DEVELOPMENT OF AN INCOMPRESSIBLE NAVIER-STOKES SOLVER WITH ALTERNATING CELL DIRECTION IMPLICIT METHOD ON STRUCTURED AND UNSTRUCTURED QUADRILATERAL GRIDS

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

BY

ONUR BAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

SEPTEMBER 2007

Approval of the thesis:

DEVELOPMENT OF AN INCOMPRESSIBLE NAVIER-STOKES SOLVER WITH ALTERNATING CELL DIRECTION IMPLICIT METHOD ON STRUCTURED AND UNSTRUCTURED QUADRILATERAL GRIDS

submitted by **ONUR BAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

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

Prof. Dr. İsmail Hakkı Tuncer
Head of Department, **Aerospace Engineering** _____

Prof. Dr. İsmail Hakkı Tuncer
Supervisor, **Aerospace Engineering Dept., METU** _____

Dr. Ali Ruhşen Çete
Co-supervisor, **Turbotek Inc.** _____

Examining Committee Members:

Prof. Dr. Yusuf Özyörük
Aerospace Engineering Dept., METU _____

Prof. Dr. İsmail Hakkı Tuncer
Aerospace Engineering Dept., METU _____

Dr. Ali Ruhşen Çete
TURBOTEK Inc. _____

Prof. Dr. Ünver Kaynak
Electrical and Electronical Engineering Dept., TOBB ETU _____

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.

Name, Last Name: Onur Baş

Signature :

ABSTRACT

DEVELOPMENT OF AN INCOMPRESSIBLE NAVIER-STOKES SOLVER WITH ALTERNATING CELL DIRECTION IMPLICIT METHOD ON STRUCTURED AND UNSTRUCTURED QUADRILATERAL GRIDS

Baş, Onur

M.S., Department of Aerospace Engineering

Supervisor: Prof. Dr. İsmail Hakkı Tuncer

Co-Supervisor: Dr. Ali Ruhşen Çete

September 2007, 58 pages

In this research, the Alternating Cell Direction Implicit method is used in temporal discretisation of the incompressible Navier-Stokes equations and compared with the well known and widely used Point Gauss Seidel scheme on structured and quadrilateral unstructured meshes. A two dimensional, laminar and incompressible Navier-Stokes solver is developed for this purpose using the artificial compressibility formulation. The developed solver is used to obtain steady-state solutions with implicit time stepping methods and a third order data reconstruction scheme (U-MUSCL) is added to obtain high order spatial accuracy. The Alternating Cell Directions Implicit method and Point Gauss Seidel scheme is compared in terms of convergence iteration number and total computation time using test cases with growing complexity, including laminar flat plate, single and multi-element airfoil calculations. Both structured and quadrilateral unstructured grids are used in single element airfoil calculations. In these test cases, it is seen that a reduction between 13% and 20% is obtained in total computation time

by usage of Alternating Cell Directions Implicit method when compared with the Point Gauss Seidel method.

Keywords: Alternating Cell Direction Implicit Method, Artificial Compressibility Method, High Order, U-MUSCL, Unstructured, Quadrilateral.

ÖZ

YAPISAL VE YAPISAL OLMAYAN DÖRTGEN ÇÖZÜM AĞLARINDA DEĞİŞKEN HÜCRE YÖNLÜ KAPALI METOD KULLANAN SIKIŞTIRILAMAZ NAVIER-STOKES ÇÖZÜCÜ GELİŞTİRİLMESİ

Baş, Onur

Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. İsmail Hakkı Tuncer

Ortak Tez Yöneticisi: Dr. Ali Ruhşen Çete

Eylül 2007, 58 sayfa

Bu araştırmada, sıkışamaz Navier-Stokes denklemlerinin zamanda ayrıklaştırılmasında Değişken Hücre Yönlü Kapalı metod kullanılmış ve yapısal ve yapısal olmayan dörtgen ağlarda, yaygın olarak kullanılan Noktasal Gevşeme metodu ile karşılaştırılmıştır. Bu amaçla yapay sıkışabilirlik formülasyonunu kullanan iki boyutlu ve laminer bir sıkışamaz Navier-Stokes çözücüsü geliştirilmiştir. Geliştirilen çözücü ile kapalı zaman adımı kullanılarak durağan çözümler alınmış ve uzayda yüksek hassasiyetli sonuçlar alınması için geliştirilen koda üçüncü dereceden akışkan değişkeni oluşturma (U-MUSCL) şeması eklenmiştir. Değişken Hücre Yönlü Kapalı metodu ve Noktasal Gevşeme metodu, yakınsama adım sayısı ve toplam çözüm zamanı açısından artan karmaşıklıktaki düz levha ve tek ve iki elemanlı kanat profili üzerindeki akışta karşılaştırılmıştır. Tek elemanlı kanat profili çözümlerinde hem yapısal hem de dörtgen yapısal olmayan çözüm ağı kullanılmıştır. Bu test problemlerinde, Noktasal Gevşeme metodu ile karşılaştırıldığında

Değişken Hücre Yönlü Kapalı metod kullanımı toplam çözüm zamanında %13 ile %20 arasında düşüş sağlamıştır.

Anahtar Kelimeler: Değişken Hücre Yönlü Kapalı Metod, Yapay Sıkışabilirlik Metodu, Yüksek Mertebeli, U-MUSCL, Yapısal Olmayan, Dörtgen.

Dedicated to my family ...

ACKNOWLEDGMENTS

I would like to express my gratitude to Prof. Dr. İsmail Hakkı TUNCER for his supervision and to Dr. Ali Ruhşen ÇETE for his continuous support and guidance at every step of this thesis.

I would like to thank Prof. Dr. Ünver KAYNAK and Prof. Dr. Haluk Aksel for their suggestions and motivation.

I would also like to thank to my colleagues and friends Özlem Ceyhan, Kerem Denk, Emre Gürdamar, Ali Aktürk, Güneş Nakiboğlu, Umut Can Dinçgez and Kıvanç Ülker for their friendship and support. I also appreciate the continuous LaTeX assistance of Tahir Turgut during the writing process.

I am grateful to Dr. Mine Yumuşak, Suzan Koç and other colleagues at ROKETSAN for their patience and support.

Special thanks go to my uncle Ali Ünver for creating my engineering enthusiasm.

I owe a special thank to my beloved fiancee Meryem Meltem Tatlıer for her precious spiritual support and patience for sharing my stress.

Finally, I want to express my deepest thanks to my parents and brother for their loving care and

efforts to whom this thesis is devoted. This work wouldn't be done without the support of them.

This study was partially supported by the Scientific and Technical Research Council of Turkey (TÜBİTAK) within TİDEB project 3040091.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS

ROMAN SYMBOLS

Q	Flow variables
u, v	Velocity components
P	Pressure
Re	Reynolds number
c	Artificial speed of sound
I	Flux passing through an edge
$\vec{V}$	Velocity vector
$\vec{I}$	Unit vector
$\vec{F}$	Flux vector
$\vec{r}$	Position vector
$\vec{R}$	Residual vector
$\hat{n}$	Normal vector
$\overline{A}$	Flux Jacobian
A	Area
S	Length
m	maximum edge number

GREEK SYMBOLS

ϕ	Velocity tangent to the edge
β	Artificial compressibility parameter
λ	Eigenvalues
χ	U-MUSCL parameter
μ	Kinematic viscosity
τ	Shear stress

SUBSCRIPTS

c	Convective
v	Viscous
x	x-component
y	y-component
L	Left state
R	Right state
Roe	Computed using Roe averaged values

CHAPTER 1

INTRODUCTION

1.1 Background

Computational Fluid Dynamics (CFD) is a branch of science, that analysis fluid flow problems numerically by solving discretised equations. The value of CFD has increased considerably in parallel to increasing computer capabilities. In recent years, it has become possible to solve practical fluid flow problems using sophisticated numerical methods and fast computers. Because of that, for most cases, CFD takes a role both in design and analysis of bodies immersed in fluids. There are numerous examples for the usage of computational fluid dynamics, where it becomes an invaluable tool by considerably reducing the need for experiments and their costs. CFD is constituted from a wide range of numerical solution methods from potential to viscous flow solutions varying in complexity and computation time. Among these methods, the solution of the Navier-Stokes equations is one of the most widely used techniques for compressible and incompressible fluid flows.

1.2 Incompressible Flow

Incompressibility assumption is valid for most practical fluid flow problems where flow speeds are low. Industrial fluid flows such as melts, flows around low speed airplanes, cars, trains and buildings, water flows and flows of most biomedical fluids such as blood can be given as exam-

ples to the incompressible fluid flows. Also in recent years, there has been a growing interest in the low speed unmanned air vehicles for which, the flow can also be considered as incompressible.

Incompressibility can be thought as the limit of the compressible Navier-Stokes equations where density is constant all over the flow domain and the propagation speed of pressure waves are infinitely high. For low Mach number flows, compressible Navier-Stokes equation solvers suffer from severe stability and accuracy problems due to the difference between the flow and the wave propagation speeds. To overcome this problem, it is possible to use the modified Navier-Stokes equations with incompressibility assumption.

A simplification occurs in the incompressible N-S equations owing to the decoupling of the energy equation from the momentum and the continuity equations. Because of this decoupling, it is not necessary to solve the conservation of energy equation in each solution step but it is possible to solve this equation easily when a heat or work interaction occurs in the problem.

1.3 Solution Methodologies for the Incompressible N-S Equations

Although the incompressible Navier-Stokes equations constitute a subset of the compressible N-S equations, they are treated separately and in general are more difficult to solve. The major difficulty faced in the solution of the incompressible Navier-Stokes equations is the lack of time evolution term of density in the continuity equation, which shows an elliptic character. Due to the missing term, a straight forward application of time stepping schemes used for compressible N-S solvers is not possible without modifications. Also, the incompressible Navier-Stokes equations pose a special kind of problem because of the continuity equation's elliptic character and the infinite speed of pressure waves. Without any modification, the well known Courant-Frederichs-Lewy (CFL) stability condition will force a maximum allowed time step of zero.

Various methods can be found in the literature for the solution of incompressible Navier-Stokes equations. Different solution approaches can be put into three main groups:

- Vorticity-Stream function method
- Pressure based methods
- Artificial compressibility method

In the first category, the incompressible N-S equations are written in terms of the stream function and vorticity, and velocities and pressure are determined according to these values [1]. The major advantage of this method is that the continuity equation is satisfied authomatically, and two equations are solved instead of three equations for two dimensional problems. On the other hand, the boundary conditions are implemented for the stream function and vorticity values instead of the primitive variables. This method is suitable for 2-D problems but extension to 3-D is not straight-forward because, a unique stream function does not exist in 3-D. Although some formulations such as vorticity-velocity approach [2] exist in the literature, selection of boundary conditions is challenging [3].

The second category is the pressure based methods which consist of mainly, Marker and Cell (MAC) method due to Harlow and Welch [4], the projection method due to Chorin [5] and the popular Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) family of schemes due to Patankar and Spalding [6]. The MAC method was first proposed by Harlow and Welch in 1965. In this method, pressure is used as a parameter to satisfy the continuity equation and this is the first primitive variable method using the Poisson equation for pressure [7] where the Poisson equation is obtained by taking the divergence of the momentum equations. In the MAC method, it is necessary to choose the correct pressure field at the current time step which satisfies the continuity equation in the next time level. In this method the strict requirement of generating a divergence free velocity field at each time step may decrease the overall computational efficiency significantly [7].

For pressure correction methods, it is possible to simplify the estimation method of pressure and velocity corrections in cases where only steady state solution is required. In such cases, the correct pressure field becomes necessary only when the solution is converged. The best known

method using this approach is the SIMPLE method. Despite its empiricism in estimating the pressure and velocity correction methods, it is used successfully for many steady-state computations [8], [9]. Although pressure correction methods are applied successfully to a variety of problems, accuracy and performance of their solutions are highly dependent on the performance of the pressure Poisson equation solver. This can be very expensive for time accurate calculations over complex geometries [3].

The third category can be classified as "density like methods" which is also called as the artificial compressibility approach. This method was originally proposed by Chorin [10] for steady problems and it was extended to unsteady flows using a method of subiterations by Peyret [11]. For a truly incompressible flow, the acoustic speed is infinite and the difference between the acoustic speed and convective speed makes the resulting system of equations ill-conditioned. In the artificial compressibility method, the system is relaxed by defining a finite acoustic speed and the resulting system becomes similar to the systems obtained by methods developed for transonic and supersonic compressible flows. This similarity allows the use of some advanced methods developed for the compressible flows on incompressible flows. In the artificial compressibility formulation, a pseudo-time derivative of pressure is added to the continuity equation to relax the strict requirement of satisfying the mass conservation. The resulting set of equations becomes a hyperbolic-parabolic system. The artificial compressibility approach has already been applied widely for both the steady [12], [13], [14], [15] and the unsteady problems [16], [17], [18], [19].

1.4 Time integration

The temporal discretisation methods can be grouped into two main categories as the explicit and implicit schemes. In general, it is known that explicit methods are easy to implement and solution steps are not computationally expensive whereas implicit methods have better convergence characteristics and are not as sensitive to computational grids as the explicit methods.

In most cases, large matrices occur in fully implicit schemes for which solutions become very time consuming. For this reason, fast implicit methods such as the Approximate Factorization (AF), Alternating Direction Implicit (ADI), Point Gauss Seidel (PGS) and Line Gauss Seidel (LGS) methods are developed.

It is also possible to find numerous ways to time-discretise the governing equations of incompressible N-S equations in the literature. For instance, a second order leap-frog scheme is used in the originally suggested scheme by Chorin [10] and Euler implicit schemes are used by Sato et. al. [20] and Chan and Anastasiou [21] for unstructured meshes.

Also, there are some publications comparing the implicit schemes based on artificial compressibility formulations such as Rogers [22] and Yuan [23] for structured grids. For unstructured meshes, Point Gauss Seidel approach is compared with a preconditioned GMRES algorithm by Anderson et. al. [24]. It is observed in those references that, a considerable increase in efficiency can be obtained by using more sophisticated methods than the PGS method.

1.5 Present Approach and Major Goals

The first objective of this study is to develop a robust and time efficient two dimensional incompressible Navier-Stokes solver using cell centered finite volume method over quadrilateral unstructured and structured meshes. For this purpose, a 2-D artificial compressibility formulation is chosen because of its wide usage and its similarity with the compressible flow equations. This similarity allows the usage of most of the methods developed and successfully implemented in the compressible flow solvers.

For the convective fluxes, a high order upwind methodology is chosen above the artificial dissipation schemes because of its improving diagonal dominance of the resulting system and inherent dissipation character.

The second objective is to apply the Alternating Cell Directions Implicit (ACDI) method and investigate its benefits in terms of convergence time and iteration number when compared with the Point Gauss Seidel (PGS) method.

The line implicit methods such as the AF, ADI and LGS are applicable only on structured grids because of the need for the curvilinear coordinates as the solution bands. However, a new line implicit scheme which is proposed by Çete [25] is suitable for both the structured and quadrilateral unstructured meshes. This method was applied to the unsteady heat equation and an extensive stability and accuracy analysis both analytically and numerically are given by Çete [26]. In this method, the cell direction concept is used as a replacement of the curvilinear coordinates used in the methods listed. Within this concept, lines passing through mutual edges are formed and combined to generate solution bands over the sequential cells. When this process finishes, these bands pass twice from each quadrilateral element which allows implementation of fast implicit methods with some modifications. A sample grid portion with constructed solution bands are given in Fig 1.1.

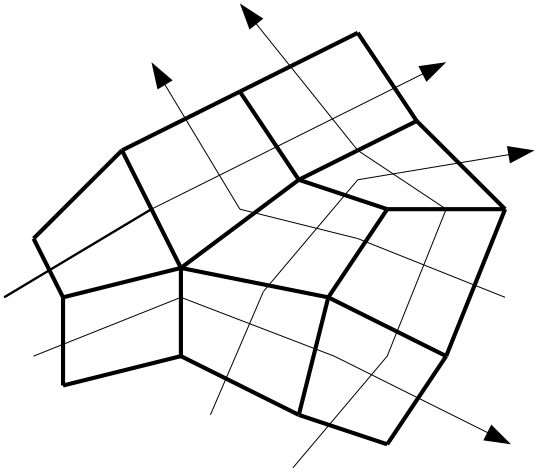

Figure 1.1: A Sample Grid Portion with Solution Bands

The Alternating Cell Directions Implicit (ACDI) method is very similar to the line implicit

methods which are widely used on solvers using structured grids, so that a considerable increase in efficiency is expected. Details of the ACDI method and the results of its application to incompressible N-S equations are given in Chapter 2 and Chapter 3.

1.6 Outline

The present work consists of 4 chapters and in the remaining sections of the present work, the following outline is followed:

Chapter 2 presents the numerical method. This chapter starts with the governing equations of incompressible Navier-Stokes equations with artificial compressibility modification. It includes the details of the spatial and temporal discretisations which are formed by upwinding, a high order reconstruction of flow variables, the characteristic based boundary conditions and the details of the PGS and ACDI temporal discretisation methods. The mesh generation methodology and the data management system are also explained in this chapter.

The results are presented on Chapter 3. In this chapter, several test cases, used for code validation are given including flow over a flat plate, over an airfoil and over a multi element airfoil. Also, the results related with the implementation of the ACDI method and the effects of the flow variable reconstruction order on the spatial accuracy are given in this chapter.

Finally, the conclusions are drawn and some recommendations for future work are made in Chapter 4.

CHAPTER 2

NUMERICAL METHOD

2.1 Governing Equations

The governing equations used in this work are two dimensional incompressible Navier-Stokes equations. Laminar and constant viscosity flows without body forces are considered. The uniform density is taken as unity in the non-dimensionalisation procedure and the resulting set of equations for an arbitrary mesh is given in integral form as

$$\frac{\partial}{\partial t}\int_A Q dA + \oint_S (\vec{F_c} - \vec{F_v}) \cdot \hat{n} dS = 0 \tag{2.1}$$

where Q is the vector of independent variables, F_c is the convective flux and F_v is the viscous flux vectors.

$$Q = \begin{pmatrix} 0 \\ u \\ v \end{pmatrix} \quad , \quad \vec{F_c} = \begin{pmatrix} (u)\cdot\vec{i} + (v)\cdot\vec{j} \\ (u^2+p)\cdot\vec{i} + (uv)\cdot\vec{j} \\ (uv)\cdot\vec{i} + (v^2+p)\cdot\vec{j} \end{pmatrix} \quad , \quad \vec{F_v} = \begin{pmatrix} 0 \\ \tau_{xx}\cdot\vec{i} + \tau_{yx}\cdot\vec{j} \\ \tau_{xy}\cdot\vec{i} + \tau_{yy}\cdot\vec{j} \end{pmatrix}$$

In Eqn (2.1), $\hat{n}$ is the outward-pointing unit normal of the corresponding control volume boundary.

The conservation of energy equation does not appear in the system of equations (2.1) because of its decoupling from the momentum equations with the incompressibility assumption. The energy equation can be solved separately when needed.

In this study, the flow variables are non-dimensionalised by using the following relations.

$$t^\star = \frac{tU_\infty}{c} \ , \ x^\star = \frac{x}{c} \ , \ u^\star = \frac{u}{U_\infty} \ , \ p^\star = \frac{p - p_\infty}{\rho U_\infty^2}$$

Where c is the chord length, U_∞ is the free stream velocity and p_∞ is the free stream pressure. Substituting these equations yields the non-dimensional form of the incompressible Navier-Stokes equations.

For a Newtonian fluid, the shear stress components are related to the velocity gradients with the constitutive laws and the expressions of them are given as

$$\tau_{xx} = 2\mu\frac{\partial u}{\partial x} \ , \ \tau_{xy} = \tau_{yx} = \mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) \ , \ \tau_{yy} = 2\mu\frac{\partial v}{\partial y} \tag{2.2}$$

After inserting the expressions of shear stress terms and dropping the terms equal to the continuity equation, the resulting non-dimensional viscous flux terms are given below [27]

$$\vec{F_v} = \frac{1}{Re} \begin{pmatrix} 0 \\ \frac{\partial u}{\partial x} \cdot \vec{i} + \frac{\partial u}{\partial y} \cdot \vec{j} \\ \frac{\partial v}{\partial x} \cdot \vec{i} + \frac{\partial v}{\partial y} \cdot \vec{j} \end{pmatrix}$$

2.2 Method of Artificial Compressibility

As stated in the previous chapter, the elliptic nature of the continuity equation is the major difficulty for the numerical solution of incompressible N-S equations. Because of the lack of time derivative in the continuity equation, simple time stepping schemes are not suitable for the original governing equations. As proposed on the original paper by Chorin [10], the steady state formulation of the artificial compressibility method is formed by adding a time-like (pseudo time) derivative of the pressure to the continuity equation and changing the time derivatives of the momentum equations to pseudo time derivatives. As a result of these modifications, although the converged results are correct, the time dependent results become wrong. For this reason, the resulting set of equations is called as "$false\ transient$". The modified continuity equation is given below in differential form.

$$\frac{1}{\beta}\frac{\partial p}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{2.3}$$

The β parameter used in Eqn (2.3) is named as artificial compressibility parameter and $\sqrt{\beta}$ is named as artificial speed of sound due to the analogy with the compressible flow equations.

Each of the governing equations, modified according to the artificial compressibility concept, has hyperbolic or hyperbolic-parabolic character. For this reason, most of the methods used in numerical solution of compressible Navier-Stokes equations can be used in this equation set without modification. The integral form of the modified governing equations, which are used in this work, are given below,

$$\frac{\partial}{\partial t}\int_A Q dA + \oint_S (F_c - F_v)\cdot \hat{n} dS = 0 \tag{2.4}$$

where,

$$Q = \begin{pmatrix} p \\ u \\ v \end{pmatrix} \quad , \quad \vec{F_c} = \begin{pmatrix} \beta\vec{V} \\ (u^2+p)\cdot\vec{i} + (uv)\cdot\vec{j} \\ (uv)\cdot\vec{i} + (v^2+p)\cdot\vec{j} \end{pmatrix} \quad , \quad \vec{F_v} = \frac{1}{Re} \begin{pmatrix} 0 \\ \frac{\partial u}{\partial x}\cdot\vec{i} + \frac{\partial u}{\partial y}\cdot\vec{j} \\ \frac{\partial v}{\partial x}\cdot\vec{i} + \frac{\partial v}{\partial y}\cdot\vec{j} \end{pmatrix}$$

2.3 Flux Evaluations

In this study, finite volume method is used for spatial discretisation of the governing equations over quadrilateral elements. The main reason of using quadrilateral elements is explained extensively in the temporal discretisation section.

A cell centered scheme is used for discretisation in which flow variables are associated with the cell centers. Derivatives are evaluated using Green-Gauss theorem and first, second and third order variable reconstruction schemes are applied for flux evaluation over the edges. In spatial discretisations, each term must be treated in the governing equations with a suitable technique for its character to obtain a stable scheme. To show the characteristics of each term, the differential form of the governing equations is more suitable and given as

$$\underbrace{\frac{\partial Q}{\partial t}}_{Term1} + \underbrace{(\frac{\partial f_c}{\partial x} + \frac{\partial g_c}{\partial y})}_{Term2} = \underbrace{\frac{1}{Re}\,(\frac{\partial f_v}{\partial x} + \frac{\partial g_v}{\partial y})}_{Term3} \tag{2.5}$$

where,

$$Q = \begin{pmatrix} p \\ u \\ v \end{pmatrix}, f_c = \begin{pmatrix} (\beta\cdot u) \\ (u^2+p) \\ (uv) \end{pmatrix}, g_c = \begin{pmatrix} (\beta\cdot v) \\ (uv) \\ (v^2+p) \end{pmatrix}, f_v = \frac{1}{Re}\begin{pmatrix} 0 \\ \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial x} \end{pmatrix}, g_v = \frac{1}{Re}\begin{pmatrix} 0 \\ \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial y} \end{pmatrix}$$

The second term of Eqn (2.5) is called as the convective flux. This term has a hyperbolic

character with the time derivative term (first term) and forms the incompressible Euler equations. Because of its hyperbolic character, central discretisation does not form a stable scheme and a special treatment is needed. The discretisation of this term will be given in detail in "Convective Fluxes" subsection.

On the other hand, the third term of the Eqn (2.5) is named as the viscous fluxes. This term has an elliptic character on its own and a parabolic character with the time derivative term. Central differencing is suitable for this term and the details will be given in the "Viscous Fluxes" subsection.

2.3.1 Convective Fluxes

It is a well known fact that central discretisation of the convective terms causes oscillations and does not form a stable solution with the Euler time stepping schemes. Artificial dissipation and upwinding are two of the mostly used methods to solve this problem.

Artificial dissipation method is proposed by Jameson [28] and in general consists of a fourth order dissipation term added on the whole domain and a second order dissipation term added on regions with high gradients or discontinuities. The artificial dissipation is a method which is in general easy to implement and in most situations is less costly to form compared with the upwinding. On the other hand, care must be taken to guarantee the artificial dissipation term's being much smaller than the physical viscosity for accuracy and being large enough to suppress the unphysical oscillations.

In this work, upwinding, that has an inherent dissipative character is used because of its advantages over artificial dissipation such as the elimination of the artificial dissipation parameter that needs to be defined by the user. Also, upwinding increases the diagonal dominance considerably and improves greatly the robustness of the code. Although upwinded fluxes are more costly to form, the speed up in convergence results in significant savings in computational time

[29].

2.3.1.1 Upwinding

The upwinding methods are essentially grouped in two categories with their approaches to identify the upwind directions. These two approaches are generally referred as the "*Riemann approach* (*Godunov approach*)" and the "*Boltzmann approach*" [30] in which the respective numerical schemes derived from these approaches are generally called as the "$Flux\ Difference\ Splitting\ Methods$" (FDS) and "$Flux\ Vector\ Splitting\ Methods$" (FVS).

The upwinding procedure used in the FVS methods are achieved with less effort than in Godunov-type methods, leading to simpler and somehow more efficient schemes [31]. This property makes the FVS schemes very popular within a large community, but the reduced sophistication results in poorer resolution when compared with the Godunov-type schemes. It is reported by Van Leer, Thomas and Roe [32] that their FVS scheme is considerably less accurate than Godunov's method with Roe's approximate Riemann Solver.

The scheme used in this work for upwinding was derived originally by Roe [33] as an approximate Riemann solver for the compressible Navier-Stokes equations. This scheme is based on one dimensional theory.

Before giving detailed explanation of the flux difference splitting schemes for incompressible Navier-Stokes equations, it is useful to state the non-applicability of the flux vector splitting schemes such as the Van Leer or Steger-Warming as the reason for using a more complex scheme.

Flux vector splitting schemes have many types such as the Van Leer [34], Steger-Warming [35], AUSM [36], etc. and they are widely used in the literature, especially for the solution of compressible fluid flows. The key feature of the FVS schemes is its reliance on a special property of the fluxes, namely the "*homogeneity*" property [31].

This property can be shown with the necessity of the equality given below.

$$F = \frac{\partial F}{\partial Q} \cdot Q \tag{2.6}$$

The homogeneity property is a strict requirement for the flux vector splitting schemes which is shown below.

$$F = F^+ + F^-$$

where,

$$F^+ = \left(\frac{\partial F^+}{\partial Q}\right) \cdot Q \ , \ F^- = \left(\frac{\partial F^-}{\partial Q}\right) \cdot Q$$

As shown below, the f_c and g_c convective fluxes do not have the homogeneity property and flux vector splitting schemes do not form a stable scheme for artificial compressibility formulation.

$$f_c = \begin{pmatrix} (\beta \cdot u) \\ (u^2 + p) \\ (uv) \end{pmatrix} \neq \begin{pmatrix} 0 & \beta & 0 \\ 1 & 2u & 0 \\ 0 & v & u \end{pmatrix} \begin{pmatrix} p \\ u \\ v \end{pmatrix} = \begin{pmatrix} (\beta \cdot u) \\ (2u^2 + p) \\ (2uv) \end{pmatrix}$$

$$g_c = \begin{pmatrix} (\beta \cdot v) \\ (uv) \\ (v^2 + p) \end{pmatrix} \neq \begin{pmatrix} 0 & 0 & \beta \\ 0 & v & u \\ 1 & 0 & 2v \end{pmatrix} \begin{pmatrix} p \\ u \\ v \end{pmatrix} = \begin{pmatrix} (\beta \cdot v) \\ (2uv) \\ (2v^2 + p) \end{pmatrix}$$

The general representation of the flux difference splitting schemes to compute fluxes over edges is given by Eqn (2.7).

$$F_{i+\frac{1}{2}} = \frac{1}{2} \cdot [F(Q_L) + F(Q_R) - S_m^+ \cdot (Q_R - Q_L)] \tag{2.7}$$

The S_m^+ term is the positive wave speed and the evaluation method of this term determines the constructed flux difference splitting type.

Evaluation of the convective fluxes on each edge of the control volumes using Roe upwinding is given as

$$F_{i+\frac{1}{2}} = \frac{1}{2} \cdot [F(Q_L) + F(Q_R) - \left| \overline{A}_{Roe} \right| \cdot (Q_R - Q_L)] \tag{2.8}$$

The $\overline{A}_{Roe}$ term is named as the Roe averaged flux jacobian matrix. A special averaging of the left and right state variables is used for each different governing equation set in Roe upwinding which is called as the Roe averaging. In the case of incompressible Navier-Stokes equations modified according to the artificial compressibility method, Roe averaging is equal to the simple arithmetic averaging.

The definition of the flux jacobian matrix is given below.

$$\overline{A} = \frac{\partial F}{\partial Q} = \begin{pmatrix} 0 & \beta\vec{i} & \beta\vec{j} \\ \vec{i} & \vec{V} + u\vec{i} & u\vec{j} \\ \vec{j} & v\vec{i} & \vec{V} + v\vec{j} \end{pmatrix}$$

The eigenvalues of the flux jacobian matrix are $\lambda_1 = \vec{V}$, $\lambda_2 = (\vec{V} + c)$ and $\lambda_3 = (\vec{V} - c)$ where "c" corresponds to the artificial speed of sound and equal to $\sqrt{\vec{V}^2 + \beta}$.

The formulae used for evaluation of the term $|\overline{A}_{Roe}|$ on each edge is given as

$$|\overline{A}| = T|\Lambda|T^{-1} \tag{2.9}$$

In Equation (2.9), the terms T and T^{-1} are corresponding to the right eigenvectors and inverse of the right eigenvectors of the flux jacobian. On the other hand, the $|\Lambda|$ term is a diagonal matrix whose terms are the absolute values of the eigenvalues. These matrices are given as

$$|\Lambda| = \begin{pmatrix} |\lambda_1| & 0 & 0 \\ 0 & |\lambda_2| & \\ 0 & 0 & |\lambda_3| \end{pmatrix}$$

$$T = \begin{pmatrix} 0 & -c[\vec{V}-c] & c[\vec{V}+c] \\ -\vec{j} & (c\vec{i}-\phi\vec{j}) & -(c\vec{i}+\phi\vec{j}) \\ \vec{i} & (c\vec{j}+\phi\vec{i}) & -(c\vec{j}-\phi\vec{i}) \end{pmatrix} \quad T^{-1} = \begin{pmatrix} -\frac{\phi}{c^2} & -\frac{\phi\vec{V}\vec{i}+c^2\vec{j}}{c^2} & -\frac{\phi\vec{V}\vec{j}-c^2\vec{i}}{c^2} \\ \frac{1}{2c^2} & \frac{(\vec{V}+c)\vec{i}}{2c^2} & \frac{(\vec{V}+c)\vec{j}}{2c^2} \\ \frac{1}{2c^2} & \frac{(\vec{V}-c)\vec{i}}{2c^2} & \frac{(\vec{V}-c)\vec{j}}{2c^2} \end{pmatrix}$$

where,

$$\vec{V} = u \cdot \vec{i} + v \cdot \vec{j}$$

$$\phi = -u \cdot \vec{j} + v \cdot \vec{i}$$

2.3.1.2 High Order Flow Variable Reconstruction

Flow variable reconstruction is the evaluation of the Q_L and Q_R terms occurring in the discretisation of the convective fluxes. Properties of the reconstruction are the main source of the spatial order of accuracy of the generated scheme for finite volume methods.

The simplest variable reconstruction method is to take the left and right state variables Q_L and Q_R as equal to the cell center values of the corresponding sides. This reconstruction generates a first order scheme for upwinded convective terms and a second order scheme for the central discretisation of the viscous terms. Second order discretisation of the viscous terms is the most popular way in the literature, but the first order discretisation of the convective flux terms

causes strong numerical diffusion errors.

To reduce the diffusion error, high order variable reconstruction schemes are used. In general high order schemes can be classified in two categories as fixed and variable stencil schemes. In incompressible flows discontinuities such as shocks do not occur, so that fixed stencil schemes are used in this study.

In general, structured grid solvers use the advantage of the structure in the grid for high order reconstruction. One of the most popular reconstruction scheme used in structured grids is the Monotone Upstream-Centered Scheme for Conservation Laws (MUSCL). MUSCL is a scheme having one parameter to vary the accuracy. It is possible to generate accuracy from second to third order by changing this parameter. The main idea of MUSCL scheme, while obtaining third order accuracy, is using the variable value from the forward side without increasing the length of the stencil. For all of the high order schemes, one of the major problem is the boundary condition implementation and the compactness of the scheme is an important property. Short stencil length is one of the major advantages of MUSCL scheme.

On the other hand, most fixed stencil unstructured solvers are limited to approximately second-order spatial accuracy and achieve this accuracy by piecewise linear reconstruction. In this work, the U-MUSCL scheme is used [37]. This scheme is a modified MUSCL scheme which is also applicable to unstructured grid solvers. A sample edge is given in Fig 2.1

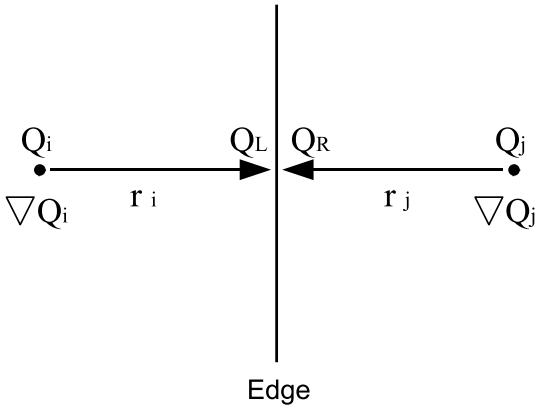

Figure 2.1: Edge Variable Extrapolation

The variable reconstruction formula for U-MUSCL scheme with the variables shown in Fig 2.1 is given below

$$Q_L = Q_i + \frac{\kappa}{2}(Q_j - Q_i) + (1-\kappa) \cdot \nabla Q_i \cdot \vec{r_i} \tag{2.10}$$

The parameter κ, occurring in Eqn (2.10), is the U-MUSCL parameter and different diffusion error levels or accuracies can be obtained for different κ values. Also, it is seen from Eqn (2.10) that for $\kappa = 0$, the U-MUSCL scheme reduces to piecewise linear reconstruction and the order of accuracy is two for this case. The results of variation in κ is numerically analyzed in Chapter 3.

2.3.2 Viscous Fluxes

The viscous term shown in Eqn (2.5) is a second order differential term. Because of that, the velocity gradients over the edges are needed to be calculated in finite volume methods. "$Average\ of\ gradients$" and "$element-based\ gradients$" methods are the most commonly used methods for this purpose [38].

These two different techniques for the evaluation of the gradients over the edges have their

own advantages and disadvantages. The average of gradients method is particularly attractive if the flow gradients are already calculated for the high order variable reconstruction [38]. In this method, simple averaging can be used for the evaluation of the gradients at the midpoints of the edges as given below,

$$\left(\frac{\partial Q}{\partial x}\right)_{i+\frac{1}{2}} = \frac{1}{2}\left[\left(\frac{\partial Q}{\partial x}\right)_{i} + \left(\frac{\partial Q}{\partial x}\right)_{j}\right] \tag{2.11}$$

No additional data storage is needed in this method but a wide stencil with unfavorable weighting occurs [38]. Also in this method, the gradients of the generated ghost cells should be known for boundary condition implementations.

The second way of forming gradients over an edge is defining an auxiliary control volume centered at this edge and evaluating the gradients by using Green-Gauss theorem over the new control volumes. This method is named as the element based gradients method. The major disadvantage of this method is its extending the memory requirements because of the additional information needed about the newly formed control volumes.

In this study, despite the additional memory requirements, the element based gradients method is chosen because of its advantages in generating the coefficients of the left hand side for the viscous terms on implicit time stepping procedure.

The viscous terms given in Eqn (2.5) as $Term3$ can be shown as

$$Term3 = \frac{1}{Re}\begin{pmatrix} 0 \\ \nabla^2 u \\ \nabla^2 v \end{pmatrix}$$

Laplace (∇^2) is a linear operator so that it is possible to represent this term as the multiplication

of the flow variables and some geometric constants. Calculation of these geometric parameters only once at the initialization stage increases the speed of the solver and it is possible to use these constants at the left hand side directly.

An example of the auxiliary control volume generated for gradient evaluation over the edge centers is given in Figure 2.2.

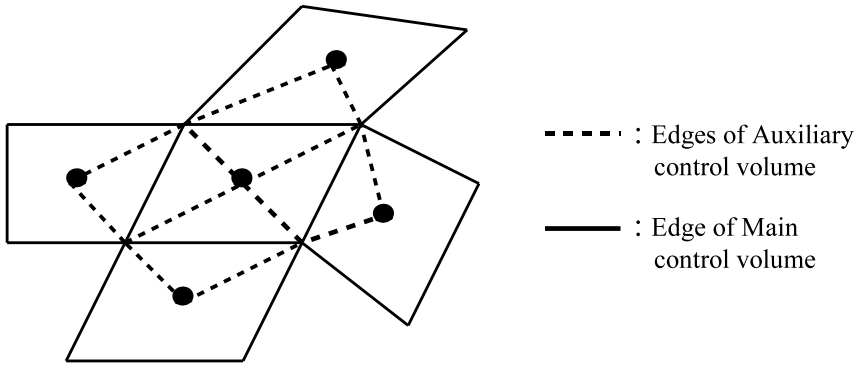

Figure 2.2: Example of Auxiliary Control Volume

The methodology given in Çete [26] is used in this work for the calculation of the geometric parameters. This procedure is explained briefly and Reference [26] is recommended for details and derivations. An edge based systematic approach is followed in Reference [26] for easiness and not to form too long equations.

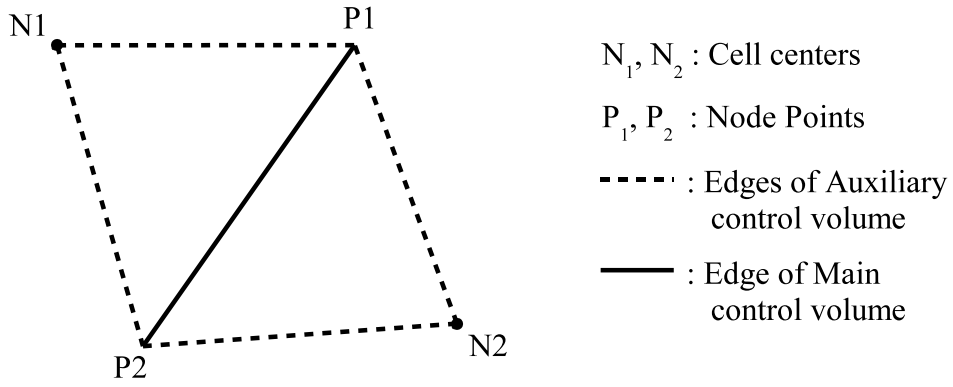

Figure 2.3: A Sample Edge to Demonstrate the Calculation of the Geometric Parameters

The viscous flux term for the sample edge shown on Fig 2.3 can be demonstrated as,

$$I_{N1N2} = L_{N1} \cdot Q_{N1} + L_{N2} \cdot Q_{N2} + L_{P1} \cdot Q_{P1} + L_{P2} \cdot Q_{P2} \tag{2.12}$$

where the geometric constants are,

$$L_{N1} = \frac{(\Delta y_{P1P2} \cdot \Delta y_{P2N1} + \Delta y_{P1P2} \cdot \Delta y_{N1P1} + \Delta x_{P1P2} \cdot \Delta x_{N1P1} + \Delta x_{P1P2} \cdot \Delta x_{P2N1})}{2A_{P1N1P2N2}}$$

$$L_{N2} = \frac{(\Delta y_{P1P2} \cdot \Delta y_{P1N2} + \Delta y_{P1P2} \cdot \Delta y_{N2P2} + \Delta x_{P1P2} \cdot \Delta x_{N2P2} + \Delta x_{P1P2} \cdot \Delta x_{P1N2})}{2A_{P1N1P2N2}}$$

$$L_{P1} = \frac{(\Delta y_{P1P2} \cdot \Delta y_{P1N2} + \Delta y_{P1P2} \cdot \Delta y_{N1P1} + \Delta x_{P1P2} \cdot \Delta x_{N1P1} + \Delta x_{P1P2} \cdot \Delta x_{P1N2})}{2A_{P1N1P2N2}}$$

$$L_{P2} = \frac{(\Delta y_{P1P2} \cdot \Delta y_{P2N1} + \Delta y_{P1P2} \cdot \Delta y_{N2P2} + \Delta x_{P1P2} \cdot \Delta x_{N1P1} + \Delta x_{P2P2} \cdot \Delta x_{P2N1})}{2A_{P1N1P2N2}}$$

2.3.3 Boundary Conditions

The incompressible Navier-Stokes equations with artificial compressibility modification forms a hyperbolic-parabolic system. This method relaxes the infinite speed of propagation to a finite value and because of the finite speed of propagation, non-reflecting boundary conditions are needed during the solution process. In this work, characteristic boundary conditions are used.

Before describing the boundary conditions separately, it is important to describe the implementation of the Dirichlet and Neumann type boundary conditions.

For the convective flux evaluation, ghost cell concept is used. In Dirichlet type boundary conditions, the values are predefined at the boundary edges. The values of the flow variables at the ghost cells are taken such that the average with the interior cell gives the required value. For Neumann type boundaries, the flow variables are taken as the same with the interior cells to obtain zero normal gradient.

In the case of viscous fluxes, in the Neumann boundary conditions, simply the normal gradients are taken to be zero. For the Dirichlet type boundary conditions, the geometric parameters described in viscous fluxes section are changed. The parameters for boundary cells are given below.

$$L_{P1} = \frac{(\Delta y_{P1P2}{}^2 + \Delta y_{P1P2} \cdot \Delta y_{N1P1} + \Delta x_{P1P2}{}^2 + \Delta x_{P1P2} \cdot \Delta x_{N1P1})}{2A_{P1N1P2N2}}$$

$$L_{P2} = \frac{(\Delta y_{P1P2}{}^2 + \Delta y_{P1P2} \cdot \Delta y_{P2N1} + \Delta x_{P1P2}{}^2 + \Delta x_{P1P2} \cdot \Delta x_{P2N1})}{2A_{P1N1P2N2}}$$

$$L_{N1} = \frac{(\Delta y_{P1P2} \cdot \Delta y_{P2N1} + \Delta y_{P1P2} \cdot \Delta y_{N1P1} + \Delta x_{P1P2} \cdot \Delta x_{N1P1} + \Delta x_{P1P2} \cdot \Delta x_{P2N1})}{2A_{P1N1P2N2}}$$

Dirichlet type boundary conditions are forced through the boundary nodes instead of the ghost cells in the viscous fluxes. The values of these boundary nodes are kept constant during the solution process.

2.3.3.1 Wall Boundary Condition

For the solution of the Navier-Stokes equations, no-slip boundary condition is used on the wall boundaries. According to the Dirichlet type boundary condition, boundary node velocity values are kept constant as zero. For the pressure, Neumann type boundary condition is used and the pressure values of the ghost cells are extrapolated from the interior cells. A sample cell with wall boundary condition is shown on Fig 2.4.

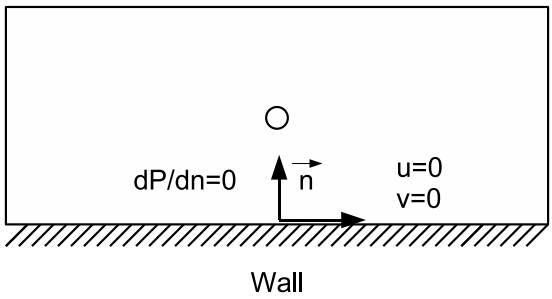

Figure 2.4: Wall Boundary Conditions for Velocities and Pressure

2.3.3.2 Inflow Boundary Condition

Inflow boundary conditions are used for the flows entering into the computation domain. For two dimensional incompressible Navier-Stokes equations, three characteristic waves occur. At inflow boundaries, two of these characteristic waves travel into the domain and one of them travels out of the domain as shown in Fig 2.5. While evaluating the flow variables on the ghost cells, the two velocity components are taken from the free-stream conditions and pressure is extrapolated from the interior cells for inflow boundary conditions.

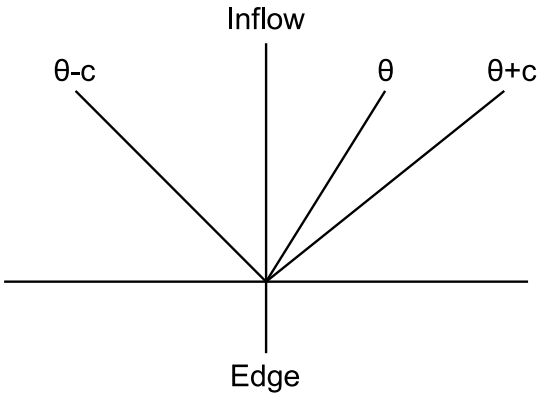

Figure 2.5: Characteristic Waves for Inflow Boundaries

2.3.3.3 Outflow Boundary Condition

Outflow boundary conditions are used for the flows leaving the computation domain. At outflow boundaries, one of these characteristic waves travels into the domain and two of them out of the domain as shown in Fig 2.6. For this case, while evaluating the flow variables on the ghost cells, the pressure is taken from the free stream condition and the two velocity components are extrapolated from the interior cells.

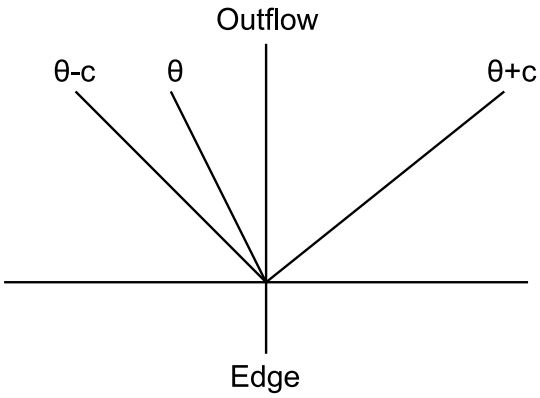

Figure 2.6: Characteristic Waves for Outflow Boundaries

2.3.3.4 Symmetry Boundary Condition

Symmetry boundary conditions are used to decrease the number of computational points by using the already known symmetry planes in CFD calculations. These types of boundary conditions are applied by setting the pressure and parallel velocity component of the ghost cells equal to the interior ones and the normal velocity component as the negative of the interior cells. By these ghost cell values, the velocity vectors' being parallel to the symmetry plane is forced. A sample symmetry plane is shown below.

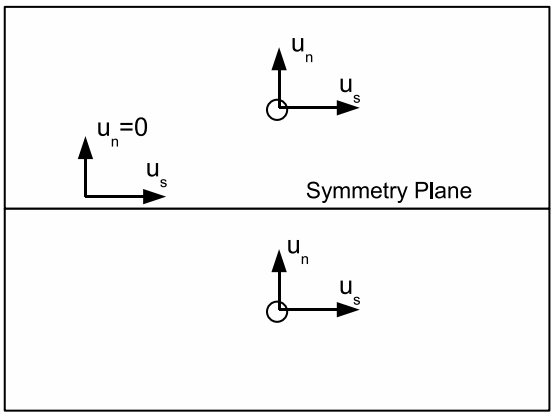

Figure 2.7: Symmetry Boundary Condition

2.4 Temporal Discretisation

The incompressible Navier-Stokes equations form a system of ordinary differential equations after completing the spatial discretisation as described in Section 2.3. The resulting set of equations is given in vectorial form as

$$\frac{d\vec{Q}}{dt} = -\vec{R} \tag{2.13}$$

The schemes used for temporal discretisation can be classified in two main groups consisting of explicit and implicit time stepping. Although the explicit schemes are easy to code and each iteration time is short, these schemes severely suffer from stability and robustness problems. On the other hand, the fully implicit time stepping methods need very high computer resources for complex problems and time efficiency decreases with the increasing number of computational points. Because of these problems, some fast implicit methods are investigated and used extensively in computational fluid dynamics applications. In this study, a relatively new method named as Alternating Cell Directions Implicit method (ACDI) and the well known Point Gauss Seidel method (PGS) are implemented and compared for different test cases.

Both of these methods are shown to be unconditionally stable for the governing equations used in this work. The details of the convergence histories of these schemes for different CFL numbers are given in Chapter 3.

The implicit formulation is obtained by the evaluation of the right hand side of the Eqn (2.13) at the new time level. The resulting system of equations becomes non-linear and a linearization of the right hand side is needed. The linearization method is given as

$$\vec{R}_i^{n+1} \approx \vec{R}_i^n + \left(\frac{\partial \vec{R}}{\partial \vec{Q}}\right)_i \cdot \Delta \vec{Q}^{n+1} \tag{2.14}$$

where,

$$\Delta \vec{Q}^{n+1} = \vec{Q}^{n+1} - \vec{Q}^n$$

The $\left(\frac{\partial \vec{R}}{\partial \vec{Q}}\right)$ term shown in Eqn (2.14) is named as the $flux\ Jacobian$. The equation obtained by substitution of Eqn (2.14) into Eqn (2.13) is given as

$$\left[\left(\frac{\vec{I} \cdot A}{\Delta t}\right) + \left(\frac{\partial \vec{R}}{\partial \vec{Q}}\right)\right]_i \cdot \Delta \vec{Q}^{n+1} = -\vec{R}_i^n \tag{2.15}$$

The system of equations shown in Eqn (2.15) is known as delta form of the governing equations. The main advantage of using delta form is the possibility to use different discretisation orders or even different techniques on the left and right hand sides of the equations. The evaluation of the flux Jacobian on the left hand side is generally made by a simpler discretisation method than used on the right hand side while it still reflects the most important features of the spatial discretisation [38]. Using low order methods on the left hand side reduces the bandwidth of the resulting linear system of equations while high spatial order of accuracy is achieved on the

converged results.

In this work, for the evaluation of the convective flux Jacobian term on the left hand side, first order Roe upwinding is used. Although Roe upwinding is a complex method, using the same upwinding method both on the left and right hand sides improves the convergence character [39].

According to the methodology used in discretisation of the viscous terms, these terms are discretised as geometric parameters multiplied by the flow variables. For this reason, these geometric parameters form the flux Jacobians used on the left hand side directly.

The calculation procedure of the flux Jacobian of the convective fluxes are explained below.

The convective part of the right hand side of Eqn (2.13) is given as

$$\vec{R}_c = \sum_{i=1}^{m} \hat{F}_{i+\frac{1}{2}}/A = \sum_{i=1}^{m} \frac{\Delta S_i}{2} \cdot [F(Q_L) + F(Q_R) - \left|\overline{A}_{Roe}\right|_i \cdot (Q_R - Q_L)]/A \tag{2.16}$$

The term $\left(\frac{\partial \vec{R}}{\partial \vec{Q}}\right)_c \cdot \Delta\vec{Q}^{n+1}$ of Eqn (2.15) is calculated by taking the derivative of the Eqn (2.16) with respect to the flow variables which is evaluated by the chain rule and given as.

$$\begin{aligned}\left(\frac{\partial \vec{R}}{\partial \vec{Q}}\right)_c \cdot \Delta\vec{Q}^{n+1} = \sum_{i=1}^{m} \frac{\Delta S_i}{2} \Big\{ &\left(\overline{A}_c\right)_{L,i} \Delta\vec{Q}^n_{L,i} + \left(\overline{A}_c\right)_{R,i} \Delta\vec{Q}^n_{R,i} \\ &- \frac{\partial}{\partial \vec{Q}_{L,i}} \left[\left|\overline{A}_{Roe}\right|_i \left(\vec{Q}^n_{R,i} - \vec{Q}^n_{L,i}\right)\right] \Delta\vec{Q}^n_{L,i} \\ &- \frac{\partial}{\partial \vec{Q}_{R,i}} \left[\left|\overline{A}_{Roe}\right|_i \left(\vec{Q}^n_{R,i} - \vec{Q}^n_{L,i}\right)\right] \Delta\vec{Q}^n_{R,i} \Big\}/A\end{aligned} \tag{2.17}$$

where $\left(\overline{A}_c\right)$ is the convective flux Jacobian. The evaluation method for the left and right state convective flux Jacobians are given below.

$$\left(\overline{A}_c\right)_{L,i} = \frac{\partial \vec{R}_{L,i}}{\partial \vec{Q}} \quad , \quad \left(\overline{A}_c\right)_{R,i} = \frac{\partial \vec{R}_{R,i}}{\partial \vec{Q}}$$

Equation (2.17) can be simplified by assuming locally constant wave speeds [40]. By assuming the $\left|\overline{A}_{Roe}\right|_i$ term's being constant, $-\frac{\partial}{\partial \vec{Q}_{L,i}}\left[\left|\overline{A}_{Roe}\right|_i \left(\vec{Q}^n_{R,i} - \vec{Q}^n_{L,i}\right)\right]$ term becomes $\left|\overline{A}_{Roe}\right|_i$ and $-\frac{\partial}{\partial \vec{Q}_{R,i}}\left[\left|\overline{A}_{Roe}\right|_i \left(\vec{Q}^n_{R,i} - \vec{Q}^n_{L,i}\right)\right]$ term becomes $-\left|\overline{A}_{Roe}\right|_i$. The resulting equation becomes,

$$\left(\frac{\partial \vec{R}}{\partial \vec{Q}}\right)_c \cdot \Delta \vec{Q}^{n+1} \approx \sum_{i=1}^{m} \frac{\Delta S_i}{2} \left\{ \left(\overline{A}_c\right)_{L,i} \Delta \vec{Q}^n_{L,i} + \left(\overline{A}_c\right)_{R,i} \Delta \vec{Q}^n_{R,i} - \left|\overline{A}_{Roe}\right|_i \left(\Delta \vec{Q}^n_{R,i} - \Delta \vec{Q}^n_{L,i}\right) \right\} / A \tag{2.18}$$

which can be written in a more compact form as,

$$\left(\frac{\partial \vec{R}}{\partial \vec{Q}}\right)_c \cdot \Delta \vec{Q}^{n+1} \approx \sum_{i=1}^{m} \frac{\Delta S_i}{2} \left\{ \left(\left(\overline{A}_c\right)_{L,i} + \left|\overline{A}_{Roe}\right|_i \right) \Delta \vec{Q}^n_{L,i} + \left(\left(\overline{A}_c\right)_{R,i} - \left|\overline{A}_{Roe}\right|_i \right) \Delta \vec{Q}^n_{R,i} \right. \tag{2.19}$$

It is seen from Eqn (2.19) that $\left(\frac{\partial \vec{R}}{\partial \vec{Q}}\right)$ term is a 3x3 matrix and the resulting left hand side becomes

$$\left[\left(\frac{\vec{I}\cdot A}{\Delta t}\right) + \left(\frac{\partial \vec{R}}{\partial \vec{Q}}\right)\right]_i \Delta \vec{Q} = \begin{pmatrix} \frac{A}{\Delta t} + \frac{\partial (R_1)_i}{\partial (Q_1)_i} & \frac{\partial (R_1)_i}{\partial (Q_2)_i} & \frac{\partial (R_1)_i}{\partial (Q_3)_i} \\ \frac{\partial (R_2)_i}{\partial (Q_1)_i} & \frac{A}{\Delta t} + \frac{\partial (R_2)_i}{\partial (Q_2)_i} & \frac{\partial (R_2)_i}{\partial (Q_3)_i} \\ \frac{\partial (R_3)_i}{\partial (Q_1)_i} & \frac{\partial (R_3)_i}{\partial (Q_2)_i} & \frac{A}{\Delta t} + \frac{\partial (R_3)_i}{\partial (Q_3)_i} \end{pmatrix} \Delta \vec{Q}_i + \sum_{j=1}^{N_F} \begin{pmatrix} \frac{\partial (R_1)_i}{\partial (Q_1)_j} & \frac{\partial (R_1)_i}{\partial (Q_2)_j} & \frac{\partial (R_1)_i}{\partial (Q_3)_j} \\ \frac{\partial (R_2)_i}{\partial (Q_1)_j} & \frac{\partial (R_2)_i}{\partial (Q_2)_j} & \frac{\partial (R_2)_i}{\partial (Q_3)_j} \\ \frac{\partial (R_3)_i}{\partial (Q_1)_j} & \frac{\partial (R_3)_i}{\partial (Q_2)_j} & \frac{\partial (R_3)_i}{\partial (Q_3)_j} \end{pmatrix} \Delta \vec{Q}_j \tag{2.20}$$

Where the indices "i" and "j" corresponds to the cell itself and the neighboring cells respectively.

The off-diagonal terms of this matrix provides the coupling between the continuity and momentum equations. While this coupling reduces the necessary convergence iteration number, each element of the resulting matrix becomes a 3x3 matrix leading to the need of a block matrix solution. For this set of governing equations, the increase in solution of each iteration step offset the gain in total iteration number as shown in Chapter 3. It is possible to solve the segregated equation set by using the LHS given below instead of the coupled equations.

$$\left[\left(\frac{\vec{I}\cdot A}{\Delta t}\right)+\left(\frac{\partial \vec{R}}{\partial \vec{Q}}\right)\right]_i \Delta\vec{Q} = \begin{pmatrix} \frac{A}{\Delta t}+\frac{\partial(R_1)_i}{\partial(Q_1)_i} & 0 & 0 \\ 0 & \frac{A}{\Delta t}+\frac{\partial(R_2)_i}{\partial(Q_2)_i} & 0 \\ 0 & 0 & \frac{A}{\Delta t}+\frac{\partial(R_3)_i}{\partial(Q_3)_i} \end{pmatrix} \Delta\vec{Q}_i + \sum_{j=1}^{N_F} \begin{pmatrix} \frac{\partial(R_1)_i}{\partial(Q_1)_j} & 0 & 0 \\ 0 & \frac{\partial(R_2)_i}{\partial(Q_2)_j} & 0 \\ 0 & 0 & \frac{\partial(R_3)_i}{\partial(Q_3)_j} \end{pmatrix} \Delta\vec{Q}_j \quad (2.21)$$

A comparison of the coupled and segregated solvers for the ACDI method is given in Chapter 3.

For a structured grid, it can be seen from Eqn (2.19) that, a pentadiagonal matrix is obtained consisting of the cell itself and four neighboring cells. Solution of pentadiagonal matrices are not time efficient. For this reason, different techniques are developed and used to obtain diagonal or tri-diagonal matrices in the literature. The implementation details of two of these techniques, namely the Point Gauss Seidel (PGS) and Alternating Cell Directions Implicit (ACDI) methods are given in the following subsections.

The boundary condition implementations are another important property influencing the convergence characteristics. With the help of defining the values of the ghost cells related to the interior cells, the influences of these ghost cells to the left hand sides can also be related to the

interior cells. Not only in the ACDI but also in the PGS method, the boundary conditions are treated implicitly in this way.

2.4.1 Point Gauss Seidel Method (PGS)

Point Gauss Seidel is a widely-used relaxation method which solves a diagonal system by sending all the non-diagonal terms to the right hand side with their most recent values. Low numerical complexity and modest memory requirements are the main advantages of this method. PGS method has been widely applied for flow solutions on structured and unstructured grids [24] [22].

The left hand side of the discretised equations are separated to diagonal and non-diagonal elements and the non-diagonal terms are sent to the right hand side. $[D]^n$ and $[O]^n$ represent the diagonal and non-diagonal terms of the total $\left[\left(\frac{\vec{I}\cdot A}{\Delta t}\right)+\left(\frac{\partial \vec{R}}{\partial \vec{Q}}\right)\right]$ matrix respectively.

$$\left[\left(\frac{\vec{I}\cdot A}{\Delta t}\right)+\left(\frac{\partial \vec{R}}{\partial \vec{Q}}\right)\right]_i = [D]^n \quad , \quad \left(\frac{\partial \vec{R}}{\partial \vec{Q}}\right)_j = [O]^n$$

As it is shown in Eqn (2.19), for 2-D problems with quadrilateral elements, a linear system of equations with five non-zero elements in each row is formed. In PGS, only the coefficient of the cell of computation is remained on the left hand side, which forms the main diagonal. The non-diagonal terms are approximately sent to the right hand side by using the most recent flow variable values in the RHS calculation.

The resulting equation system is given as.

$$[D]^n \cdot \Delta Q^n = -R^{(n+1/n)} \tag{2.22}$$

where $-R^{(n+1/n)}$ represents the evaluation of the right hand side with the most recent values of the corresponding flow variables.

PGS method is unconditionally stable for this equation set as shown in Chapter 3 and given in Reference [22].

2.4.2 Alternating Cell Directions Implicit Method (ACDI)

Alternating Cell Directions Implicit (ACDI) Method is proposed by Çete [26] in 2004. The main idea lying behind this method is the usage of the sequential cell directions as the lines used in line implicit methods.

For cells having even number of edges, it is possible to define directions passing through the mutual edges. Some examples for these cell directions are shown in Fig 2.8.

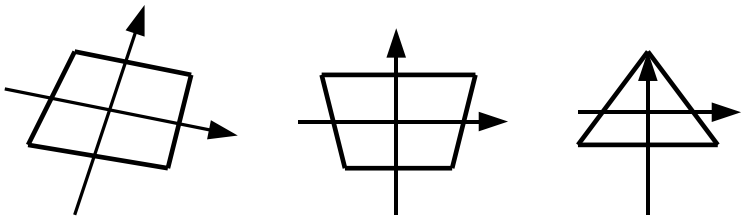

Figure 2.8: Examples for the Cell Directions

It is also seen that, for cells having odd number of edges, an edge is assumed to have zero length and a cell direction is ended on that edge. By combining these cell directions in a grid, it is possible to construct the solution bands passing through the sequential cells.

In Figure 2.9, the dotted lines represent the initial triangular mesh, the solid lines show the quadrilateral mesh and the lines with arrows show the solution bands.

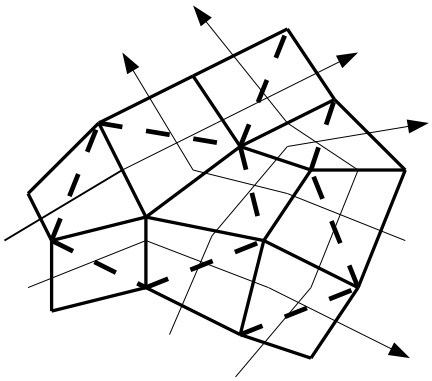

Figure 2.9: An Example of Unstructured Grid with Solution Bands

For single blocked structured grids, the sequential cell directions coincide with the curvilinear coordinates but for unstructured meshes, it is possible to obtain a direction concept by this method. After the generation of solution bands, it is possible to apply line implicit methods with some modifications.

In the ACDI method, a temporal discretisation similar to Line Gauss Seidel is used with a line distribution similar to Alternating Directions Implicit (ADI) method where solution bands pass from each cell twice. The general cell structure is shown in Fig 2.10 for one of the solution directions on a sample cell.

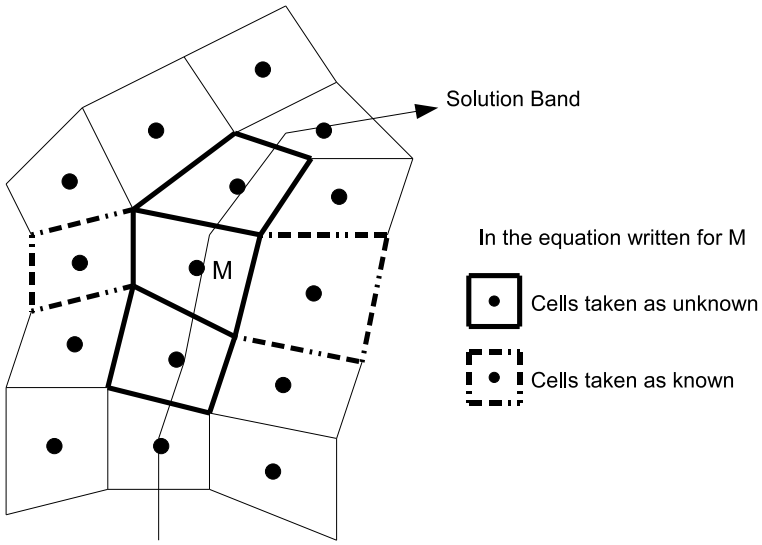

Figure 2.10: General Cell Structure

As it is seen in Figure 2.10, the three cell centers aligned with the solution band are taken as unknowns to form a tridiagonal matrix and solved for an entire solution band at once. The representation of the ACDI scheme is shown below.

$$[A, B, C]^n \cdot \Delta Q^n = \left[-R^{(n+1/n)}\right] \tag{2.23}$$

where $R^{(n+1/n)}$ again represents the most recent values of the corresponding flow variables and $[A, B, C]^n$ corresponds to the three elements of the $[A, B, C, D, E]^n$ where it represents each of the five elements of the flux Jacobian term $\sum_{i=1}^{m} \left[\left(\frac{\vec{I} \cdot A}{\Delta t}\right) + \left(\frac{\partial \vec{R}}{\partial \vec{Q}}\right)\right]_i$

Stability character of the ACDI method is quite similar to the line Gauss Seidel method where the main criteria of the stability is that the sum of all off-diagonal terms are smaller than the diagonal terms. As shown in Chapter 3, ACDI method is also unconditionally stable for this equation set.

2.5 Mesh Generation and Data Management

2.5.1 Mesh Generation

Mesh generation is an important and one of the mostly time consuming processes needed for numerical solutions of the governing equations for geometries with complex boundaries. By transforming a complex computational domain to simpler regions using mesh generation techniques, complications due to the shape of the physical region are removed. Although various mesh cell types are available, usually triangular or quadrilateral elements are chosen for discretisation of the 2-D domains.

There exist basically two types of grids. These grid types are given below with their most common generation methods:

- Structured Grids
  - Algebraic Methods
  - Partial Differential Equations Methods
    - Hyperbolic
    - Elliptic
    - Parabolic
- Unstructured grids
  - Delaunay
  - Advancing Front

Each of these grid generation methods has their own advantages and disadvantages. In addition to these methods, there are also some techniques that combine more than one method together such as, the advancing front and Delaunay triangulation.

In this work, structured and quadrilateral unstructured computational grids are generated and used.

2.5.1.1 Structured Mesh Generation

For structured mesh generation, a hyperbolic grid generator developed by Dinçgez [41] is used. In hyperbolic grid generation, explicit marching in one dimension is done between two surfaces. At each step, a new layer is formed using the already known surface point distribution. This method provides approximate grid orthogonality over the whole domain. Although generating grids by this method is computationally inexpensive, the shape of the outer boundaries can not be controlled exactly due to the nature of the hyperbolic equations. Hyperbolic grid generation is one of the mostly used techniques on external flow calculations where the outer boundaries do not have significant effect on the problems. The details of the hyperbolic grid generator used in this work can be found in Dinçgez [41].

2.5.1.2 Unstructured Mesh Generation

Unstructured quadrilateral mesh generation algorithms can be categorized in two main groups, namely indirect and direct methods. The indirect methods rely on generating quadrilateral meshes by combining or dividing initial triangular elements. On the other hand, the direct methods generate the quadrilateral elements directly with the methods such as advancing front or mapping.

For unstructured mesh generation purpose, both of the indirect and direct methods are used in this study. Merging the triangular meshes to form quadrilateral elements is used as the indirect method and paving algorithm is used as the direct method.

For the indirect method, a public domain code *Delaundo* is used to generate initial triangular grid and they are turned into quadrilateral elements by merging two by two. *Delaundo* is an advancing front Delaunay triangulation tool and details of its algorithm can be found in Mueller [42]. After the merging operation, to remove the remaining triangles, these triangular cells are divided into three quadrilateral elements. This process is followed by dividing all quadrilateral elements into four to remove the hanging nodes. A sample mesh generation

process including these steps is shown in Fig 2.11.

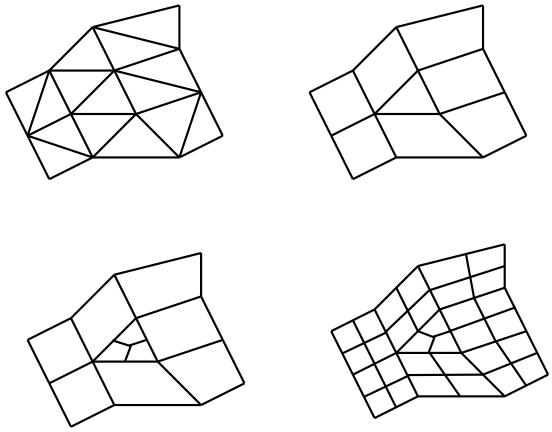

Figure 2.11: Steps for Obtaining Quadrilateral Elements from Triangular Meshes

In an ideal mesh used on finite volume method, the midpoints of the edges should coincide with the centers of the lines connecting the centroids of the corresponding control volumes, which means that the flow variables are extrapolated to the midpoints of the edges. For the situations where the flow variables are extrapolated to different locations, a first order error term is introduced which is proportional to the distance between the ideal and extrapolated locations [37]. An example of highly distorted unstructured mesh is given below.

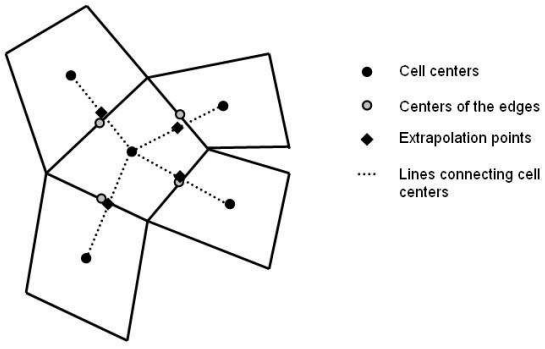

Figure 2.12: A Sample Highly Distorted Mesh

To reduce the error term originated from the cell distortion, Laplacian smoothing is used as the last step of unstructured quadrilateral mesh generation process. The difference between a sample unstructured mesh with and without Laplacian smoothing is given below

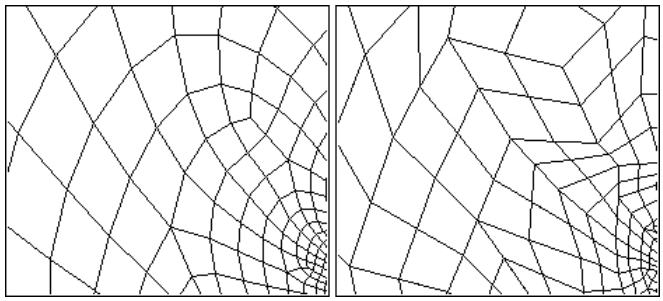

Figure 2.13: A Sample Unstructured Mesh With and Without Laplacian Smoothing

The initial triangular and final quadrilateral mesh over NACA 0008 airfoil profile, which is used in Chapter 3 as a test case, is also shown in Fig 2.14.

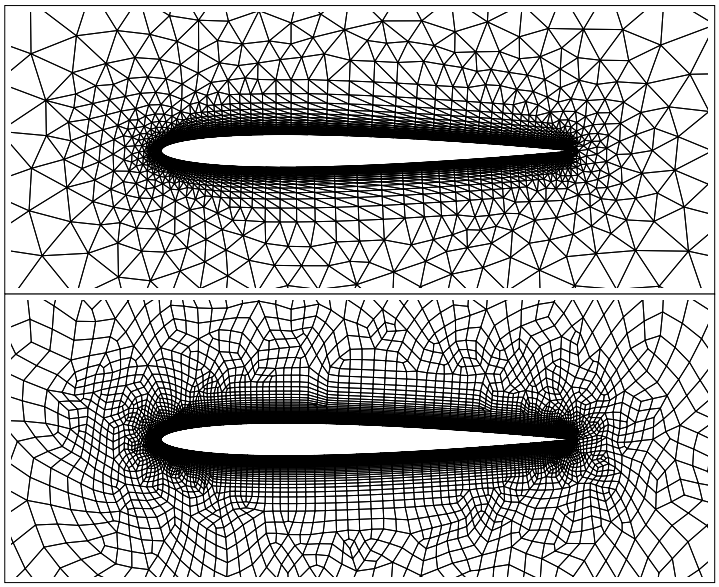

Figure 2.14: Triangular and Quadrilateral Mesh over NACA 0008 Profile

Although good quality unstructured meshes are generated by using this method for single element airfoils, satisfactory results can not be taken for multi element airfoil test case. For NACA 23012b multi-element airfoil profile, the commercial code GAMBIT is used to generate quadrilateral unstructured meshes directly.

GAMBIT is the geometry and mesh generation tool of the CFD software FLUENT. This meshing software is both capable of generating triangular and quadrilateral meshes. For the quadrilateral mesh generation, paving algorithm, which is introduced by Blacker and Stephenson [43], is used. In this method, complete rows of elements starting from the boundaries are formed.

Another important capability of the GAMBIT is defining size functions for faces and edges which increases the user control over the resulting mesh. By using different size functions for main airfoil and flap, meshes with good quality can be generated for multi element airfoils.

The connection between the developed code and GAMBIT is established via the output of GAMBIT in *neutral* file format. In this format, nodal coordinates, element node data and boundary nodes are written in turn.

2.5.2 Data Management

The "*cell based quad edge*" data management system is chosen as the most suitable one for this scheme [26]. In this system, the data are stored in two one dimensional arrays where one of these arrays contains the node information and the other array contains the face information. Face data array also contains the information of its four node points. In this system, it is possible to form or to follow up the solution bands automatically from the data of starting cell and the edge of that band.

It is possible to follow the solution band generation over Fig 2.15 from the data in Table 2.1. In this table, it is seen that from the boundary cell and its starting edge numbers, the sequential cells are found until another boundary is reached.

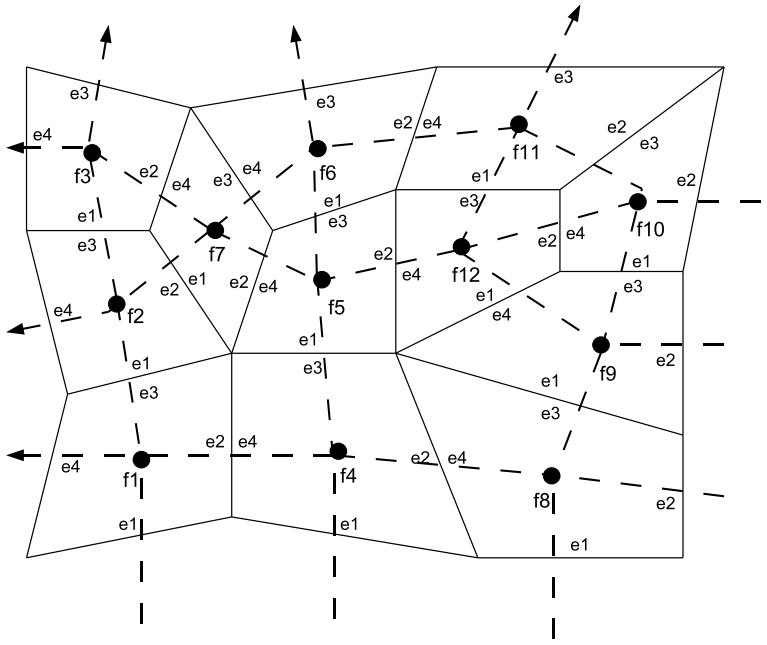

Figure 2.15: Data Management System over a Sample Mesh

Table 2.1: Data Management for Figure 2.15

e1,f1,e3	e1,f2,e3	e1,f3,e3				
e1,f4,e3	e1,f5,e3	e1,f6,e3				
e1,f8,e3	e1,f9,e3	e1,f10,e3	e2,f11,e4	e2,f6,e4	e3,f7,e1	e2,f2,e4
e2,f9,e4	e1,f12,e3	e1,f11,e3				
e2,f10,e4	e2,f12,e4	e2,f5,e4	e2,f7,e4	e2,f3,e4		

For more information about the data management system used in this work, Reference [26] is recommended.

CHAPTER 3

RESULTS & DISCUSSION

A detailed description of the formulation and algorithm used in this work are given in the previous chapters and the algorithm is programmed in a computer code using the FORTRAN 90 programming language in this study. For all CFD codes, including the one developed in this work, validation with proper test cases is an important step to carefully analyze all potential error sources in the algorithm and discretisation processes. For this reason, various test cases are selected in the order of growing complexity. These test cases are the laminar flow over a flat plate, the flow over an airfoil and over a multi element airfoil, respectively. It is possible to find validation data for the selected problems in the literature including experimental, analytical or numerical results. Experimental data are used for validation where available while analytical or numerical results are used on the remaining cases.

Single and multi element airfoil test cases are also used for detailed analysis about the effects of different variable reconstruction and temporal discretisation schemes on the robustness, time efficiency and spatial accuracy of the developed Navier-Stokes solver.

In this study, a MUSCL type variable reconstruction scheme modified to be used in unstructured grids [37] is used. To show the effect of U-MUSCL data reconstruction scheme in the related

test cases, the variation of the spatial accuracy with the U-MUSCL parameter is analyzed for possible gains and the order of accuracy of the optimum case is calculated numerically. The graphical representations of the accuracy variations are also given in single and multi element airfoil test cases.

For the effect of the temporal discretisation schemes, ACDI and PGS schemes are implemented and tested in this work as stated in the previous chapters. Improvements in both the total computation time and the convergence iteration number are expected to be achieved by implementation of the Alternating Cell Directions Implicit method [25], that is similar to the line relaxation methods. The comparisons of these schemes are used for evaluation of the benefits of ACDI method.

In all test cases given in this chapter, the ACDI time integration method with 3rd order U-MUSCL data reconstruction scheme is used with a time step equal to 10^6 and the artificial compressibility parameter β is chosen to be unity unless otherwise is stated. The computing times reported in this work are the CPU time of a personal computer with an AMD Athlon 3000+ processor with 1 GB of RAM.

3.1 Laminar Flow over a Flat Plate

The first test case used for the validation of the developed Navier-Stokes equations solver is the laminar flow over a flat plate with zero pressure gradient. The well known Blasius solution of the boundary layer equations are used as the validation data [44]. A typical 190x70 structured grid with clustering near the leading edge and boundary layer is used for the solution of a semi infinite flat plate. The domain consists of five plates length long and two plates length high. The computational domain with the boundary conditions and the computational grid is given below.

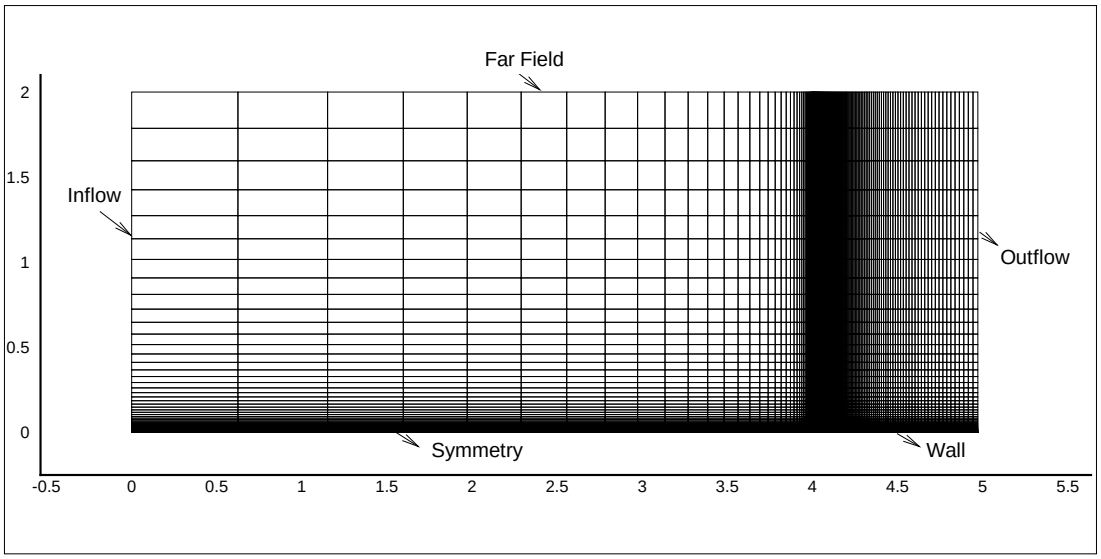

Figure 3.1: Computational Mesh and Boundary Conditions Used for Flat Plate Calculation

A calculation for a plate Reynolds number of 10^4 is used in this validation case. It is a known property that Blasius solution is not valid near the leading edge of the flat plate, so that the location of $\frac{X}{L} = 0.75$ is chosen for the comparison of the velocity profiles.

Figure 3.2 shows a comparison of the computed u velocity profile with the Blasius solution. The u velocity normalized by the free stream velocity is plotted against $\eta = y\sqrt{\frac{U_\infty}{\nu x}}$ in this figure.

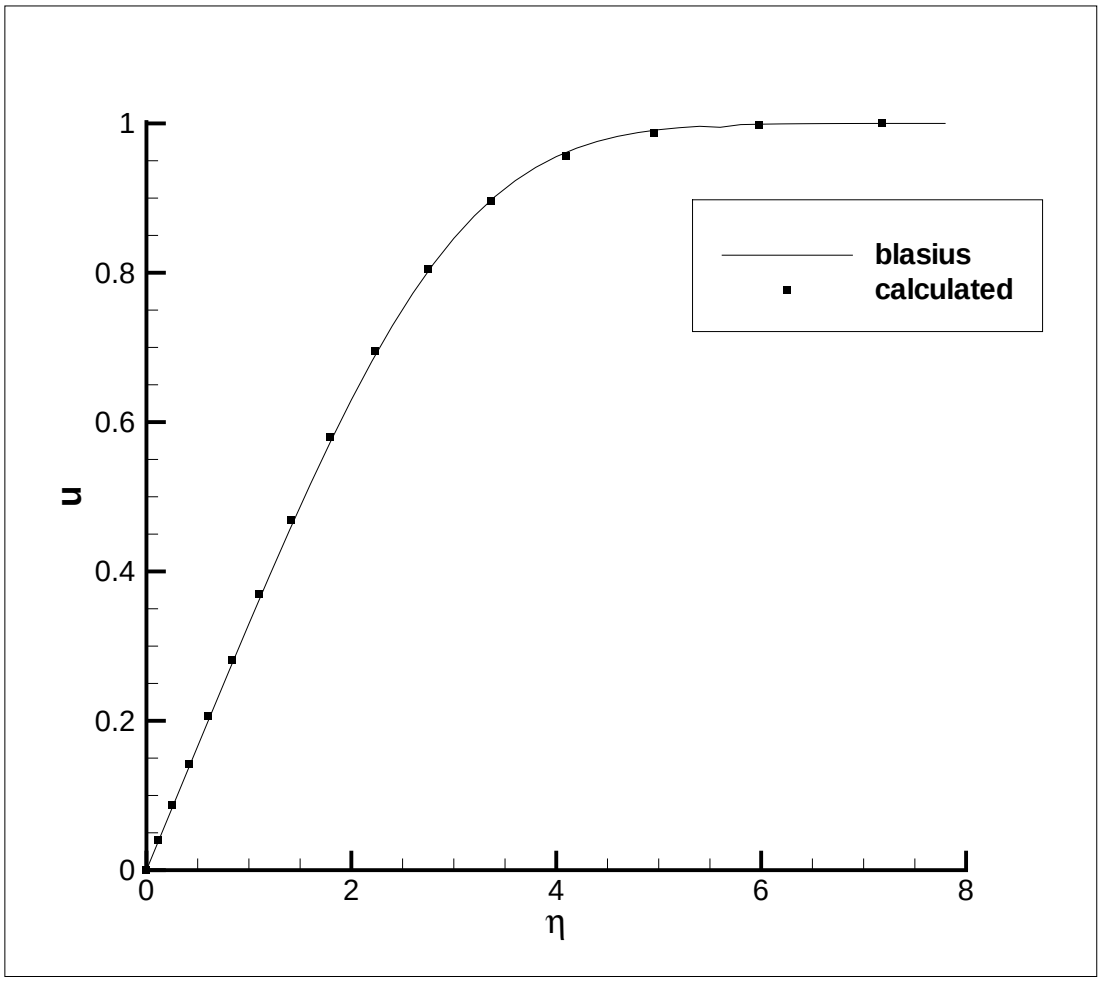

Figure 3.2: Comparison of the Computed Velocity Profile with the Blasius Solution

The variation of local skin friction coefficient with the local Reynolds number is compared with the Blasius solution of the skin friction coefficient calculated by the formula

$$c_f = \frac{0.664}{\sqrt{Re_x}}$$

where $Re_x = \frac{\rho U x}{\mu}$ is the local Reynolds number.

The comparison of the local skin friction variation with the local Reynolds number is given in Fig 3.3 with a logarithmic scale for both of the axes.

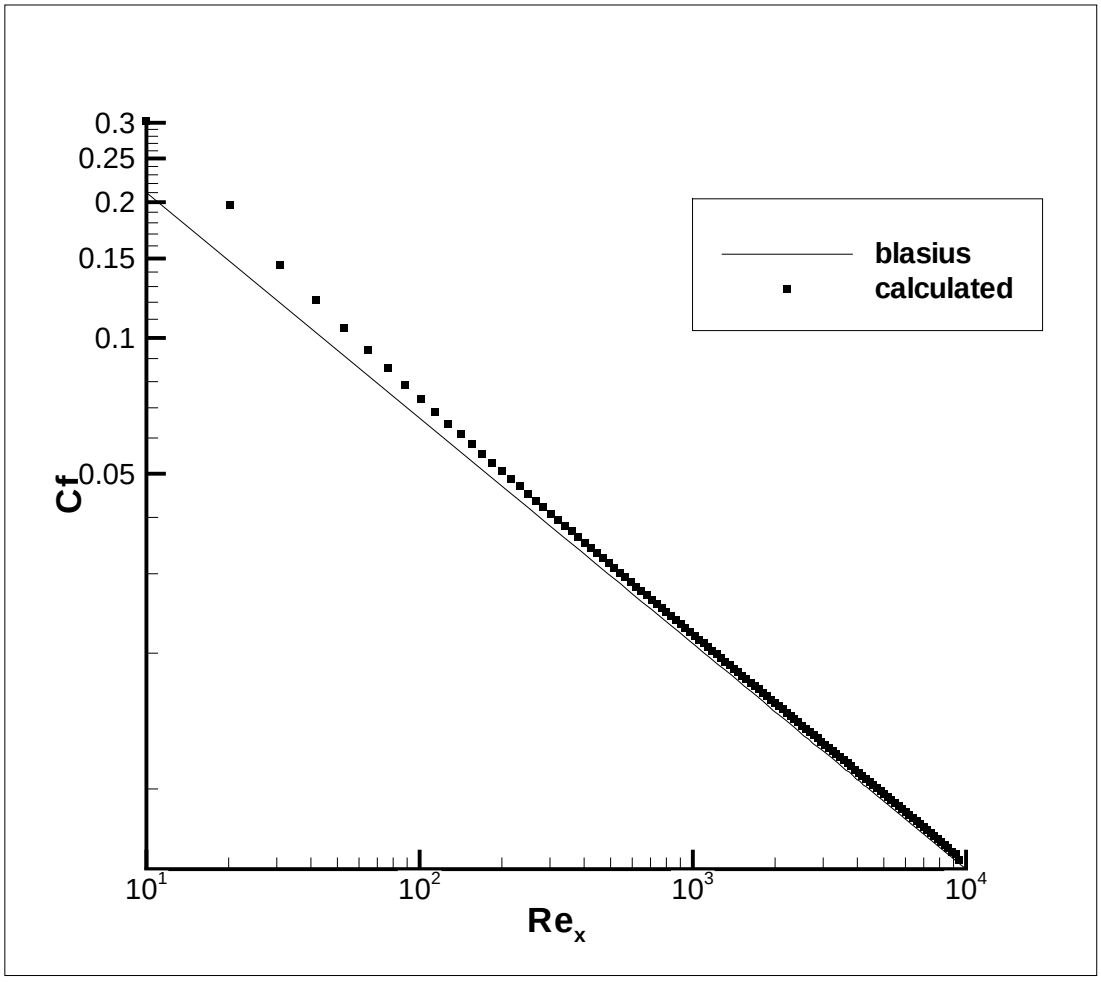

Figure 3.3: Comparison of the Computed Skin Friction Coefficient with the Blasius Solution

Also the theoretical total drag coefficient of the flat plate with a zero pressure gradient is given by $C_D = \frac{1.328}{\sqrt{Re}}$ where Re is the plate Reynolds number. The total drag coefficient computed is 0.013468 for this test case which is 1.4% different from the theoretical value.

A good agreement with the Blasius solution is obtained for both the velocity profile at the location $\frac{X}{L} = 0.75$ and the variation of the local skin friction coefficient in this test case which validates the developed N-S solution methodology.

3.2 Flow over an Airfoil

The second test case used for the validation of the developed Navier-Stokes equations solver is the low Reynolds number flow over the NACA 0008 airfoil profile. Low Reynolds number, low Mach number flows have attracted considerable attention with the growing interest in micro air vehicle (MAV) applications in recent years. Solving flows in this regime with the compressible flow solvers is difficult that, this is one of the mostly used validation test cases for the incompressible flow solvers. The flow solution obtained by the INS2D flow solver is used as the validation data [45]. The flow conditions chosen for this test case is a Reynolds number of 6000 and an angle of attack of 2 degrees.

Both structured and quadrilateral unstructured meshes are generated and used for this test case. C-type structured grids are first generated using the hyperbolic grid generator and used for grid dependency test. This test is conducted in two parts where the grid density along and normal to the wall boundary are varied. The C-part of the computational domain is chosen to have a radius of eight plates length and the wake region length is taken as twelve plates long. Characteristic boundary conditions are used in the outer boundaries and no-slip boundary condition is used on the wall.

The 260x40 C-type structured computational grid and its enlarged view are shown in Fig 3.4.

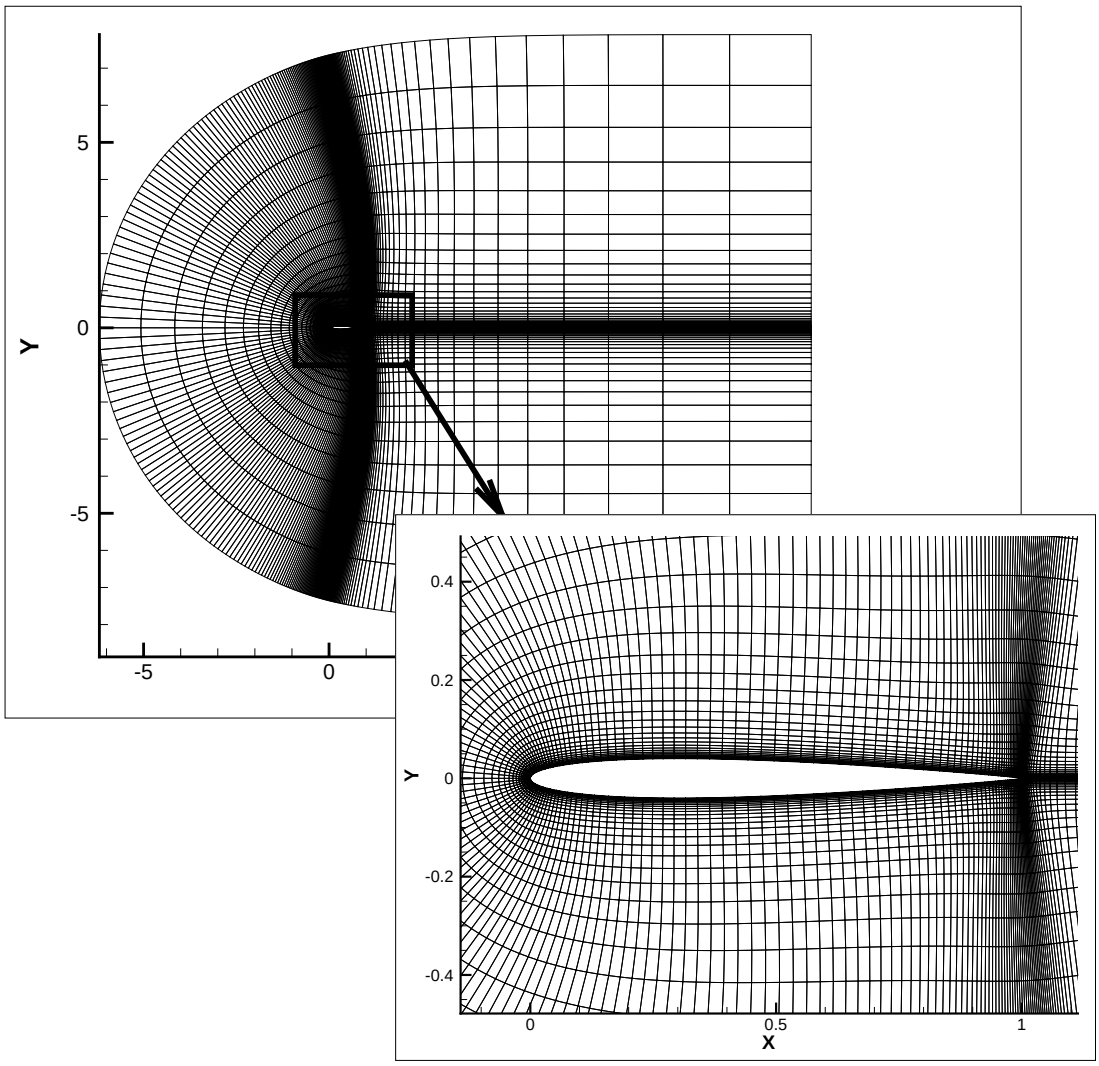

Figure 3.4: 260x40 Structured Grid Around Naca 0008 Airfoil Profile

The generated 260x40 structured grid consists of 258 solution bands where the minimum band length is 39 and the maximum band length is 259 cells.

In order to analyze the effect of grid density normal to the wall boundary, four different grids are generated with 30, 40, 50 and 60 points where the point number on the wall boundary is taken as 260. The variation of pressure coefficient distribution is given in Fig 3.5.

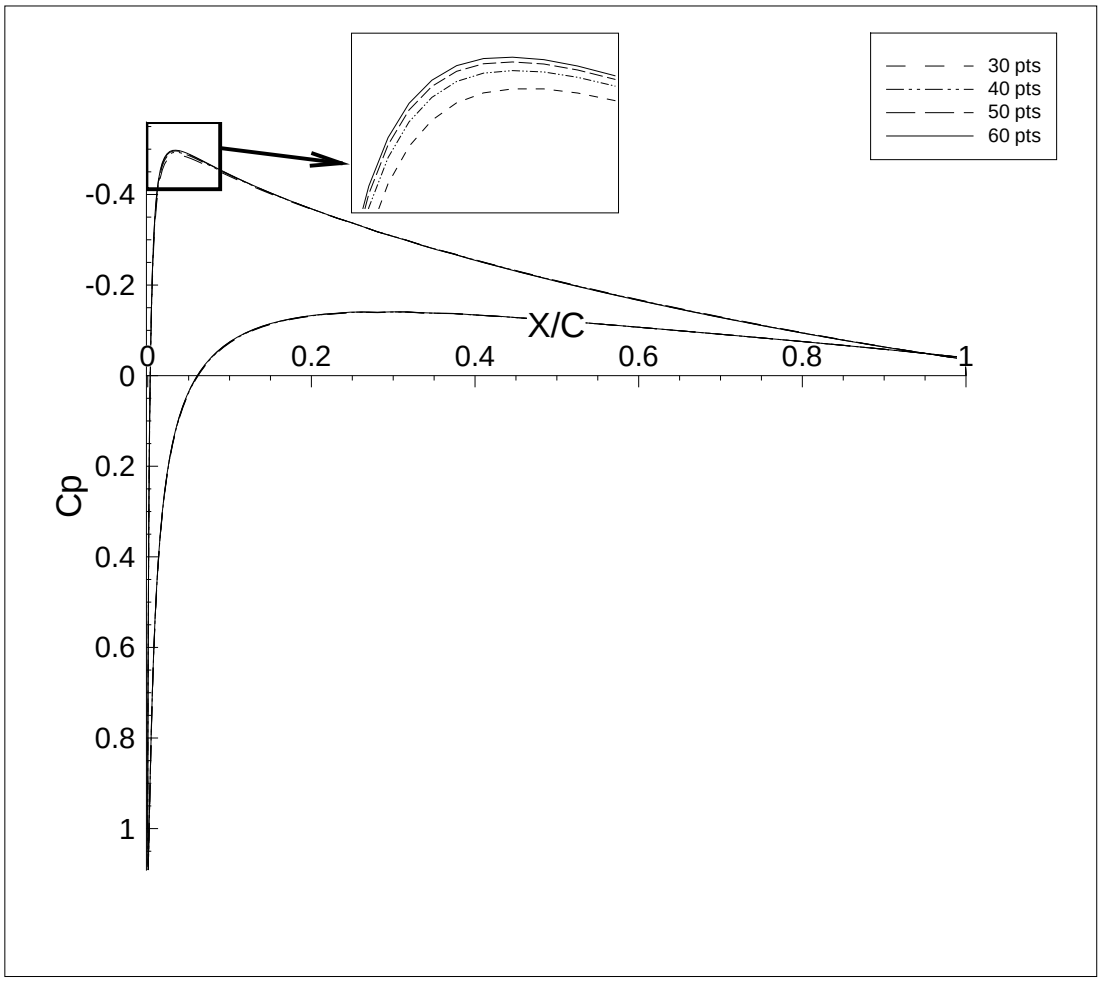

Figure 3.5: Variation of Pressure Coefficient Distribution with Varying Number of Point Normal to the Wall

The number of points located along the airfoil surface is another important parameter for accuracy. It is essential for the solver to converge to a correct solution when the grid density is increased. The pressure coefficient distributions for grids with 100, 140, 180, 220 and 260 points over airfoil profile and 40 points normal to the wall boundary are given in Fig 3.6.

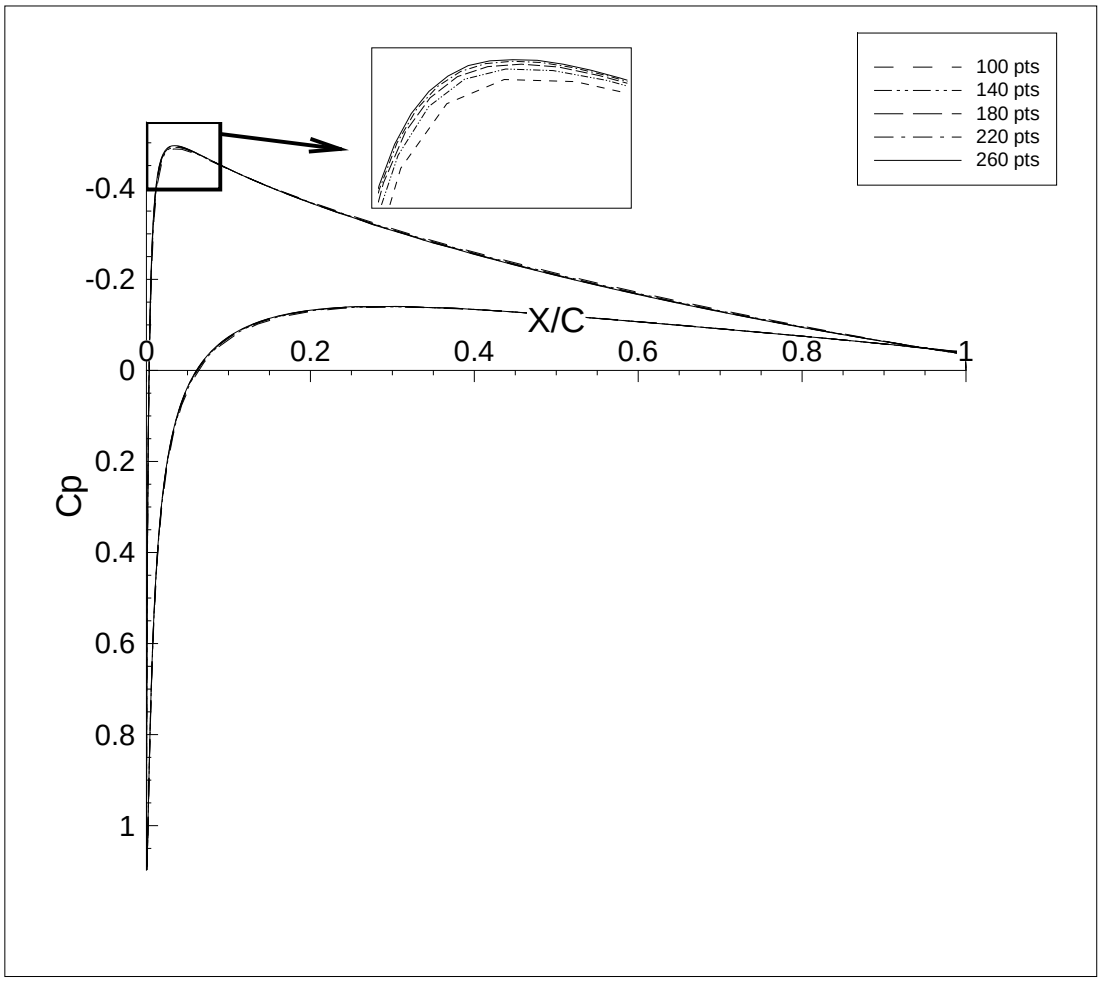

Figure 3.6: Variation of Pressure Coefficient Distribution with Varying Number of Point Along the Airfoil Surface

It is seen from Fig 3.5 and Fig 3.6 that as the number of grid cells increase, the solution converges to a fixed solution which proves that the developed flow solver is mesh independent.

The resulting surface pressure coefficient distribution over the airfoil profile for the 260x40 structured grid is compared with the INS2D result in Fig 3.7.

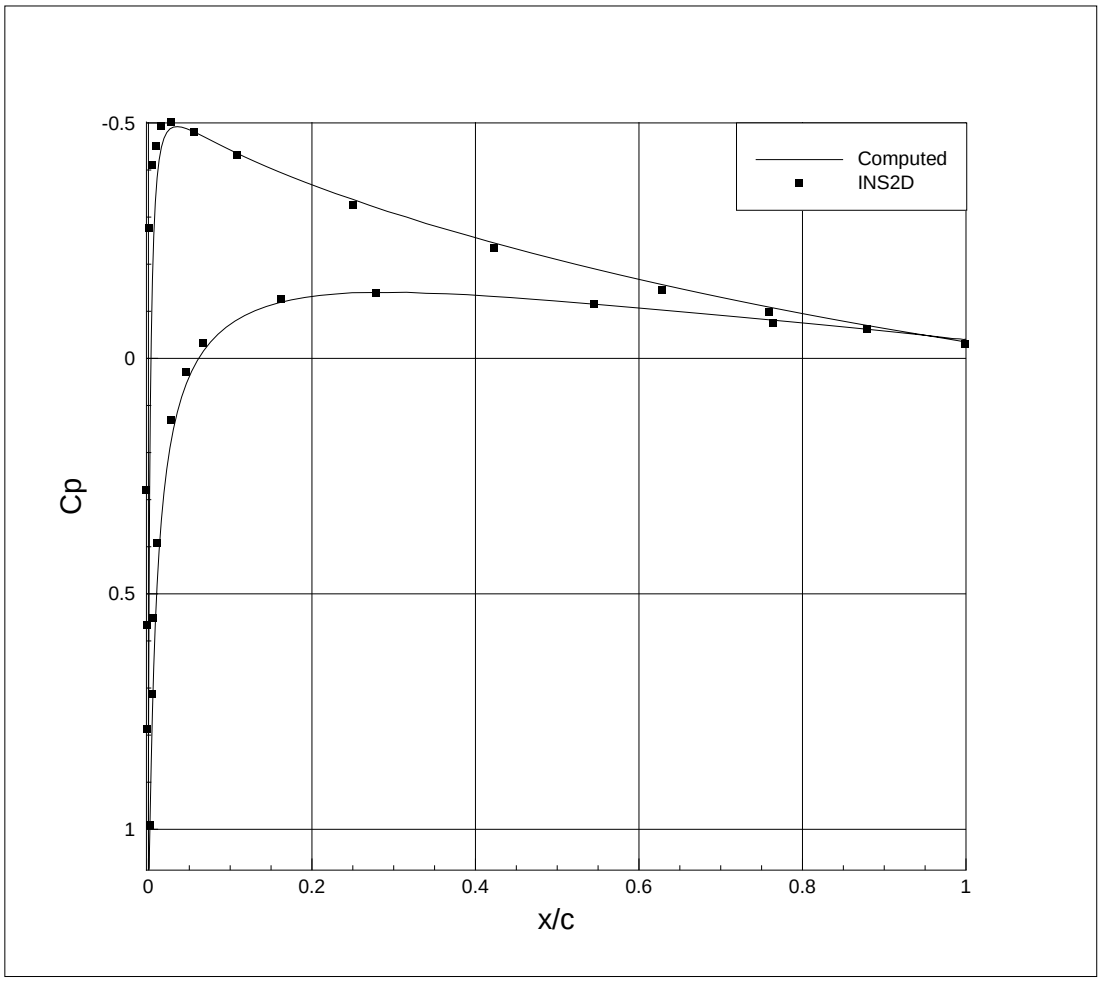

Figure 3.7: Calculated Pressure Coefficient Distribution Using 260x40 Structured Grid

The pressure coefficient distribution over the whole airfoil and enlarged view of the leading edge are shown in Fig 3.8 as flood contours.

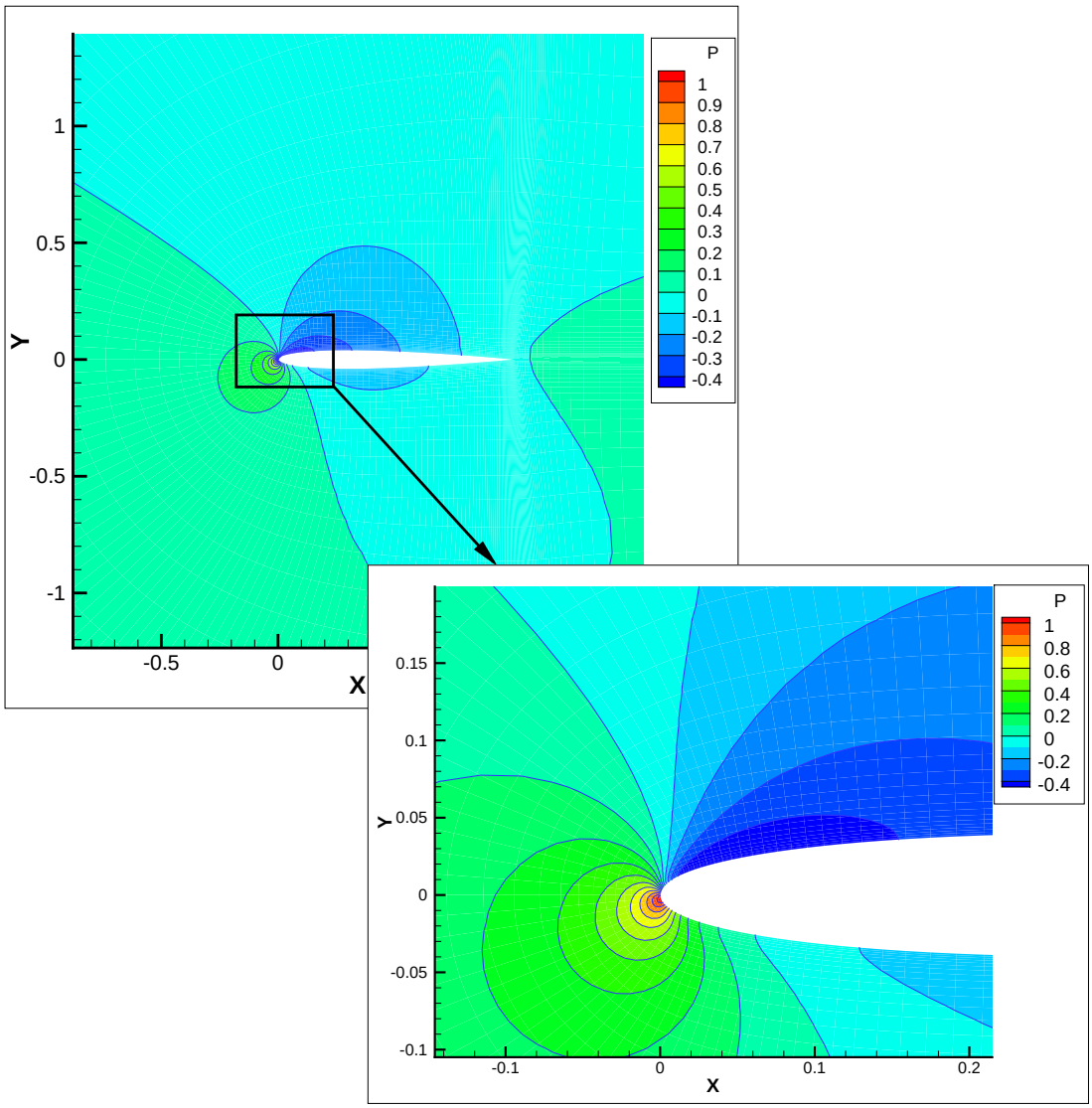

Figure 3.8: Flood Contour of the Pressure Coefficient on 260x40 Structured Grid over NACA 0008 Airfoil Profile

For the solution using unstructured grid, a computational mesh with 11584 points is generated using the indirect methodology explained in Section 2.5.1.2. The radius of the outer boundary of the computational domain is taken as seven plates long. Characteristic boundary conditions are used in the outer boundary and no-slip boundary condition is used on the wall as in the case of solution with the structured grid. The generated unstructured computational mesh and the enlarged view of the leading edge region are shown in Fig 3.9.

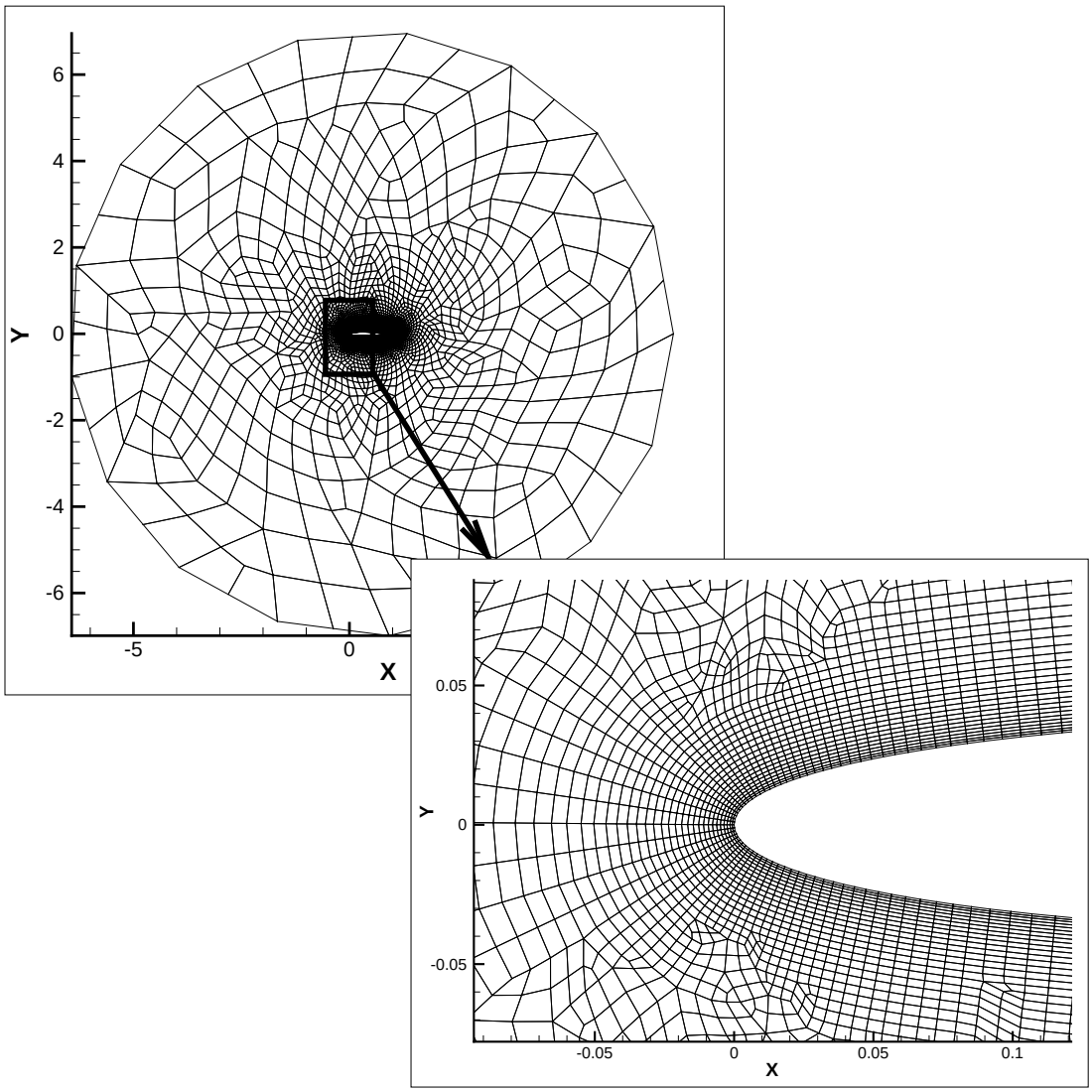

Figure 3.9: Unstructured Mesh Used for Naca 0008 Airfoil Calculation with 11584 Points

For the unstructured quadrilateral meshes, solution bands are generated using the sequential cell directions and these bands of a portion of the generated mesh are shown in Fig 3.10. The generated unstructured grid consists of 125 solution bands where the minimum band length is 13 and the maximum band length is 1793 cells. It is seen from this figure that two solution bands are passing from each cell no matter what the shape of the cell is, as long as it is quadrilateral.

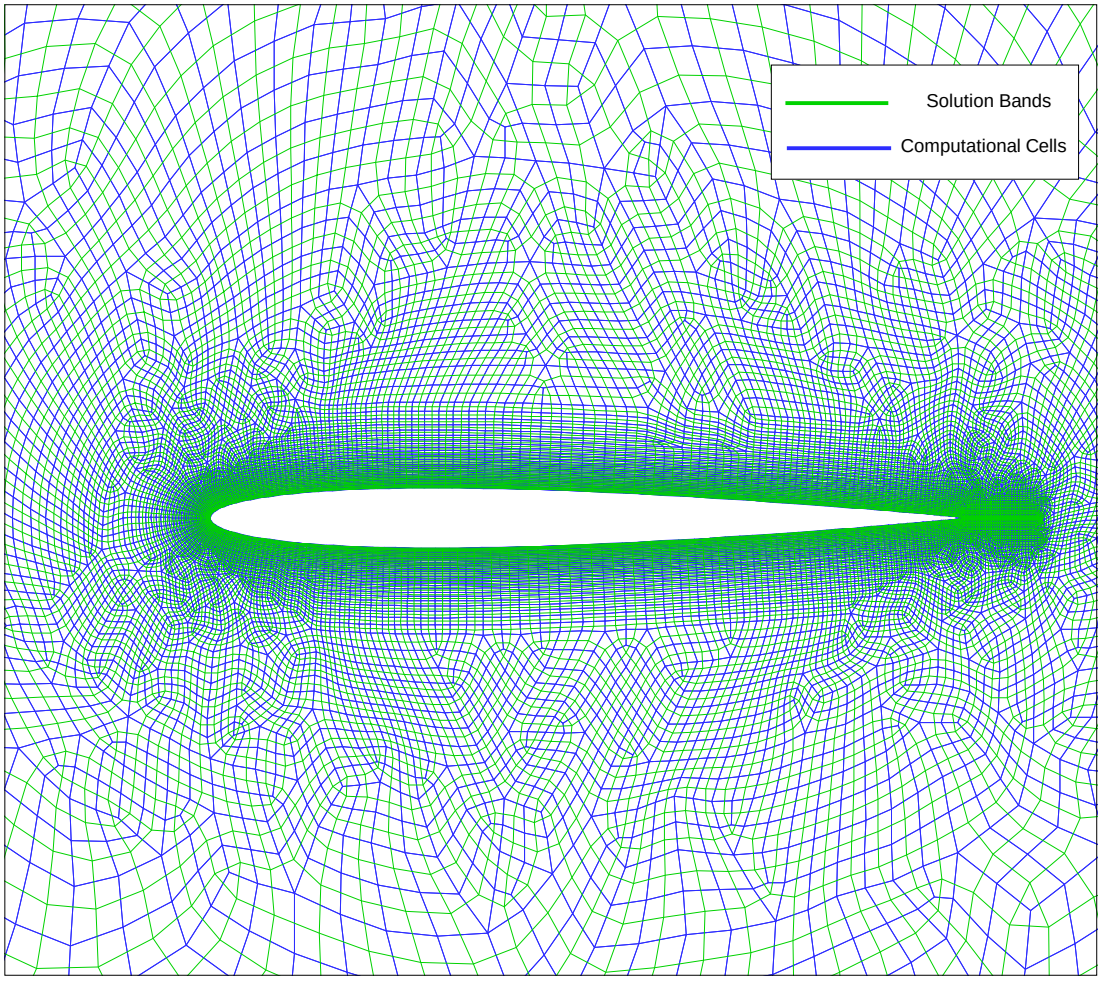

Figure 3.10: Solution Bands of the Unstructured Mesh for Naca 0008 Airfoil Calculation

The resulting surface pressure coefficient distribution over the airfoil profile for the unstructured mesh is given below,

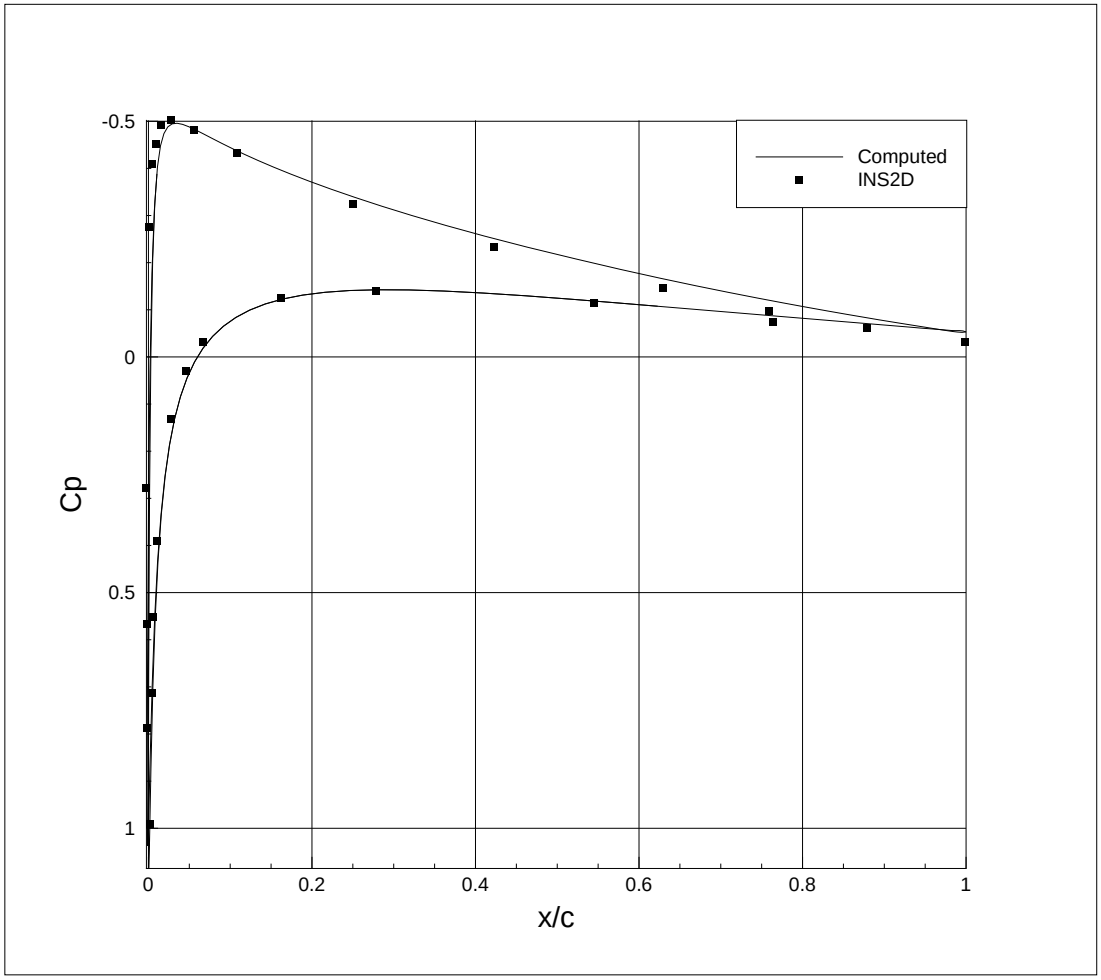

Figure 3.11: Calculated Pressure Coefficient Distribution Using Unstructured Mesh with 11584 Points

The pressure coefficient distribution over the whole airfoil and close-up leading edge is shown in Fig 3.12 as flood contours for the unstructured mesh.

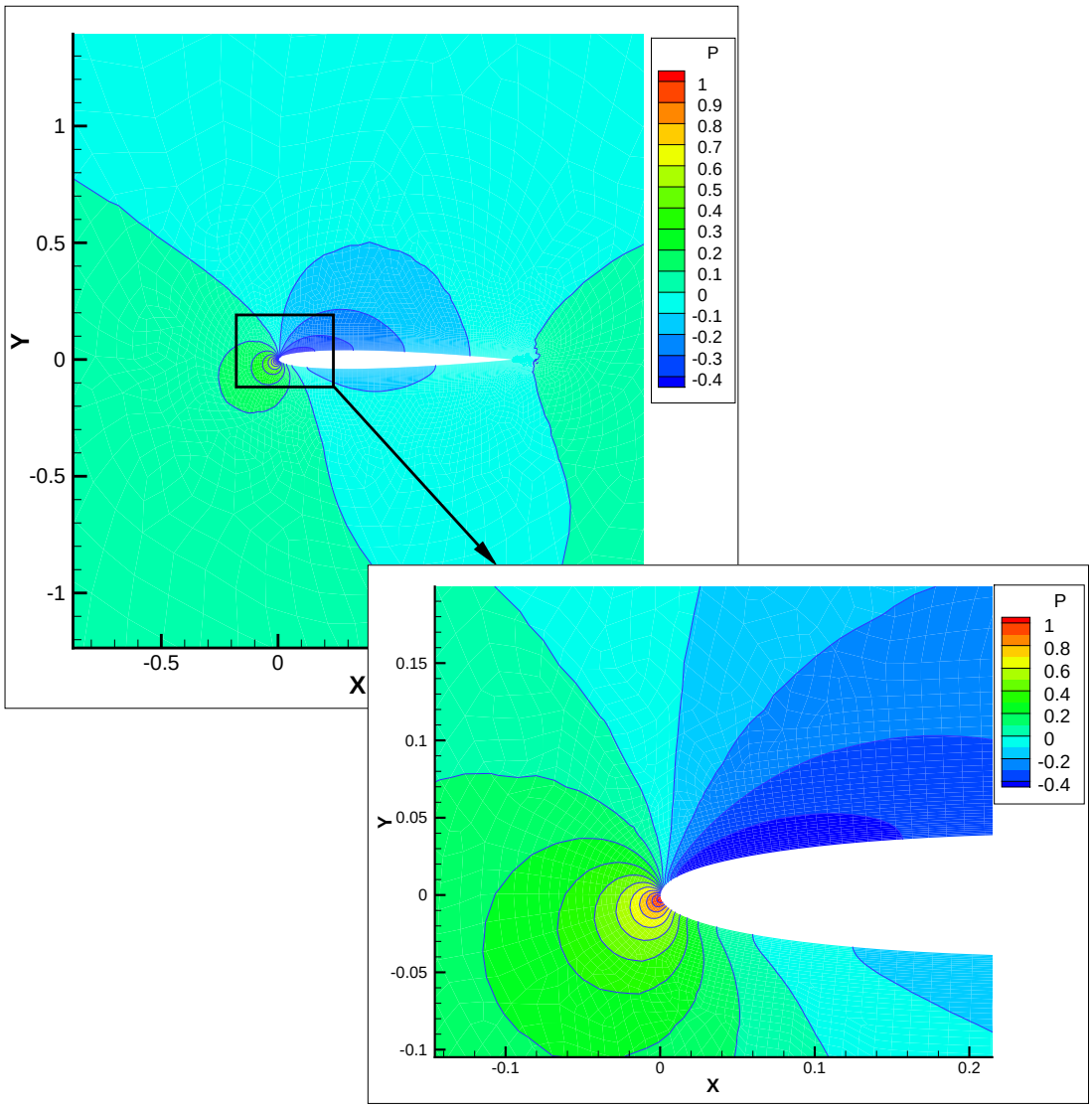

Figure 3.12: Flood Contour of the Pressure Coefficient on Unstructured Mesh over NACA 0008 Airfoil Profile with 11584 Points

It is seen from Fig 3.7 and Fig 3.11 that good agreement with the INS2D results are obtained for the surface pressure coefficient distributions both with the structured and unstructured meshes.

The low Reynolds number flow over NACA0008 airfoil is chosen as the main test case for the comparison study of the spatial and temporal discretisation schemes because it is easy to generate both structured and unstructured meshes over this geometry. The following two subsections include the results for comparison cases of different spatial and temporal discretisation schemes.

3.2.1 Flux Evaluation Comparison

In this work, U-MUSCL [37] scheme is used and an analysis is done about the influence of the U-MUSCL parameter. The same 260x40 structured grid is chosen for discretisation to eliminate the possible errors coming from the meshes such as distortion. The INS2D solution is taken as the reference data and L1 norm error is calculated between the computed results and these data. Variation of error with the χ value is given below,

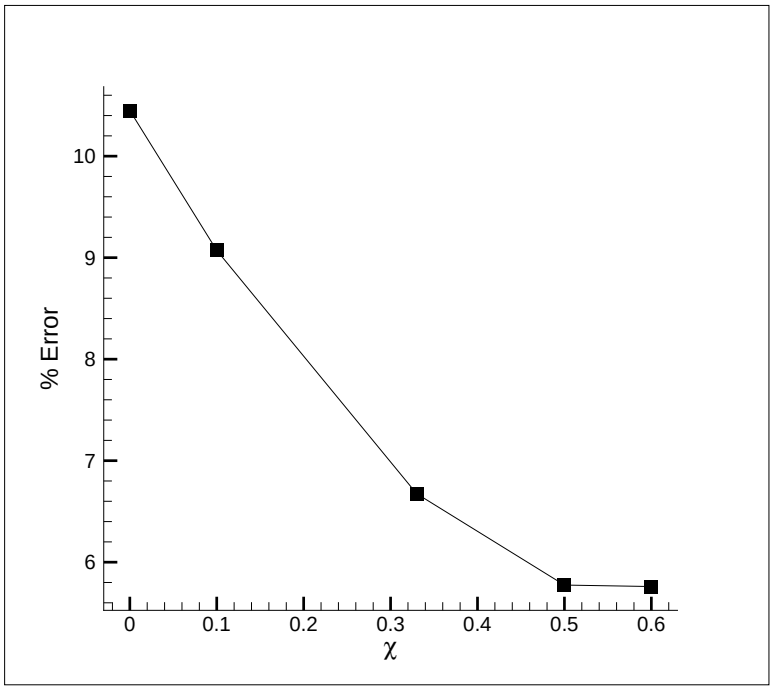

Figure 3.13: Variation of Average Error with χ

Although very similar error values are obtained by 0.5 and 0.6 χ values, convergence characteristics get worse with growing χ values and it is even not possible to achieve a converged solution when χ is taken as 0.7 . Variation of convergence history with varying χ values is shown below,

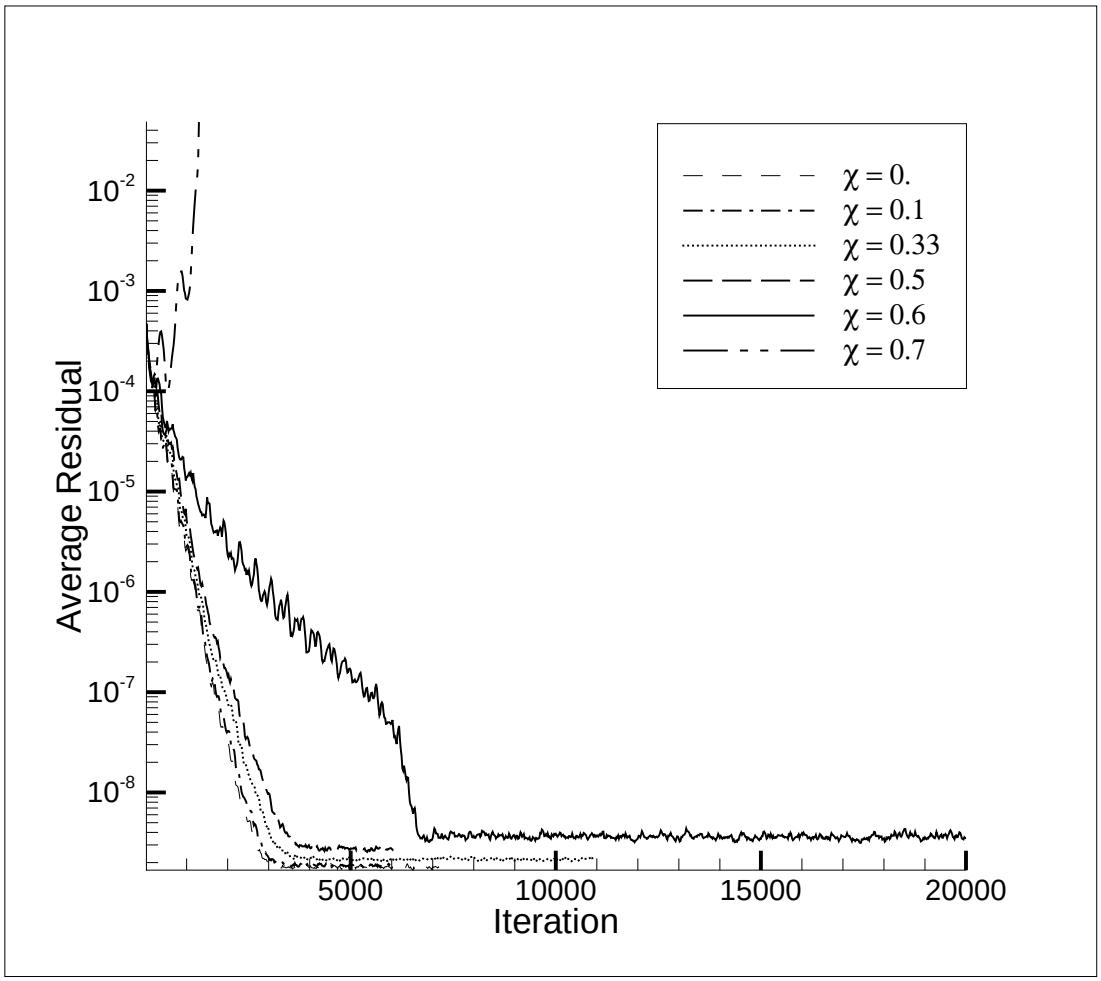

Figure 3.14: Convergence Characteristics for Different χ Values

After it is determined that the optimum value of χ is 0.5, spatial accuracy of that χ value is verified by analyzing the variation of error on 220x30, 260x40 and 300x50 structured grids. The L1 Norm error is plotted against the characteristic lengths of these three grids, their square and cube respectively in Fig 3.15. It is seen from Figure 3.15 that the error is approximately linearly proportional to the cube of the characteristic length which proves that this scheme's order of accuracy is three.

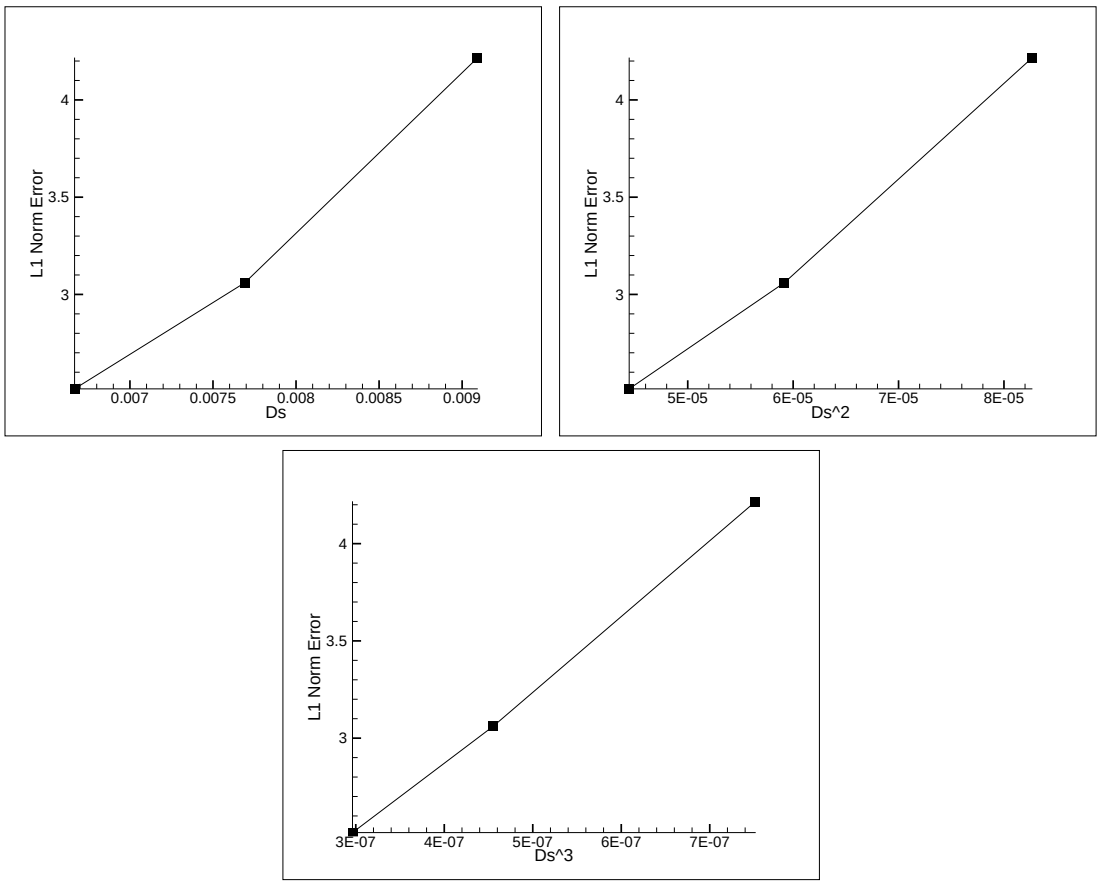

Figure 3.15: Variation of L1 Norm Error with Characteristic Length, its Square and Cube

The accuracy gain for the 260x40 structured grid and the unstructured grid with 11584 point over NACA 0008 airfoil profile are shown graphically on Fig 3.16 and Fig 3.17.

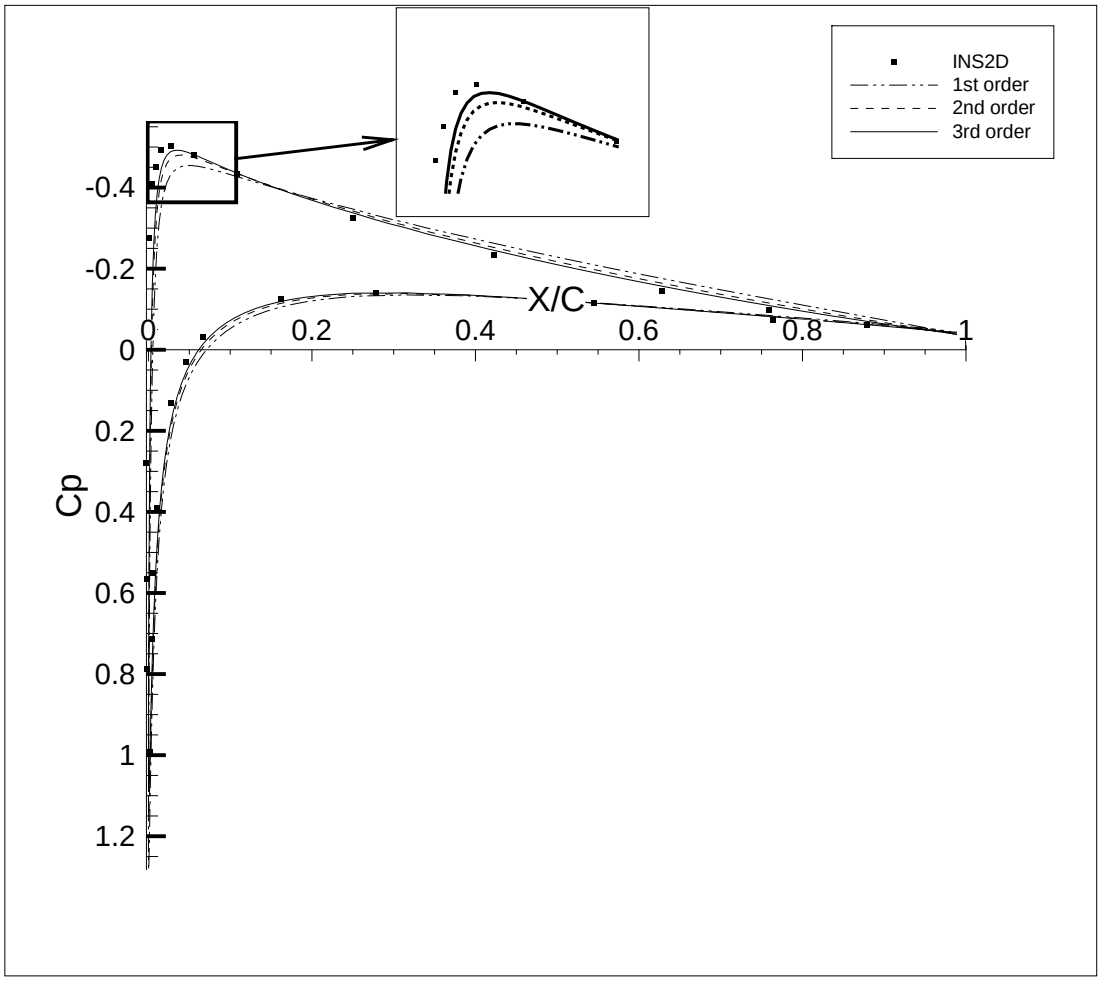

Figure 3.16: Graphical Representation of the Accuracy Gain on 260x40 Structured Mesh over NACA 0008 Airfoil Profile

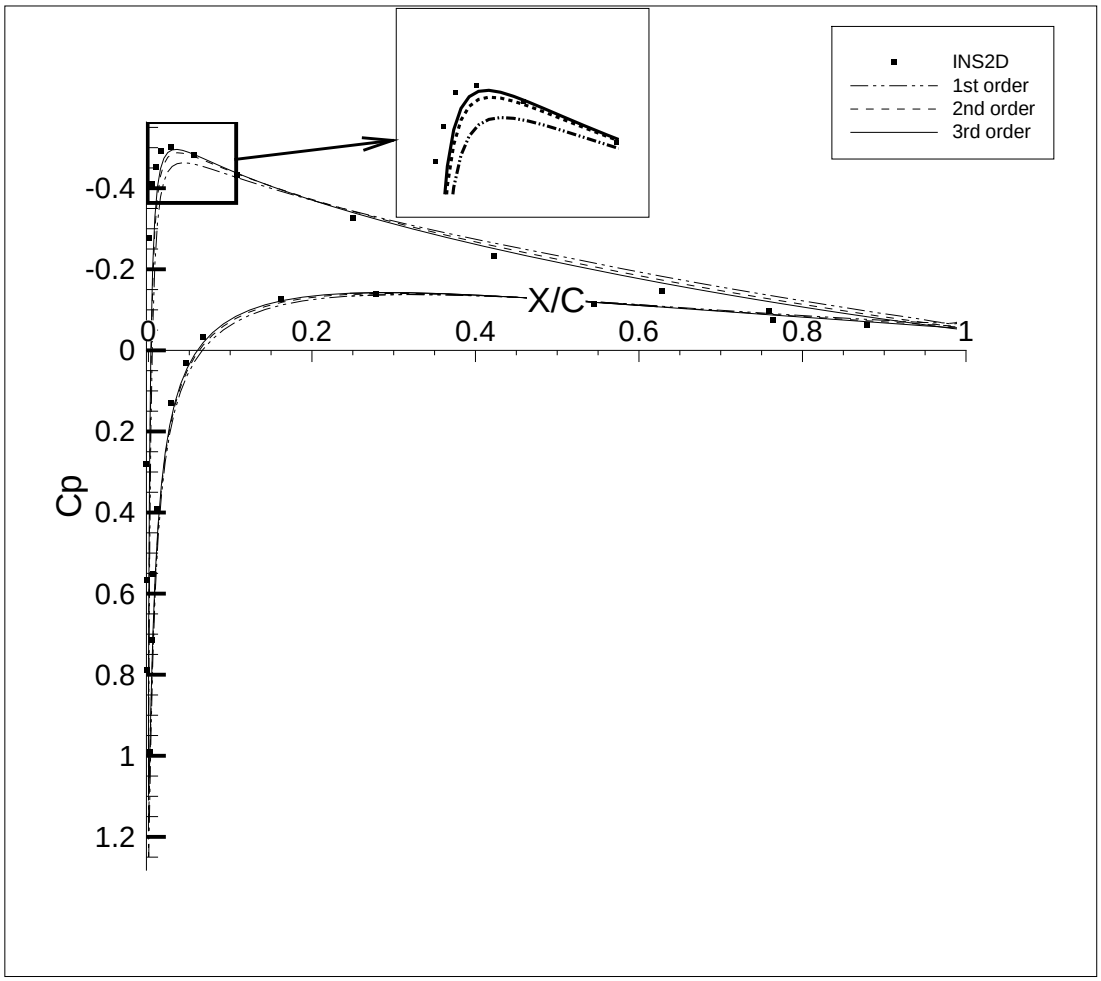

Figure 3.17: Graphical Representation of the Accuracy Gain on Unstructured Mesh over NACA 0008 Airfoil Profile with 11584 Points

It is seen from these figures that a considerable accuracy gain is obtained both in structured and unstructured meshes using the U-MUSCL flow variable reconstruction scheme.

3.2.2 Temporal Discretisation Comparison

Before starting the comparison of the ACDI and PGS methods, CFL number versus convergence iteration number graphs are plotted for both the PGS and ACDI methods to find an optimum CFL number to be used in the computations. Figure 3.18 is given for that purpose and it is seen that as CFL number increases, convergence iteration number reduces until a minimum value is reached. This character is known to be valid for the relaxation methods.

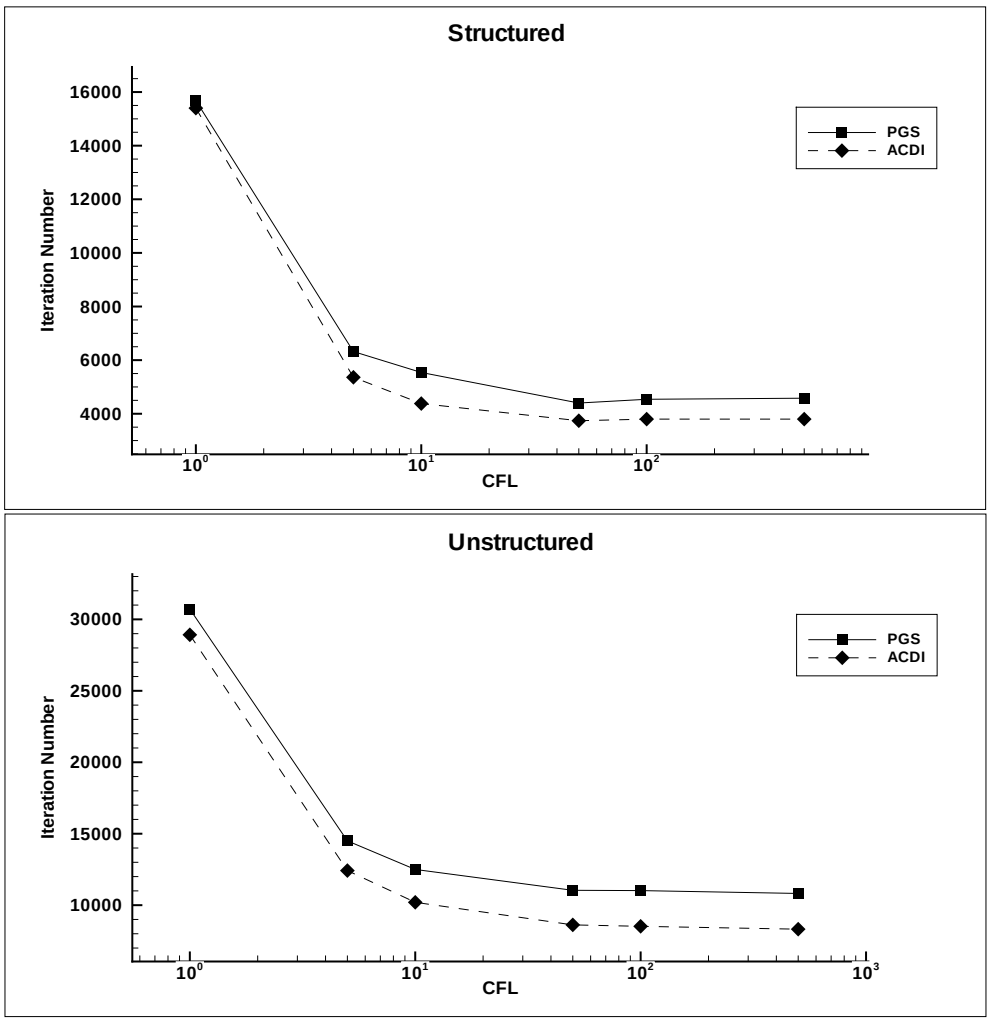

Figure 3.18: Variation of Convergence Iteration Time with CFL Number on 260x40 Structured Grid and Unstructured Grid with 11584 Points

In this figure, the solution is considered to be converged when the average residual decreases below the 10^{-8} value. It is also seen from Fig 3.18 that a sufficiently large CFL number is always optimum. For this reason, CFL number is taken to be 10^6 in all of the calculations given in this chapter.

The coupled and segregated solvers are also compared for this test case. For the segregated solver, the off-diagonal terms of the LHS matrix is neglected and by this way, tri-diagonal matrices are obtained as the final linear system of equations. For the coupled solver, the resulting

system becomes a block tri-diagonal matrix where each element is a 3x3 matrix. For the solution of the resulting linear system of equations, a block tri-diagonal matrix solver is used. The comparison of the average residual versus iteration step and computation times are given in Fig 3.19 and Fig 3.20 for structured and unstructured meshes.

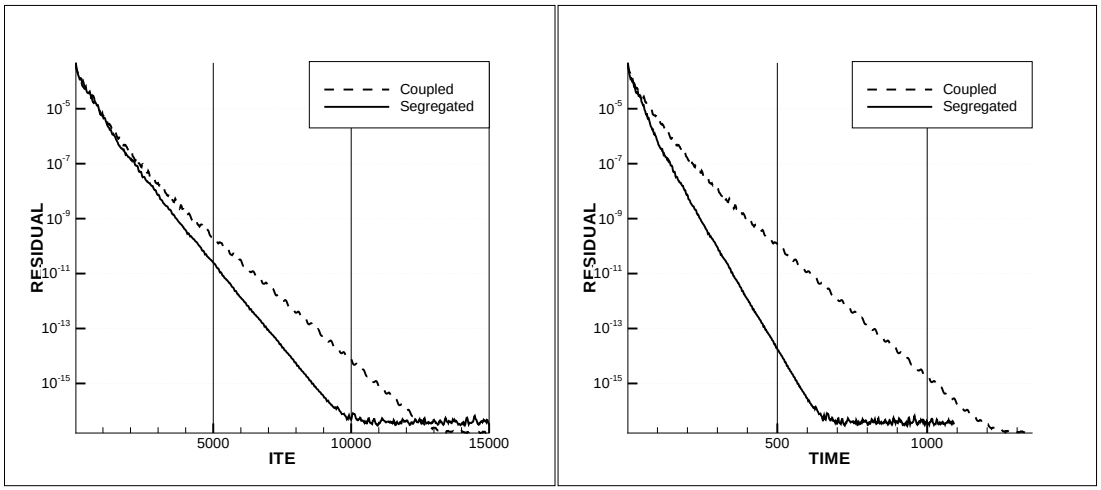

Figure 3.19: Comparison of the Coupled and Segregated Solutions on 260x40 Structured Grid over NACA 0008 Airfoil Profile

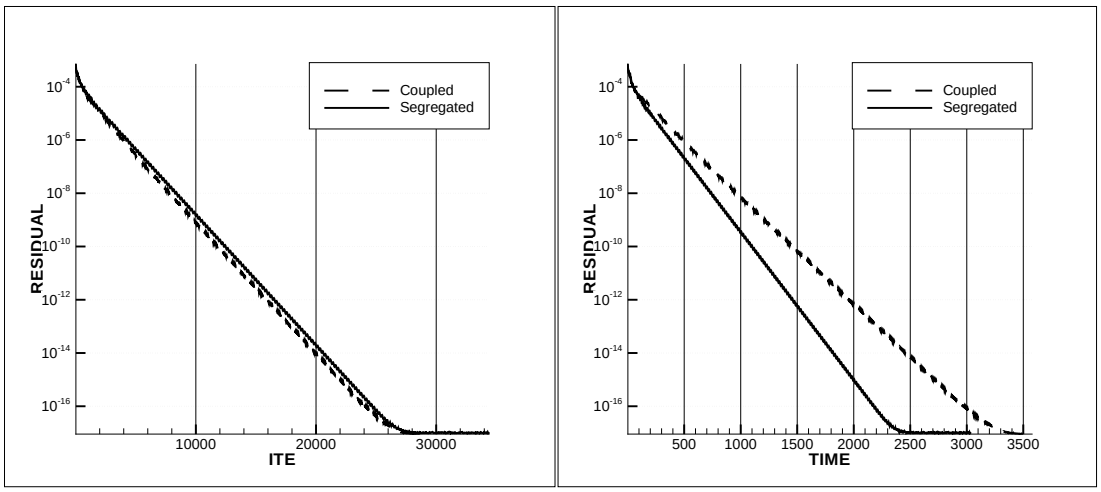

Figure 3.20: Comparison of the Coupled and Segregated Solutions on Unstructured Grid over NACA 0008 Airfoil Profile with 11584 Points

It is seen from these figures that, the convergence iteration number of the coupled solver can be slightly better than the segregated solver in some cases. But when the total computation time is concerned, it is observed that the segregated solver is more efficient.

After it is determined that the segregated solver is more time efficient, the comparison of the average residual versus iteration step and computation time of PGS and ACDI schemes are given in Fig 3.21 and Fig 3.22 for the 260x40 structured grid and the unstructured grid with 11584 points, respectively. When the convergence criteria is chosen to be 10^{-10}, a reduction of 24.1% in iteration number and 20.2% in total CPU time is obtained for the structured grid and a reduction of 24.3% in iteration number and 19.9% in total CPU time is obtained for the unstructured mesh.

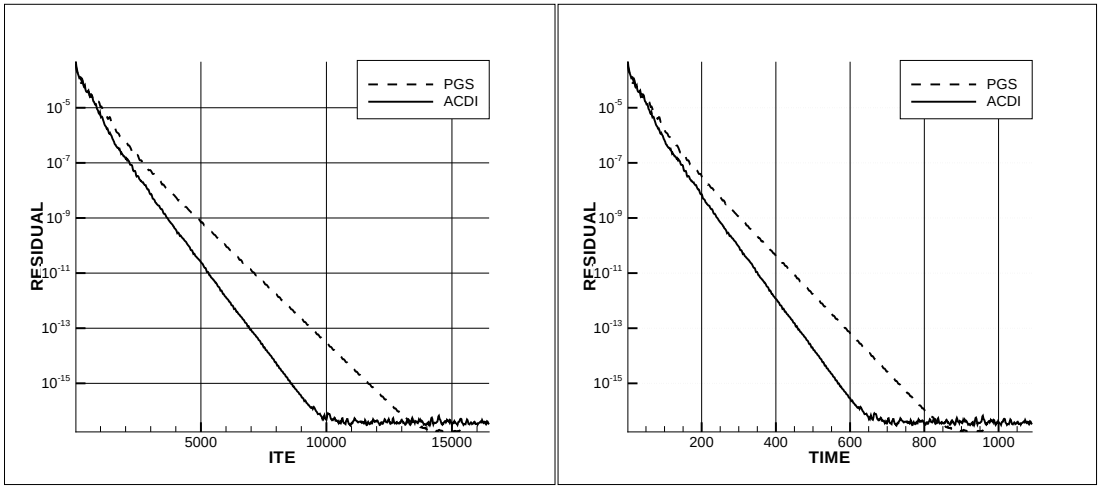

Figure 3.21: Average Residual versus Iteration Number and CPU Time for 260x40 Structured Grid over NACA 0008 Airfoil Profile

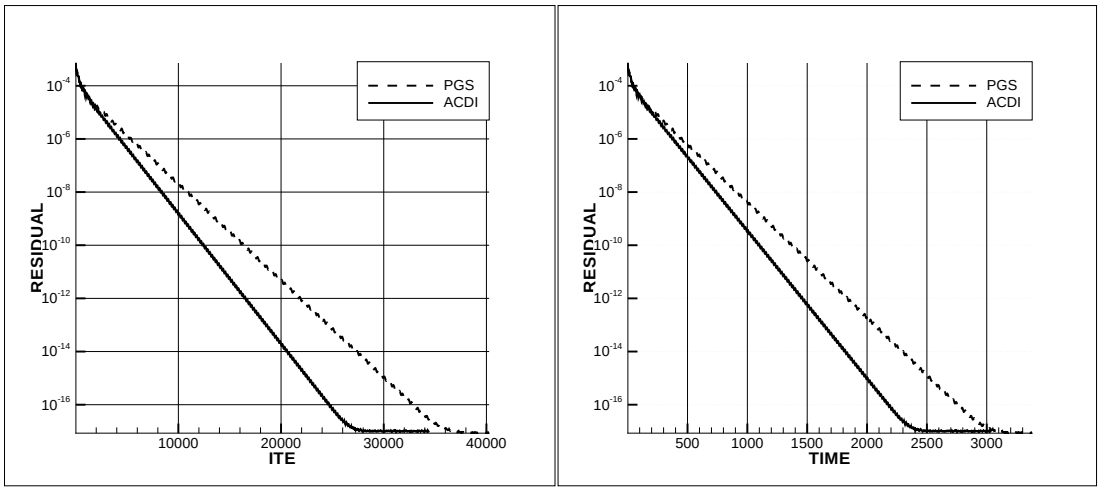

Figure 3.22: Average Residual versus Iteration Number and CPU Time for Unstructured Grid over NACA 0008 Airfoil Profile with 11584 Points

3.3 Flow over a Multi Element Airfoil

The last test case used for the validation of the developed solver is the high Reynolds number flow over the NACA 23012b multi element airfoil profile. Multi element airfoil calculations are one of the major cases showing the effectiveness of the unstructured meshes. Although it is easy to generate unstructured meshes over multi element airfoils, sophisticated techniques such as multi-block and Chimera methods should be employed for structured grid generation. The flow condition chosen for this test case is a Reynolds number of 1.46×10^6 with a 1.0 degree of angle of attack for the main airfoil. Experimental data taken from Reference [46] is used as the validation data for this test case. There are experimental data for different angles of attack and flap deflection angles in Reference [46]. For the flap deflection, 10 degrees is chosen. Because of high Reynolds number, a turbulent flow is expected over the multi element airfoil which is beyond the capabilities of the developed solver. A separation at the trailing edge of the main airfoil is observed for the laminar solution which is damped by the enhanced viscosity of the turbulent flow in real physics. For that reason, inviscid solution is taken for comparison with the validation data.

An unstructured mesh with 20075 points is used for the solution. The generated unstructured computational mesh consists of 544 solution bands where the minimum band length is 4 and the maximum band length is 951 cells. The computational mesh is shown in Fig 3.23.

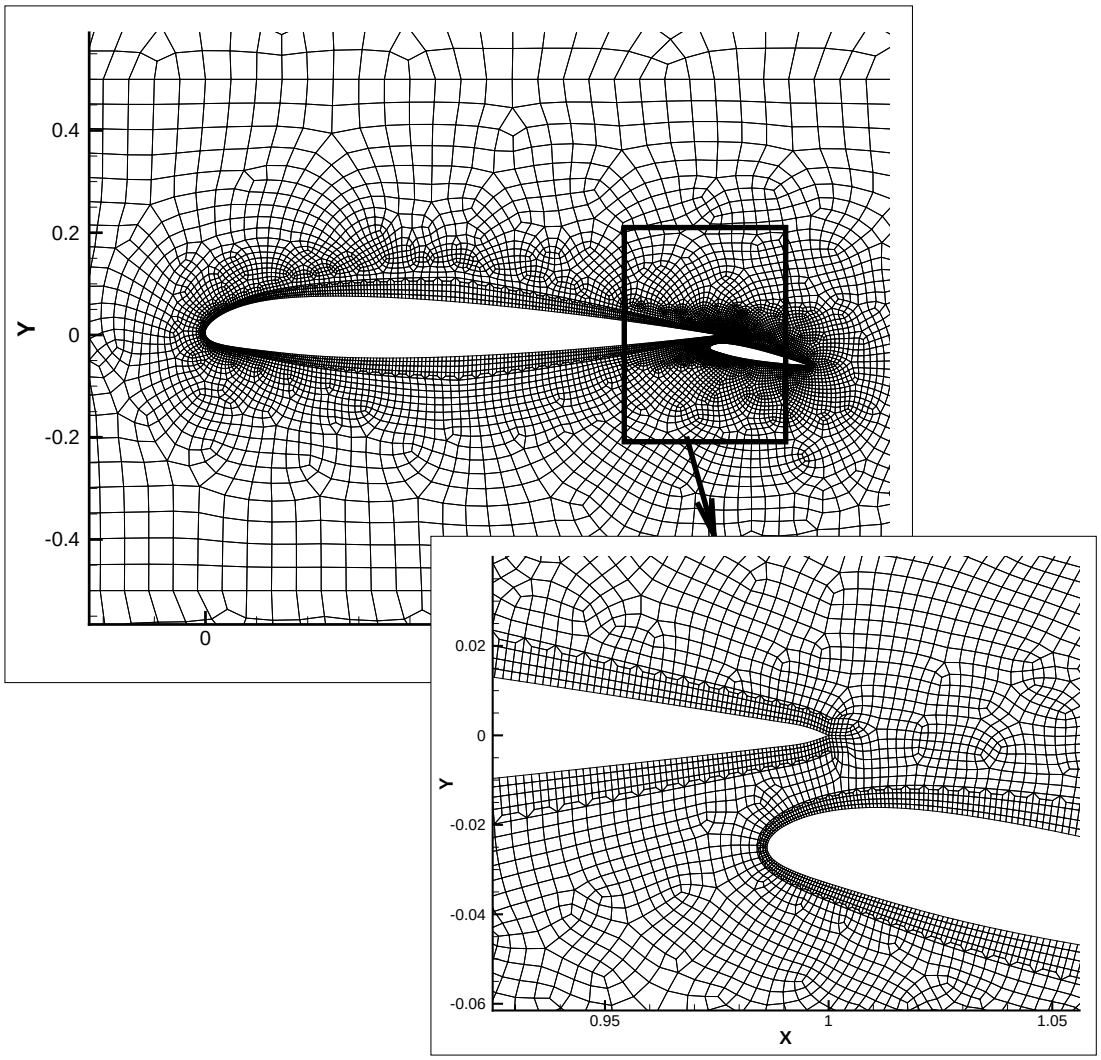

Figure 3.23: Unstructured Mesh Used for NACA 23012b Multi Element Airfoil Calculation with 20075 Points

The resulting pressure coefficient distribution over the airfoil profile for the unstructured mesh is given below with the experimental data.

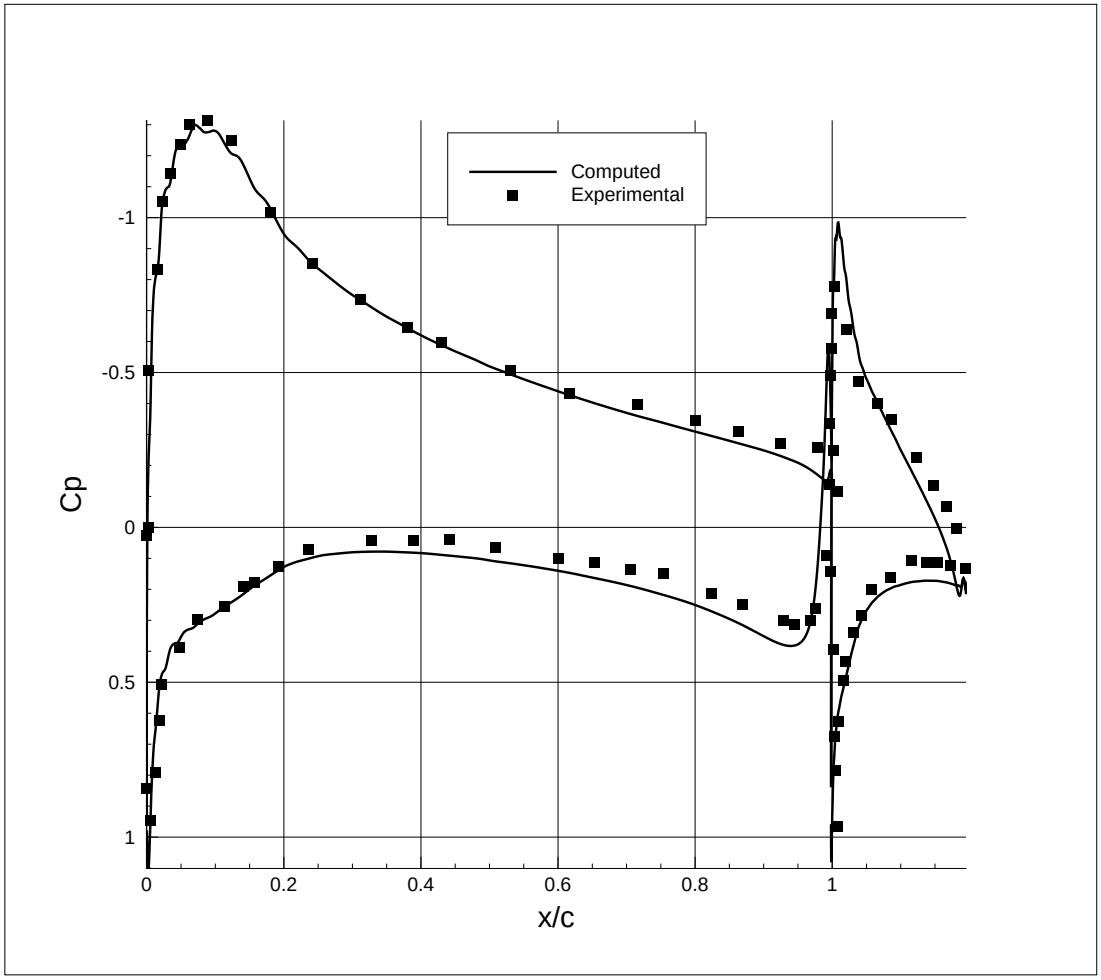

Figure 3.24: Calculated Pressure Coefficient Distribution over NACA 23012b Airfoil Using Unstructured Mesh with 20075 Points

The pressure coefficient distribution on the whole domain around the NACA 23012b multi element airfoil and close-up the gap between main airfoil and flap are shown in Fig 3.25 as flood contours.

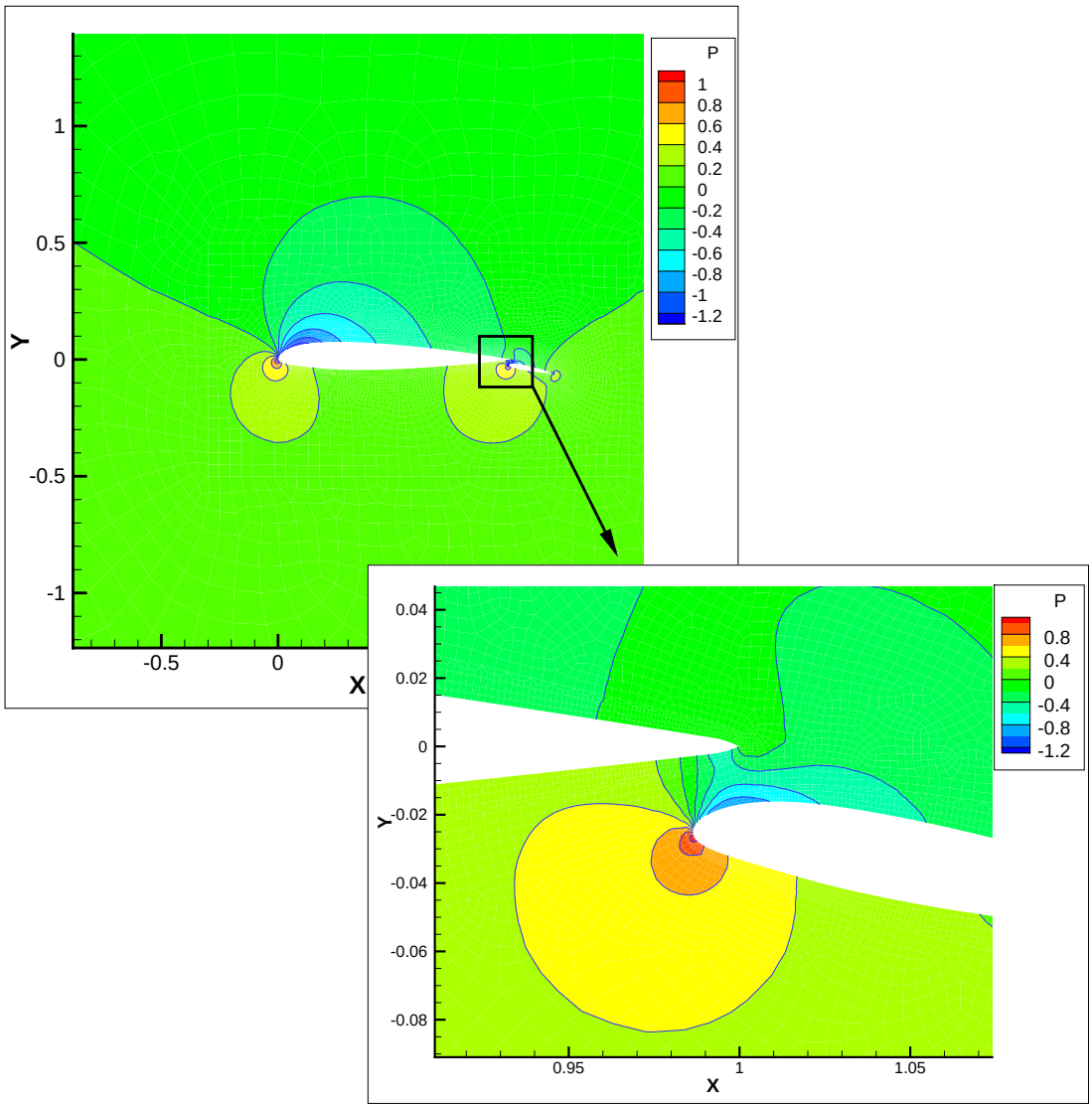

Figure 3.25: Flood Contour of the Pressure Coefficient on Unstructured Mesh over NACA 23012b with 20075 Points

It is seen from Figure 3.24 that although inviscid flow solution is taken for a turbulent case, good agreement with the experimental data is obtained for the pressure coefficient distribution.

3.3.1 Flux Evaluation Comparison

The comparison of the first, second and third order variable reconstruction schemes on the unstructured grid over NACA23012b airfoil profile is shown graphically in Fig 3.26.

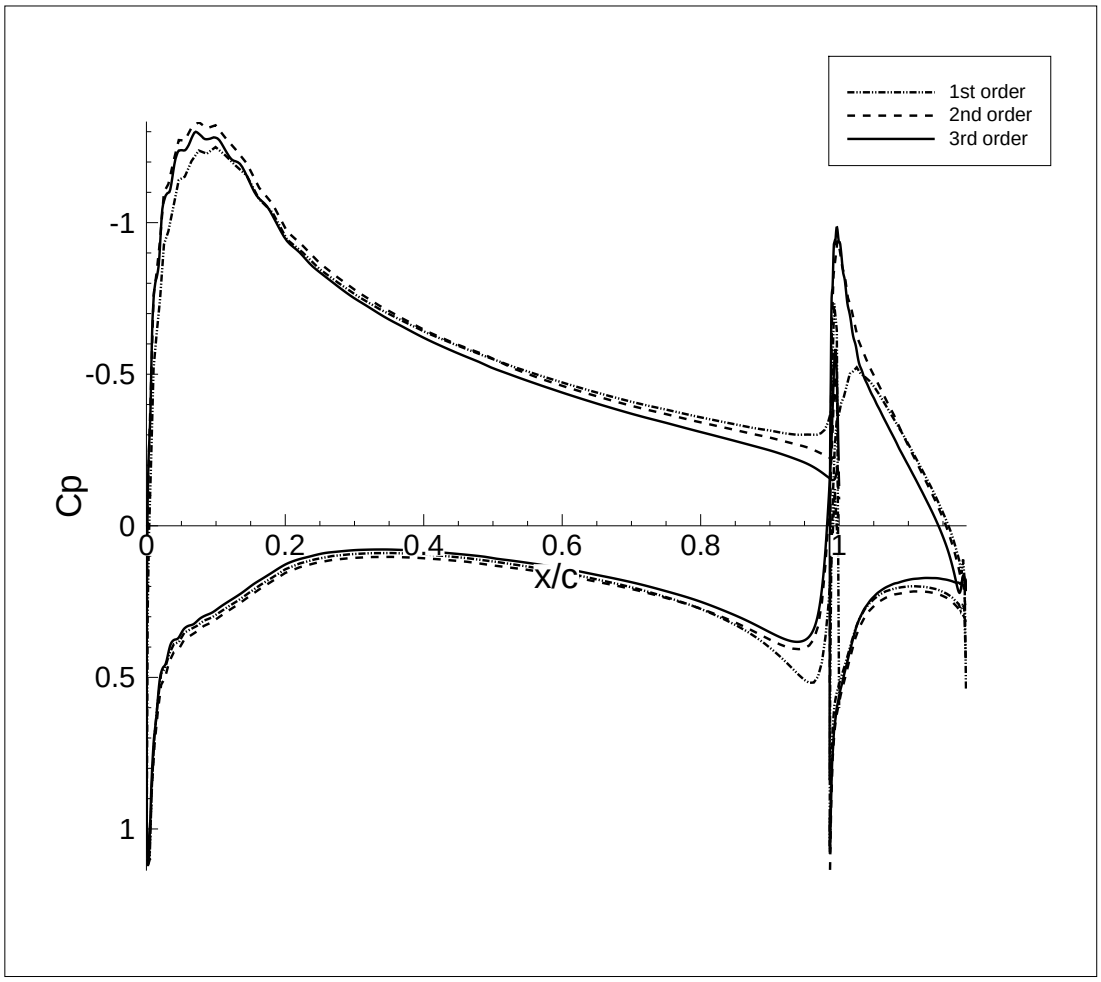

Figure 3.26: Graphical Representation of the Accuracy Gain on Unstructured Mesh over NACA 23012b Multi-Element Airfoil Profile with 20075 Points

3.3.2 Temporal Discretisation Comparison

The comparison of the residual versus iteration step and computation time for the multi-element airfoil are given in Fig 3.27. It is seen from Fig 3.27 that, a reduction of 18.3% in convergence iteration number and 13.8% reduction in total computation time is obtained for this test case when the convergence criteria is chosen to be 10^{-10}.

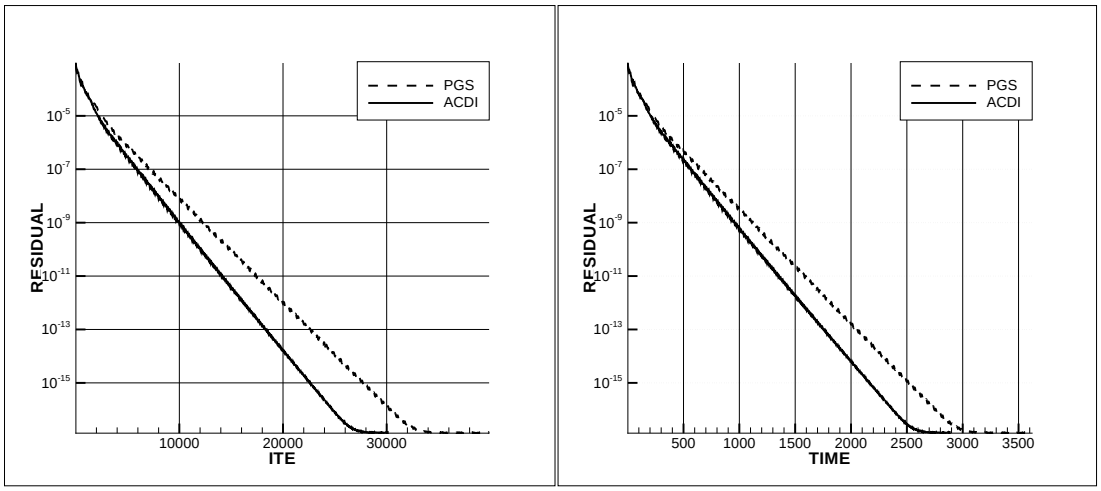

Figure 3.27: Average Residual versus Iteration Number and Computation Time for Unstructured Grid over NACA23012b Multi-Element Airfoil Profile with 20075 Points

CHAPTER 4

CONCLUSIONS

An implicit algorithm for computing the incompressible Navier-Stokes equations on structured and quadrilateral unstructured grids has been developed and presented. The artificial compressibility and Alternating Cell Directions Implicit methods with a computationally efficient cell-centered algorithm are used in this study. The main features of the developed code are given below.

- Artificial compressibility form of the incompressible Navier-Stokes equations are used.
- Alternating Cell Directions Implicit and Point Gauss Seidel Methods are used for temporal discretisation.
- Structural and quadrilateral unstructured meshes are used.
- U-MUSCL variable reconstruction scheme and Roe approximate Riemann solver are used for the computation of the convective fluxes.
- Element-based gradient evaluation method is used for the viscous fluxes.

For validation of the developed code, numerical computations were presented for the incompressible solutions of three different test cases including the laminar flow over a flat plate and

flows over single and multi element airfoils. The results showed good agreement with the reference data for the selected test cases.

The major contribution of the present work is the implementation of the Alternating Cell Direction Implicit method to the incompressible Navier-Stokes equations. Computational efficiency of the developed algorithm/code is compared with the PGS scheme in terms of CPU time and number of iterations required to obtain a converged solution. These comparisons show that the algorithm used in this study reduces the computation times 13-20% for the selected test cases.

For future developments, implementation of turbulence models is definitely a recommended course of action. Turbulence models are essential especially for accurate solution of high Reynolds number viscous flows. Additional recommendations for future work include obtaining time accuracy via dual time stepping method and extension to three dimensions. The method developed can easily be extended to 3-D flow solver.

REFERENCES

[1] Justesen, P., "*A Numerical Study of Oscillating Flow Around a Circular Cylinder*", Journal of Fluid Mechanics, Vol. 222, pp. 157-196, 1991.

[2] Osswald, G.A., Ghia, K.N, Ghia, U., "*A Direct Algorithm for Solution of Incompressible Three-Dimensional Unsteady Navier-Stokes Equations*", AIAA Paper 87-1139, 1987.

[3] Ahn, H.T., "*A New Incompressible Navier-Stokes Method with General Hybrid Meshes and Its Application to Flow/Structure Interaction*", PhD Thesis, Dept. of Aerospace Engineering and Engineering Mechanics, The University of Texas, 2005.

[4] Harlow, F.H. and Welch, J.E., "*Numerical Calculation of Time-Dependent Viscous Incompressible Flow with Free Surface*", Physics of Fluids, Vol. 8, pp. 2182-2189, 1965.

[5] Chorin, A.J., "*Numerical Solution of the Navier-Stokes Equations*", Math. Comp., Vol. 22, pp. 742-762, 1968.

[6] Patankar, S.V. and Spalding, D.B., "*A Calculation Procedure for Heat, Mass and Momentum Transfer*", Intl. Journal of Heat and Mass Transfer, Vol. 15, pp. 1787-1806, 1972.

[7] Hafez, M.M., "*Numerical Simulations of Incompressible Flows*", World Scientific Pub. Co. Inc., 2003.

[8] Patankar, S. V., "*Numerical Heat Transfer and Fluid Flow*", Hemisphere Publishig Co., New York, 1980.

[9] Chen, Y.S., Shang, H. M. and Chen, C. P., "*Unified CFD Algorithm with a Pressure Based Method*", 6th Intl. Symposium on Comp. Fluid Dyn., 1995.

[10] Chorin, A.J., "*A Numerical Method for Solving Incompresible Viscous Flow Problems*", J. Comp. Phy., Vol. 2, pp. 12-26, 1967.

[11] Peyret, R., "*Unsteady Evolution of a Horizontal Jet in a Stratified Fluid*", Journal of Fluid Mechanics, Vol. 78, pp. 49-63, 1976.

[12] Steger, J. L. and Kutler, P., "*Implicit Finite-Difference Procedures for the Computation of Vortex Wakes*", AIAA J., Vol. 15, No. 4, pp. 581-590, 1977.

[13] Kwak, D., Chang, J. L. C., Shanks, S. P., and Chakravarty, S., "*A Three-Dimensional Incompressible Navier-Stokes Flow Solver using Primitive Variables*", AIAA J, Vol. 24, No. 3, pp. 390-396, 1986.

[14] Chang, J. L. C., Kwak, D., Rogers, S. E. and Yang, R-J, "*Numerical Simulation Metods of Incompressible Flows and an Application to the Space Shuttle Main Engine*", Int. J. Numerical Method in Fluids, Vol. 8, pp. 1241-1268, 1988.

[15] Choi, D. and Merkle, C. L., "*Application of Time-Iterative Schemes to Incompressible Flow*", AIAA J., Vol. 23, No. 10, pp. 1518-1524, 1985.

[16] Merkle, C. L. and Athavale, M., "*Time-Accurate Unsteady Incompressible Flow Algorithms Based on Artificial Compressibility*", AIAA Paper 87-1137, 8th Computational Fluid Conference, 1987.

[17] Rogers, S. E. and Kwak, D., "*An Upwind Differencing Scheme for the Time-Accurate Incompressible Navier-Stokes Equations*", AIAA Paper 88-2583, AIAA 6th Applied Aerodynamics Conference, Williamsburg, VA, June 6-8, 1988.

[18] Rogers, S. E. Kwak, D. and Kiris, C., "*Steady and Unsteady Solutions of the Incompressible Navier-Stokes Equations*", AIAA J. Vol. 29, No. 4, pp. 603-610, 1991.

[19] Belov, A. Martinelli, L. and Jameson, A., "*A New Implicit Algorithm with Multigrid for Unsteady Incompressible Flow Calculations*", AIAA Paper 95-0049, 33rd Aerospace Sciences Meeting and Exhibit, 1995.

[20] Sato, Y., Hino T. and Hinatsu M., "*Unsteady Flow Simulation Around a Moving Body by an Unstructured Navier-Stokes Solver*", Proc. Numerical Towing Tank Simulation, 2003.

[21] Chan, C. T. and Anastasiou K., "*Solution of Incompressible Flows With or Without a Free Surface Using the Finite Volume Method on Unstructured Triangular Meshes*", Int. J. Numerical Method in Fluids, Vol. 29, pp. 35-57, 1999.

[22] Rogers, S., E., "*A Comparison of Implicit Schemes for the Incompressible Navier-Stokes Equations with Artificial Compressibility*", AIAA Paper 95-0567, 33rd Aerospace Sciences Meeting and Exhibit, 1995.

[23] Yuan, L. "*Comparison of Implicit Multigrid Schemes for Three Dimensional Incompressible Flows*", J. Comp. Phy., Vol. 177, pp. 134-155, 2002.

[24] Anderson, W. K., Rausch, R.D., and Bonhaus, D. L., "*Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids*", J. Comp. Phy., Vol. 128, pp. 391-408, 1996.

[25] Çete, A. R. and Kaynak, U., "*A New Approximate Factorization Method Suitable for Structured and Unstructured Grids*", AIAA Paper 2006-3789, 9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference 2006.

[26] Çete, A.R., "*Yapısız Ayrık Yönler Metodu*", Doktora Tezi, İstanbul Teknik Üniversitesi, 2004.

[27] Hoffmann, K.A., Chiang, S.T., "*Computational Fluid Dynamics*", Engineering Education System Co., Wichita, 2000.

[28] Jameson, A., Schmidt, W., Turkel, E., "*Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes*", AIAA Paper 81-1259, 14th Fluid and Plasma Dynamics Conference, 1981.

[29] Rogers, S.E., "*Numerical Solution of the Incompressible Navier-Stokes Equations*", PhD Thesis, Dept. of Aeronautics and Astronautics, Stanford University, 1989.

[30] Harten, A., Lax, P.D., Van Leer, B., "*On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws*", SIAM Review, 25(1):35-61, 1983.

[31] Toro, E.F., "*Riemann Solvers and Numerical Methods for Fluid Dynamics*", Springer-Verlag, Berlin, 1997.

[32] Van Leer, B., Thomas, J.L., Roe, P.L., "*A Comparison of Numerical Flux Formulas for the Euler and Navier-Stokes Equations*", AIAA Paper 87-1104, 8th Computational Fluid Dynamics Conference, 1987.

[33] Roe, P.L., "*Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes*", J.Comput. Phys., Vol. 43, pp. 357-372, 1981.

[34] Van Leer, B., "*Flux-Vector Splitting for the Euler Equations*", Proc. 8th Int. Conf. on Numerical Methods in Fluid Mechanics, pp. 507-512, 1982.

[35] Steger, J. L. and Warming, R. F., "*Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite Difference Methods*", J. Computational Physics, Vol. 40, pp. 263-293, 1981.

[36] Liou, M., Steffen, C. J., "*A New Flux Splitting Scheme*", J. Computational Physics, Vol. 107, pp. 23-39, 1993.

[37] Burg, C.O.E., "*Higher Order Variable Extrapolation For Unstructured Finite Volume RANS Flow Solvers*", AIAA Paper 2005-4999, 17th Computational Fluid Dynamics Conference, 2005.

[38] Blazek, J, "*Computational Fluid Dynamics: Principles and Applications*", Elsevier Science Ltd., Oxford, 2001.

[39] Onur Ö., "*Effect of the Jacobian Evaluation on Direct Solutions of the Euler Equations*", MSc dissertation, Middle East Technical University, 2003.

[40] Venkatakrishnan, V., "*Preconditioned Conjugate Gradient Methods for the Compressible Navier-Stokes Equations*", AIAA J. Vol. 29, No. 7, pp. 1092-1100, 1991.

[41] Dinçgez, U., "*Three Dimensional Hyperbolic Grid Generation*", MSc dissertation, Middle East Technical University, 2006.

[42] Mueller, J.D., "*On Triangles and Flow*", PhD Thesis, The University of Michigan, 1996.

[43] Blacker, T.D., Stephenson, M.B., "*Paving: A New Approach to Automated Quadrilateral Mesh Generation*", Int. J. Numerical Method in Engineering, Vol. 32, pp. 811-847, 1991.

[44] Schlichting, H., "*Boundary Layer Theory*", McGRAW Hill, 1960.

[45] Kunz, P.J., Kroo, I.M., "*Analysis, Design, and Testing of Airfoils for Use at Ultra-Low Reynolds Numbers*", in "*Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications*", Mueller, T.J., AIAA, 2002.

[46] Wenzinger, J. C., "*Pressure Distribution over an N.A.C.A 23012 Airfoil with an N.A.C.A. 23012 External-Airfoil Flap*", N.A.C.A Report No. 614, 1960.