THREE-DIMENSIONAL LAMINAR COMPRESSIBLE NAVIER STOKES SOLVER FOR INTERNAL ROCKET FLOW APPLICATIONS

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

BY

KORHAN COŞKUN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MECHANICAL ENGINEERING

DECEMBER 2007

Approval of the thesis:

THREE-DIMENSIONAL COMPRESSIBLE NAVIER STOKES EQUATIONS SOLVER FOR INTERNAL ROCKET FLOW APPLICATIONS

submitted by KORHAN COŞKUN¸ in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering Department, Middle East Technical University by,

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

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

Prof. Dr. ˙M. Haluk AKSEL _____
Supervisor, Mechanical Engineering Dept., METU

Dr. H. Tuğrul TINAZTEPE _____
Co-supervisor, Roketsan Missiles Industries Inc.

Examining Committee Members:

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

Prof. Dr. ˙M. Haluk AKSEL _____
Mechanical Engineering Dept., METU

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

Dr. H. Tuğrul TINAZTEPE _____
Roketsan Missiles Industries Inc.

Dr. K.Atılgan TOKER _____
Roketsan Missiles Industries Inc.

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 : Korhan COŞKUN

Signature :

ABSTRACT

THREE-DIMENSIONAL COMPRESSIBLE NAVIER STOKES EQUATIONS SOLVER FOR INTERNAL ROCKET FLOW APPLICATIONS

COŞKUN, Korhan
M. Sc., Department of Mechanical Engineering
Supervisor: Prof. Dr. M. Haluk AKSEL
Co-Supervisor: Dr. H. Tuğrul TINAZTEPE

December 2007, 72 pages

A three dimensional, Navier-Stokes finite volume flow solver which uses Roe's upwind flux differencing scheme for spatial and Runge-Kutta explicit multi-stage time stepping scheme and implicit Lower-Upper Symmetric Gauss Seidel (LU-SGS) iteration scheme for temporal discretization on unstructured and hybrid meshes is developed for steady rocket internal viscous flow applications. The spatial accuracy of the solver can be selected as first or second order. Second order accuracy is achieved by piecewise linear reconstruction. Gradients of flow variables required for piecewise linear reconstruction are calculated with both Green-Gauss and Least-Squares approaches.

The solver developed is first verified against the three-dimensional viscous laminar flow over flat plate. Then the implicit time stepping algorithms are compared against two rocket motor internal flow problems. Although the solver is intended for internal flows, a test case involving flow over an airfoil is also given. As the last test case, supersonic vortex flow between concentric circular arcs is selected.

Key-words: Navier-Stokes Equations, Implicit, Second Order Spatial Accuracy, Finite Volume Method, Lower-Upper Symmetric Gauss Seidel, Least Squares Method, Venkatakrishnan's Limiter.

ÖZ

ROKET İÇ AKIŞ UYGULAMALARI İÇİN ÜÇ BOYUTLU SIKIŞTIRILABİLİR NAVIER-STOKES ÇÖZÜCÜSÜ

COŞKUN, Korhan
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. M. Haluk AKSEL
Ortak Tez Yöneticisi: Dr. H. Tuğrul TINAZTEPE

Aralık 2007, 72 sayfa

Roket motorları ağdalı iç akış uygulamalarının zamandan bağımsız, durağan çözümlerinin yapılabilmesi için, düzensiz ve hibrit çözüm ağını uzayda Roe'nun yön hassas (upwind) akı ayrımına dayalı yöntemini, zamanda ise Runge-Kutta çok kademeli ve implisit Alt-Üst Simetrik Gauss-Seidel İterasyon yöntemini kullanarak ayrıştıran, üç boyutlu bir sonlu hacim Navier-Stokes denklemleri çözücüsü geliştirilmiştir. Çözücü uzayda hem birinci hem de ikinci dereceden doğrulukla çözüm yapabilmektedir. İkinci dereceden doğruluk doğrusal yeniden yapılandırma yöntemiyle elde edilmiştir. Doğrusal yeniden yapılandırma için gerekli olan akış değişkenlerinin türevleri Green-Gauss ve en küçük kareler yöntemi kullanılarak bulunmuştur.

Öncelikle geliştirilen çözücünün üç boyutta ağdalı laminar akış çözümleme yeteneği üç boyutlu düz plaka problemi üzerinde gerçekleştirilmiştir. Implisit zamanda ilerleme algoritmasının doğruluğu iki roket motoru iç akış problemleri ile sınanmıştır. Her ne kadar çözücü, iç akışlar için hazırlanmış olsa da, kanat üzeri akış çözümü de yapılmıştır. Son olarak, eşeksenli silindirler arasındaki supersonic vorteks akış çözümleri verilmiştir.

Anahtar Kelime: Navier-Stokes Denklemleri, İmplisit, Uzayda İkinci Derece Doğruluk, Sonlu Hacim Metodu.

ACKNOWLEDGEMENTS

I would like to express my deepest thanks and gratitude to Prof. Dr. Haluk AKSEL for his supervision, encouragement, understanding and constant guidance.

I would like to thank to Dr. Tuğrul TINAZTEPE for crucial advises on flow solvers, which help combining the mathematical work to real physics of the rocket propulsion.

I would like to express my sincere appreciation to my colleagues Onur BAŞ, Güneş NAKİBOĞLU and for their crucial advises and invaluable efforts during the preparation of this thesis.

I would like to thank to Dr. Kemal Atılgan TOKER, for supplying the 3-dimensional hybrid flow solver and help me extend its capabilities.

I would like to thank to ROKETSAN for partially supporting this study.

My gratitude is endless for my family whom this thesis is devoted to.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

a	speed of sound
C_p	pressure coefficient, specific heat at constant pressure
C_f	local friction coefficient
D	diagonal matrix
e	total internal energy per unit volume
F	x-component of flux vector, cell face flux
G	y-component of flux vector, mass flux
$\vec{H}$	convective flux vector with components F, G, K
$\vec{H}_v$	diffusive flux vector with components F_v, G_v, K_v
H	total enthalpy, flux at a face
I	identity matrix
$\vec{i}$	unit normal vector in x-direction
$\vec{j}$	unit normal vector in y-direction
$\vec{k}$	unit normal vector in z-direction
K	right eigenvector of convective flux jacobian matrix, z-component of flux vector
$L(i)$	lower neighbors of a cell
l	length
$\dot{m}$	mass flow rate
M	Mach number
N	number of faces
$\vec{n}$	normal vector
Pr	Prandtl number
p	pressure

Re	Reynolds number
R	gas constant, residual
S	surface
ΔS	face surface area
t	time
T	temperature
$U(i)$	upper neighbors of a cell
u	x-component of velocity vector
q	heat flux
Q	column vector of conservative variables
v	y-component of velocity vector
$\bar{V}$	velocity vector
w	z-component of velocity vector
x,y,z	Cartesian space coordinates

Other Symbols:

ρ	density
γ	specific heat ratio
∂	partial derivative operator
Ω	control volume
λ	eigenvalue, spectral radius of flux jacobian matrix
τ	shear stress
μ	absolute viscosity
κ	thermal conductivity, limiter function parameter
η	Blasius similarity variable
α	wavestrength, Runge-Kutta method coefficients

Superscripts:

L	value at the left hand side of a face

n	quantity at discrete time level n
R	value at the left hand side of a face
*	non-dimensional quantity

Subscripts:

i, I	cell number
L	value at the left hand side of a face
R	value at the left hand side of a face
j, J	neighbor cell number
inj	injection property
n	normal component
0	stagnation state
ref	reference
max	maksimum value
min	minimum value
∞	freestream value
x	x component
y	y component
z	z component

CHAPTER 1

INTRODUCTION

1.1 PURPOSE OF THE STUDY

The purpose of this study is to develop a three-dimensional flow solver, which can be used for solid propellant rocket motor internal flow applications in order to analyze its performance.

Flow solver is capable of solving Navier-Stokes equations in three dimensions for steady state problems. It uses the finite volume method, with upwind flux calculation of Roe. Moreover, the solver can operate on structured and hybrid grids. For the integration of governing equations in time, solver can use both explicit and implicit methods. Accuracy of spatial discretization can be selected to be first and second order.

The developed solver, which is described in this thesis, is validated with the results of previously validated performance prediction codes. One test is a flow over a flat plate, which is used to test the solvers capability to resolve laminar flow. Two rocket internal flow problems are employed to test the implicit time stepping algorithm of the solver. Fourth test case, flow over an airfoil, is investigated which involves shock flow field. Last test case is inviscid, supersonic vortex between two concentric circular arcs.

1.2 CONTENTS OF THE THESIS REPORT

In Chapter 2, the methods and common practices used in simulation of fluid flow problems are introduced. Discretization techniques used in literature are briefly described. Spatial and temporal discretization methods together with different grid types such as structured, unstructured and hybrid grids, are presented in this chapter.

Chapter 3 contains the governing flow equations with the spatial and temporal discretization methods used in this thesis. The details of spatial discretization is presented for convective and diffusive fluxes separately. Convective fluxes are calculated using the Riemann solver of Roe. Also, how second order of accuracy of convective flux calculation is achieved is given in Chapter 3. Both explicit and implicit time discretization methods are used, which are Multi-Stage Runge Kutta Method and Lower-Upper Symmetric Gauss Seidel Iteration, respectively. Chapter 3 concludes with the description of boundary conditions used in the solver developed.

Validation of the solver is presented in Chapter 4. First, solver's capability of solving laminar flow is investigated with three dimensional flow over a flat plate. Both explicit and implicit solutions are performed to compare the performance of time discretization methods. Then, test cases involving rocket internal flow are used to further validate the solver. The last test case is the flow over an airfoil where there are shock waves. This test case's aim is to investigate the capability of the solver to resolve the shock structures. Finally, supersonic vortex flow between concentric circular arcs is investigated to verify second order accuracy of the solver.

CHAPTER 2

LITERATURE SURVEY

2.1 NUMERICAL DISCRETIZATION TECHNIQUES

In Computational Fluid Dynamics (CFD) there have been mainly three discretization techniques for the solution of the governing equations of the fluid motion. These methods have advantages and disadvantages with respect to the area of application, and will be covered in the subsequent sections.

2.1.1 Finite Difference Method

Finite Difference Method (FDM) is the earliest discretization technique applied in CFD, and for this reason it has been the most commonly used technique until today. This method is simple to apply and uses Taylor Series expansion of derivatives of flow variables. However, this method requires highly regular structured grids to obtain accurate results.

2.1.2 Finite Element Method

Finite Element Method (FEM) is firstly developed for the analysis of structural elements and later adapted to the fluid flow problems. This method uses the weak variational form of the governing equations to convert high order derivatives to lower ones. Also, the variation of flow variables in a control volume is handled with use of polynomial shape functions which gives the flexibility in increasing the accuracy of the solution.

2.1.3 Finite Volume Method

Finite Volume Method (FVM) is based on the physical concept of using macroscopic control volumes to numerically solve the conservation laws of fluid motion. In FVM, the integral form of the governing equations are used in the discretization. Since discretization of conservation laws is done in integral form, mass, momentum and energy is conserved at each control volume in the domain.

In FVM method, the grid topology alternatives are countless. Almost any arbitrary grid structure can be used in the formulation of FVM [1]. The combination of the formulation of a flow problem on control volumes, which is the most physical way to obtain a discretization, with the flexibility in the geometric representation of the domain and the flexibility in defining the discrete flow variables, makes FVM extremely popular in engineering applications.

A common feature of most upwind codes is their finite volume representation. In each cell, the flow variables are supposed to be distributed in a specified way; piecewise constant; piecewise linear; or some higher-order representation.

Frink used finite volume formulation with upwind differencing for the solution of different aeerodynamic problems [2, 3]. Batina [4] also used FVM formulation for complex aircraft aerodynamic analysis introducing the dynamic mesh algorithms into it.

Besides its advantages, FVM has a handicap in the definition of the derivatives, which are required for viscous flow simulations. Since the computational grid may not be orthogonal or equally spaced for unstructured grids, the definition of the derivatives of flow variables based on Taylor series expansion is impossible. Also, weak formulation in FEM used to convert high order derivatives to lower ones, can not be applied in FVM. For the definition of derivatives, methods such as Green-

Gauss (GG) and Least-Squares (LSQ) are the most commonly methods used in FVM.

There are mainly two approaches for the approximation of mass, momentum and energy fluxes over the surface of control volumes in computational domain:

2.1.3.1 Cell Vertex Schemes

In the cell vertex scheme, the flow properties are assigned at the vertices of the cells of the grid. Flow properties are directly calculated at the nodes which gives the advantage of not distributing the flow variables from cell center values to nodes after numerical calculations are ended. Cell-vertex discretization offer advantages in accuracy especially on non-regular grids.

Since flow variables are assigned at the nodes of the grid, control volumes need to be defined around the nodes. The generation of these control volumes are requires algorithms.

Cell vertex formulation has been applied by Barth [5], Venkatakrishnan [6] and Mavriplis [7] to several applications of steady and unsteady computations.

2.1.3.2 Cell Centred Schemes

In the cell centred formulation, the flow properties are directly calculated at the center of the computational cell which itself is the control volume for finite volume discretization. Therefore, there is no need to generate new control volumes. However, since the flow variables at cell centers need to be extrapolated to nodes in the domain, some numerical errors may be introduced during this extrapolation.

Frink [8,9] and Batina [4] applied cell center approach in their steady state aerodynamic simulation methods.

2.2 NUMERICAL SCHEMES

2.2.1 Spatial Discretization

Most of the algorithms were based on either finite element or central differencing which need implementation of artificial dissipation for acceptable results. Central differencing schemes can not take into account the direction of the flow, because the physical propagation of flow information along the characteristics is not considered. For smooth flows, which have no big discontinuities, central schemes based on Taylor series expansion can be used. However, when discontinuities appear significantly, information of the flow direction is required to resolve the non-linear behaviour of the flow.

To overcome this difficulty, upwind schemes are developed which take the direction of the flow in consideration. Nowadays, upwind schemes are the major spatial discretization technique of main research and commercial codes [3,10].

Upwind differencing uses the propagation of information with the theory of characteristics in constructing the information traveling in opposite directions in a separate and stable manner.

Second order central schemes require scalar artificical dissipation to damp oscillations generated near the high gradient regions. Higher order upwind methods require limiter functions for second order accuracy in space.

Characteristic theory is easy to understand in one-dimensional flows, but for 2-D and 3-D flow problems, the flow direction is not clearly identified. However, choosing the upwind direction being normal to the face of the computational cell

across which the fluxes are computed is a commonly used way for 2-D and 3-D flows.

Upwind schemes may be divided into two categories as Flux Vector Splitting Type Schemes and Flux Difference Splitting (Godunov) Type Schemes.

2.2.1.1 Flux Vector Splitting Schemes

Upwind discretization is obtained by splitting the flux vector into two parts based on information coming from upwind and downwind of the cell face in flux vector splitting algorithms [11]. In other words, the flux terms are split according to the sign of associated with propagation speeds.

The main drawback of flux vector splitting methods is evident in the vicinity of sonic conditions since the splitting of flux is done with respect to the sign of the Mach number or the velocity vector.

2.2.1.2 Flux Difference Splitting (Godunov) Schemes

In the flux difference splitting schemes, local Riemann problem is solved on each face of cells. The flow variables are taken as constant over the left and right states of the cell face. Using left and right states of the face, local Riemann problem is solved to achieve convective fluxes at the face. In the original Godunov scheme [12], the local Riemann problem is solved exactly. Since this approach is computationally expensive, some approximate Riemann solvers have been built by Roe [13], Osher and Solomon [14], Toro [15] and Harten [16].

2.2.2 Temporal Discretization

There are basically two types of time stepping methods used both for integrating governing equations in time to obtain solutions for steady and unsteady flows. There are Explicit Time Stepping Algorithms and Implicit Time Stepping Algorithms.

Due to the simplicity both in physically sound procedure and numerical implementation, explicit time stepping methods are widely used for flow problem simulations [17,18]. These methods are simple and can be easily vectorized for parallel processing.

The explicit time-stepping schemes may be efficient methods for coarse meshes and for the simulation of Euler equations. However, solution of Navier-Stokes equations requires fine meshes especially within the boundary layer. For fine meshes, explicit methods lose their attractivity in terms of rate of convergence. Batina [4] stated that for cases where fine meshes are used, an implicit time discretization which can advance the solution in large time steps is much more favorable to obtain steady state solutions in a computationally efficient way.

In contrast of their high convergence rate, implicit methods require large amount of memory. Also, the application of some complicated implicit time stepping methods is quite complicated especially for viscous flows, since complicated viscous Jacobian matrices have to be derived and coded.

Alternating Direction Implicit (ADI) is one of the first iterative implicit schemes [19]. ADI method can only be used on structured grids. Lower-Upper Symmetric Gauss-Seidel (LU-SGS) scheme has a low numerical complexity level and modest memory requirement. LU-SGS was first introduced by Jameson and Yoon [20].

2.3 COMPUTATIONAL GRID

2.3.1 Structured Grids

In structured grids, neighboring grid points in the physical space are the same as the neighboring points in the computational space. In other words, structured grids have an implicit connectivity that allows specifying the grid locations by its computational coordinates.

Structured grids have the advantage of simple coding when numerical algorithms are concerned but it is difficult and sometimes impossible to produce structured grids for complex domains. Multi-block grids are used in order to overcome this difficulty but still it takes too much time to generate a grid around a complex geometry.

Also, the dense mesh which is required especially at wall boundaries for viscous calculations has to be carried all away into the far field for structured grids which obviously increases the memory requirements of the numerical code.

2.3.2 Unstructured Grids

Solution of flow fields around complex geometries directly brings the use of unstructured grids, since unstructured grids do not need any implicit connectivity like structured grids. Unstructured grids composed of triangular and tetrahedral elements in 2-D and 3-D, respectively, offer the designer the ability to model flows around complex bodies and to incorporate adaptive procedures to the solution.

Unstructured meshes are also computationally feasible when the grid generation time is concerned. It can be said that the size of the mesh, which directly affects the memory requirements of the numerical algorithm, can only be minimized by using unstructured grids.

For applications having complex geometries, the unstructured formulation appears to be the method of choice.

The most important drawback of unstructured meshes is low resolution of boundary layer and viscous wakes. The skewness of meshes near boundary creates high amount of numerical diffusion.

2.3.3 Hybrid Grids

The hybrid grid definitions combine the best practices of both structured and unstructured grids.

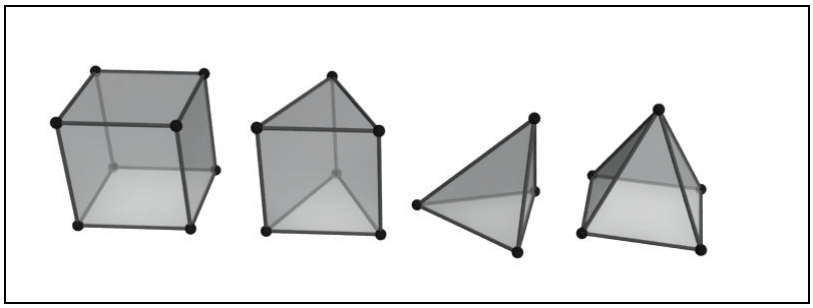

Figure 2.1 Hybrid grid elemens

Hybrid grids offer usage of structured high quality grids near the boundaries and usage of unstructured grids away from the boundaries. By this way, a mesh which is dense enough to observe the boundary layer and which is coarse enough far from the boundaries is obtained.

Near boundaries, layers of hexahedral cells can be used, for other parts of the domain, the field is filled with tetrahedral cells and pyramid and wedge type of cells which are the bridge between tetrahedral and hexahedral elements.

CHAPTER 3

NUMERICAL ALGORITHM

In this chapter, the numerical algorithm developed for the solution of the Navier-Stokes equations will be presented. First, governing equations will be described together with the nondimensionalization method used. Then, spatial and temporal discretization methods used in the solver will be presented. Spatial discretization is done according to the method called “Riemann solver of Roe”. Second order spatial discretization of governing equations are also given in this chapter. Time discretization is done both explicitly and implicitly. Explicit time integration of governing equations is done by Multi-Stage Runge Kutta Method, whereas the implicit method used is the Lower-Upper Symmetric Gauss-Seidel iteration. Finally, different types of boundary conditions used in this work will be presented.

The solver developed in this study is based on a three-dimensional Euler solver which is explicit in nature and first order accurate in space, for hybrid grids. The solver was upgraded to a Navier-Stokes solver which is second order accurate in space, and has the capability of implicit time integration.

3.1 GOVERNING EQUATIONS

Motion of the flow field in this study is modeled by the Navier-Stokes equations for a viscous, heat conducting, compressible gas without any external body force or heat generation with respect to a stationary reference frame. Equations are written in conservation law form in three-dimensions. The compressible gas is assumed to be a perfect gas. Governing equations will be presented in the following sections of this chapter in the detailed form as implemented in the numerical algorithm.

3.1.1 Three Dimensional Navier-Stokes Equations

The three-dimensional compressible Navier-Stokes equations in cartesian coordinates are given below in conservation form.

$$\frac{\partial Q}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}+\frac{\partial K}{\partial z}=\frac{\partial F_v}{\partial x}+\frac{\partial G_v}{\partial y}+\frac{\partial K_v}{\partial z} \tag{3.1}$$

$$Q=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ e\end{bmatrix}\quad F=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ \rho uH\end{bmatrix}\quad G=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vw\\ \rho vH\end{bmatrix}\quad K=\begin{bmatrix}\rho w\\ \rho uw\\ \rho vw\\ \rho w^2+p\\ \rho wH\end{bmatrix} \tag{3.2}$$

$$F_v=\begin{bmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ \tau_{xz}\\ \tau_{xx}u+\tau_{xy}v+\tau_{xz}w-q_x\end{bmatrix} \tag{3.3}$$

$$G_v = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ \tau_{yx}u + \tau_{yy}v + \tau_{yz}w - q_y \end{bmatrix} \tag{3.4}$$

$$K_v = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ \tau_{zx}u + \tau_{zy}v + \tau_{zz}w - q_z \end{bmatrix} \tag{3.5}$$

The shear stress components are expressed as;

$$\tau_{xx} = \mu\left(\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\left(\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)\right) \quad \tau_{yy} = \mu\left(\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\right)\right) \tag{3.6}$$

$$\tau_{zz} = \mu\left(\frac{4}{3}\frac{\partial w}{\partial z} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right) \qquad \tau_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{3.7}$$

$$\tau_{xz} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \qquad \tau_{yz} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \tag{3.8}$$

$$\tau_{yx} = \tau_{xy} \qquad \tau_{zx} = \tau_{xz} \qquad \tau_{yz} = \tau_{zy} \tag{3.9}$$

Components of heat flux are given as below.

$$-q_x = \kappa\frac{\partial T}{\partial x} \quad -q_y = \kappa\frac{\partial T}{\partial y} \quad -q_z = \kappa\frac{\partial T}{\partial z} \tag{3.10}$$

Finally, the eqation of state is written in terms of pressure as follows:

$$p = (\gamma - 1)\left(e - \frac{1}{2\rho}\left((\rho u)^2 + (\rho v)^2 + (\rho w)^2\right)\right) \tag{3.11}$$

Enthalpy per unit mass is defined as,

$$H = \frac{e + p}{\rho} \tag{3.12}$$

where ρ is the density, u, v and w are the velocity components in the x, y and z directions respectively, T is the temperature, e is the total internal energy per unit volume, μ is the viscosity coefficient, κ is the thermal conductivity and γ is the ratio of specific heats.

3.1.2 Non-Dimensionalization of Governing Equations

A set of non-dimensional variables can be defined from the dimensional quantities by using the reference quantities defined above. In this context the nondimensionalized variables are denoted by using a subscript star(*).

$$x^* = \frac{x}{l_{ref}} \qquad y^* = \frac{y}{l_{ref}} \qquad z^* = \frac{z}{l_{ref}} \tag{3.13}$$

$$u^* = \frac{u}{a_{ref}} \qquad v^* = \frac{v}{a_{ref}} \qquad w^* = \frac{w}{a_{ref}} \tag{3.14}$$

$$\rho^* = \frac{\rho}{\rho_{ref}} \qquad p^* = \frac{p}{\rho_{ref} a_{ref}^{\ 2}} \qquad e^* = \frac{e}{\rho_{ref} a_{ref}^{\ 2}} \tag{3.15}$$

$$H^* = \frac{H}{a_{ref}{}^2} \qquad t^* = \frac{t a_{ref}}{l_{ref}} \qquad \mu^* = \frac{\mu}{\mu_{ref}} \tag{3.16}$$

where ρ_{ref} is the reference density, a_{ref} is the reference speed of sound, l_{ref} is the reference length and μ_{ref} is the reference viscosity coefficient.

As a result of the above selection of non-dimensionalization, the following nondimensional parameters appear in the governing equations:

$$\mathrm{Re}_{ref} = \frac{\rho_{ref} a_{ref} l_{ref}}{\mu_{ref}} \qquad \mathrm{Pr} = \frac{\mu C_p}{\kappa} \tag{3.17}$$

where Re_{ref} is the reference Reynolds number and Pr is the Prandtl number.

Also following relation are used in the above eqations:

$$C_p = \frac{\gamma R}{\gamma - 1} \qquad a_{ref} = \sqrt{\gamma R T_{ref}} \tag{3.18}$$

where C_p is specific heat under constant pressure and R is the gas constant.

Resulting non-dimensional Navier-Stokes equations are given below:

For convenience, the superscript (*) is dropped.

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial K}{\partial z} = \frac{\partial F_v}{\partial x} + \frac{\partial G_v}{\partial y} + \frac{\partial K_v}{\partial z} \tag{3.19}$$

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \quad F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho uH \end{bmatrix} \quad G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho vH \end{bmatrix} \quad K = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho wH \end{bmatrix} \tag{3.20}$$

$$F_v = \frac{1}{\mathrm{Re}_{ref}} \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{xx}u + \tau_{xy}v + \tau_{xz}w - q_x \end{bmatrix} \tag{3.21}$$

$$G_v = \frac{1}{\mathrm{Re}_{ref}} \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ \tau_{yx}u + \tau_{yy}v + \tau_{yz}w - q_y \end{bmatrix} \tag{3.22}$$

$$K_v = \frac{1}{\mathrm{Re}_{ref}} \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ \tau_{zx}u + \tau_{zy}v + \tau_{zz}w - q_z \end{bmatrix} \tag{3.23}$$

The shear stress components are expressed as;

$$\tau_{xx} = \mu\left(\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\left(\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)\right) \quad \tau_{yy} = \mu\left(\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\right)\right) \tag{3.24}$$

$$\tau_{zz} = \mu\left(\frac{4}{3}\frac{\partial w}{\partial z} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right) \qquad \tau_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{3.25}$$

$$\tau_{xz} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \qquad \tau_{yz} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \tag{3.26}$$

$$\tau_{yx} = \tau_{xy} \qquad \tau_{zx} = \tau_{xz} \qquad \tau_{yz} = \tau_{zy} \tag{3.27}$$

Components of heat flux are given as below.

$$q_x = -\frac{\mu}{\Pr}\frac{1}{\gamma - 1}\frac{\partial T}{\partial x} \qquad q_y = -\frac{\mu}{\Pr}\frac{1}{\gamma - 1}\frac{\partial T}{\partial y} \qquad q_z = -\frac{\mu}{\Pr}\frac{1}{\gamma - 1}\frac{\partial T}{\partial z} \tag{3.28}$$

3.1.3 Integral Form of Navier-Stokes Equations

The integral form of the Navier-Stokes equations, which is suitable for numerical calculations is given by;

$$\frac{\partial}{\partial t}\iiint_{\Omega}\vec{Q}\,\mathrm{d}\Omega + \iint_{S}\vec{H}\cdot\vec{n}\,\mathrm{d}S = \iint_{S}\vec{H}_v\cdot\vec{n}\,\mathrm{d}S \tag{3.29}$$

where the column vector $\vec{Q}$ represents the conservative variables, column vector $\vec{H}$ represents the convective flux vector, and column vector $\vec{H}_v$ represents the viscous diffusive flux vector. The definition of convective and viscous diffusive flux vectors are given in equations (3.30) and (3.31).

$$\vec{H} = \vec{F}i + \vec{G}j + \vec{K}k \tag{3.30}$$

$$\vec{H}_v = \vec{F}_v i + \vec{G}_v j + \vec{K}_v k \tag{3.31}$$

3.2 SPATIAL DISCRETIZATION

A cell centred finite volume discretization is applied to equation (3.20), which is in integral form. Time rate of change of conservative variable vector $\vec{Q}$ within a computational domain Ω, is balanced by the net convective and diffusive fluxes across the boundary surface *S*. For this purpose the computational domain is divided into finite number of hybrid/unstructured elements. Each element serves as a computational cell for cell centred approach.

In the finite volume formulation, for a constant control volume, equation (3.29) becomes:

$$\Omega_{\mathrm{i}} \frac{\partial Q_i}{\partial t} + \sum_{j=1}^{N_i} \left[H_n\right]_{i,j} \Delta S_{i,j} - \sum_{j=1}^{N_i} \left[H_{v,n}\right]_{i,j} \Delta S_{i,j} = 0 \tag{3.32}$$

where H_n and $H_{v,n}$ are defined as

$$H_n = \vec{H} \cdot \vec{n} \qquad H_{v,n} = \vec{H}_v \cdot \vec{n} \tag{3.33}$$

3.2.1 Convective Fluxes

Convective fluxes are computed using original Roe's method by following Toro [15]. The left hand side of the cell face is denoted by the subscript "*L*" and the right hand of the cell face is denoted by the subscript "*R*". The structure of the exact solution of the Riemann problem for the x-split three-dimensional Euler equations is given by Toro [15]. The vectors of conserved variables and fluxes are given as:

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \quad \text{and} \quad H_n = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho uH \end{bmatrix} \tag{3.34}$$

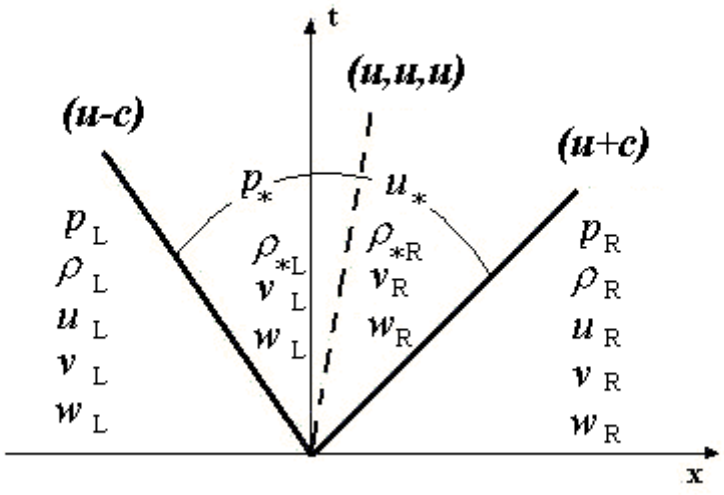

Figure 3.1 Structure of Riemann Problem for x-split Euler Equations

The piece-wise initial data is given as :

$$W_L = \begin{bmatrix} \rho_L \\ u_L \\ v_L \\ w_L \\ p_L \end{bmatrix} \quad \text{and} \quad W_R = \begin{bmatrix} \rho_R \\ u_R \\ v_L \\ w_L \\ p_L \end{bmatrix} \tag{3.35}$$

According to the original Roe's method, the inter-cell flux is given as:

$$H_n = 0.5 \cdot (H_L + H_R) - 0.5 \cdot \sum_{i=1}^{m} \overline{\alpha}_i \left| \overline{\lambda}_i \right| \overline{K}^{(i)} \tag{3.36}$$

The Roe averages of flow variables are given as

$$\overline{u} = \frac{\sqrt{\rho_L} u_L + \sqrt{\rho_R} u_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \tag{3.37}$$

$$\overline{v} = \frac{\sqrt{\rho_L} v_L + \sqrt{\rho_R} v_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \tag{3.38}$$

$$\overline{w} = \frac{\sqrt{\rho_L} w_L + \sqrt{\rho_R} w_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \tag{3.39}$$

$$\overline{h}_0 = \frac{\sqrt{\rho_L} h_{0L} + \sqrt{\rho_R} h_{0R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \tag{3.40}$$

$$\overline{c} = (\gamma - 1)\left(\overline{h}_0 - \frac{1}{2} \overline{V}^2 \right) \tag{3.41}$$

The eigenvalues of x-direction Jacobian matrix in terms of averaged values are

$$\overline{\lambda}_1 = \overline{u} - \overline{c} \quad , \quad \overline{\lambda}_2 = \overline{u} \quad , \quad \overline{\lambda}_3 = \overline{u} \quad , \quad \overline{\lambda}_4 = \overline{u} \quad , \quad \overline{\lambda}_5 = \overline{u} + \overline{c} \tag{3.42}$$

The averaged eigenvectors are

$$\overline{K}^{(1)} = \left[1 \quad \overline{u} - \overline{c} \quad \overline{v} \quad \overline{w} \quad \overline{h}_0 - \overline{u}\,\overline{c}\right]^T \tag{3.43}$$

$$\overline{K}^{(2)} = \left[1 \quad \overline{u} \quad \overline{v} \quad \overline{w} \quad \tfrac{1}{2}\overline{V}^2\right]^T \tag{3.44}$$

$$\overline{K}^{(3)} = \left[0 \quad 0 \quad 1 \quad 0 \quad \overline{v}\right]^T \tag{3.45}$$

$$\overline{K}^{(4)} = \left[0 \quad 0 \quad 0 \quad 1 \quad \overline{w}\right]^T \tag{3.46}$$

$$\overline{K}^{(5)} = \left[1 \quad \overline{u} + \overline{c} \quad \overline{v} \quad \overline{w} \quad \overline{h}_0 + \overline{u}\,\overline{c}\right]^T \tag{3.47}$$

The wave strengths are given as:

$$\overline{\alpha}_3 = \Delta u_3 - \overline{v}\Delta u_1 \tag{3.48}$$

$$\overline{\alpha}_4 = \Delta u_4 - \overline{w}\Delta u_1 \tag{3.49}$$

$$\overline{\alpha}_2 = \frac{\gamma - 1}{\overline{c}^2}\left[\Delta u_1(\overline{h}_0 - \overline{u}^2) + \overline{u}\Delta u_2 - \overline{\Delta u_5}\right] \tag{3.50}$$

$$\overline{\alpha}_1 = \frac{1}{2\overline{c}}\left[\Delta u_1(\overline{u} + \overline{c}) - \Delta u_2 - \overline{c}\,\overline{\alpha}_2\right] \tag{3.51}$$

$$\overline{\alpha}_5 = \Delta u_1 - (\overline{\alpha}_1 + \overline{\alpha}_2) \tag{3.52}$$

where

$$\overline{\Delta u_5} = \Delta u_5 - (\Delta u_3 - \overline{v}\Delta u_1)\overline{v} - (\Delta u_4 - \overline{w}\Delta u_1)\overline{w} \tag{3.53}$$

A modification to the above solution technique is required at sonic flow regions which is called entropy fix technique.

3.2.1.1 Entropy Fix

Linearized solutions to Riemann problem give results in discontinuous jumps which is preferable in shocks and contacts. However for expansion waves, this procedure creates errors. Roe's solver can be modified in order to avoid solutions with entropy violation. The Harten-Hyman entropy fix [21] solution is described in this section.

At each surface, the presence of a rarefaction wave is checked according to the relations given below. If a rarefaction wave is present, the entropy fix process is used instead of the original Roe's method.

The left speeds are calculated as

$$\lambda_1^{\,L} = u_L - c_L \qquad \text{and} \qquad \lambda_1^{\,R} = u_{*L} - c_{*L} \tag{3.54}$$

where

$$\rho_{*L} = \rho_L + \overline{\alpha}_1 \tag{3.55}$$

$$u_{*L} = \frac{\rho_L u_L + \overline{\alpha}_1 (\overline{u} - \overline{c})}{\rho_L + \overline{\alpha}_1} \tag{3.56}$$

$$p_{*L} = (\gamma - 1) \cdot \left[e_{0L} + \overline{\alpha}_1 (\overline{h}_0 - \overline{uc}) - \tfrac{1}{2} \rho_{*L} {u_{*L}}^2 \right] \tag{3.57}$$

$$c_{*L} = \sqrt{\frac{\gamma p_{*L}}{\rho_{*L}}} \tag{3.58}$$

If $\lambda_1^{\,L} < 0 < \lambda_1^{\,R}$ then the left wave is a transonic or a rarefaction wave. The following procedure is followed in such a case.

$$H_n = H_L + \hat{\lambda}_1 \overline{\alpha}_1 \overline{K}^{(1)} \tag{3.59}$$

where

$$\hat{\lambda}_1 = \lambda_1^{\,L} \left(\frac{\lambda_1^{\,R} - \overline{\lambda}_1^{\,L}}{\lambda_1^{\,R} - \lambda_1^{\,L}} \right) \tag{3.60}$$

The right speeds are calculated as

$$\lambda_5^{\,L} = u_{*R} + c_{*R} \qquad \text{and} \qquad \lambda_5^{\,R} = u_R + c_R \tag{3.61}$$

where

$$\rho_{*R} = \rho_R - \overline{\alpha}_5 \tag{3.62}$$

$$u_{*R} = \frac{\rho_R u_R - \overline{\alpha}_5 (\overline{u} + \overline{c})}{\rho_R - \overline{\alpha}_5} \tag{3.63}$$

$$p_{*R} = (\gamma - 1) \cdot \left[e_{0R} - \overline{\alpha}_5 (\overline{h}_0 + \overline{u}\,\overline{c}) - \tfrac{1}{2} \rho_{*R} u_{*R}^{\,2} \right] \tag{3.64}$$

$$c_{*R} = \sqrt{\frac{\gamma p_{*R}}{\rho_{*R}}} \tag{3.65}$$

If $\lambda_5^{\,L} < 0 < \lambda_5^{\,R}$ then the right wave is a transonic or a rarefaction wave. The following procedure is followed in such a case.

$$H_n = H_R - \hat{\lambda}_5 \overline{\alpha}_5 \overline{K}^{(5)} \tag{3.66}$$

where

$$\hat{\lambda}_5 = \lambda_5^{\,R}\left(\frac{\overline{\lambda}_5 - \lambda_5^{\,L}}{\lambda_5^{\,R} - \lambda_5^{\,L}}\right) \tag{3.67}$$

3.2.1.2 First Order Accuracy

Accuracy of the convective fluxes are defined by the approximation used to find values of flow variables at the left and right of the cell face. For a first order approximation, the cell center values are assigned to the cell faces.

3.2.1.3 Second Order Accuracy

Higher order approximation is achieved by assuming that the flow variables change inside the control volume. If the variation of flow variables is linear then the accuracy becomes second order. The method used to achieve second order accuracy is "Piecewise Linear Reconstruction" developed by Barth and Jespersen [22]. In this method, it is assumed that the solution is piecewise linearly distributed over the control volume. The left and right state of the face for a cell centred scheme are found using the relations;

$$Q_L = Q_I + \Psi_I\left(\nabla \vec{Q}_I . \vec{r}_L\right) \tag{3.68}$$

$$Q_R = Q_J + \Psi_J \left(\nabla \vec{Q}_J . \vec{r}_R \right) \tag{3.69}$$

where Q_L and Q_R are the left and right states of the face, respectively, and Q_I and Q_J are the center values for two neighboring cells, respectively. Ψ denotes a limiter function. The vectors $\vec{r}_L$ and $\vec{r}_R$ point from the cell-centroid to the face mid-point.

This method requires the gradient of flow variables at the cell center. In this study, the gradients are calculated by two methods, Green-Gauss and Least-Squares Approach, which are given in detail in the subsequent sections. The computational burden of both approaches are almost the same. However, it was shown by Blazek and Haselbacher [23] that Green-Gauss approach fails on hybrid grids. The gradient can become highly inaccurate, particularly where different element types meet. On the other hand, least-squares approach is shown to be first order accurate on general grids [24].

3.2.1.3.1 Green-Gauss Approach

This method approximates the gradient of flow variables as the surface integral over the control volume. Flow variables at the face are computed by averaging of two neighboring cells who share the face in consideration. Gradient at the cell center is calculated as,

$$\nabla \vec{Q} \approx \frac{1}{2} \sum_{j=1}^{N_i} \left(Q_i + Q_j \right) \vec{n}_{i,j} \Delta S_{i,j} \tag{3.70}$$

where Q is any flow variable, $\vec{n}_{i,j}$ is the unit normal vector of cell face and $\Delta S_{i,j}$ is the face area. The integration is carried out by looping over all faces of the control volume.

3.2.1.3.2 Least-Squares Approach

The evaluation of gradients by the least-squares approach was first introduced by Barth [25]. The least squares approach is based upon the use of a first-order Taylor series approximation for each cell center which is a neighbor of cell whose gradients are to be calculated. The change of the solution along the line connecting two cell centers can be computed from

$$\nabla \vec{Q}_I . \vec{r}_{IJ} = Q_J - Q_I \tag{3.71}$$

where $\vec{r}_{IJ}$ represents the vector from center of cell *I* to center of cell *J*. When the above relation is applied to all neighboring cells of cell *I*, the following over-constrained system of linear equations is obtained.

$$\begin{bmatrix} \Delta x_{I1} & \Delta y_{I1} & \Delta z_{I1} \\ \vdots & \vdots & \vdots \\ \Delta x_{IJ} & \Delta y_{IJ} & \Delta z_{IJ} \\ \vdots & \vdots & \vdots \\ \Delta x_{IN_A} & \Delta y_{IN_A} & \Delta z_{IN_A} \end{bmatrix} \begin{bmatrix} \dfrac{\partial Q}{\partial x} \\ \dfrac{\partial Q}{\partial y} \\ \dfrac{\partial Q}{\partial z} \end{bmatrix}_I = \begin{bmatrix} Q_1 - Q_I \\ \vdots \\ Q_J - Q_I \\ \vdots \\ Q_{N_A} - Q_I \end{bmatrix} \tag{3.72}$$

with $\Delta(\circ)_{IJ} = (\circ)_J - (\circ)_I$. Further N_A denotes the number of neighbor cells of the cell *I*. The above system can be written as,

$$\overline{A}\vec{x} = \vec{b} \tag{3.73}$$

Solving equation (3.73) for the gradient vector $\vec{x}$ requires the inversion of the matrix $\overline{A}$. To prevent problems with ill-conditioning, Anderson and Bonhaus suggested to decompose $\overline{A}$ into the product of an orthogonal matrix $\overline{Q}$ and an upper triangular matrix $\overline{R}$ using the Gram-Schmidt process [26]. Hence, solution to equation (3.73) follows as

$$\vec{x} = \overline{R}^{-1}\overline{Q}^{T}\vec{b} \tag{3.74}$$

If equation (3.74) is written explicitly, the following equation is obtained for the gradient values.

$$\vec{\nabla}Q_I = \sum_{J=1}^{N_A} \vec{\omega}_{IJ}\left(Q_J - Q_I\right) \tag{3.75}$$

with the vector of weights $\vec{\omega}_{IJ}$ defined as,

$$\vec{\omega}_{IJ} = \begin{bmatrix} \alpha_{IJ,1} - \dfrac{r_{12}}{r_{11}}\alpha_{IJ,2} + \beta\alpha_{IJ,3} \\ \alpha_{IJ,2} - \dfrac{r_{23}}{r_{22}}\alpha_{IJ,3} \\ \alpha_{IJ,3} \end{bmatrix} \tag{3.76}$$

The terms in equation (3.76) are given by

$$\begin{aligned} \alpha_{IJ,1} &= \frac{\Delta x_{IJ}}{r_{11}^{\,2}} \\ \alpha_{IJ,2} &= \frac{1}{r_{22}^{\,2}}\left(\Delta y_{IJ} - \frac{r_{12}}{r_{11}}\Delta x_{IJ}\right) \\ \alpha_{IJ,3} &= \frac{1}{r_{33}^{\,2}}\left(\Delta z_{IJ} - \frac{r_{23}}{r_{22}}\Delta y_{IJ} + \beta\Delta x_{IJ}\right) \\ \beta &= \frac{r_{12}r_{23} - r_{13}r_{22}}{r_{11}r_{22}} \end{aligned} \tag{3.77}$$

Also, the below matrix entries are calculated and stored for each cell in the domain.

$$
\begin{aligned}
r_{11} &= \sqrt{\sum_{J=1}^{N_N} \left(\Delta x_{IJ}\right)^2} \\
r_{12} &= \frac{1}{r_{11}} \sum_{J=1}^{N_N} \Delta x_{IJ} \Delta y_{IJ} \\
r_{22} &= \sqrt{\sum_{J=1}^{N_N} \left(\Delta y_{IJ}\right)^2 - {r_{12}}^2} \\
r_{13} &= \frac{1}{r_{11}} \sum_{J=1}^{N_N} \Delta x_{IJ} \Delta z_{IJ} \\
r_{23} &= \frac{1}{r_{22}} \left(\sum_{J=1}^{N_N} \Delta y_{IJ} \Delta z_{IJ} - \frac{r_{12}}{r_{11}} \sum_{J=1}^{N_N} \Delta x_{IJ} \Delta z_{IJ} \right) \\
r_{33} &= \sqrt{\sum_{J=1}^{N_N} \left(\Delta z_{IJ}\right)^2 - \left({r_{13}}^2 + {r_{23}}^2\right)}
\end{aligned}
\tag{3.78}
$$

3.2.1.3.3 Limiter Function

Second and higher order upwind spatial discretizations require the use of limiter functions in order to prevent the generation of oscillations and spurious solutions in regions of high gradients, such as shocks. By using limiter functions, the scheme becomes monotonicity preserving. In other words, maxima in the flow field must be non-increasing, minima non-decreasing, and no new local extrema should be created during the advancement of the solution in time.

On unstructured and hybrid grids, the aim of a limiter is to reduce the gradient used to reconstruct the left and right state at the face of the control volume. The limiter function must be zero at strong discontinuities, in order to obtain a first order upwind scheme which guarantees monotonicity. Also, the original unlimited reconstruction has to be preserved in smooth flow regions to keep the amount of numerical dissipation as low as possible.

As the limiter function Venkatakrishnan's [27,28] limiter is used, which is explained below in detail. The limiter reduces the reconstructed gradient $\vec{\nabla}Q_I$ at the center of cell *I* by the factor

$$\Psi_I = \min_J \begin{cases} \frac{1}{\Delta_2}\left[\frac{\left(\Delta^2_{1,\max}+\varepsilon^2\right)\Delta_2 + 2\Delta^2{}_2\Delta_{1,\max}}{\Delta^2_{1,\max}+2\Delta^2{}_2+\Delta_{1,\max}\Delta_2+\varepsilon^2}\right] & \Delta_2 > 0 \\ \frac{1}{\Delta_2}\left[\frac{\left(\Delta^2_{1,\min}+\varepsilon^2\right)\Delta_2 + 2\Delta^2{}_2\Delta_{1,\min}}{\Delta^2_{1,\min}+2\Delta^2{}_2+\Delta_{1,\min}\Delta_2+\varepsilon^2}\right] & \Delta_2 < 0 \\ 1 & \Delta_2 = 0 \end{cases} \tag{3.79}$$

where

$$\Delta_{1,\max} = Q_{\max} - Q_I \tag{3.80}$$

$$\Delta_{1,\min} = Q_{\min} - Q_I \tag{3.81}$$

In the equations (3.80) and (3.81), $Q_{\max}$ and $Q_{\min}$ stand for the maximum / minimum values of all surrounding cells including the cell *I*. Definition of Δ_2 is given in equation (3.82) as

$$\Delta_2 = \vec{\nabla}Q.\vec{r}_L \tag{3.82}$$

where $\vec{r}_L$ is given as the vector from the cell centroid to the midpoint of the corresponding cell face.

The parameter ε^2 is intended to control the amount of limiting. In practice, ε^2 is taken as proportional to a local length scale;

$$\varepsilon^2 = \left(\kappa\Delta h\right)^3 \tag{3.83}$$

where κ is a constant and Δh is the cube-root of the volume of the cell.

3.2.2 Diffusive Fluxes

To compute the diffusive fluxes, the gradient of the velocity vector is needed. Since, the gradients of flow variables are calculated in the Green-Gauss and Least-Squares methods, to achieve second order accuracy in space, the velocity gradient is not calculated separately, instead, velocity gradient is calculated from gradients of flow variables using the chain rule. A sampe calculation procedure is given by equation (3.84).

$$\frac{\partial(\rho u)}{\partial x} = \rho \frac{\partial u}{\partial x} + u \frac{\partial \rho}{\partial x} \tag{3.84}$$

Temperature gradient is also required for the calculation of diffusive fluxes Temperature gradient is handled by Green-Gauss and Least Squares methods.

Calculated velocity and temperature gradients are calculated at the cell center, however, since gradients at the cell faces are needed, the gradients at the cell center are extrapolated to the cell faces. For all boundary conditions, except wall boundary condition, velocity and temperature gradients at the cell faces are taken as the same as the gradients at the cell center.

For a no-slip wall boundary condition, gradients at the cell faces are taken as the same as the cell center values. However, the temperature gradient in the normal direction of the cell face is taken as zero, since adiabatic wall assumption is made.

3.3 TIME DISCRETIZATION

Time discretization in the present solver is handled by two different approaches. The solver can integrate governing equations in time by either explicitly or implicitly. For the explicit time stepping, Runge-Kutta Multi-Stage scheme is used, whereas, for the implicit time stepping Lower-Upper Symmetric Gauss-Seidel approach is used. Implementation of both methods are given in oncoming sections in detail.

3.3.1 Explicit Time Stepping

If governing equations are discretized in time, equation (3.32) takes the following form,

$$\Omega_i \frac{Q_i^{n+1} - Q_i^n}{\Delta t} = R_i^n \tag{3.85}$$

where residual R_i^n is defined as,

$$R_i^n = \sum_{j=1}^{N_i} \left[H_n\right]^n{}_{i,j} \Delta S_{i,j} - \sum_{j=1}^{N_i} \left[H_{v,n}\right]^n{}_{i,j} \Delta S_{i,j} \tag{3.86}$$

After rearranging, equation (3.85) becomes,

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Omega_i} R_i^n \tag{3.87}$$

The solution at time step *n+1* can be evaluated starting from the initial data at time step *n* by using an explicit algorithm through Runge-Kutta (RK) Method [17]. The general RK scheme of order *m* is given by:

$$
\begin{aligned}
&Q^{(0)} = Q^{n} \\
&Q^{(k)} = Q^{(0)} - \alpha_k \frac{\Delta t}{\Omega} R(Q^{(k-1)}) \qquad k = 1,2..m \\
&\mathrm{Q}^{(\mathrm{n}+1)} = Q^{(m)}
\end{aligned} \tag{3.88}
$$

where the constants are given as $0 < \alpha_k < 1$ and $\alpha_m = 1$.

In the present flow solver, fourth order scheme is used with the standard constants

$$\alpha_1 = 1/4 \;\; , \;\; \alpha_2 = 1/3 \;\; , \;\; \alpha_3 = 1/2 \;\; , \;\; \alpha_4 = 1 \tag{3.89}$$

where CFL is the Courant number coefficient and defined as follows

$$CFL = \frac{\Delta t}{\Delta x} V^{n}{}_{\max} \tag{3.90}$$

$V^{n}{}_{\max}$ is defined as the maximum wave speed throughout the domain at time level n for each cell.

3.3.2 Implicit Time Stepping

Implicit method used in this thesis is Lower-Upper Symmetric Gauss Seidel approximate factorization scheme, which is developed by Jameson and Yoon [20]. Later this method has been extended and used for hybrid and mixed grids[29-33]. Application of this method for unstructured grids is given in detail below.

For convenience, the governing equations are repeated here, as

$$\Omega_{\mathrm{i}} \frac{\partial Q_i}{\partial t} + \sum_{j=1}^{N_i} \left[H_n \right]_{i,j} \Delta S_{i,j} - \sum_{j=1}^{N_i} \left[H_{v,n} \right]_{i,j} \Delta S_{i,j} = 0 \tag{3.91}$$

Applying backward Euler scheme for time integration, the following equation is obtained;

$$\frac{\Omega_i}{\Delta t}\left(Q_i^{n+1} - Q_i^{n}\right) + \sum_{j=1}^{N_i}\left[H_n\right]^{n+1}_{i,j}\Delta S_{i,j} - \sum_{j=1}^{N_i}\left[H_{v,n}\right]^{n+1}_{i,j}\Delta S_{i,j} = 0 \tag{3.92}$$

It should be noted that convective and diffusive fluxes are calculated at time level $n+1$. However, the face values required to calculate fluxes at time level $n+1$ are not known. For this reason, the flux values at time level $n+1$ are linearized about the time level n.

$$\begin{aligned}\sum_{j=1}^{N_i}\left[H_n\right]^{n+1}_{i,j}\Delta S_{i,j} - \sum_{j=1}^{N_i}\left[H_{v,n}\right]^{n+1}_{i,j}\Delta S_{i,j} = \sum_{j=1}^{N_i}\left[H_n\right]^{n}_{i,j}\Delta S_{i,j} - \sum_{j=1}^{N_i}\left[H_{v,n}\right]^{n}_{i,j}\Delta S_{i,j} \\ + \left(\sum_{j=1}^{N_i}\Delta F_{ij}{}^{n}\Delta S_{i,j} - \sum_{j=1}^{N_i}\Delta F_{v,ij}{}^{n}\Delta S_{i,j}\right)\end{aligned} \tag{3.93}$$

After the linearization of fluxes, equation (3.92) becomes,

$$\frac{\Omega_i}{\Delta t}\left(\Delta Q_i^{n}\right) + \left(\sum_{j=1}^{N_i}\Delta F_{ij}{}^{n} - \sum_{j=1}^{N_i}\Delta F_{v,ij}{}^{n}\right) = -R_i^{n} \tag{3.94}$$

where residual R_i^n and ΔQ_i^n are defined by equations (3.86) and (3.95), respectively.

$$\Delta Q_i^{n} = Q_i^{n+1} - Q_i^{n} \tag{3.95}$$

In equation (3.94), $\Delta F_{ij}{}^{n}$ and $\Delta F_{ij,v}{}^{n}$ are differences in convective and diffusive fluxes between time level $n+1$ and n at each face. These flux differences can be approximated considering that these flux values depend both on cell i and j.

$$\begin{aligned}\Delta F_{ij}^{\,n} - \Delta F_{ij,v}^{\,n} = &\left[F\left(Q_i^{\,n+1}, Q_j^{\,n+1}\right) - F\left(Q_i^{\,n}, Q_j^{\,n+1}\right)\right] \\ &+ \left[F\left(Q_i^{\,n}, Q_j^{\,n+1}\right) - F\left(Q_i^{\,n}, Q_j^{\,n}\right)\right] \\ &- \left[F_v\left(Q_i^{\,n+1}, Q_j^{\,n+1}\right) - F_v\left(Q_i^{\,n}, Q_j^{\,n+1}\right)\right] \\ &\left[F_v\left(Q_i^{\,n}, Q_j^{\,n+1}\right) - F_v\left(Q_i^{\,n}, Q_j^{\,n}\right)\right]\end{aligned} \tag{3.96}$$

Linearizing the first and third terms on the right hand side of equation (3.96), it is possible to obtain

$$F\left(Q_i^{\,n+1}, Q_j^{\,n+1}\right) - F\left(Q_i^{\,n}, Q_j^{\,n+1}\right) = \frac{\partial F_{ij}}{\partial Q_i}\Delta Q_i^{\,n} \tag{3.97}$$

$$F_v\left(Q_i^{\,n+1}, Q_j^{\,n+1}\right) - F_v\left(Q_i^{\,n}, Q_j^{\,n+1}\right) = \frac{\partial F_{v,ij}}{\partial Q_i}\Delta Q_i^{\,n} \tag{3.98}$$

Substituting equations (3.97) and (3.98) back into equation (3.94), one can get

$$\begin{aligned}&D\Delta Q_i^{\,n} + \sum_{j=1}^{N_i}\left[F\left(Q_i^{\,n}, Q_j^{\,n} + \Delta Q_j^{\,n}\right) - F\left(Q_i^{\,n}, Q_j^{\,n}\right)\right]\Delta S_{i,j} \\ &- \sum_{j=1}^{N_i}\left[F_v\left(Q_i^{\,n}, Q_j^{\,n} + \Delta Q_j^{\,n}\right) - F_v\left(Q_i^{\,n}, Q_j^{\,n}\right)\right]\Delta S_{i,j} = -R_i^{\,n}\end{aligned} \tag{3.99}$$

where diagonal matrix D is given by equation (3.100) , and I is the identity matrix.

$$D = \frac{\Omega_i}{\Delta t} I + \sum_{j=1}^{N_i}\left(\frac{\partial F_{ij}}{\partial Q_i} - \frac{\partial F_{v,ij}}{\partial Q_i}\right)\Delta S_{i,j} \tag{3.100}$$

In equation (3.99), the residual, $R_i^{\,n}$,on the right hand side is calculated using Roe's approximate Riemann solver. However, convective and diffusive fluxes on the left hand side are calculated by a first order approximation.

The first order numerical fluxes in the left hand side of equation (3.99) are chosen as,

$$F_{ij} - F_{v,ij} = \frac{1}{2}\left[H_n(Q_i) + H_n(Q_j) - \lambda_{ij}(Q_j - Q_i)\right] \tag{3.101}$$

where λ_{ij} is the spectral radius of flux Jacobian matrix at the cell face and defined as,

$$\lambda_{ij} = \vec{V}.\vec{n}_{ij} + a + \frac{2\mu}{\rho\left|\vec{n}_{ij}.\vec{r}_{ji}\right|} \tag{3.102}$$

In equation, $\vec{n}_{ij}$ is the normal unit vector of the face, $\vec{V}$ is the velocity vector, a is the speed of sound, μ is the kinematic viscosity and $\vec{r}_{ji}$ is the vector pointing from cell center of i to the cell center of neighbor j.

Because each control volume is closed, for cell i one can obtain,

$$\sum_{j=1}^{N_i}\left(\frac{\partial H_{n,i}}{\partial Q_i}\right)\Delta S_{i,j} = 0 \tag{3.103}$$

Substituting equation (3.103) into equation (3.100) and using equations (3.101) and (3.102), diagonal matrix becomes an identity matrix with a scale factor,

$$D = \left(\frac{\Omega_i}{\Delta t} + \frac{1}{2}\sum_{j=1}^{N_i}\lambda_{ij}\Delta S_{i,j}\right)I \tag{3.104}$$

When the approximation given in equation (3.101) for numerical fluxes are applied to equation (3.99), final form of the scheme is obtained as

$$D\Delta Q_i^{\,n} + \frac{1}{2}\sum_{j=1}^{N_i}\left[H_n\left(Q_j^{\,n} + \Delta Q_j^{\,n}\right) - H_n\left(Q_j^{\,n}\right) - \lambda_{ij}\Delta Q_j^{\,n}\right]\Delta S_{i,j} = -R_i^{\,n} \tag{3.105}$$

Equation (3.105) is solved using one sweep of symmetric Gauss-Seidel iteration as follows.

Forward Sweep

$$D\Delta Q_i^* = -R_i^n - \frac{1}{2}\sum_{j\in L(i)}\left[H_n\left(Q_j^n + \Delta Q_j^*\right) - H_n\left(Q_j^n\right) - \lambda_{ij}\Delta Q_j^*\right]\Delta S_{i,j} \tag{3.106}$$

Backward Sweep

$$\Delta Q_i^n = \Delta Q_i^* - \frac{1}{2}D^{-1}\sum_{j\in U(i)}\left[H_n\left(Q_j^n + \Delta Q_j^n\right) - H_n\left(Q_j^n\right) - \lambda_{ij}\Delta Q_j^n\right]\Delta S_{i,j} \tag{3.107}$$

where $L(i)$ and $U(i)$ represent the lower and upper neighbor cells of cell i according to the cell ordering.

3.4 BOUNDARY CONDITIONS

3.4.1 Injection Boundary Condition

At the injecting surface, pressure gradient is assumed to be zero. The injected gas density is found through the solution of relations given below.

$$T_o = T + \frac{V^2}{2C_p} \tag{3.108}$$

$$T_o = \frac{p}{\rho R} + \frac{(G/\rho)^2}{2C_p} \tag{3.109}$$

where T_o is the flame temperature and G is the injection mass flux given as input data. Once the density is calculated, momentum flux terms can be calculated. The energy flux is calculated by using the flame temperature.

3.4.2 Symmetry Boundary Condition

The normal flux term passing through the surface is set to zero whereas the gradient of other flux terms are set to zero for the symmetry boundary condition. This is achieved through ghost cells on the symmetry boundary where flux terms tangent to the surface are kept as they are and the mirror image of the normal terms are taken.

3.4.3 Wall Boundary Condition

For an inviscid flow, all the flux terms passing through the surface are set to zero, whereas the tangential flux terms are kept as they are. For a viscous flow, all the flux terms passing through the surface are set to zero.

3.4.4 Subsonic Outflow Boundary Condition

Subsonic flow should reach the ambient pressure as the code converges. Therefore pressure value at the boundary is enforced by the ambient pressure. The velocity and temperature gradients are normal to the surface are kept zero.

$$\frac{\partial \vec{V}}{\partial \vec{n}} = 0 \text{ and } \frac{\partial T}{\partial \vec{n}} = 0 \tag{3.110}$$

The density value at the boundary should be computed at the ambient pressure and temperature.

3.4.5 Supersonic Outflow Condition

Supersonic flows do not interact with outflow conditions. Thus, all the exit flow parameters are extrapolated from interior points.

3.4.6 Far-Field Boundary Condition

For the ghost cells on the boundary, depending on the direction and speed of the flow, the ghost cell values are extrapolated from the interior and freestream conditions. If the flow is supersonic and entering the domain all ghost flow variables are taken as freestream conditions. If the flow is subsonic and entering into the domain, pressure is interpolated from the interior and all other variables are taken from freestream conditions. When flow is supersonic and leaving the domain all ghost cell variables are taken from the interior. If the flow is subsonic and leaving the domain, pressure is taken as the freestream pressure and all other variables are extrapolated from the interior.

CHAPTER 4

VALIDATION

In this section, results obtained for the validation of the numerical algorithm will be presented. The discretization accuracy of viscous flux vector is tested by laminar flow over a flat plate as the first test case.The numerical algorithm for the solution of full Navier-Stokes equations has no directional dependence therefore a purely two-dimensional problem for which the analytical Blasius solution can be used for comparison.

To assess the accuracy improvement switching from the first order to second order spatial accuracy, a subsonic internal flow is selected which has an exact solution and results of a validated two-dimensional solver, IBS2D. This test case is also used for the comparison of gradient calculation methods, Green-Gauss and Least-Sqaures methods for structured and unstructured grids.

The third test case is used to show that the numerical algorithm is working properly for what it is aimed, which is an rocket internal flow. The improvement in accuracy and the sensitivity of the solution to different gradient calculation methods are also discussed.

Although, the solver in intended to be used for internal flows, the last test case is chosen as an external flow over an airfoil. For this test case, the accuracy improvement is investigated together with the solution sensitivity to the limiter function parameters.

4.1 LAMINAR FLOW OVER A FLAT PLATE

To verify the Navier-Stokes solver, primarily the two dimensional laminar flat plate problem is solved. The freestream Reynolds number of the flow is 35000 and the freestream Mach number is 0.3. For such flows, the external velocity is independent from the streamwise space variable and such flows are mentioned as constant pressure flows. A grid of 121x81x5 points (5 chords length, 3 chords height and half chord depth) is used. There are 58320 nodes and 47600 cells. The first Δy is 0.0001 and exponentionally grows towards the top boundary. There are approximately 25-30 grid points in the boundary layer. At the leading edge cells are clustered algebraically with the first Δx value being 0.0003. This clustering is necessary for resolving the streamwise pressure gradient that exists only at the leading edge. There are 55 points upstream of the leading edge, 52 points on the plate surface and 13 points in the wake region.

Implicit time stepping with both first and second order solutions are performed. For the second order solution, the Green-Gauss method is used in the calculation of gradients flow variables. A CFL number of 10^6 is used for both solutions. Although the domain is three-dimensional, two-dimensional representation of the grid is given in Figure 4.1, since the grid does not show any variation in the third dimension.

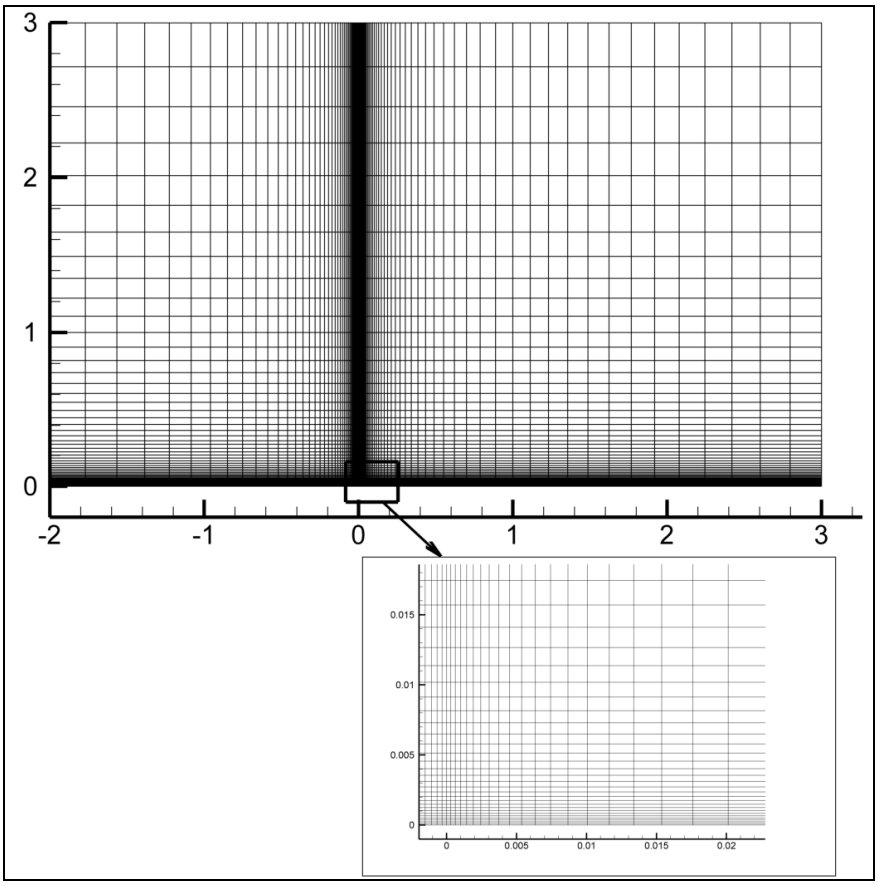

Figure 4.1 Grid of Laminar Flat Plate Test Case

Similarity parameter $\eta = y\sqrt{\frac{U_\infty}{\nu x}}$ is used for comparison with the analytical solution of streamwise velocity. First and second order accurate laminar numerical solution at $\frac{x}{L} = 0.5$ compares well with self similar Blasius solution [34] results which are given in Figure 4.2. Moreover, the improvement in the comformance to the analytical solution, for the second order numerical solution is easily seen. A comparison of local skin friction C_f with the finite length expression is given by [34];

$$C_f = 0.664\,\mathrm{Re}_x^{-0.5} + 1.334\,\mathrm{Re}_x^{-0.875} \tag{4.1}$$

$$\mathrm{Re}_x = \frac{\rho U_\infty x}{\mu} \tag{4.2}$$

showed that the solver is successful in computing the analytical results as shown in Figure 4.3. In this figure, it can be seen that the second order solution gives more accurate results than first order solution at the trailing edge of the flat plate.

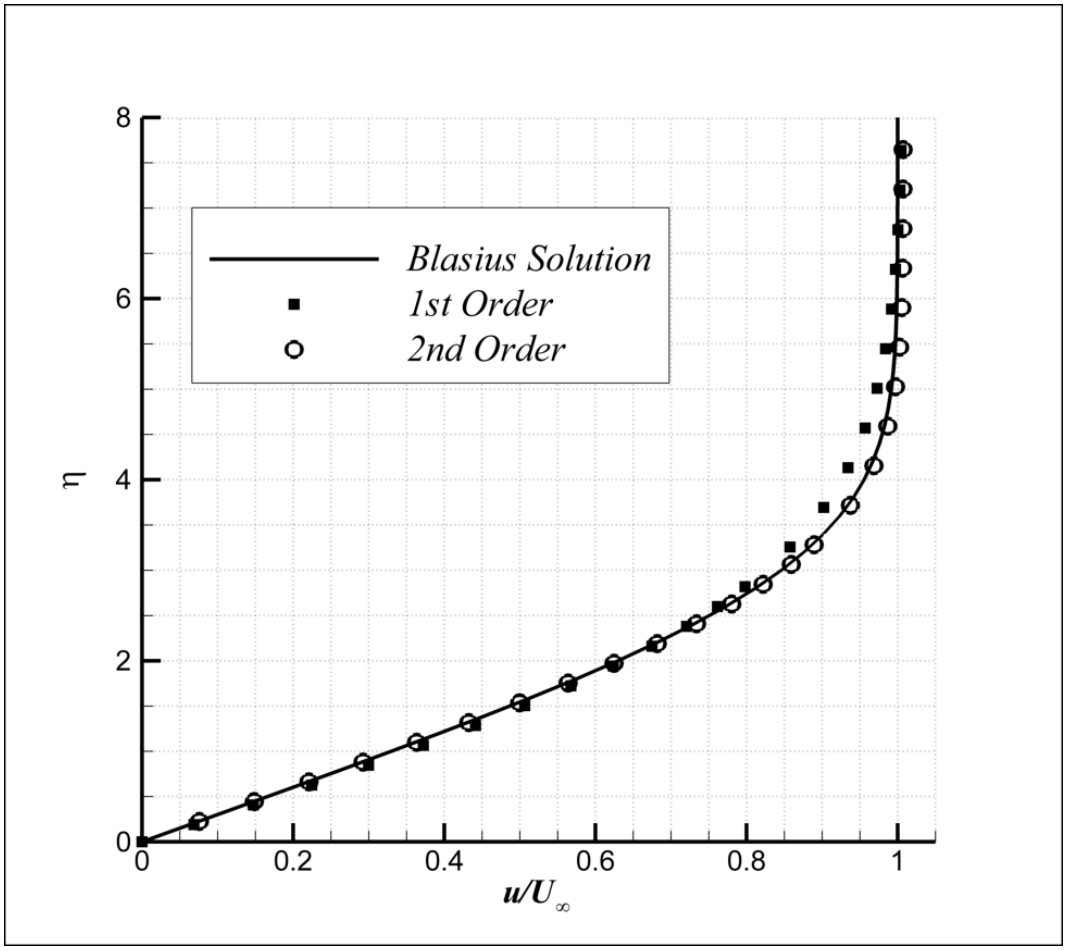

Figure 4.2 Comparison of Axial Velocity Component Results with Blasius Solution.

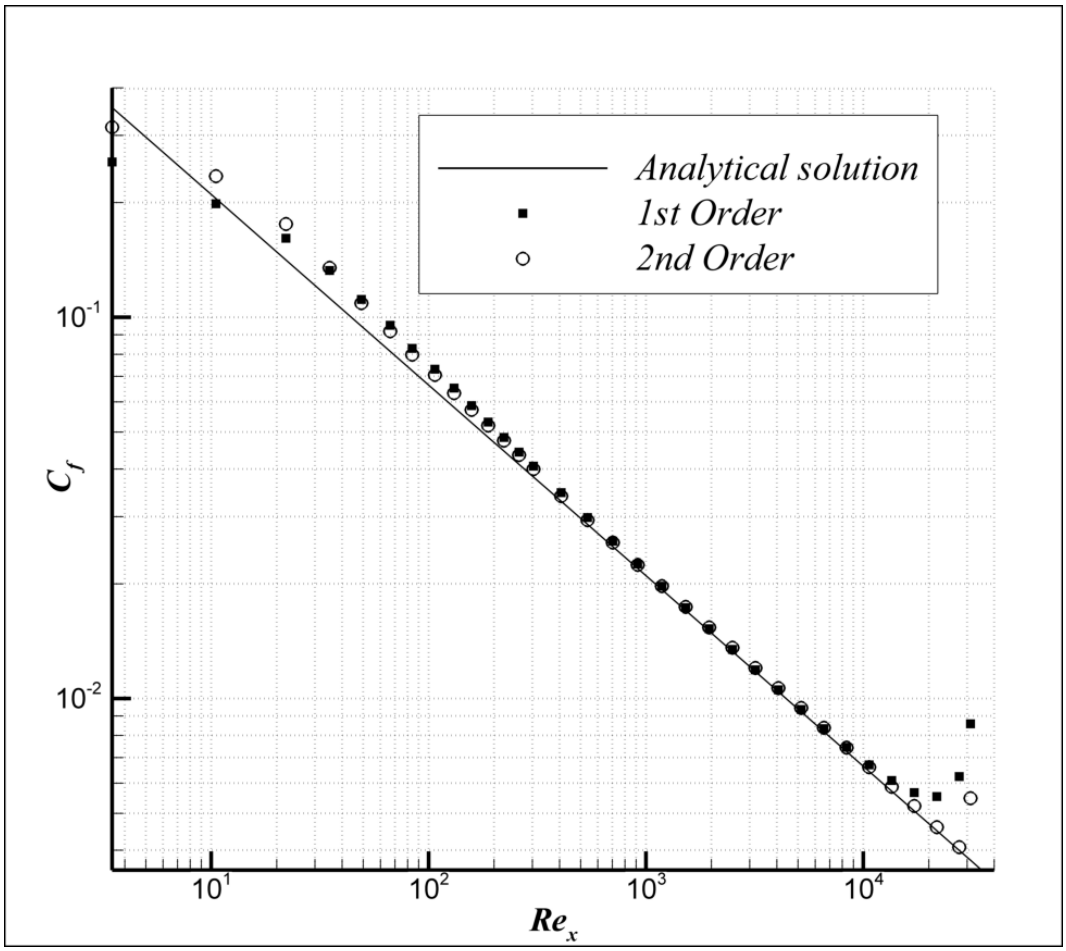

Figure 4.3 Comparison of Skin Friction Coefficient Results with Analytical Solution.

Agreement between Blasius solution and the velocity profile is at an acceptable level. Also, good conformance of local skin friction coefficient with the analytical solution is obtained. These results validate the developed Navier-Stokes solution method applied in the solver.

4.2 T-108 TEST CASE 1A

T-108 project [35] is aimed to develop and validate two-dimensional Euler solvers. In this project, two-dimensional Euler solver (IBS2D) developed by Yumuşak [35] is validated. Test case 1A is a planar injection problem where the flow is fully subsonic. For this test case, an exact solution is available [36]. IBS2D is second order accurate in space. The results obtained will be compared to the exact solution and the results of IBS2D.

First and second order accurate solutions with a structured grid were performed to see the effect of second order accurate method applied in the solver. For this test case, Green-Gauss method is used to calculate the gradients of flow variables.. Mach number distribution for first and second order solutions and results of IBS2D are investigated.

The geometry and the boundary conditions are given in Figure 4.4. The physical and geometrical parameters used for this test case are shown in Table 4.1.

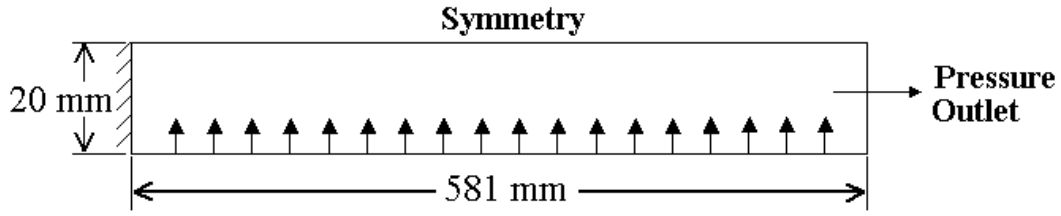

Figure 4.4 Geometry of Test Case 1

	2-D Planar	3-D Solver
Geometric Parameters		
Length	581 mm	581 mm
Height	20 mm	20 mm
Width	-	20 mm
Grid Size	51 x 16	51 x 16 x 16
Flow Parameters		
Mass flux	2.42 kg/m^2/s	
Exit Pressure	150000 Pa	
Gamma	1.4	
Gas Constant	286.7 J/kgK	
Reference pressure	100000 Pa	
Flame Temperature	303 K	

Table 4.1 Geometric and Flow Parameters for Test Case 1

The exact solution for this test case is given in Figure 4.5 and by equations (4.3) and (4.4).

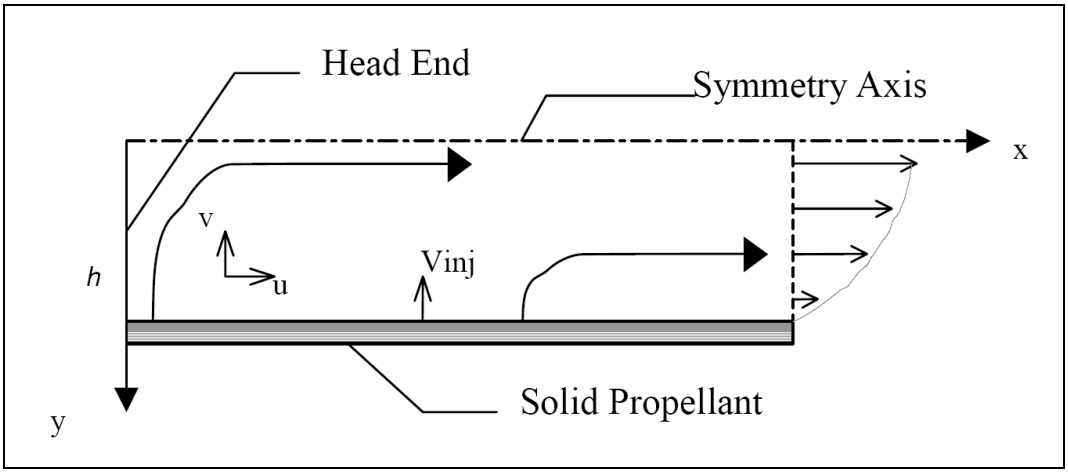

Figure 4.5 Definitions for the Exact Solution for Test Case 1

$$u = \frac{\pi}{2}\frac{x}{h}\left|v_{inj}\right|\cos\left(\frac{\pi}{2}\frac{y}{h}\right) \tag{4.3}$$

$$v = -v_{inj}\sin\left(\frac{\pi}{2}\frac{y}{h}\right) \tag{4.4}$$

The grids used by IBS2D and the present solver are given in Figure 4.6 and Figure 4.7.

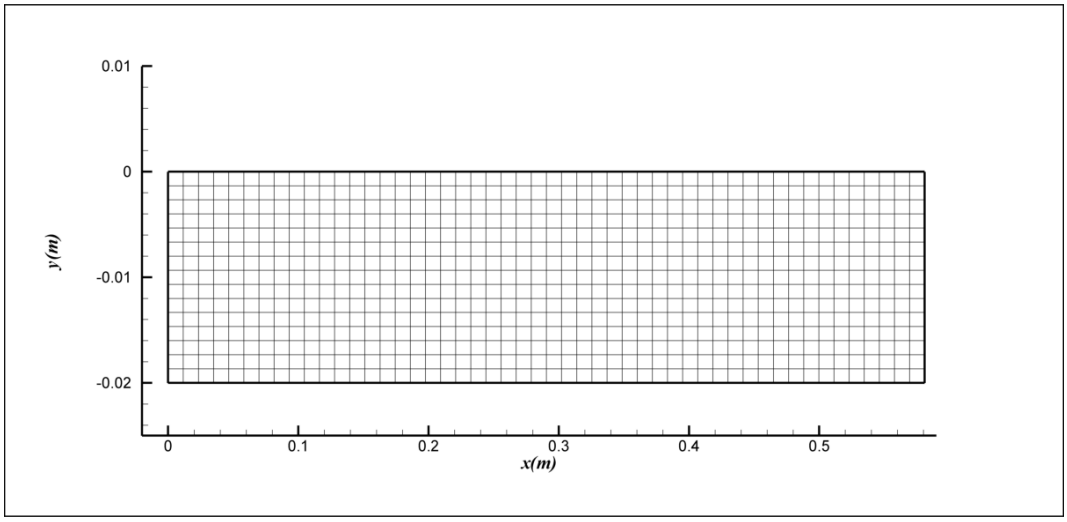

Figure 4.6 Two-Dimensional grid of Test Case 1 (IBS2D)

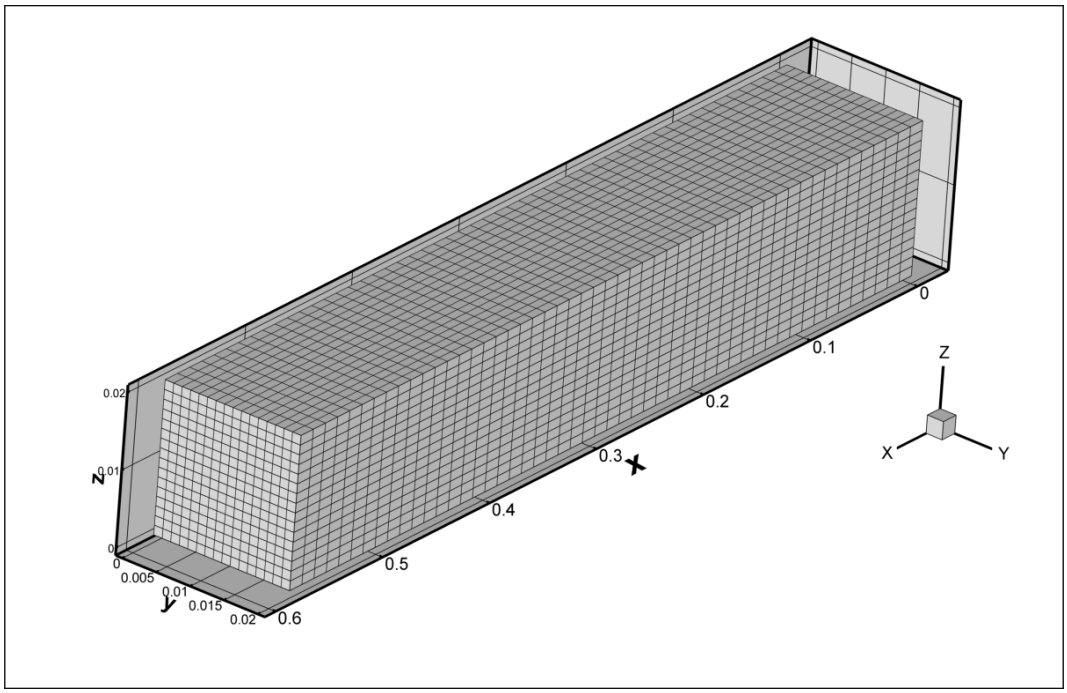

Figure 4.7 Three-Dimensional grid of Test Case 1 (Present Solver)

When Figure 4.8 is investigated, the accuracy improvement in the second order solution is evident. Athough, first and second order solutions give the same results for very low Mach numbers, as the Mach number increases, the second order solution matches with the solutions of IBS2D which is a second order accurate solver.

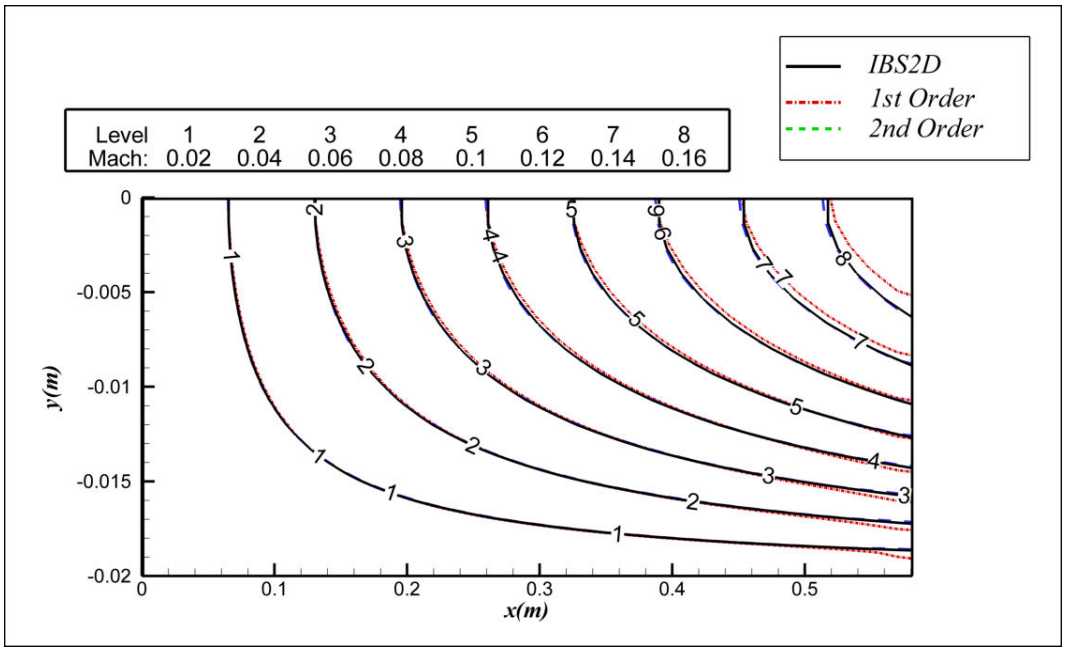

Figure 4.8 Test Case 1 Mach Number Contours

In Figure 4.9, static pressure at the symmetry axis is given for IBS2D and first and second order solutions for the present solver. The pressure prediction of second order solution, closely resembles the IBS2D solution. Although the difference in the pressure prediction between the first and second order solutions seems to be very small for this test case, the flow is fully subsonic and maximum Mach number is low. For a real rocket motor internal flow, the difference in pressure estimation may be more pronounced.

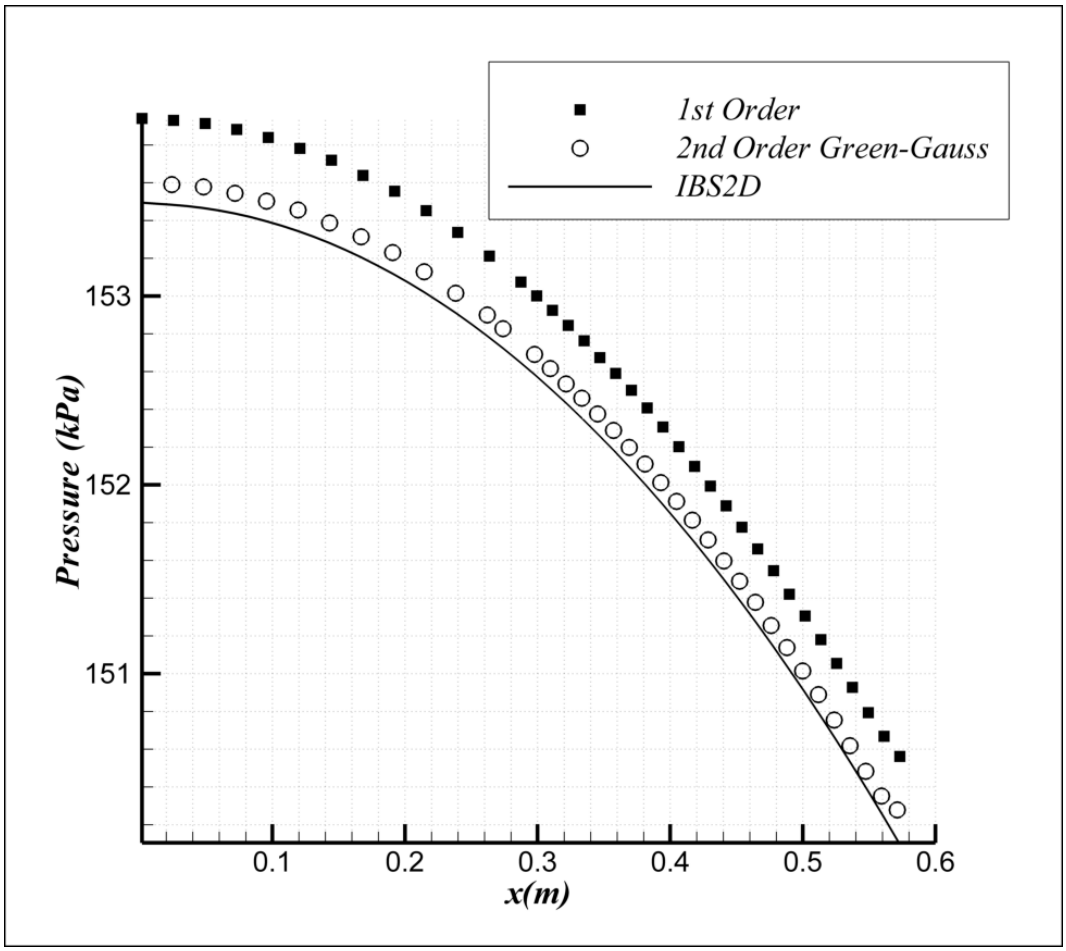

Figure 4.9 Static Pressure at the Symmetry Axis

4.3 T-108 TEST CASE 2

Test case 2 of T-108 project is an axisymmetric mini solid rocket motor problem where the flow is subsonic to supersonic, which is a characteristic of a rocket motor.

For this test case, first of all, a comparison is made between the explicit and implicit time stepping methods used in the solver. To reveal the pure effect of the time stepping methods on the performance of the solver, a first order inviscid solution is performed with the explicit Runge-Kutta method and implicit Lower-Upper Symmetric Gauss-Seidel method. Total iterations and time to reach

convergence are compared, to find the speed up of the solver when switching from explicit to implicit time stepping. This test case is selected because the solver is intended for rocket motor internal flows. The results will be compared to the IBS2D solution.

Second order inviscid solutions are performed to distinguish the accuracy gain with two gradient calculation methods, GG and LSQ. The results are compared with the IBS2D solution.

In order to compare results with axi-symmetric two-dimensional results of IBS2D, a three-dimensional grid with tetrahedron cells are generated. Three-dimensional results are given as a two-dimensional slice at the *x*-*y* plane so that a comparison can be made. The graphical representation of grids, which are used by IBS2D and the present solver, are given in Figure 4.10. The physical properties of the flow and grid parameters are given in Table 4.2.

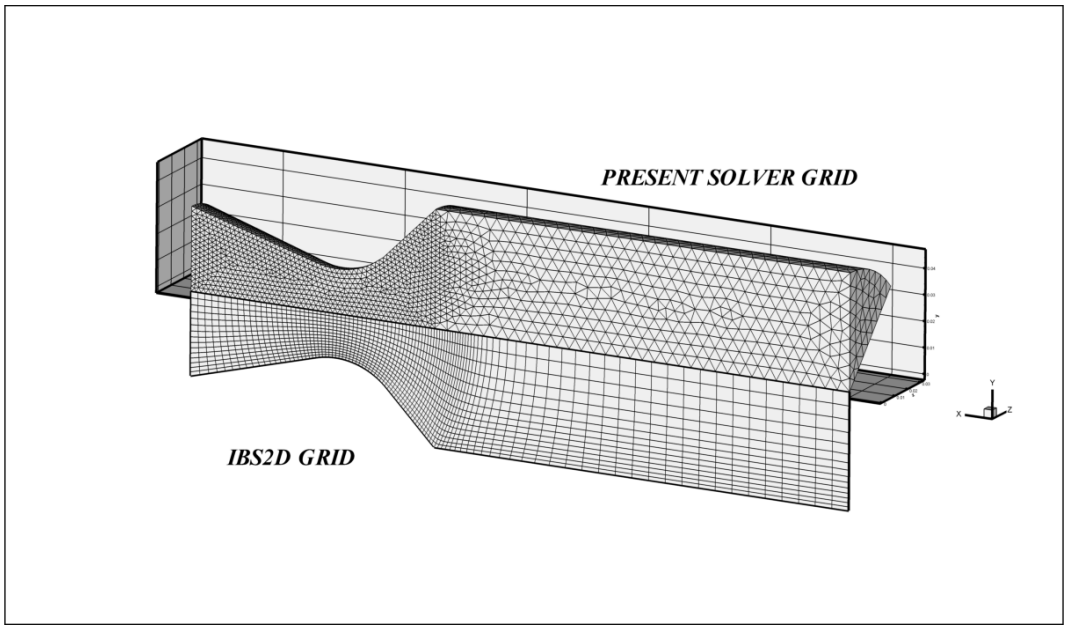

Figure 4.10 Grids used by IBS2D and Present Solver

	2-D Axi-symmetric	3-D Solver
Geometric		
Length	270 mm	270 mm
Radius	20 mm	20 mm
Grid Size	99 x 16	7271 Nodes 32332 Elements
Flow		
Mass flux	11.39 kg/m^2/s	
Exit Pressure	100000 Pa	
Gamma	1.14	
Gas Constant	299.5 J/(kg.K)	
Reference pressure	100000 Pa	
Flame Temperature	3387 K	

Table 4.2 Geometric and Flow Parameters for Test Case 2

4.3.1 Time Stepping Method Comparison

With the present solver, both implicit and explicit first order solutions are performed to see the impact of the time stepping method on the robustness of the solver. For these solutions, a computer with an Intel Xeon 3.4 GHz processor with 2 GB of RAM.

The CFL number for the explicit solution is 0.95, whereas a CFL number of 10^6 is used for the implicit method.

The Mach number contours for both solutions are given in Figure 4.11. It is evident that both explicit and implicit methods produced the same results. That is, no extra numerical diffusion error is introduced with the implicit time stepping method applied in the solver. The residual and CPU time plots are given in Figure 4.12. The decrease in the total iteration number is 80%, whereas the CPU time decrease

is 90%. The reasons for such a drastic drop in CPU time and total iteration number are, (I), for each iteration only one flux evaluation is done for the implicit method, whereas four flux evaluations are performed in the explicit method, and (II), the speed up due to the CFL number.

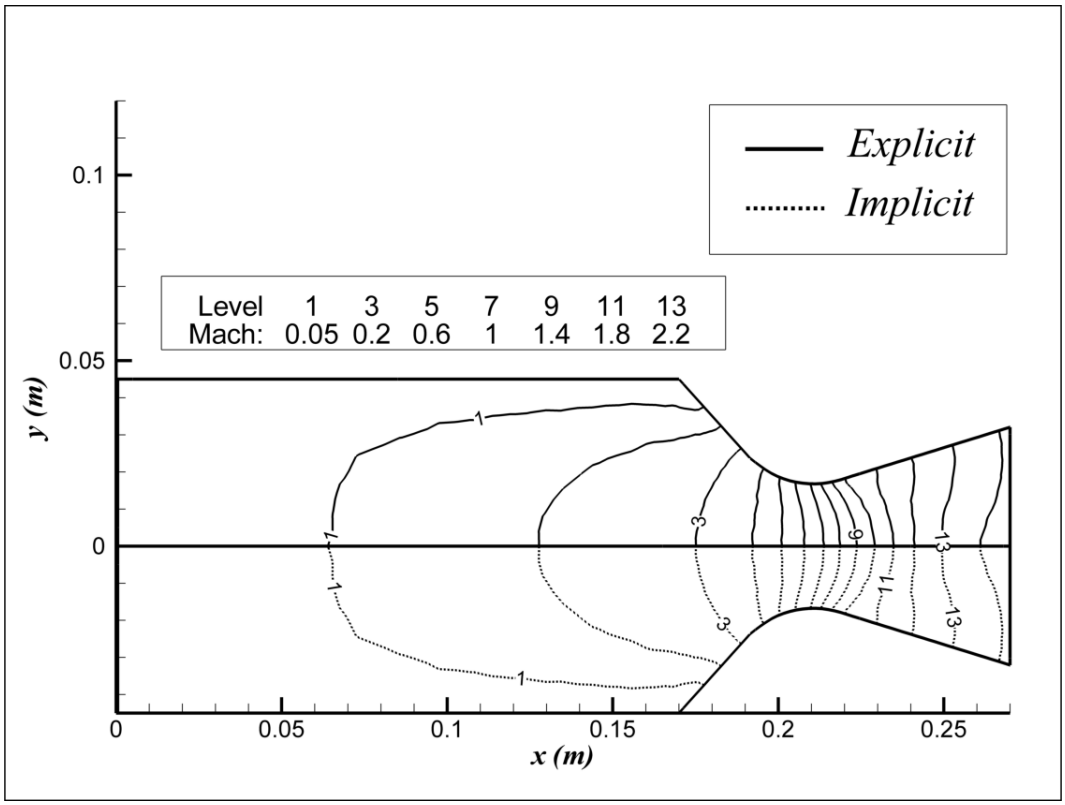

Figure 4.11 Mach Number Contours for Explicit and Implicit Solutions

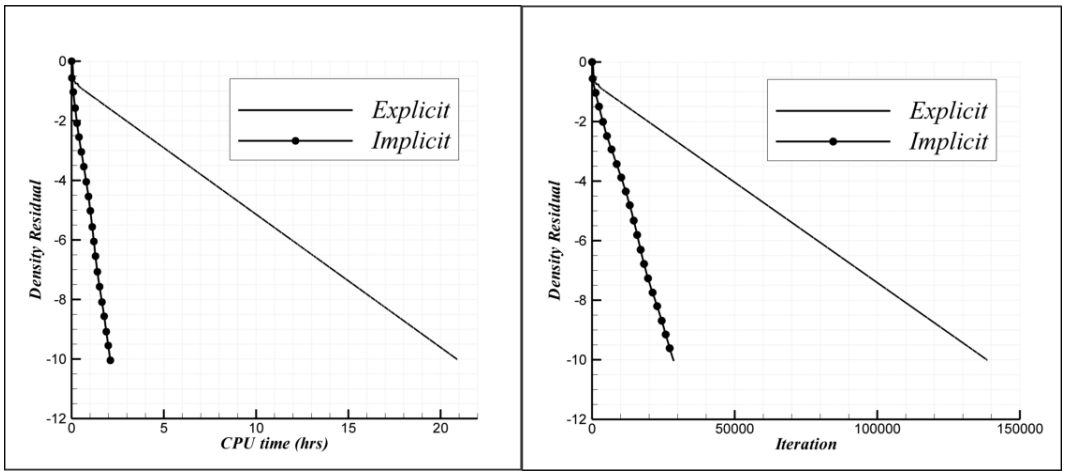

Figure 4.12 Residual and CPU Time History of Explicit and Implicit Solutions

4.3.2 Comparison of Gradient Calculation Methods

Second order solutions are performed with both GG and LSQ gradient calculation methods. For both solutions, a CFL number of 10^6 is used. The results are compared with the results of IBS2D.

Mach number contours are given in Figure 4.13 for first order solution with the IBS2D results. In this figure, it can be seen that the discrepancy between the first order solution and IBS2D solution increase as the flow speed increases in the diverging section of the nozzle, where the flow is supersonic. In Figure 4.14, Mach number contours of second order solutions with GG and LSQ methods and IBS2D solution are shown. The accuracy recovery with the second order solutions are easily seen in the figure. However, if the supersonic region is closely examined, it can be observed that the LSQ method performs better than GG method.

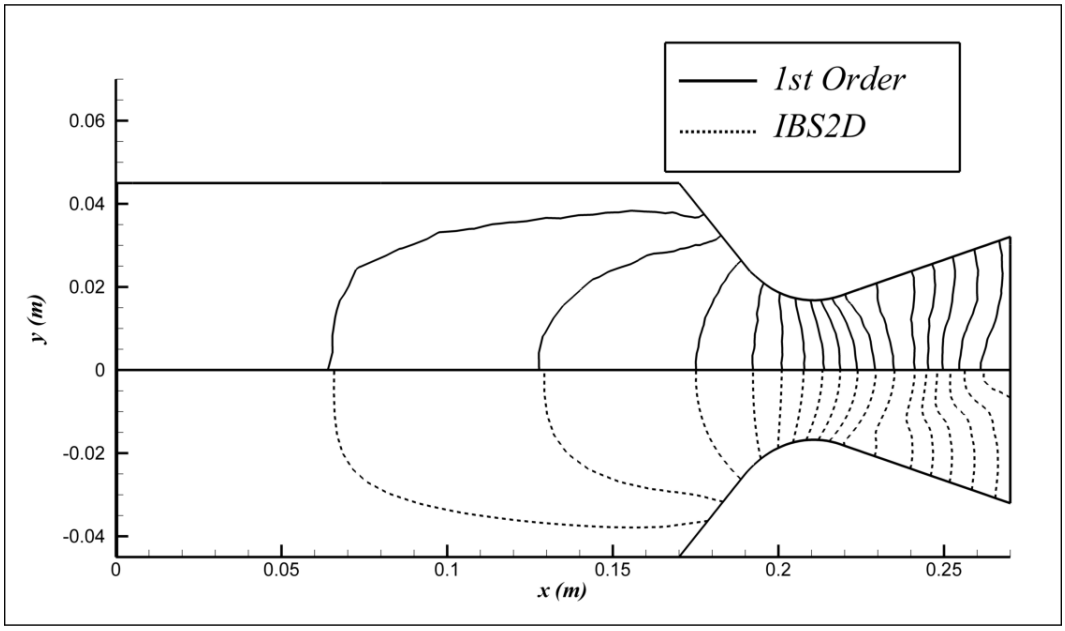

Figure 4.13 Mach Number Contours of 1st Order and IBS2D Solutions

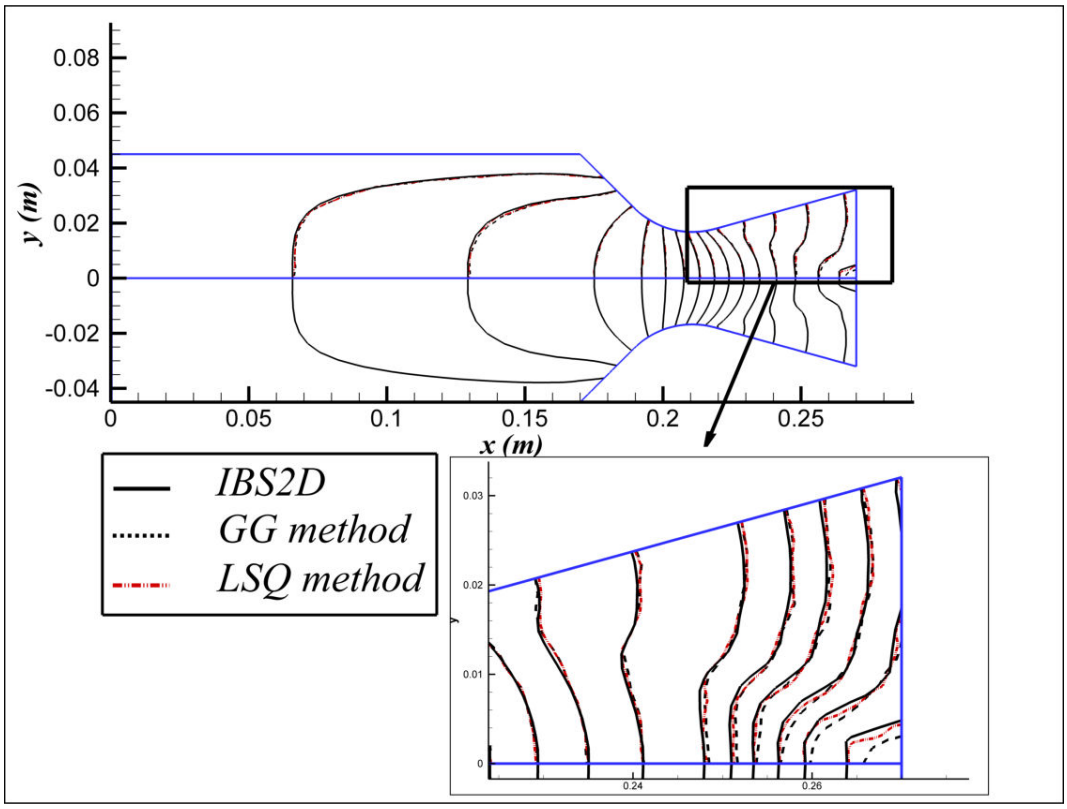

Figure 4.14 Mach Number Contours of Second Order and IBS2D Solutions

4.4 FLOW OVER NACA0012 AIRFOIL

This test case involves an airfoil, over which there are flow transitions from subsonic to supersonic. The freestream Mach number is 0.85 and the angle of attack is 1°. Outer radius of the solution domain is 15 chords. The reference solution for this test case is given in [37]. In this test case, the performance of the solver is investigated when the accuracy is switched to the second order. Besides, the assessment of the accuracy gain and the effect of the variation of the limiter function parameter κ, on the convergence and pressure coefficient are explored.

An unstructured grid is used, where there are 320 and 80 cells on the airfoil and outer boundary, respectively. The grid is shown in Figure 4.15. The grid used for the reference solution is an O-type grid with 320x64 points.

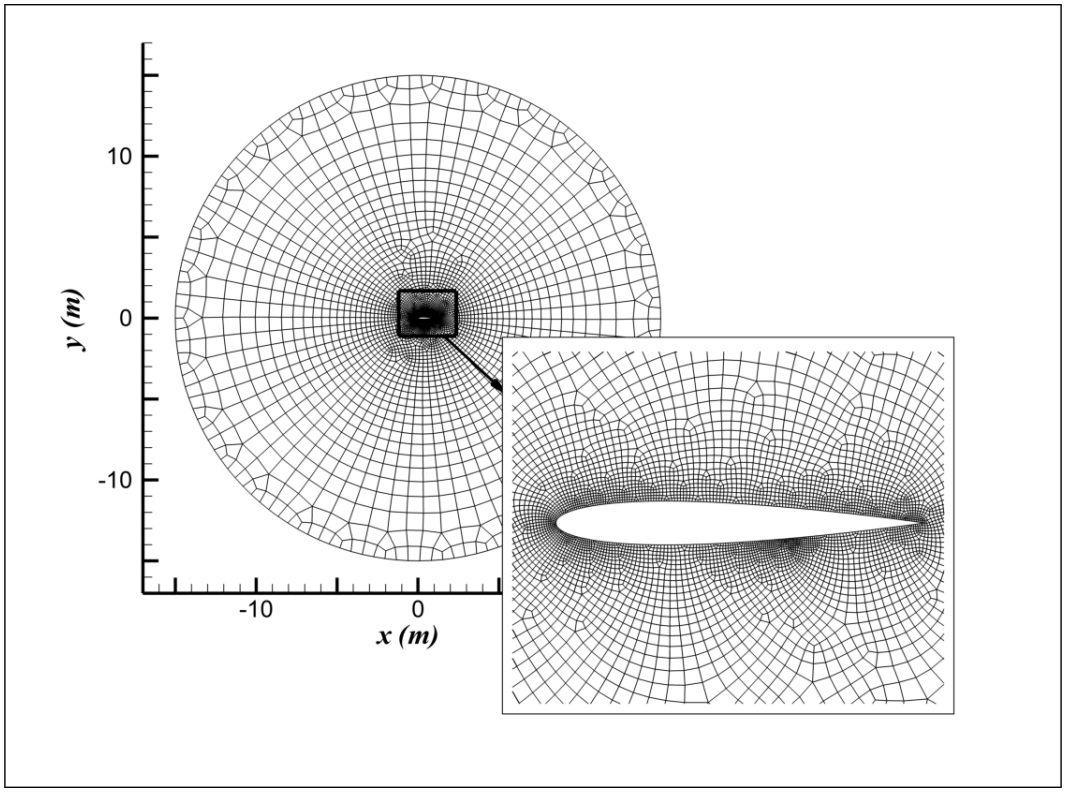

Figure 4.15 Grid of NACA0012 airfoil test case

Pressure coefficient results for the first order solution and the reference solution are given in Figure 4.16. In this figure, it is evident that the first order solution deviates from the reference solution for a considerably large amount. Eventhough, the pressure coefficient values do not comform with the reference solution, the shock locations are close for both solutions. In regions close to the head end of the airfoil, pressure coefficient values show oscillatory behavior. This shows that the grid refinement in these regions, is not good enough for the first order solution. The oscillations in near the head end of the airfoil can be seen in Figure 4.17.

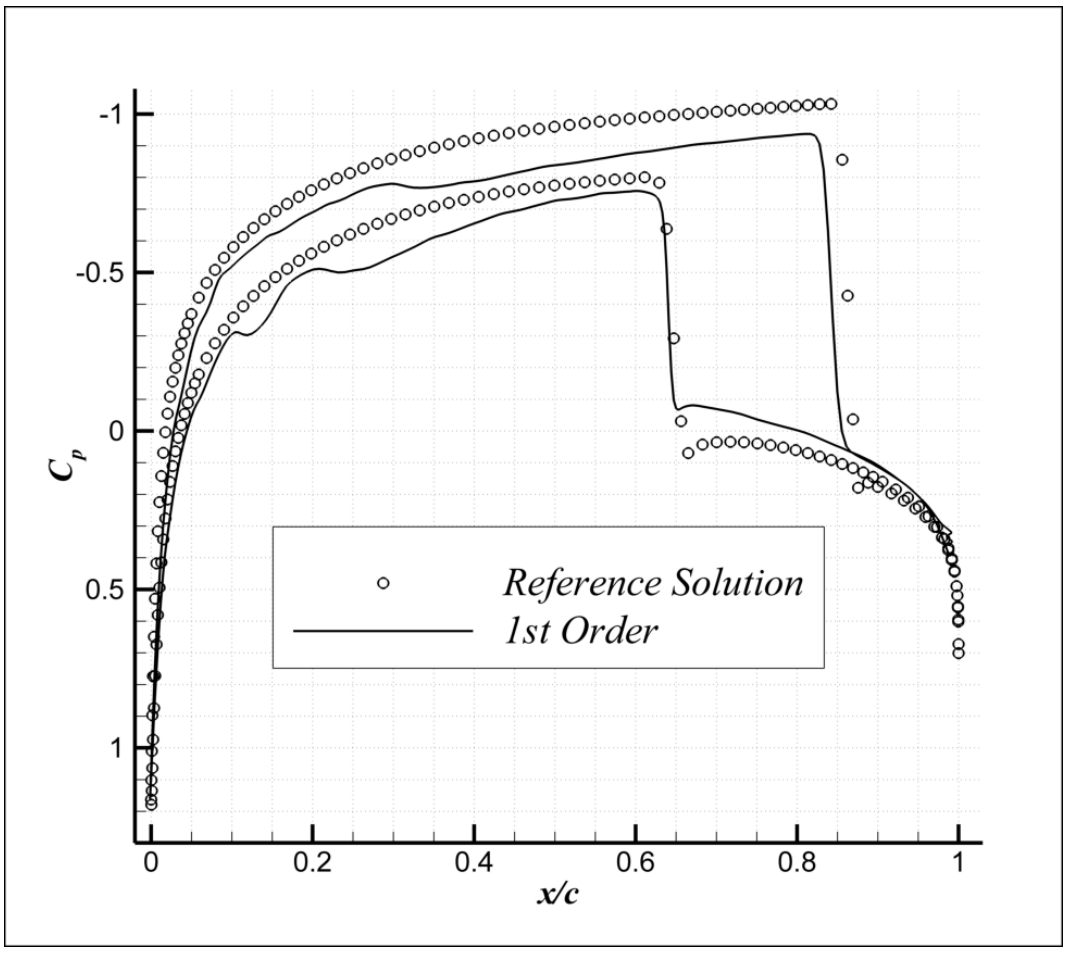

Figure 4.16 Pressure Coefficient Distribution on NACA0012 airfoil, of First Order and Reference Solutions

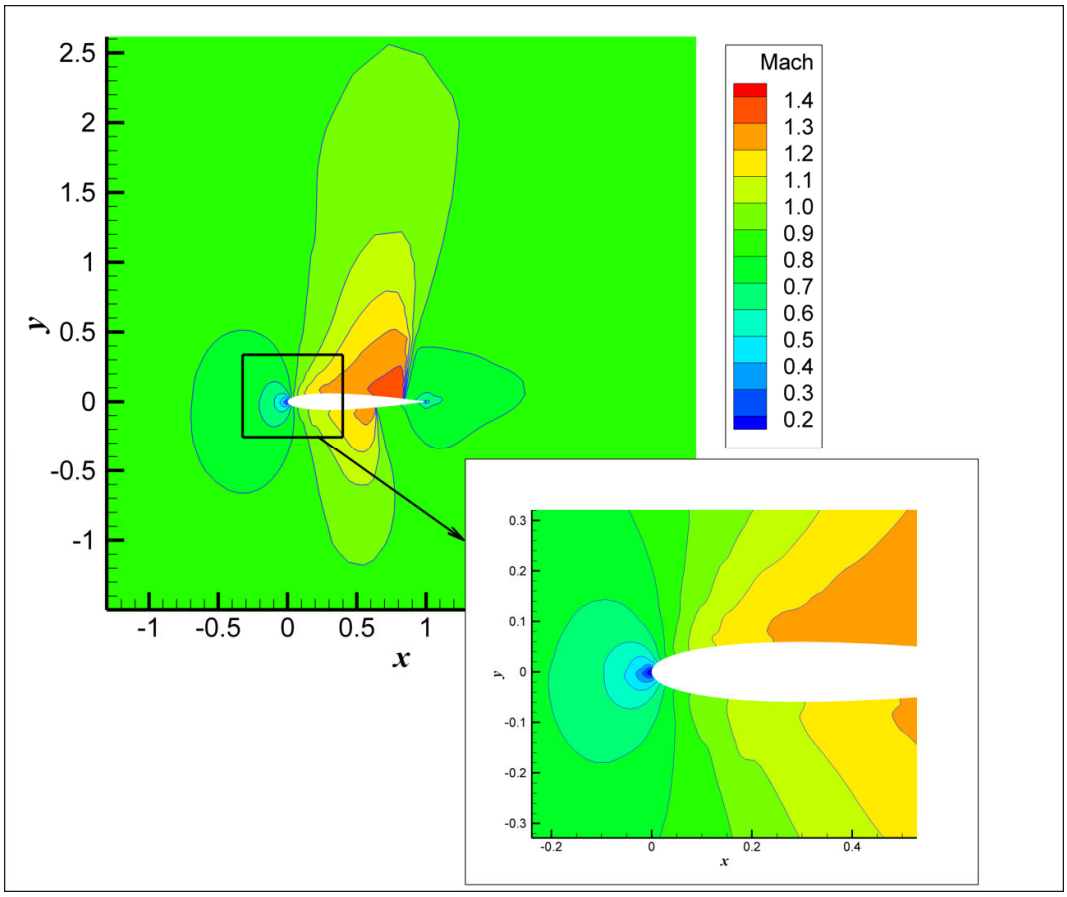

Figure 4.17 Mach Number Distribution around NACA0012 airfoil, First Order Solution

Second order solutions are performed with the application of the limiter function and behavior of different limiter function parameters is analyzed. Three solutions with κ value of 0, 5 and 10 are obtained. If κ is close to the zero, full limiting is achieved. However, if κ approaches large values, simply, there is no limiting at all. The residual history of all solutions is shown in Figure 4.18. All second order solutions are initiated using the results of first order solution. For a κ value of 0, no converged solution could be obtained. For κ being equal to 5 and 10, converged solutions are achieved.

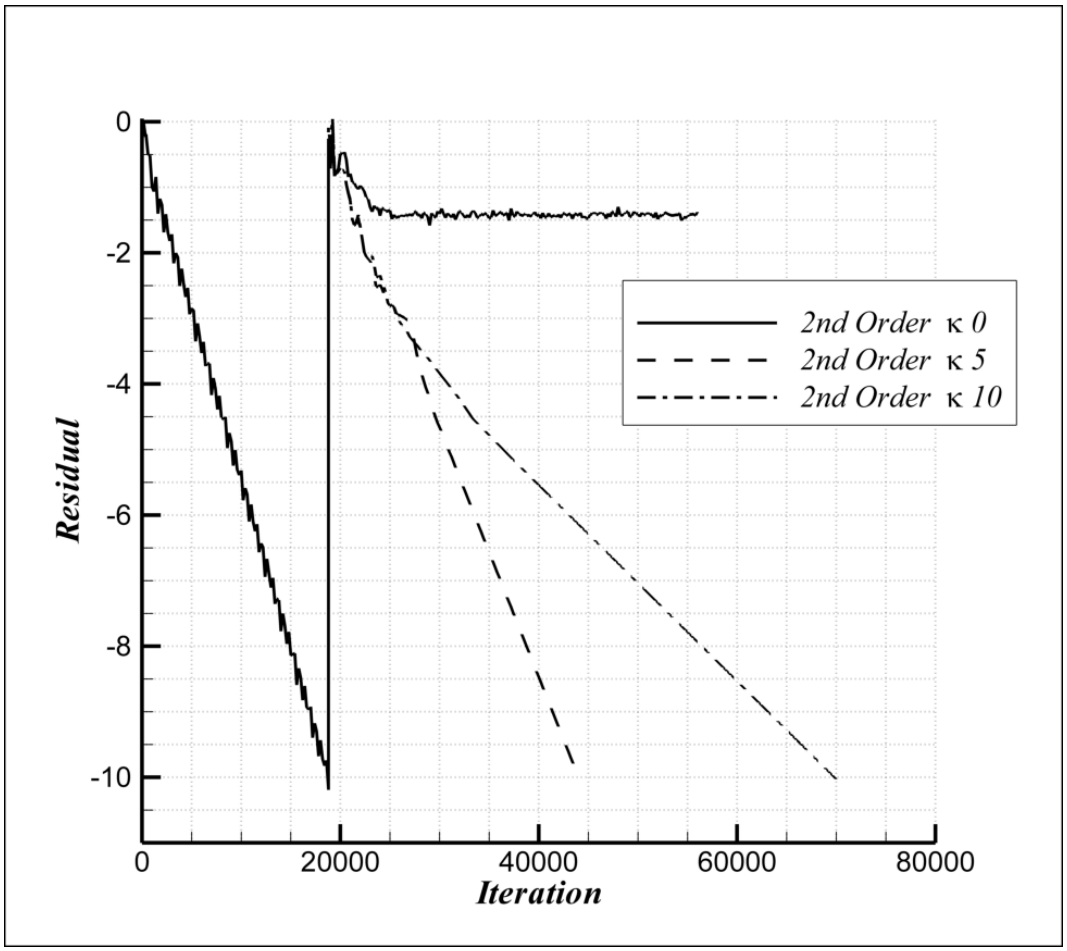

Figure 4.18 Residual History of Second Order Solutions with Different κ values

The pressure coefficient results for second order solutions are given in Figure 4.19. Although high values of κ helped to achieve to gain convergence, overshoots occurred in the near regions of shocks on the upper and lower part of the airfoil. As κ is increased from 5 to 10, the overshoots are more pronounced. The Mach contour distribution for the second order solution with κ being equal 5 is given in Figure 4.20.

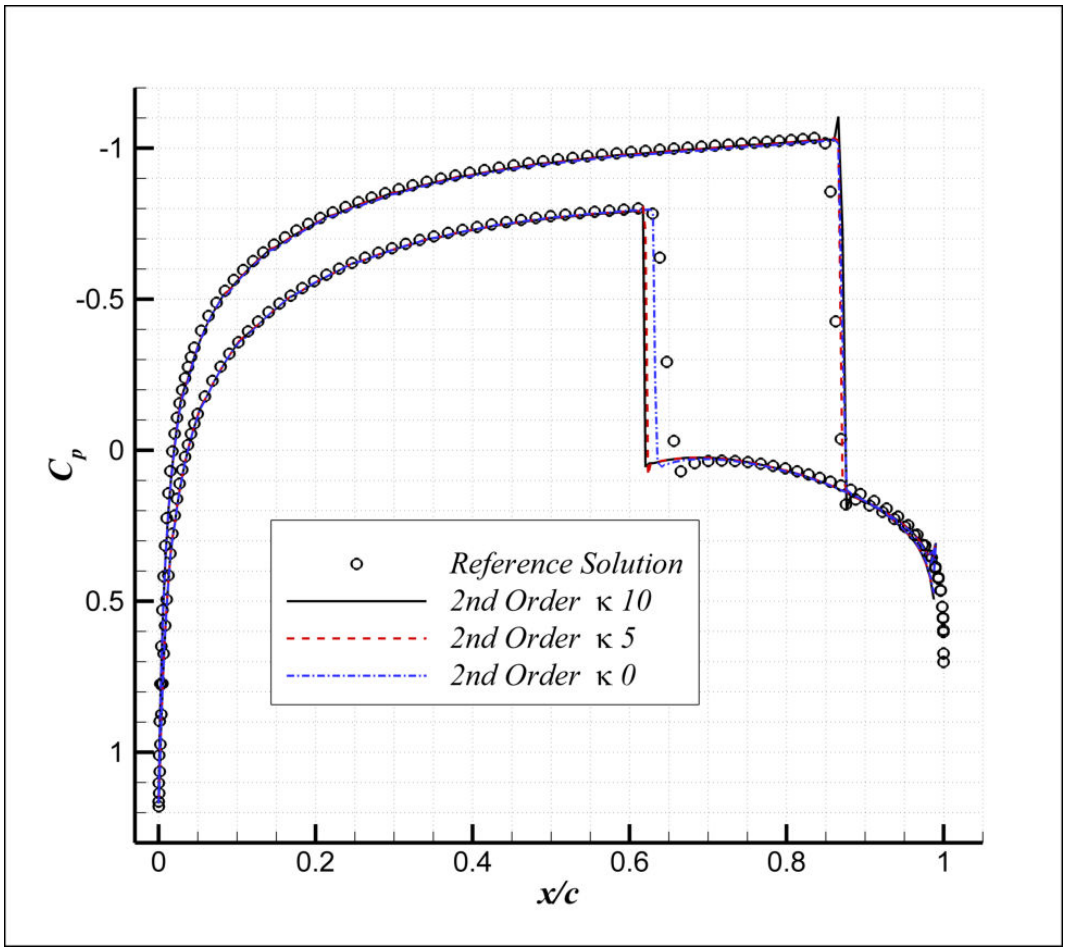

Figure 4.19 Pressure Coefficient Distribution on NACA0012 airfoil, of Second Order and Reference Solutions

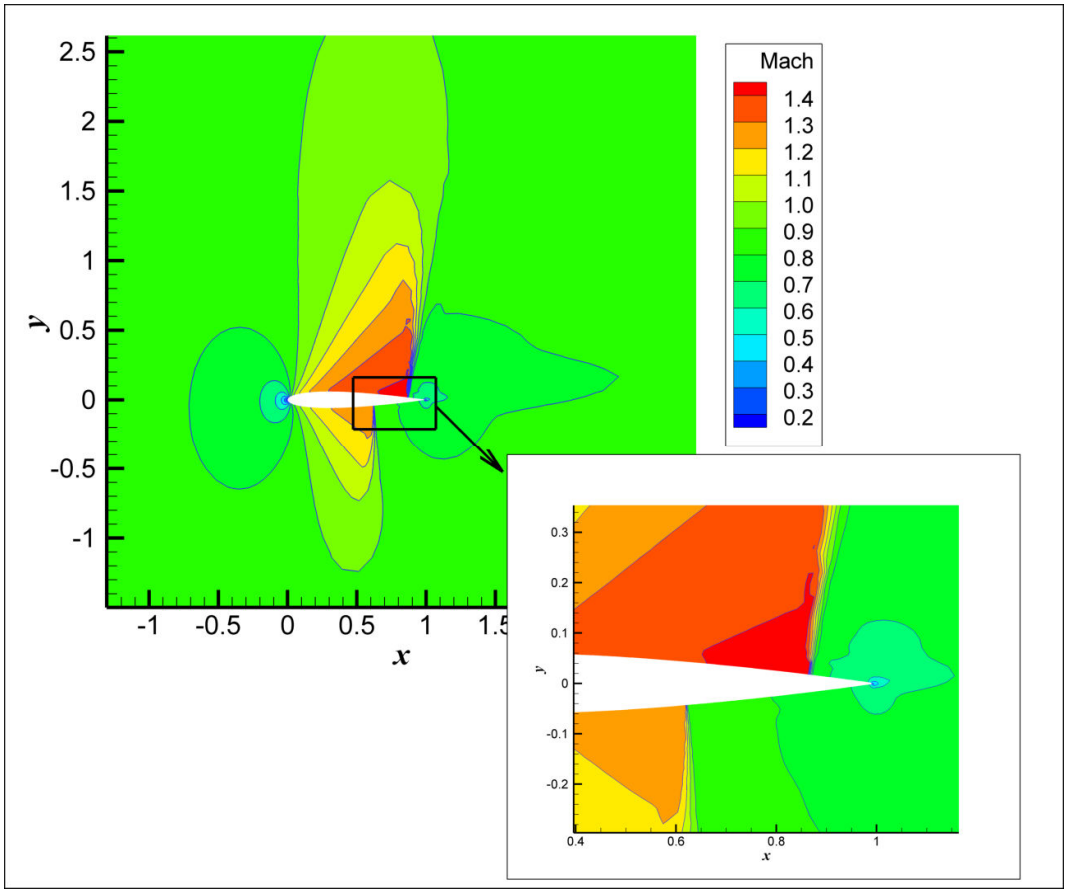

Figure 4.20 Mach Number Distribution around NACA0012 airfoil, Second Order Solution, κ equals 5

4.5 SUPERSONIC VORTEX FLOW

The last test case is an inviscid, isentropic, supersonic flow between two concentric circular arcs as illustrated in Figure 4.21. The flow inlet is located at the right boundary and the flow exit is at the bottom boundary. The circular arcs are taken as solid walls. The inner radius has been taken as $r_{inner} = 1$ and the outer radius as $r_{outer} = 1.384$. The flow is supersonic throughout the domain, with $M_{inner} = 2.25$.

The flow parameters vary only with radius r and are given by [38];

$$\rho(r) = \left[1 + \frac{\gamma - 1}{2} M^2{}_{inner} \left(1 - \left(\frac{r_{inner}}{r} \right)^2 \right) \right]^{\frac{1}{\gamma - 1}} \tag{4.5}$$

$$P(r) = \frac{1}{\gamma} \left[1 + \frac{\gamma - 1}{2} M^2{}_{inner} \left(1 - \left(\frac{r_{inner}}{r} \right)^2 \right) \right]^{\frac{\gamma}{\gamma - 1}} \tag{4.6}$$

Four different grids (Figure 4.22 to Figure 4.24) are used for this test case, to find how the error changes with varying grid size. First and second order solutions are performed on all grids and L2 norm error is calculated for discrete solution points using the results of the solver and the exact solution. L2 norm errors are given in Table 4.3

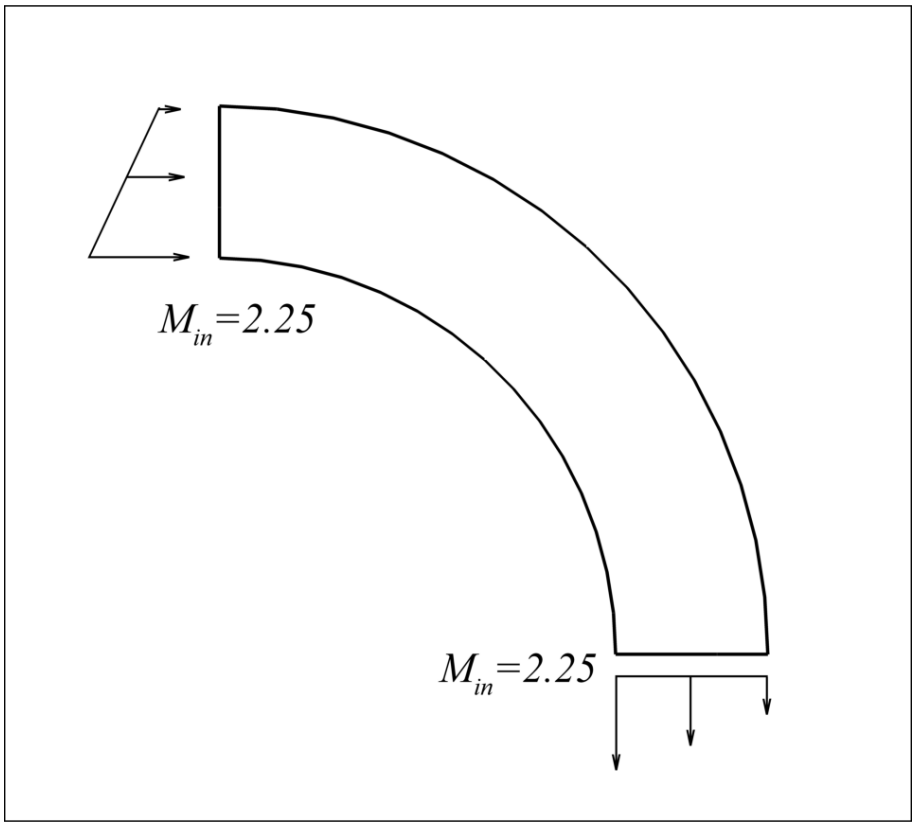

Figure 4.21 Description of Supersonic Vortex Flow

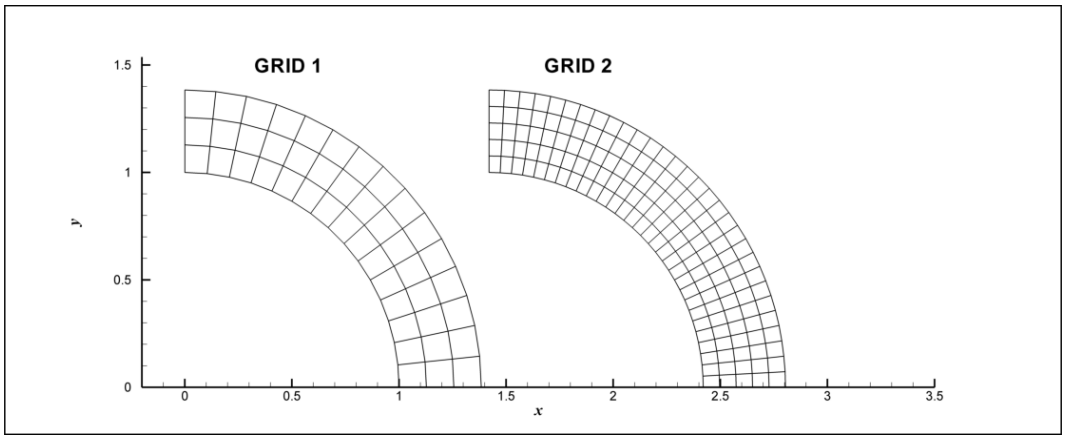

Figure 4.22 Grid 1 and 2 of Supersonic Vortex Flow

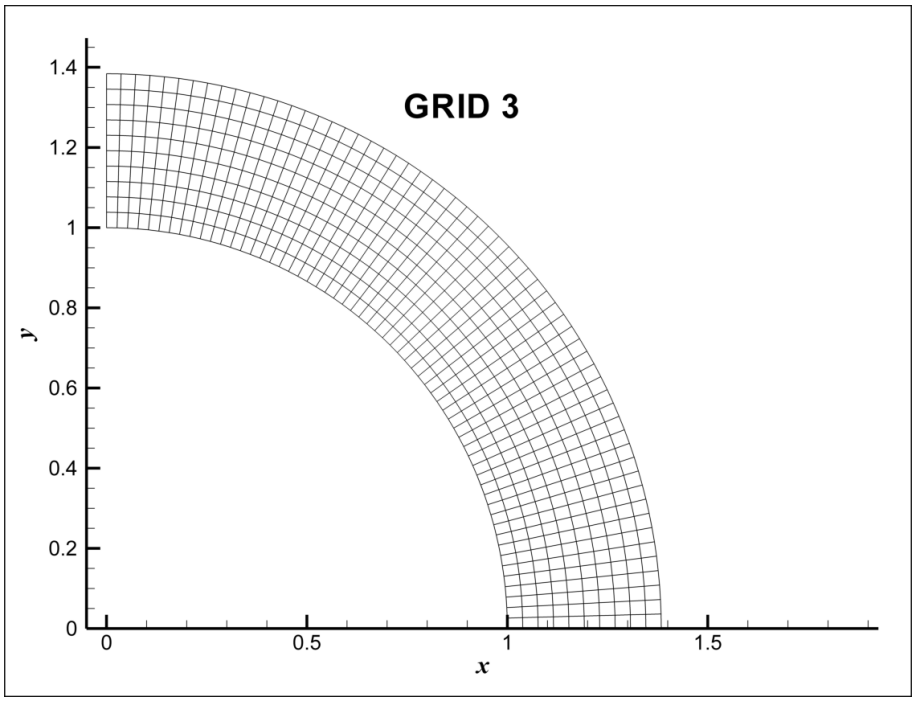

Figure 4.23 Grid 3of Supersonic Vortex Flow

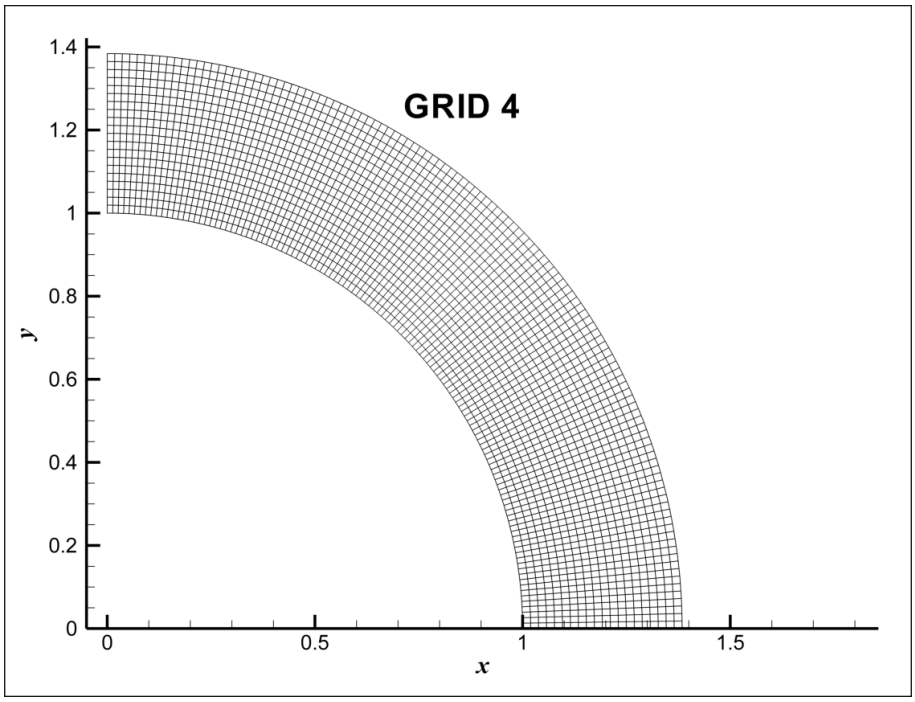

Figure 4.24 Grid 4of Supersonic Vortex Flow

In

Table 4.3, it can be seen that as the grid spacing is halved, L2 global norms of error in density decreases by approximately 50% and 75% for the first order solution and second order solutions respectively, for the first three grids. These results confirm that the solver is able to give second order accurate results.

GRID	$1/\sqrt{cell\_number}$	L2 Norm Error (%)	
		First Order	Second Order
Grid 1 (15x3x3 cells)	0.195	64.31	14.60
Grid 2 (30x5x5 cells)	0.110	28.44	2.85
Grid 3 (60x10x10 cells)	0.055	4.25	0.63
Grid 4 (120x20x20 cells)	0.028	2.63	0.24

Table 4.3 L2 Global Norms of Error in Density

The density contours of second order solutions for four grids are given in Figure 4.25 to Figure 4.28. The improvement in the solution as the grid spacing decreases can be seen in these figures.

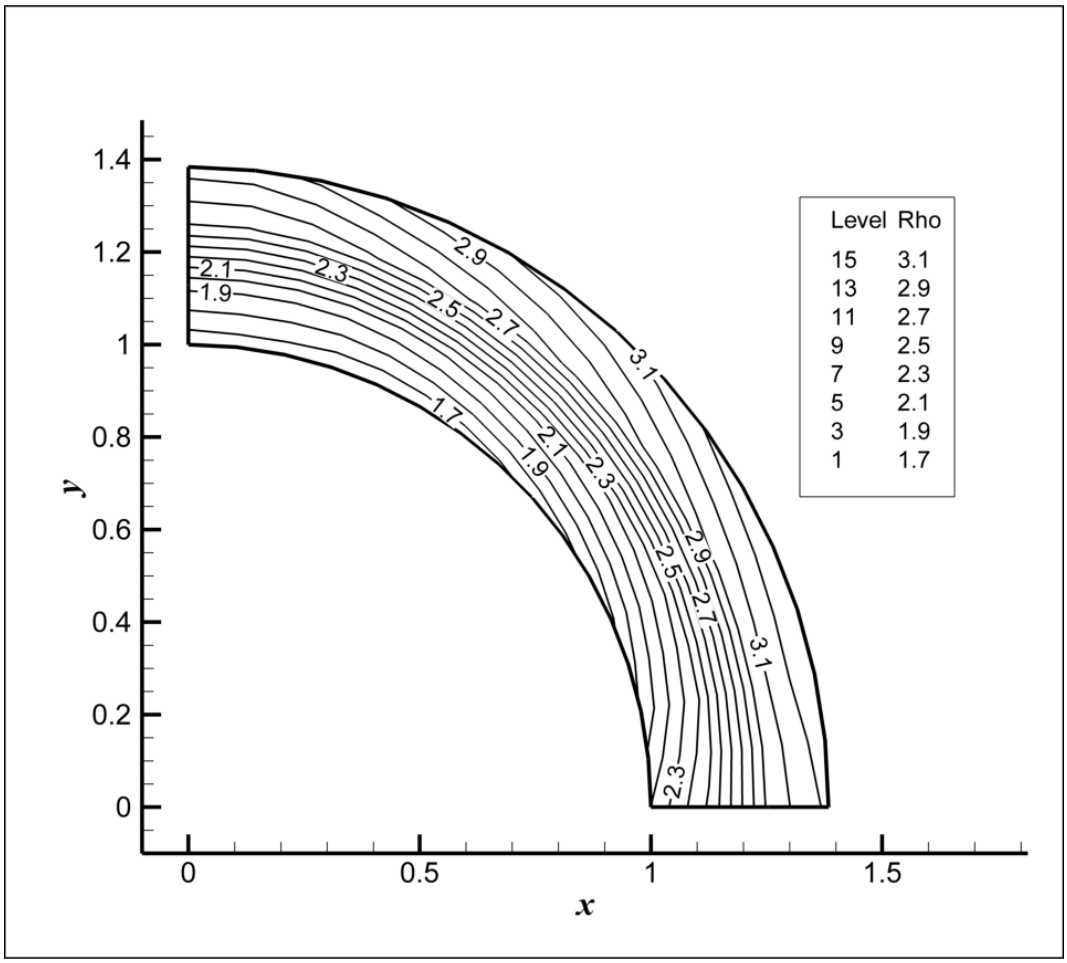

Figure 4.25 Second Order Solution for Grid 1

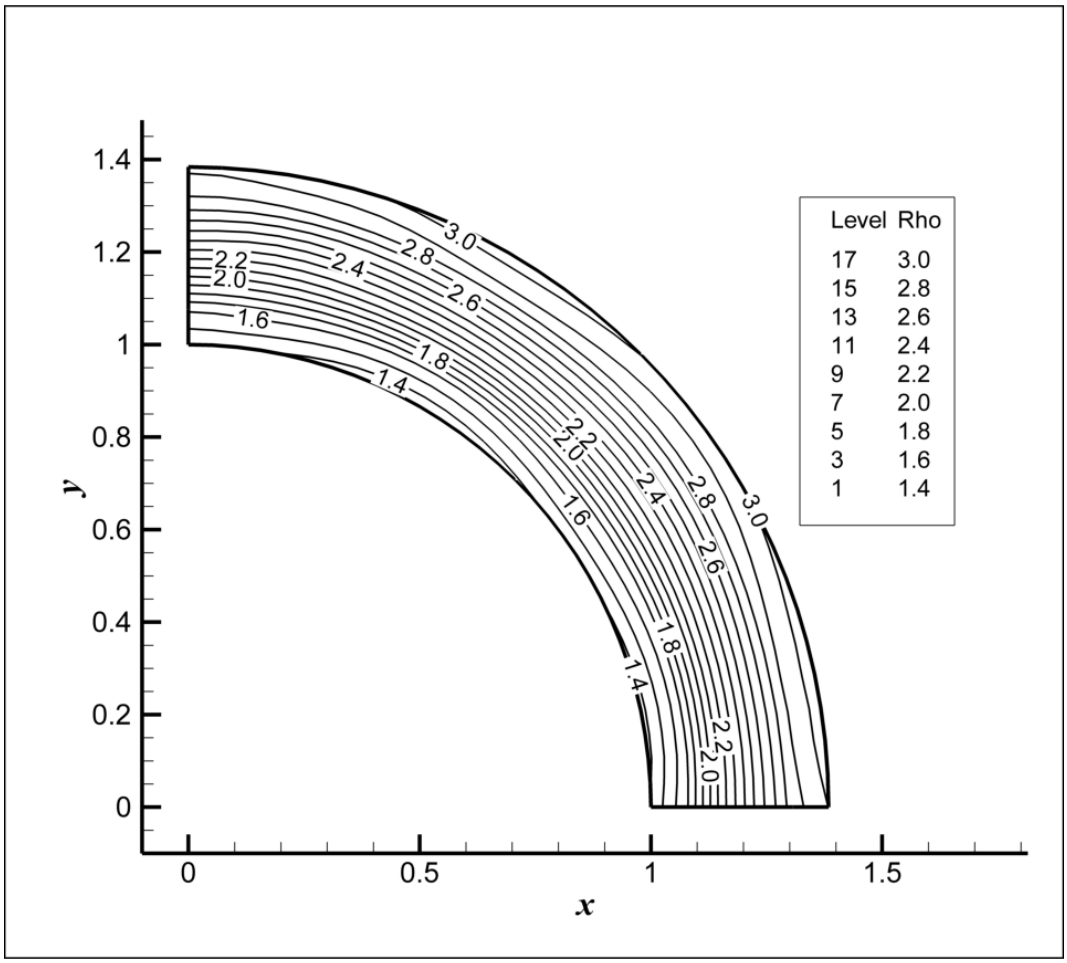

Figure 4.26 Second Order Solution for Grid 2

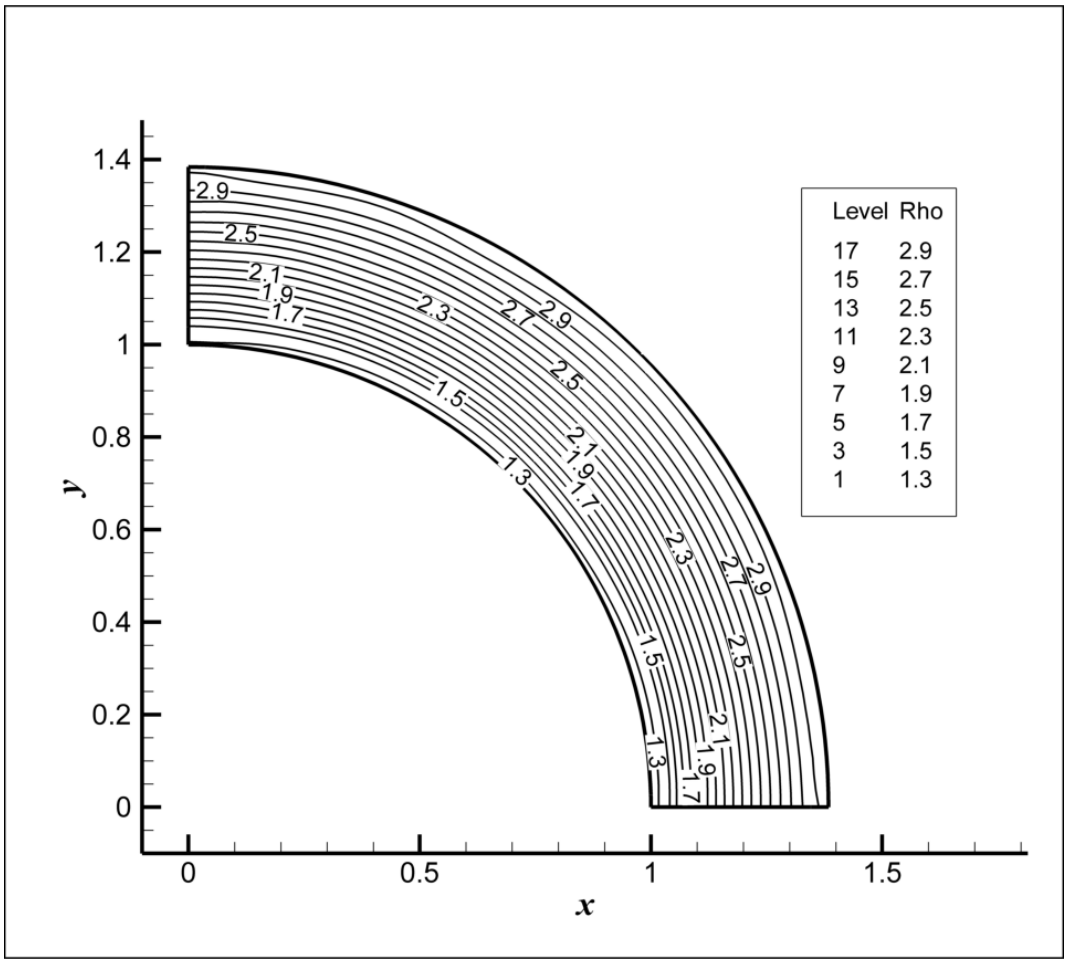

Figure 4.27 Second Order Solution for Grid 3

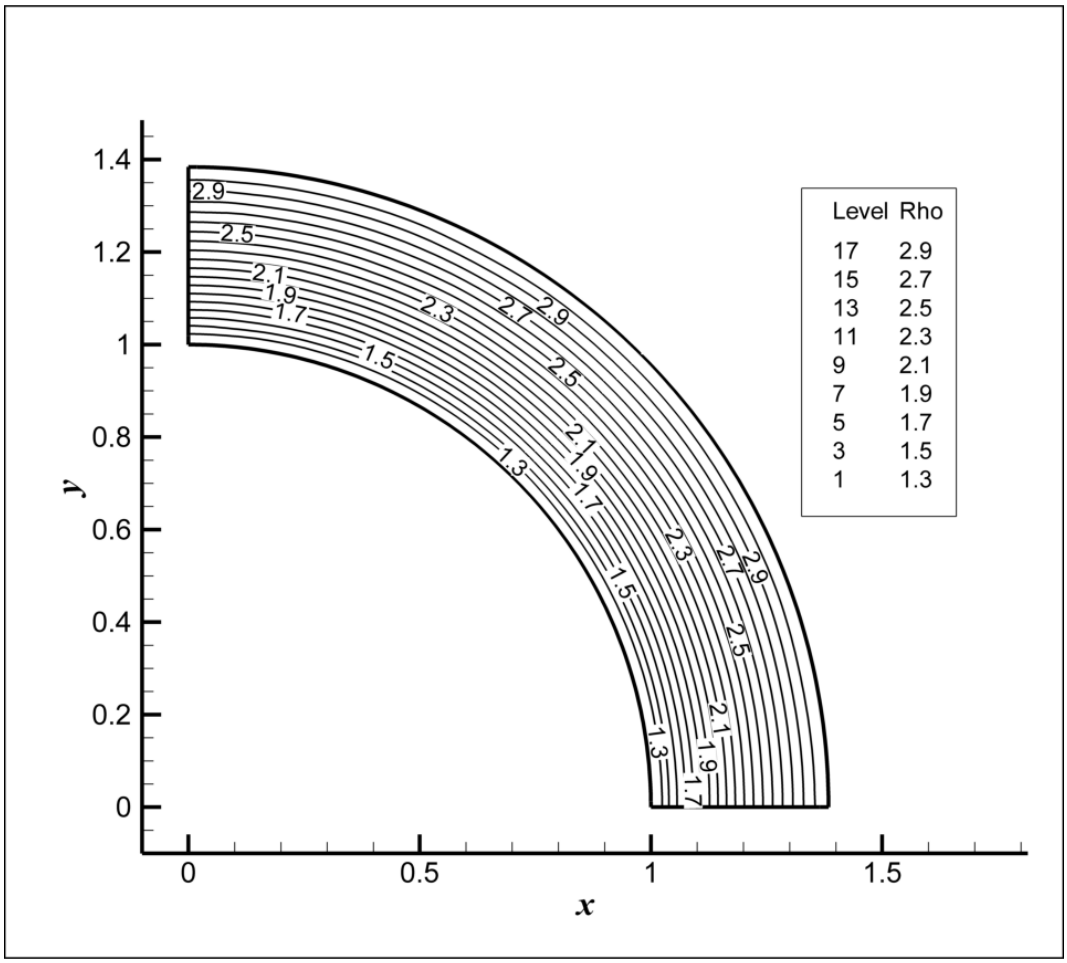

Figure 4.28 Second Order Solution for Grid 4

CHAPTER 5

CONCLUSION AND DISCUSSION

An implicit algorithm for computing the compressible Navier-Stokes equations on hybrid grids has been developed and presented. Lower-Upper Symmetric Gauss-Seidel implicit method with a computationally efficient cell-centered algorithm are used in this study. The main features of the developed code are given below.

Second order accuracy in space is achieved by piecewise linear reconstruction method.

Two methods, Green-Gauss and Least-Squares, are used to compute the gradients of flow variables and it has been shown that both of these methods were found to be suitable for this purpose. Least-Squares method gives more accurate results for unstructured grids. Venkatakrishnan's limiter is used for second order accurate solutions. Effect of the limiter parameters on the solution is investigated. Effect of the implicit method on the convergence speed is shown by comparing the results with explicit solution.

For future developments, implementation of turbulence models may be a course of action. Also, the code can be upgraded so that it can run on parallel processors. Another improvement may be the use of multigrid methods to further speed up the convergence. Moreover, implementation of dual- time stepping may increase the capability of solver for unsteady flows.

REFERENCES

[1] Hirsch, C., *"Numerical Computation of Internal and External Flows"*, Vol 1&2, John Wiley&Sons,1989.

[2] Frink, N. T., *"Assessment of an Unstructured-Grid Method for Predicting 3-D Turbulent Flows"*, AIAA 96-0292, 1996.

[3] Frink, N. T., *"Upwind Scheme for Solving the Euler Equations on Unstructured Grids"*, AIAA Journal Vol. 30, No. 1, pp. 70-77, January, 1992.

[4] Batina, J. T., *"Implicit Upwind Solution Algorithms for Three-Dimensional Unstructured Meshes"*, AIAA Journal Vol. 31, No. 5, pp. 801-805, May, 1993.

[5] Barth, T. J., *"Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier-Stokes Equations"*, NASA Ames Research Center, Moffet Field, Ca., USA, Von Karman Institute for Fluid Dynamics, Lecture Series 1994-05, Computational Fluid Dynamics, March 21-25, 1994.

[6] Venkatakrishnan V., *" Implicit Schemes and Parallel Computing in Unstructured Grid CFD"*, ICASE Report No. 95-28 NASA Langley Research Center, 1995.

[7] Jothiprasad, G., Mavriplis, D. J., and Caughey, D. A., *"Higher Order Time Integration Schemes for the Unsteady Navier-Stokes Equations on Unstructured Meshes"*, NASA/CR-2002-211967 ICASE Report No. 2002-44, 2002.

[8] Pirzadeh, S. Z., and Frink, N. T., *"Assessment of the Unstructured Grid Software TetrUSS for Drag Prediction of the DLR-F4 Configuration"*, AIAA 2002-0839, January, 2002.

[9] Frink, N. T. and Pirzadeh, S. Z., *"Tetrahedral Finite-Volume Solutions to the Navier-Stokes Equations on Complex Configurations"*, International Journal for Numerical Methods in Fluids, Vol. 31, pp. 175-187, 1999.

[10] Dadone, A., and Grossman, B., *"Characteristic-Based, Rotated Upwind Scheme for the Euler Equations"*, AIAA Journal Vol. 30, No. 9, pp.2219-2226, September, 1992.

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

[12] Godunov, S. K., *"A Difference Method for the Numerical Computation of Discontinuous Solutions of Hydrodynamic Equations"*, Math Sbornik, Vol. 47, pp. 271-306, 1959.

[13] Roe, P. L., *"Discrete Models for the Numerical Analysis of Time-Dependent Multidimensional Gas Dynamics"*, Journal of Computational Physics, Vol. 63, pp. 458-476, 1986.

[14] Osher, S., and Solomon, F., *"Upwind Schemes for Hyperbolic Systems of Conservation Laws"*, Mathematics of Computation, Vol. 38, No. 158, pp. 339-377, 1982.

[15] Toro, E. F., *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 2nd Edition, Springer, 1999.

[16] Harten, A., *"High Resolution Schemes for Hyperbolic Conservation Laws"*, Journal of Computational Physics, Vol. 49, pp. 357-393, 1983.

[17] Jameson, A., Schmidt, W., and Turkel, E., *"Numerical Simulation of the Euler Equations by Finite Volume Methods using Runge-Kutta Time Stepping Schemes"*, AIAA 81-1259, 1981.

[18] Batina, J. T., *"Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex-Aircraft Aeordynamic Analysis"*, AIAA Journal Vol. 29, No. 3, pp. 327-333, March, 1991.

[19] Briley, W.R., and McDonald, H., *"Solution of the Multi-Dimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method"*, Journal of Computational Physics, Vol. 24, pp. 372-397, 1977.

[20] Yoon, S., and Jameson, A., *"A Multigrid LU-SSOR Scheme for Approximate Newton-Iteration Applied to the Euler Equations"*, NASA CR-17954, 1986.

[21] Harten, A., and Hyman, J.M., *"Self Adjusting Grid Methods for One-Dimensional Hyperbolic Conservation Laws"*, Journal Computational Physics, Vol. 50, pp. 235-269, 1983.

[22] Barth, T.J., and Jespersen, D.C., *"The Design and Application of Upwind Schemes on Unstructured Meshes"*, AIAA Paper 89-0366, 1989.

[23] Haselbacher, A., and Blazek, J., *"On the Accurate and Eflcient Discretisation of the Navier-Stokes Equations on Mixed Grids"*, AIAA Journal, 38, pp. 2094-2102, 2000.

[24] Delanaye, M., *Polynomial Reconstruction Finite Volume Schemes for the Compressible Euler and Navier-Stokes Equations on Unstructured Adaptive Grids*, PhD Thesis, The University of LiBge, Belgium, 1996.

[25] Barth, T., J., *"A 3-D Upwind Euler Solver for Unstructured Meshes"*, AIAA Paper 91-1548,1991.

[26] Anderson, W.K., *Bonhaus, D.L. "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids"*, Computers & Fluids, Vol. 23, pp. 1-21, 1994.

[27] Venkatakrishnan, V., *"On the Accuracy of Limiters and Convergence to Steady State Solutions"*, AIAA Paper 93-0880, 1993.

[28] Venkatakrishnan, V., *"Convergence to Steady-State Solutions of the Euler Equations on Unstructured Grids with Limiters"*, Journal of Computational Physics, Vol. 118, pp. 120-130, 1995.

[29] Tomaro, R.F., Strang, W.Z., and Sankar, L.N., *"An Implicit Algorithm for Solving Time Dependent Flows on Unstructured Grids"*, AIAA Paper 97-0333, 1997.

[30] Sharov, D., and Nakahashi, K., *"Reordering of 3-D Hybrid Unstructured Grids for Vectorized LU-SGS Navier-Stokes Calculations"*, AIAA Paper 97-2102, 1997.

[31] Kano, S., and Kazuhiro, N., *"Navier-Stokes Computations of HSCT Off-Design Aerodynamics Using Unstructured Hybrid Grids"*, AIAA Paper 98-0232, 1998.

[32] Sharov, D., and Nakahashi, K., *"Low Speed Preconditioning and LU-SGS Scheme for 3-D Viscous Flow Computations on Unstructured Grids"*, AIAA Paper 98-0614, 1998.

[33] Strang, W.Z., Tomaro, R.F., and Grismer, M.J., *"The Defining Methods of Cobaltso: a Parallel, Implicit, Unstructured Euler/Navier-Stokes Flow Solver"*, AIAA Paper 99-0786, 1999.

[34] Schlichting, H., *Boundary-Layer Theory*, McGraw-Hill Book Company, 7th edition, (ISBN 0-07-055334-3), pp. 147-148, 1987.

[35] Vuillot, F., Yumuşak M., Taşkınoğlu E. S., and Tınaztepe T., *"Technical Report of RTO-AVT T-108 Support Project"*, Roketsan A.Ş., 1998.

[36] Taylor G., *"Fluid Flow in Regions Bounded by Porous Surfaces"*, Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, Vol. 234, Issue 1199, pp. 456-475, 1956.

[37] AGARD-AR-211, *Test Cases for Inviscid Flow Field Methods*, May 1985.

[38] C. Ilinca, X.D. Zhang, J.-Y. Trepanier , and R. Camarero, *"A Comparison of Three Error Estimation Techniques for Finite-Volume Solutions of Compressible Fows"*, Comput. Methods Appl. Mech. Engrg. , Vol. 189, pp.1277-1294, 2000.

[39] Toker K.A., *"Three-Dimensional Retarding Walls and Flow in Their Vicinity"*, Ph.D. Thesis Submitted to the Graduate School of Natural and Applied Sciences of the Middle East Technical University, Dec 2004.