plate

To verify the viscous terms modeled correctly, the laminar compressible boundary layer flow is calculated. For this problem, the freestream Mach number is 0.5 and Reynolds number is 10000. The characteristic length is the width of the flat plate.

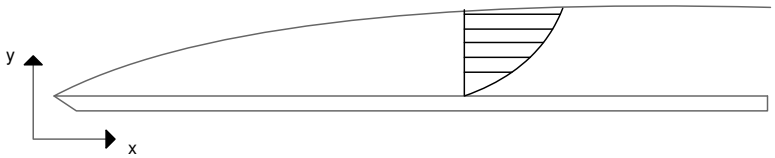

Figure 3.11: 2D boundary layer over a flat plate

The length of the flat plate is calculated from the Reynolds number. Freestream conditions are total pressure of 100000 Pa and total temperature of 288.16 K. In this test case, 81x101 mesh is used. As can be seen from Figure 3.12, at the leading edge of the plate and near solid wall of the plate, grid refinement process is performed.

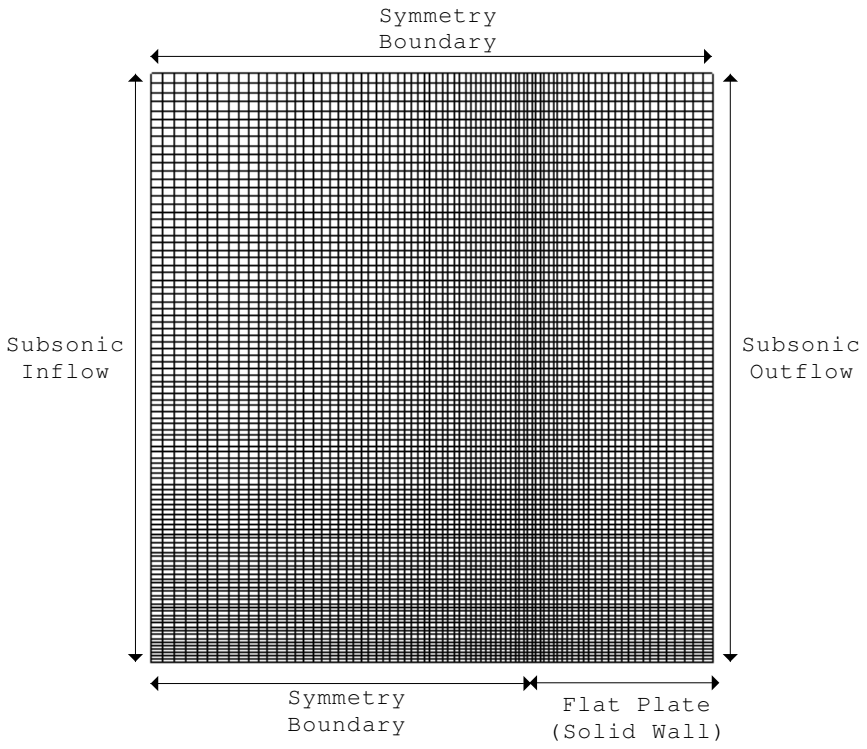

Figure 3.12: Grid used in boundary layer solution

The numerical velocity profile (Figure 3.13) is plotted using similarity variable and compared with the Blasius solution [43]. Calculation of the similarity variable is presented in [43] in detail.

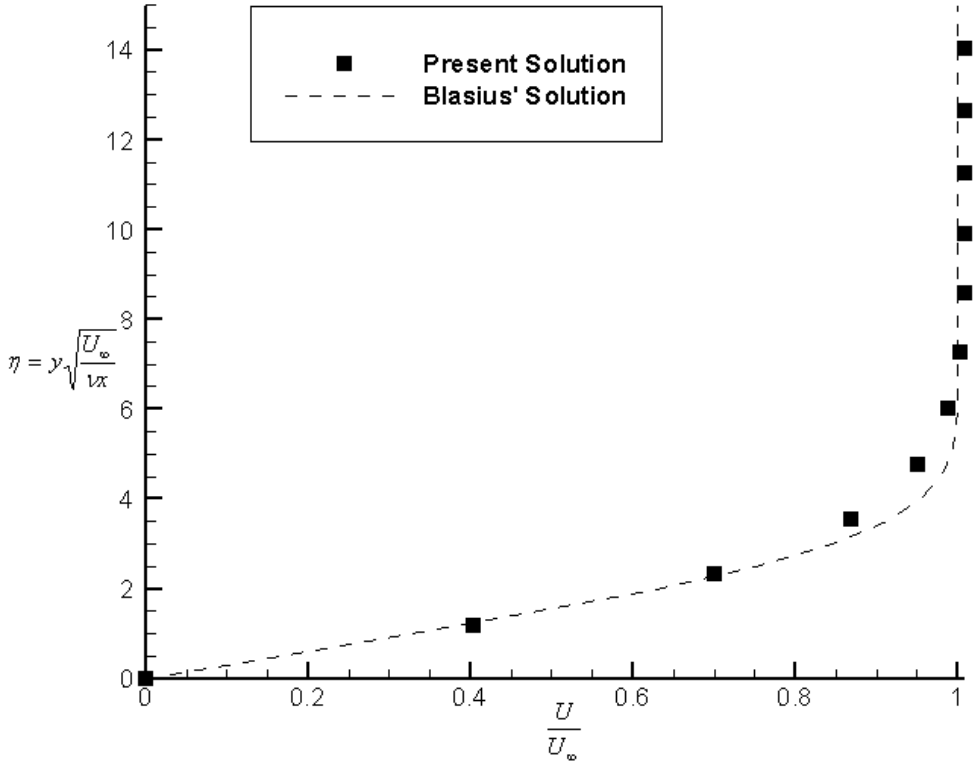

Figure 3.13: Comparison of numerical solution with the Blasius' solution

As can bee seen from Figure 3.13, results agree with Blasius solution except some points near the edge of the boundary layer where $U/U_\infty = 1$.

3.3.2 Test Case 2: 2D Converging-Diverging Nozzle

To test the capability of the solver, 2D converging-diverging nozzle is chosen because of the geometry in which seperation occurs. However, in literature, almost all solutions of this problem are inviscid. Due to this reason, a commercial code - FLUENT - is used to validate the solution.

Because of the geometry of the nozzle, it is sufficiet to solve only the upper half of the nozzle. As an inlet boundary condition, pressure inlet is selected. North boundary is solid wall, south boundary is symmetry and outlet subsonic outflow. In Figure 3.14, the geometry and boundary types of the nozzle can be seen.

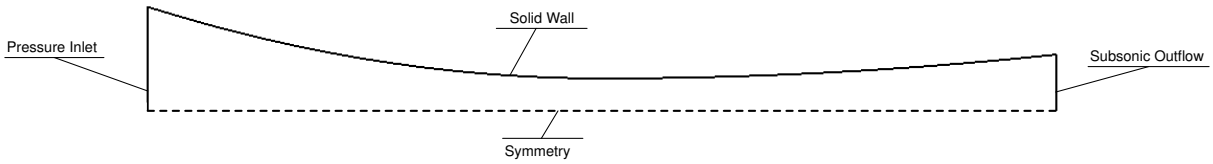

Figure 3.14: Geometry of 2D Converging-Diverging Half Nozzle

61x21 mesh used in this problem can be seen in Figure 3.15

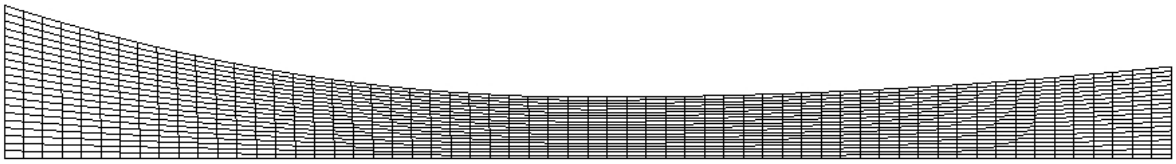

Figure 3.15: 61x21 mesh used in 2D Converging-Diverging Half Nozzle problem

Freestream total pressure and temperature values are 101325 Pa and 500 K, respectively. The Reynolds number is selected as 10000. The characteristic length in calculation of Re is height of the inlet. To calculate the static temperature and pressure, isentropic flow relations are used:

$$P_s = P_0(1 + \frac{\gamma - 1}{2} M_\infty^2)^{-\frac{\gamma}{\gamma - 1}}$$
$$T_s = T_0(1 + \frac{\gamma - 1}{2} M_\infty^2)^{-1} \tag{3.2}$$

As mentioned before, because of the reversed flow in the outlet, it is required to have special treatment on outlet boundary because reversed flow at the outlet enters the nozzle again. This means that, in this region boundary type is not an outflow anymore. Figures 3.16 and 3.17 show reversed flow at the outlet boundary.

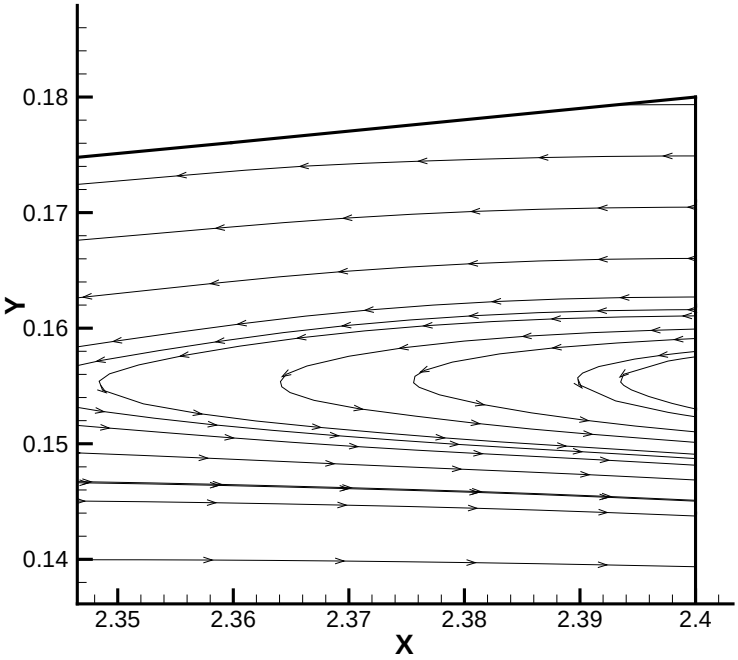

Figure 3.16: Reversed Flow at the Upper Right Corner of the Nozzle - FLUENT Solution

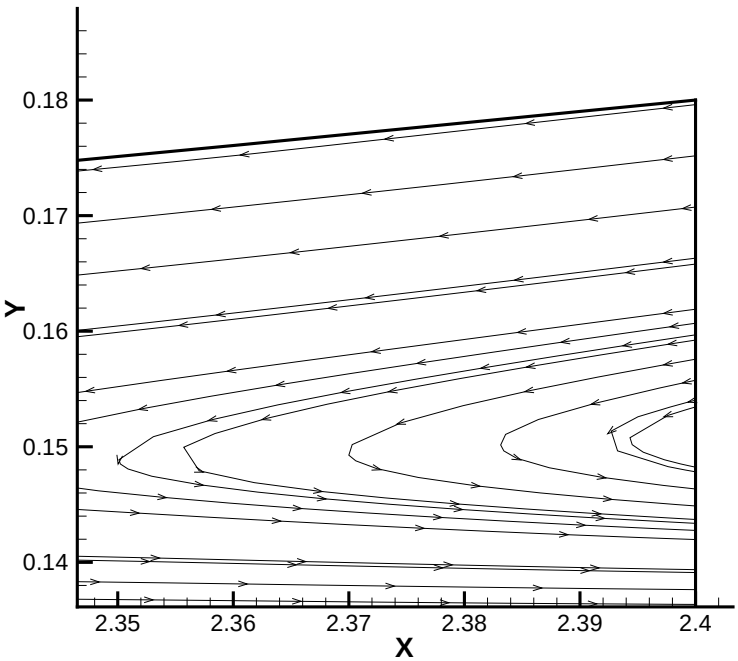

Figure 3.17: Reversed Flow at the Upper Right Corner of the Nozzle - Present Solution

In Figures 3.18, 3.19, 3.20 and 3.21, present solution's static pressure, static temperature, Mach number and density values at symmetry boundary are compared with FLUENT solution.

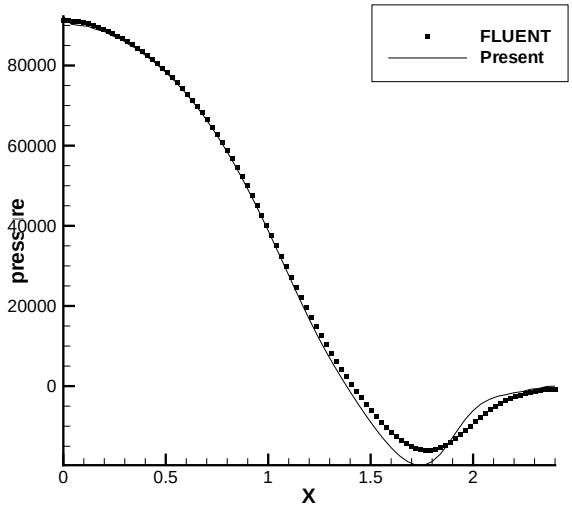

Figure 3.18: Static pressure at symmetry boundary

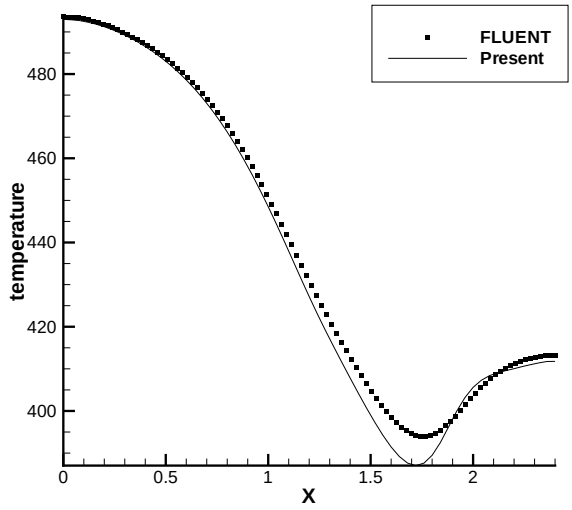

Figure 3.19: Static temperature at symmetry boundary

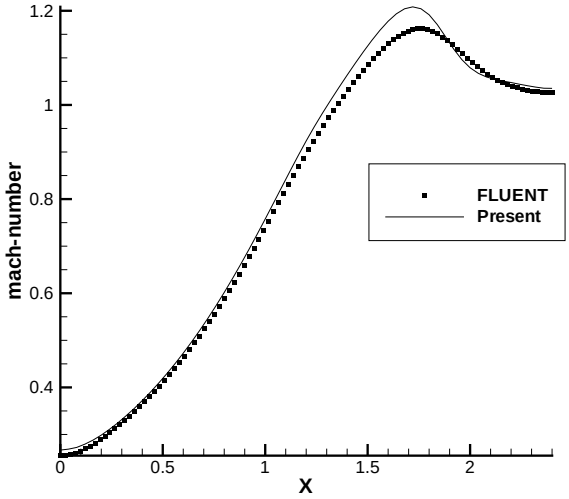

Figure 3.20: Mach Number at symmetry boundary

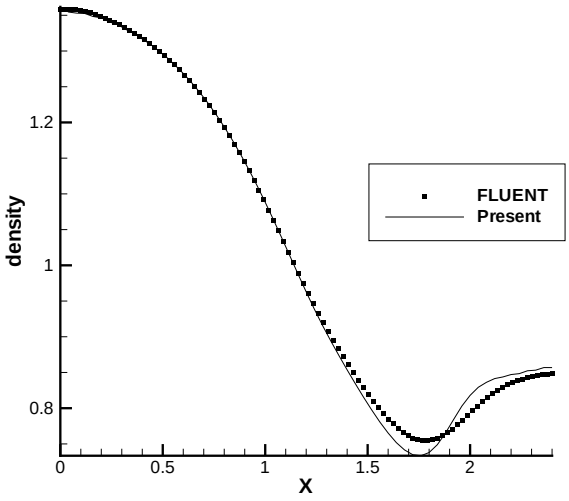

Figure 3.21: Density at symmetry boundary

The difference between the present and FLUENT solution may be caused due to different boundary condition implementations at outflow boundary. Despite of these small differences, it can be said that the developed code is capable of solving viscous compressible flows without shocks even in supersonic region.

CHAPTER 4

CONCLUSIONS

In this thesis, a pressure based solver for computing both incompressible and compressible Navier-Stokes equations is developed. To be able to use non-orthogonal grids in solutions, cross-derivative terms in fluxes coming from non-orthogonality are implemented to Spalding and Patankar's SIMPLE algorithm. To avoid pressure checkerboarding effect, Rhie and Chow momentum interpolation technique is used for the evaluation of velocities at cell faces.

For validation of the code, numerical computations are presented for incompressible and compressible solutions of four different test cases. Incompressible solutions easily conclude that the code is sufficient to solve incompressible flows. If upwind differencing is used for the calculation of convective fluxes, it is examined that the code can simulate flows upto Re=10000.

After a few tests with the code, it can be concluded that the code has not a shock capturing capability. This may be caused from the calculation of temperature field or any other error term during passing the shock. However, based on the results of compressible test cases, for calculating low compressible through compressible flow regimes without shocks, the code gives very close solutions with compared data.

REFERENCES

[1] Chorin, A. J., "*A Numerical Method for Solving Incompressible Viscous Flow Problems*", Journal of Computational Physics, Vol. 2, pp. 12-26, 1967.

[2] Harlow, F., Welsh, J."*Numerical Calculation of Time-Dependent Viscous Incompressible Flow with Free Surface*", Physics of Fluids, vol.8, pp. 2182-2189, 1965.

[3] Patankar, S. V., Spalding, D. B."*A Calculation Procedure for Heat, Mass and Momentum Transfer in Three Dimensional Parabolic Flows*",International Journal of Heat and Mass Transfer, vol.15, pp. 1787-1806, 1972.

[4] Patankar, S. V.,"*Numerical Heat Transfer and Fluid Flow*",Series in Computational Methods in Mechanics and Thermal Sciences, Hemisphere, New York, 1980.

[5] Kobayashi, M., Pereira, C."*Calculation of Incompressible Laminar Flows on a Nonstaggered, Nonorthogonal Grid*",Numerical Heat Transfer, Part B, vol.19, pp. 243-262, 1991.

[6] Peric, M., Kesler, R. and Scheurer, G."*Comparison of Finite-Volume Numerical Methods with Staggered and Colocated Grids*",Computer and Fluids, vol.16, pp. 389-403, 1988.

[7] Lapworth, B. L.,"*Examination of Pressure Oscillations Arising in the Computation of Cascade Flow Using a Boundary-Fitted Co-ordinate System*", International Journal of Numerical Methods in Fluids, vol.8, pp. 387-404, 1988.

[8] Peric, M.,"*Analysis of Pressure-Velocity Coupling on Non-orthogonal Grids*", Numerical Heat Transfer, Part B, vol.17, pp. 63-82, 1990.

[9] Jessee, J., Fiveland, W."*A Cell Vertex Algorithm for the Incompressible Navier-Stokes Equations on Non-orthogonal Grids*",International Journal of Numerical Methods in Fluids, vol.23, pp. 271-293, 1996.

[10] Choi, S., Nam, H., Lee, Y. and Cho, M.,"*An Efficient Three-Dimensional Calculation Procedure for Incompressible Flows in Complex Geometries*", Numerical Heat Transfer, Part B, vol.23, pp. 387-400, 1993.

[11] Jeng, N., Liou, Y.,"*On the Open Boudary Condition for the SIMPLE Algorithm Using Nonstaggered Grids*", Numerical Heat Transfer, Part B, vol.27, pp. 23-42, 1995.

[12] Rhie, C. M., Chow, W. L. "*Numerical Study of the Turbulent Flow Past an Airfoil With Trailing Edge Separation*", AIAA Journal, vol. 21, pp. 1525 - 1532, 1983.

[13] Melaaen, M.,"*Calculation of Fluid Flow with Staggered and Nonstaggered Curvilinear Nonorthogonal Girds - the Theory*", Numerical Heat Transfer, Part B, vol. 21, pp. 2139, 1992.

[14] Melaaen, M.,"*Calculation of Fluid Flow with Staggered and Nonstaggered Curvilinear Nonorthogonal Girds - the Comparison*", Numerical Heat Transfer, Part B, vol. 21, pp. 119, 1992.

[15] Mathur, S., Murthy, J. "*A Pressure-Based Method for Unstructured Meshes*", Numerical Heat Transfer, Part B, vol. 31, pp. 195 - 215, 1997.

[16] van Doormal, J. P., Raithyb, G.D., "*Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flow*",Numerical Heat Transfer, vol. 7, pp. 147-163, 1984.

[17] Spalding, D. B.,"*Mathematical Modelling of Fluid Mechanics, Heat Transfer and Mass Transfer Processes*", Mechanical Engineering Report, No HTS/80/1, Imperial College, London, 1980.

[18] Issa, R. I.,"*Solution of the Implicitly Discretised Fluid Flow Equations by Operator-Splitting*", Internal Report, Dept. of Min. Resources Engineering, Imperial College, London, 1982.

[19] Jameson, A., Schmidt, J. and Turkel, E.,"*Numerical Solutions to the Euler Equations by Finite Volume Methods Time Stepping*", AIAA paper, AIAA-81-1259, 1981.

[20] Steger, J. L., Warming, R. F.,"*Flux Vector Splitting of the Inviscid Gas Dynamic Equations with Application to Finite-Difference Methods*", Journal of Computational Physics, vol. 40, pp. 263-293, 1981.

[21] Godunov, S. K., "*A Difference Scheme for Numerical Solution of Discontinuos Solution of Hydrodynamic Equations*", Math. Sbornik, vol. 47, pp. 271-306, 1959.

[22] Roe, P. L., "*Characteristic-based Schemes for the Euler Equations*", Ann. Rev. Fluid Mech., vol. 18, pp. 337, 1986.

[23] Harlow, F.H., Amsden, A.A., "*A Numerical Fluid Dynamics Calculation For All Flow Speeds*", Journal of Computational Physics, vol. 8, pp. 197, 1971.

[24] Karki, K. C., Patankar, S.V.,"*Pressure Based Calculation Procedure for Viscous Flows at All Speeds in Arbitrary Configurations*", AIAA Journal, vol. 27, pp. 1167-1174, 1989.

[25] Zhou, G., Davidson, L.,"*Transonic Flow Computations Using a Modified SIMPLE Code Based on a Collocated Grid Arrangement*", In First European CFD Cenference, 1992.

[26] Demirdzic, I., Lilek, Z. and Peric, M."*A Collocated Finite Volume Method for Predicting Flows at All Speeds*", International Journal of Numerical Methods, vol. 16, pp. 1029-1050, 1993.

[27] McGuirk, J.J., Paige, G.J.,"*Shock Capturing Using a Pressure Correction Method*", AIAA Journal, vol. 28(10), pp. 1751-1757, 1990.

[28] Lien, F.S., Leschziner, M.A.,"*A Pressure Velocity Solution Strategy for Compressible Flow and Its Application to Shock/Boundary-layer Interaction Using Second Moment Turbulence Closure*", Journal of Fluids Engineering, vol. 115, pp. 717-725, 1993.

[29] Rodi, W., Majumdar, S. and Schönung, B."*Finite Volume Methods for Two-Dimensional Incompressible Flows With Complex Boundaries*", Computer Methods in Applied Mechanics and Engineering, vol. 75, pp. 369 - 392, 1989.

[30] Majumdar, S., Rodi, W. and Vanka, S.P., "*On the Use of Non-staggered pressure-Velocity Arrangement for Numerical Solution of Incompressible Flows*", Report SFB 210/T/35, University Of Kalsruhe, 1987.

[31] Ferziger, J.H., Peric, M., "*Computational Methods for Fluid Dynamics*", 3rd, rev. edition, Springer, 2002.

[32] Blazek, J., "*Computational Fluid Dynamics: Principles and Applications*", 1st edition, Elsevier, 2001.

[33] Norris, S.E., "*Doctoral Thesis*", University of Sydney, Sydney, 2000.

[34] Roychowdhury, D.G., Das, S.K. and Sundararajan, T."*A Collocated Finite Volume Method for Predicting Flows at All Speeds*", International Journal of Numerical Methods, vol. 45, pp. 741-763, 1999.

[35] Darbandi, M., Schneider, G.E. and Hosseinizadeh, S.F."*Solving Compressible Flow Using Incompressible Procedure*", 35th AIAA Thermophysics Conference, Anaheim, California, June 11-14, 2001.

[36] Ghia, U., Ghia, K. N. and Shin, C. T.,"*High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and Multigrid Method*", Journal of Computational Physics, vol. 48, pp. 387-411, 1982.

[37] Demirdzic, I, Lilek, Z. and Peric, M."*Fluid Flow and Heat Transfer Test Problems for Non-Orthogonal Grids: Bench-Mark Solutions*", International Journal of Numerical Methods in Fluids, vol. 15, pp. 329-354, 1992.

[38] Erturk, E., Dursun, B."*Numerical Solutions of 2-D Steady Incompressible Flow in a Driven Skewed Cavity*", ZAMM Z. Angew. Math. Mech. 87, vol. 5, pp. 377-392, 2007.

[39] Hirt, C. W., Amsden, A. A.,Cook, J. L., "*An Arbitrary Lagrangean-Eulerian Computing Method for All Speed Flows*", Journal of Computational Physics, vol. 14, pp. 226-253, 1974.

[40] Roache, P., "*Scaling of High Reynolds Number Weakly Seperated Channel Flows*", Numerical and Physical Aspects of Aerodynamic Flows, Springer-Verlag, 1981.

[41] Napolitano, M. and Orlandi, P., "*Laminar Flow in Complex Geometry*", International Journal for Numerical Methods in Fluids, vol. 5, pp. 667-683, 1985.

[42] Karki, K. C., "*A Calculation Procedure For Viscous Flows At All Speeds in Complex Geometries*", The University of Minnesota, 1986.

[43] Schlichting, H., "*Boundary Layer Theory*", McGraw-Hill Inc, 1979.

[44] Kho, C., "*A Unified Formulation for Mixed Incompressible/Compressible Flows*", McGill University, Quebec, Canada, 1997.