Development of a General-Purpose Compressible Flow AnuPravaha Based Solver using an Implicit Method

A Thesis Submitted
in Fulfilment of the Requirements
for the Degree of
Master of Technology

by
Saketh Chandra
Roll No.- ME14MTECH11002
M.Tech II Yr.-Fluids and Energy Systems

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Department Of Mechanical And Aerospace Engineering
Indian Institute Of Technology, Hyderabad

JULY 2016

DECLARATION

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

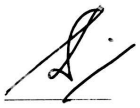
(---Saketh---)

Saketh Chandra

ME14MTECH11002

APPROVAL SHEET

This thesis entitled **Development of General Purpose Compressible Flow Solver using an Implicit method** by **Saketh Chandra** is approved for degree of Master of Technology from IIT Hyderabad.



Asst Prof. Saptarshi Majumdar
Department of Chemical Engg.
IITH

Asst Prof. Pankaj Kohle
Department of Mech and Aerospace Engg.
IITH

Prof. Vinayak Eswaran
Department of Mech and Aerospace Engg.
IITH

ACKNOWLEDGMENT

I am grateful to my teacher who has taught me how to live a reponsible life with his patience and love.

I would like to express my gratitude to my guide Prof. Vinayak Eswaran for allowing me freedom to work and at the same time keeping an eye to see that I do not drift away from the main objective. I would especially like to thank him for his moral support and calmness, at the time when I have been running on low confidence with the thesis work. His honesty and humility, has always motivated us to work harder and has kept us grounded. And simultaneously, he has been the source of motivation for all our friends. Thus, the work environment I have had has been vibrant and energetic. I would like to thank Ashwani Assam and Vatsalya Sharma for their valuable suggestions at various points of the thesis. In addition, they have helped us in understanding the AnuPravaha solver, which has been the starting point for this thesis. I would also like to thank all my friends who have always maintained an atmosphere of co-operation and friendship. Thus, have made the thesis a joyful experience. I would like to thank my guide and the entire IIT Hyderabad system for providing us an excellent computational facility to work upon.

Saketh Chandra

Abstract

With India focussing even more on Aerospace applications, research and development in compressible flow has received a boost in the country. We aim to develop a general-purpose and robust compressible flow solver to help in research in Aerospace problems.

In this thesis we aim to develop a general-purpose and robust compressible flow solver using the implicit MacCormack scheme in finite volume formulation. A system of unsteady Navier-Stokes equations are integrated to a steady state solution utilizing MacCormack's implicit numerical scheme. A new implicit boundary treatment was introduced in the MacCormack implicit scheme. The scheme is unconditionally stable and does not require solution of large systems of linear equations. It is shown that the upgrade from explicit MacCormack scheme, previously implemented in the solver, to an implicit one is very simple and straightforward.

Contents

Nomenclature

ρ	Density of gas
α	Angle of attack
γ	Ratio of specific heat
δ	Implicit operator
$\triangle$	Explicit operator
$\triangle_+, \triangle_-$	forward and backward finite differencing operator respectively
β	γ - 1
A	inviscid Jacobian, $\frac{\partial F_x}{\partial W}$
B	inviscid Jacobian, $\frac{\partial F_y}{\partial W}$
C	inviscid Jacobian, $\frac{\partial F_z}{\partial W}$
c	Speed of sound
C_v	Specific heat of gas at constant volume
C_p	Specific heat of gas at constant pressure
D_A, D_B, D_C	Characteristic Diagonal matrix
e	Total internal energy
F_x	x direction flux vector
F_y	y direction flux vector
F_z	z direction flux vector
I	Identity Matrix
M	Mach no.
P	Pressure(N/m^2)
R	Universal gas constant

Sx	Right eigen vector for x direction
Sx^{-1}	left eigen vector for x direction
Sy	Right eigen vector for y direction
Sy^{-1}	left eigen vector for y direction
Sz	Right eigen vector for z direction
Sz^{-1}	left eigen vector for z direction
T	Absolute Temperature
u	x component of velocity
v	y component of velocity
V_p	Volume of cell p
w	z component of velocity
W	Conservative vector
x	Cartesian coordinate
y	Cartesian coordinate
z	Cartesian coordinate

Chapter 1

Introduction

The fundamental basis of almost all CFD problems are the Navier–Stokes equations, which define many single-phase (gas or liquid, but not both) fluid flows. These equations can be simplified by removing terms describing viscous actions to yield the Euler equations. Further simplification, by removing terms describing vorticity yields the full potential equations. Finally, for small perturbations in subsonic and supersonic flows (not transonic or hypersonic) these equations can be linearized to yield the linearized potential equations. The basic mathematical model of fluid flow takes the form of partial differential equations which express the laws of conservation of mass, momentum and energy. While analytical solutions to these equations are possible for a few simple cases, in most cases, specially for complex geometry, the only alternative is to obtain approximate numerical solutions. Computational Fluid Dynamics (CFD, in short) is a powerful bridge between the calculus describing flow physics and high–speed computing. CFD methodology has matured over the years to an extent that it has found its way into most fluid flow research applications, notably in the aerospace industry.

Computational Fluid Dynamics (CFD) methods must satisfy stringent constraints because of the wide range of scales and frequencies in the target flows. To deal with those requirements, higher order, low dispersion and low dissipation schemes are needed. However, these schemes are also more sensitive to spurious waves generated by numerical boundary conditions.

In aerodynamics, the compressibility of a fluid is a very important factor. In nature, all the fluids are detectably compressible, but we define incompressible flows for our convenience of study. A compressible fluid will reduce its volume in the presence of an external pressure. Compressible flows (in contrast to variable density flows) are those where dynamics (i.e pressure) is the dominant factor in density change. Generally, fluid flow is considered to be compressible if the change in density relative to the stagnation density is greater than 5 %. Significant compressible effects occur beyond a Mach number of 0.3 and greater. Compressible effects are observed in practical applications like high speed aerodynamics, missile and rocket propulsion, high speed turbo compressors, steam and gas turbines, etc.

Compressible flow is divided often into four main flow regimes based on the local Mach number (M) of the fluid flow

- Subsonic flow regime ($M \leq 0.8$)
- Transonic flow regime ($0.8 \leq M \leq 1.2$)
- Supersonic flow regime ($M > 1$)
- Hypersonic flow regime ($M > 5$)

Compressible flow may be treated as either viscous or inviscid. Viscous flows are solved by the Navier-Stokes system of equations and inviscid compressible flows are solved by Euler equations. The physical behavior of compressible fluid flow is quite different from incompressible fluid flow. The solutions of Euler equation are different, due to their hyperbolic (wave-like) nature, from the solutions of the elliptic governing equations of incompressible flows. Compressible flow can have discontinuities such as shock waves. So for compressible flows special attention is required for solution methods which will accurately capture these discontinuities.

A major difference between solution methods for compressible flow and incompressible flow lies in the boundary conditions that are imposed. In compressible flow, boundary conditions are imposed based on the characteristic waves coming into the domain boundary, which is very different from the Elliptic-type boundary conditions used for incompressible flows.

For over a decade our research group has been continuously developing and modifying a CFD software called IITK-DAE ANUPRAVAHA, a genaral purpose CFD solver. The solver uses the finite volume method with a structured grid arrangement originally developed for incompressible flows, the solver was extended by previous M.Tech students (Nikhil Kalkote 2013, Ashwani Assam 2014) to compressible flows by using explicit methods. However, using an unsteady solver to obtain steady-state solutions by the false transient method is inefficient, especially if explicit time steping with time-step constraints due to numerical stability, is used. To achieve fast convergence to the steady state, an implicit time marching scheme is thought to be much better to solve the Euler equations. This the-

sis implements a scheme based on the MacCormack implicit scheme [22] so that higher courant number can be used and get faster convergence compared to the explicit method. In this thesis the results for Euler equation using the implicit scheme is validated and an efficient matrix form solving discretized equations is implemented for the code to run faster. The implicit scheme Euler equations are then extended to the Navier Stokes equations

1.0.1 IITK-DAE ANUPRAVAHA Compressible Solver

The ANUPRAVAHA Compressible solver was separated from the original ANUPRAVAHA incompressible solver to cater to aerospace applications exclusively. In this solver, the flow equations have been previously solved using the explicit MacCormack and AUSM+ schemes.The explicit MacCormack scheme, with artificial viscosity, proved to have very good accuracy and efficiency. It has been applied successfully for calculations of subsonic, transonic and supersonic flows over profiles and wings.

The main drawback of the explicit scheme is its time-step limitation due to the numerical stability condition. It becomes inefficient for unsteady flows where the global time-scale (e.g. period of oscillation of a wing) can be much larger than the time-step, and for the high-Reynolds viscous flows, where the mesh refinement in boundary layers results in extremely small time-steps. A computation with an explicit scheme requires substantial computer time.

Some implicit schemes have the advantage of being unconditionally stable, i.e., without CFL restrictions. Since the convergence to steady-state depends

on the propagation speed of the error waves, large CFL numbers accelerate the convergence to steady state. The implicit MacCormack scheme, therefore, is implemented in this thesis to facilitate faster convergence of unsteady and steady compressible flows.

1.0.2 Literature review

Hirsch (2007) has discussed the general methodology to analyze the nature of systems of partial differential equations. This systematic procedure to determine the nature of equations and the propagation of their solution is key to the understanding the implementation of boundary conditions. The second volume of Hirsch (2007) discusses almost all basic numerical schemes. such as central, upwinding and high-resolution schemes pertaining to Euler and Navier-Stokes equations. Euler equations are solved in conservative form but with boundary conditions prescribed in primitive form. In Chapter 19 Hirsch discusses the implementation of boundary conditions (both physical and numerical) from characteristic extrapolation for conservative and primitive variables, along with different extrapolation methods.

Implicit and semi-implicit schemes require a very powerful linear solver since the Jacobians usually lack diagonal dominance at least at high CFL numbers. This has an adverse effect on the convergence of many iterative solvers. Implicit solvers are still rarely used for the computation of stationary solutions to the Euler equations. However, their development has been pursued by several groups [[14], [15], [12], [22]]. Many existing schemes employ linearizable/differentiable

limiters, and are conditionally stable, and the rate of steady-state convergence deteriorates if the CFL number exceeds a certain upper bound. The scheme presented here converges for arbitrary CFL numbers despite oscillatory correction factors and the rate of steady-state convergence does not deteriorate for large CFL numbers. The implicit algorithm used in this work, avoids computationally expensive nonlinear iterations.

The development of robust and accurate boundary conditions is of primary importance, and sufficient care must be taken in the numerical implementation. The accuracy, robustness, stability, and convergence of an implicit solver are strongly influenced by the boundary treatment. A strong form of the governing equations along with boundary conditions states the conditions at every point over a domain, solution must satisfy. On the other hand a weak form states the condition that the solution must satisfy in an integral sense. Strongly imposed boundary conditions may inhibit convergence to a steady state. Thus, it is worthwhile to use flux boundary conditions of Neumann type. The weak type of boundary conditions turns out to be much more stable and flexible than its strong counterpart. When boundary conditions are prescribed in a weak sense, only the boundary integral of the weak formulation is affected by the boundary conditions, while the volume integrals remain unchanged. This is similar to the boundary treatment, which is usually implemented in finite volume schemes. In the finite volume framework the boundary fluxes are directly overwritten by the imposed boundary conditions.

The Neumann type of boundary conditions, based on the weak formulation,

can be treated implicitly and incorporated into the matrix in a physical way. It improves the convergence rates and does not affect the matrix properties or give rise to stability restrictions in contrast to the strong type of boundary conditions. According to [29], [27] a stability restriction of CFL number of 0.6 applies for an explicit implementation of weak wall boundary conditions, while the stability is significantly enhanced with a semi-implicit version up to a CFL number of 100. This emphasizes the importance of an implicit treatment of boundary conditions for the numerical performance, which is presented in this study. We recommend a boundary Riemann solver to compute the boundary fluxes in the boundary integrals to avoid unphysical effects particularly at large CFL numbers. To define a boundary Riemann problem the concept of ghost nodes is introduced. We show that a suitable treatment of boundary conditions makes it possible to achieve unconditional stability.

In the following chapters, the design procedure of an unconditionally stable finite volume scheme for the Euler equations are addressed. In the Euler equations, the treatment of boundary conditions based on a boundary Riemann solver is described, and the implicit solver is presented. Furthermore, the design procedure of implicit solver for Eulers and Navier Stokes equations are described. Finally, the numerical performance and accuracy of the proposed scheme are analyzed.

1.0.3 Objective of present work

The objectives of this thesis are manifold:

- To convert the system of governing equations to matrix form in the ANUPRAVAHA

solver, for more efficient computations

- To implement the implicit MacCormack methodology for the Euler Equations and to validate it for 2-D and 3-D geometries in sub-sonic, transonic and supersonic flows.

- To extend the implicit method solutions of Navier Stokes equations and validate the methods with solutions of supersonic wall-bounded flows.

- To integrate the explicit and Implicit Euler and Navier Stokes solvers into one general purpose code.

- To validate this solver for the three different regimes i.e., subsonic, transonic and supersonic flows.

Chapter 2

Governing Equations

2.1 The Flow and its Mathematical Description

Fluid dynamics is defined as the investigation of the interactive motion of a large number of individual particles (molecules or atoms). So, we can assume the density of the fluid is high enough and it can be approximated as a continuum. This means, even an infinitesimally small (in the sense of differential calculus) element of the fluid contains a sufficient number of particles, in terms of molecule or atoms, for which we can specify mean velocity and mean kinetic energy. In this way, we are able to define velocity, pressure, temperature, density and other important quantities at each point of the fluid.

The derivation of the principal equations of fluid dynamics depends upon the dynamical behaviour of a fluid, is determined by the following conservation laws:

1. The conservation of mass.

2. The conservation of momentum.

3. The conservation of energy.

The conservation of a certain flow quantity is based on the total variation of flow quantity inside an arbitrary volume and the net effect of the amount of the quantity being transported across the boundary due to any internal forces and sources and/or the external forces acting on the volume. The amount of the quantity crossing the boundary is called the *Flux*. The flux can be divided into two different parts: one due to the convective transport and the other one due to the molecular motion present in the fluid at rest.

Consider a general flow field as represented by streamlines in Fig. 2.1. An arbitrary finite region of the flow, bounded by the closed surface $\partial\upsilon$ and fixed in space, defines the control volume υ. We also consider a surface element dS and its associated, outward pointing unit normal vector $\vec{n}$ of the control surface which enclose the control volume υ.

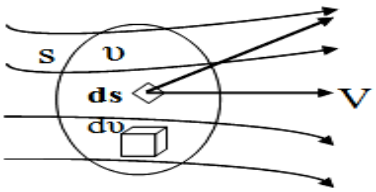


Figure 2.1: Definition of a finite control volume (fixed in space)

Let the conservation law applied to an scalar quantity per unit volume ϕ. Its variation in time within $\partial\upsilon$ can be written as,

$$\frac{\partial}{\partial t}\int_{\upsilon}\phi\, d\upsilon$$

This is equal to the sum of the contributions due to the convective flux which is the amount of the quantity ϕ entering the control volume through the boundary

with the velocity $\vec{u}$.

$$-\oint_{\partial v} \phi(\vec{u}.\vec{n})dS$$

The integral formulation of the conservation law is given by

$$\frac{\partial}{\partial t}\int_v \phi\, dv \;+\; \oint_{\partial v} \phi(\vec{u}.\vec{n})dS \;=\; 0 \tag{2.1}$$

2.1.1 Continuity Equation

If we consider only single-phase fluids, the law of mass conservation expresses as: mass cannot be created in such a fluid system, nor it can disappear. For the continuity equation, the conserved quantity ϕ is the density ρ. According to the general formulation of Eqn. 2.1, we can write the continuity equation as:

$$\frac{\partial}{\partial t}\int_v \rho\, dv \;+\; \oint_{\partial v} \rho(\vec{u}.\vec{n})\, dS \;=\; 0$$

2.1.2 Momentum Equation

The derivation of the momentum equation is based on the particular form of Newton's second law which states that the variation of momentum is caused by the net force acting on an mass element. The momentum of an infinitesimally small portion of the control volume v given by $\rho\vec{u}\, dv$. The variation in time of momentum within the control volume equals

$$\frac{\partial}{\partial t}\int_v \rho\vec{u}\, dv$$

Here $\rho\vec{u} \;=\; [\rho u \;\; \rho v \;\; \rho w]^T$, where u, v, w are the x, y and z components of the velocity, respectively.

In the conservation of momentum, the contribution of the convective tensor is given by

$$-\oint_{\partial v} \rho\vec{u}(\vec{u}.\vec{n})\, dS$$

Two types of forces act on the control volume: external volume or body forces and surface forces. Surface forces result from only two sources:

a) The pressure distribution, imposed by the outside fluid surrounding the volume.

b) The shear and normal stresses, resulting from the friction between the fluid and the surface of the volume.

Now sum up all the above contributions according to the general conservation law (Eqn. 2.1), and finally obtain the expression for momentum conservation equation

$$\frac{\partial}{\partial t}\int_v \rho\vec{u}\, dv + \oint_{\partial v} \rho\vec{u}(\vec{u}.\vec{n})\, dS = \int_v \rho\vec{f_e} - \oint_{\partial v} p\vec{n}\, dS + \oint_{\partial v} (\vec{\tau}.\vec{n})\, dS$$

where $\vec{f_e}$ body force per unit mass, p is the static pressure, τ is the stress tensor.

2.1.3 Energy Equation

The energy equation is based on the first law of thermodynamics. It states that the rate of change in the total energy inside the volume is equal to the rate of work of forces acting on the volume and by the net heat flux into it. The total energy per unit mass is defined E and we can write:

$$E = e + \frac{u^2 + v^2 + w^2}{2}$$

where e is internal energy per unit mass.

Similar to the momentum conservation equation we can write a conservative equation for the heat energy by accordingly for the rate of heat addition by conduction and volumetric heating and the work done by surface and body forces. The energy conservation equation according to the general conservation law (Eq. 2.1) is

$$\frac{\partial}{\partial t}\int_{\upsilon} \rho E \, d\upsilon + \oint_{\partial\upsilon} \rho E(\vec{u}.\vec{n}) \, dS \; = \; \oint_{\partial\upsilon} k(\nabla T.\vec{n}) \, dS \; + \; \int_{\upsilon} (\rho \vec{f_e}.\vec{u} + \dot{q_h}) - \oint_{\partial\upsilon} p(\vec{u}.\vec{n}) \, dS \; + \; \oint_{\partial\upsilon} (\vec{\tau}.\vec{u}).\vec{n} \, dS$$

where $\dot{q_h}$ is the rate of heat addition per unit volume and k is the thermal conduction of the fluid.

2.2 Euler Equations

The most general flow configuration for a non-viscous, non-heat conducting fluid is described by the set of Euler equations, obtained from the Navier Stokes equations by neglecting all shear stresses and heat conduction terms. If we collect the conservation laws of mass, momentum and energy into one system of equations neglecting the body forces and stress forces, we obtain the Euler Equations. The time-dependent Euler equations, in conservation form and in an absolute frame of reference, for the conservative variables U is:

$$\frac{\partial}{\partial t}\int_{\upsilon} U \, d\upsilon \; + \; \oint_{\partial\upsilon} \nabla.\vec{F} \, d\upsilon \; = \; 0 \tag{2.2}$$

which form a system of first order hyperbolic partial differential equations, where U is the solution vector

$$U = \begin{bmatrix} \rho \\ u \\ v \\ w \\ E \end{bmatrix}$$

and the flux vector F has the Cartesian components (f , g, h) given by equation 2.2

$$f = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (e+p)u \end{bmatrix} \; g = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (e+p)v \end{bmatrix} \; h = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (e+p)w \end{bmatrix}$$

Assuming the Control Volume (CV) is fixed in space, the governing integral equation can be written as,

$$\oint_{\partial v} (\frac{\partial U}{\partial t} + \nabla.\vec{F})\, dv = 0 \tag{2.3}$$

and further, as the CV is arbitrary, we can write,

$$\frac{\partial U}{\partial t} + \nabla.\vec{F} = 0 \tag{2.4}$$

2.3 Discretization Techniques and Grid Generation

In mathematics, discretization concerns the process of translating continuous functions, models and equations into discrete counterparts. This process is usually carried out as a first step toward making them suitable for numerical evaluation and implementation on digital computers The discretization techniques use

grids in order to discretize the governing equations 2.2, 2.4. Basically, there are two different types of grids:

- Structured Grids: each grid point (vertex, node) is uniquely identified by the indices i, j, k and the corresponding Cartesian coordinates $x_{i,j,k}, y_{i,j,k}, and z_{i,j,k}$. The grid cells are quadrilaterals in 2D and hexahedral in 3D.

- Unstructured Grids: grid cells as well as grid points have no particular ordering, i.e., neighbouring cells or grid points cannot be directly identified by their indices (usually, only a single index is used). In the past, the grid cells were triangles in 2D and tetrahedral in 3D. Nowadays unstructured grids usually consist of a mix of quadrilaterals and triangles in 2D and of hexahedral, tetrahedral, prisms and pyramids in 3D.

Here we use structured grids to solve the governing equations. The main advantage of structured grids is that the indices i, j, k represent a linear address space, since it directly corresponds to how the flow variables are stored in the computer memory. This property allows it to access the neighbours of a grid point very quickly and easily, just by adding or subtracting an integer value to or from the corresponding index (e.g. $(i+1)$, $(j-3)$, etc . see Fig. 2.2). The evaluation of gradients, fluxes, and also the treatment of boundary conditions is simplified by this feature. The same holds for the implementation of an implicit scheme, because of the well-ordered, banded flux Jacobian matrix.

But there is also a disadvantage. The disadvantage is the time-consuming and complicated task required for the generation of structured grids for complex

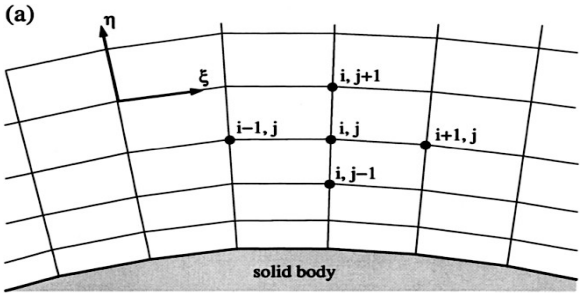


Figure 2.2: Structured, body-fitted grid approach (in two dimensions)

geometries. Another difficulty is that generating *good* grids with regular cells of moderate skewness and aspect ratios, is very difficult if the entire complex domain is fitted with a single block grid. To overcome this disadvantage we can divide the physical space into a number of topologically simpler parts or blocks (see 2.3), which can be more easily meshed. This is called the multiblock mesh. In this thesis we use multiblock approach to generate the mesh.

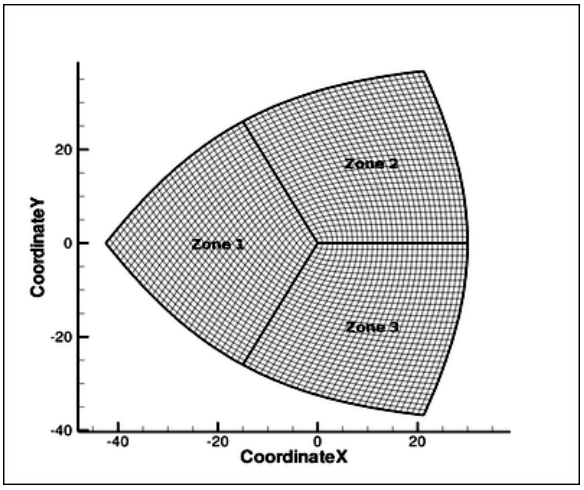


Figure 2.3: Structured, multiblock grid

The advantage of this approach is that, grid lines can be chosen separately for each block as required to be close to rectangular, or orthogonal, which increase

numerical accuracy and convergence. The another advantage of the multiblock methodology is that it allows for the possibility of using parallel computation by means of domain decomposition.

The discretization schemes used in CFD can be divided into the following main categories:

- Finite Difference Method: which can be applied to rectangular structured mesh configurations.
- Finite Volume Method: which can be applied to both structured and unstructured mesh configurations.
- Finite Element Method: which is the common method in solid mechanics, but is also applicable to fluid mechanics, which is applied to unstructured grids.

2.3.1 Finite Difference Method

The finite difference method was the first approaches applied to the numerical solution of differential equations. It was first utilized by Leonhard Euler in 1768 [17]. This method is directly applied to the differential form of the governing equations 2.4.

For a function $U(x)$, the Taylor series expansion of $U_{x_0+\Delta x}$ in x can be written as

$$U_{(x_0+\Delta x)} = U_{(x_0)} + \Delta x \left(\frac{\partial U}{\partial x}\right)_{x_0} + \frac{\Delta x^2}{2}\left(\frac{\partial^2 U}{\partial x^2}\right)_{x_0} + ...$$

From the above equation, the first derivative of U can be approximated as

$$\left(\frac{\partial U}{\partial x}\right)_{x_0} = \frac{U_{x_0+\Delta x} - U_{x_0}}{\Delta x} + \bigcirc(\Delta x) \tag{2.5}$$

The above approximation is of first order, since the truncation error (abbreviated as $\bigcirc(\Delta x)$), which is proportional to the largest term of the remainder, goes to zero with the first power of Δx.

To apply this general definition 2.5, we consider an one-dimensional space, the x-axis, and the space discretization is done with N discrete mesh points x_i, i = 0,...,N (Figure 2.4).

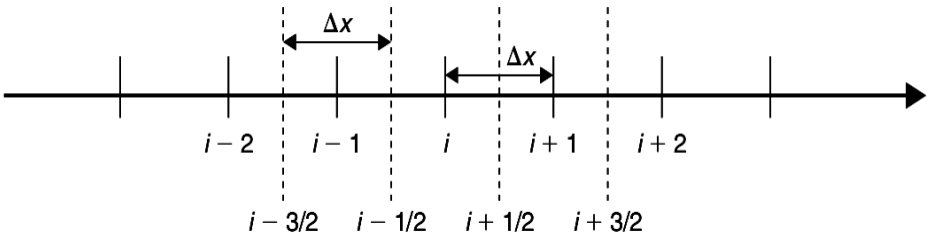


Figure 2.4: One-dimensional uniform FDM grid on the x-axis [17]

Let U_i is the value of the function U_{x_0} at the point x_i , i.e. $U_i = U_{x_i}$ and the spacing between the discrete points is constant and equal to Δx. Applying the above relation 2.5 at point i, we obtain the following finite difference approximation

$$(U_x)_i = \left(\frac{\partial U}{\partial x}\right)_i = \frac{U_{i+1} - U_i}{\Delta x} \underbrace{- \frac{\Delta x}{2}\left(\frac{\partial^2 U}{\partial x^2}\right)_i - \frac{\Delta x^2}{6}\left(\frac{\partial^3 U}{\partial x^3}\right)_i + ...}_{\text{Truncation error}} \tag{2.6}$$

$$= \frac{U_{i+1} - U_i}{\Delta x} + \bigcirc(\Delta x) \tag{2.7}$$

As this formula involves the point $(i+1)$ to the right of point i, it is called the first order forward difference for the first derivative U_{x_i}.

Now if Δx is replaced by $-\Delta x$, then the finite difference approximation is

$$(U_x)_i = \left(\frac{\partial U}{\partial x}\right)_i = \frac{U_i - U_{i-1}}{\Delta x} \underbrace{+ \frac{\Delta x}{2}\left(\frac{\partial^2 U}{\partial x^2}\right)_i - \frac{\Delta x^2}{6}\left(\frac{\partial^3 U}{\partial x^3}\right)_i + ...}_{\text{Truncation error}} \quad (2.8)$$

$$= \frac{U_i - U_{i-1}}{\Delta x} + \bigcirc(\Delta x) \quad (2.9)$$

This formula is called the first order backward difference for the derivative U_{x_i} as it involves the point (i-1) to the left of point i. If we add this two equations (eqs 2.6 and 2.8), we obtain a second order approximation

$$(U_x)_i = \frac{U_{i+1} - U_{i-1}}{2\Delta x} - \frac{\Delta x^2}{6}\left(\frac{\partial^3 U}{\partial x^3}\right)_i + ... \quad (2.10)$$

$$= \frac{U_{i+1} - U_{i-1}}{2\Delta x} + \bigcirc(\Delta x^2) \quad (2.11)$$

Equation 2.10 involves the points to the left and to the right of point i, is therefore called a central difference formula.

The important advantages of the finite difference methodology are its simplicity and the possibility to obtain high-order approximations easily to achieve greater accuracy of the spatial discretization. The main disadvantage of this method is, it requires a structured rectangular grid, so the range of application is restricted. Furthermore, the finite difference method cannot be directly applied in body-fitted i.e curvilinear coordinates. So first we have to transform the governing equations into a rectangular grid system or in other words transform the physical

to the computational space. Thus, the finite difference method can be applied only to rather simple geometries.

2.3.2 Finite Volume Formulation

The finite volume method directly makes use of the conservation laws, the integral formulation of the Euler equations(eq 2.2). It was first employed by McDonald for the simulation of 2-D inviscid flows [17]. The finite volume method discretizes the governing equations by first dividing the physical space into a number of arbitrary polyhedral control volumes. The surface integral is Equation 2.2 is then approximated by the sum of the fluxes crossing the individual faces of the control volume. The accuracy of this spatial discretization depends on the particular scheme with which the fluxes are evaluated.

Additionally, complicated boundary conditions for complex flow domains can be implemented in a relatively straight-forward manner.

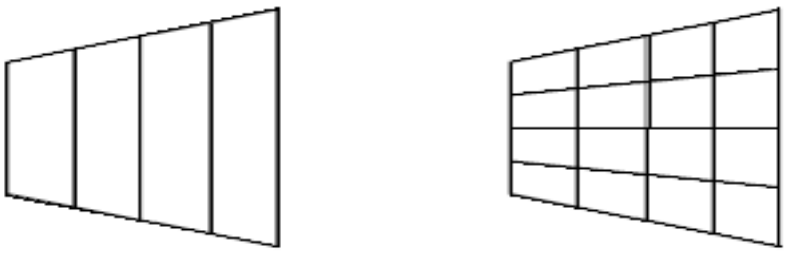


Figure 2.5: 1d (left) and 2d (right) Finite Volume discretization of an expanding domain

Figure 2.5 shows an example of a 1-D and 2-D finite volume discretization for an expanding flow domain. Algebraic equations can be obtained for each control volume by approximating the volume and surface integrals using quadrature

formulae. Volume integrals can be evaluated with second order accuracy by the product of the mean value of ϕ, assumed to be at the cell centroid, and the cell volume whilst surface integrals are calculated by summation over the sides of the cell. The integral on each face being approximated by the midpoint rule. The semi-discrete form of the governing equations are written for each cell as

$$\frac{\partial U_{cell-centered}}{\partial t} = -\frac{1}{V}\sum_{if} F_{if} A_{if}$$

with A and V being the cell edge interface area and cell volume respectively. The discretized equations applied to each control volume can be advanced in time from an initial solution once a technique for determining the interface fluxes is specified.

There are two basic approaches of defining the shape and position of the control volume with respect to the grid:

- Cell centered scheme (Fig: 2.6(a)): Here the flow quantities are stored at the centroids of the grid cells. So, the control volumes are identical to the grid cells. We use the cell-centered scheme in this thesis.

- Cell vertex scheme (Fig: 2.6(b)): Here the flow variables are stored at the grid points. The control volume can then either be the union of all cells sharing the grid point, or some volume centered around the grid point.

The main advantage of the finite volume method is that the spatial discretization is carried out directly in the physical space. Thus, there are no problems associated with transformation between the physical and the computational coordinate

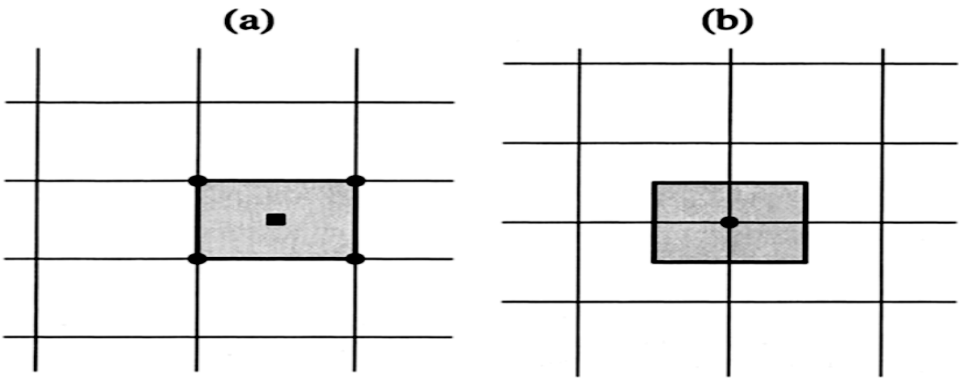


Figure 2.6: Control volume of cell centered (a) and cell vertex (b) scheme

system, as in the case of the finite difference method. Another advantage of the finite volume method, compared to the finite difference method is that it is very flexible, and can be rather easily implemented on structured as well as on unstructured grids. This makes the finite volume method particularly suitable for the treatment of flows in complex geometries.

The finite volume method is based on the direct discretization of the integral conservation laws, mass, momentum and energy, which are also conserved by the numerical scheme. So it has the ability to compute weak solutions of the governing equations correctly. This is the another important feature of the method, However, one additional condition is needed to be fulfilled in the case of the Euler equations, known as the entropy condition. It is necessary because of the non-uniqueness of the weak solutions. The entropy condition prevents the occurrence of unphysical features like expansion shocks, which violate the second law of thermodynamics (by decrease of entropy).

Under certain conditions, the finite volume method can be shown to be equiv-

alent to the finite difference method, or to a low-order finite element method.

2.4 Time Integration

For a given current flow state, the discretized equations can be advanced in time by selecting an appropriate numerical integration technique. Schemes are classified as being either explicit, implicit or a mixture of the two. Explicit integration uses knowledge of only the current flow state to determine the new state at the next time-step and as such is not very computationally intensive. The equations are advanced in small time steps governed by strict stability criteria. For example, a wave starting at a cell interface should not cross more than half of the cell width during a time step. Implicit integration uses knowledge of both the (known) current flow state and the (unknown) next time step state. So each time step is computationally more expensive than an explicit method because the equations for all cells have to be solved simultaneously. But implicit methods have advantages in stability, allowing larger time-steps to be used in the computations.

Mathematically, if $Y(t)$ is the current system state and $Y(t+\Delta t)$ is the state at the later time (Δt is a small time step), then for an explicit method for the PDE

$$\frac{\partial Y}{\partial t} \;=\; F(y)$$

is

$$Y(t+\Delta t) \;= Y(t) \;+\; F(Y(t))$$

while for an implicit method one solves an equation

$$Y(t + \Delta t) = Y(t) + F(Y(t + \Delta t))$$

The main drawback of explicit schemes is that the stability requirements can result in very short time steps and correspondingly long computation times. Implicit methods are used because many problems arising in practice are stiff, for which the use of an explicit method requires impractically small time steps to keep the error in the result bounded. For such problems, to achieve given accuracy, it takes much less computational time if we use an implicit method with larger time steps.

For the flows considered in this thesis, we use an implicit technique for time integration to reduce the computational time and compare it with explicit scheme's computational time.

2.5 Closure

In this chapter we discussed the basic nature of the governing equations for the flow problems and different discretization techniques of solving these governing equations. We also discussed the way of time integration.

Chapter 3

Navier Stokes Equation and Matrix Form

3.1 Navier Stokes Equation

The Navier-Stokes equations represent in three dimensions a system of five equations for the five conservative variables ρ, ρu, ρv, ρw, and ρE. But they contain seven unknown flow field variables, namely: ρ, u, v, w, E, p, and T. Therefore, we have to supply two additional equations, the first is the equation of state which prescribes the thermodynamic relations between the state variables, the second is an equation relating the total energy, E with the temperature, T. For example, for an ideal gas the pressure can be expressed as a function of the density and temperature, and the total energy as a function of the temperature. Beyond this, we have to provide the viscosity coefficient μ and the thermal conductivity coefficient k as functions of the state of the fluid. Clearly, the relationships depend on the kind of fluid being considered. In the following, we shall therefore show methods of closing the equations for two commonly encountered situations.

The Navier-Stokes and energy equations for compressible flow of an ideal gas are :

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} = 0 \tag{3.1}$$

$$\frac{\partial \rho u}{\partial x} + \frac{\partial (\rho u^2 + P - \tau_{xx})}{\partial x} + \frac{\partial (\rho uv - \tau_{xy})}{\partial y} + \frac{\partial (\rho uw - \tau_{xz})}{\partial z} = 0 \tag{3.2}$$

$$\frac{\partial (\rho v)}{\partial t} + \frac{\partial (\rho uv - \tau_{yx})}{\partial x} + \frac{\partial (\rho v^2 + P - \tau_{yy})}{\partial y} + \frac{\partial (\rho vw - \tau_{yz})}{\partial z} = 0 \tag{3.3}$$

$$\frac{\partial (\rho w)}{\partial t} + \frac{\partial (\rho wu - \tau_{zx})}{\partial x} + \frac{\partial (\rho wv - \tau_{zy})}{\partial y} + \frac{\partial (\rho w^2 + P - \tau_{zz})}{\partial z} = 0 \tag{3.4}$$

$$\begin{aligned} \frac{\partial (E_t)}{\partial t} + \frac{\partial ((E_t + p)u + q_x - u\tau_{xx} - v\tau_{xy} - w\tau_{xz})}{\partial x} + \\ \frac{\partial ((E_t + p)v + q_y - u\tau_{yx} - v\tau_{yy} - w\tau_{yz})}{\partial y} + \\ \frac{\partial ((E_t + p)w + q_z - u\tau_{zx} - v\tau_{zy} - w\tau_{zz})}{\partial z} = 0 \end{aligned} \tag{3.5}$$

where $P = \rho RT$,$E_t = \rho(C_v T + \frac{u^2+v^2+w^2}{2})$,$H = E + \frac{P}{\rho}$

The mathematical nature of steady and unsteady nature of Navier-Stokes equations are stated below

- Compared to the Euler equations, the presence of viscosity and heat conduction transforms the conservation laws of momentum and energy into second-order partial differential equations
- The unsteady continuity equation is hyperbolic, for compressible flow where it is considered as an equation for the density, on the other hand, the steady continuity equation is elliptic

- Unsteady momentum and energy equations are parabolic and steady momentum and energy equations have elliptic behavior

- The coupled system of the Navier-Stokes equations is therefore a hybrid system, being parabolic-hyperbolic for the unsteady case but becoming elliptic for the stationary formulation

From the computational point of view, usually we are more interested in steady state than transient solutions. Therefore, while solving the steady state equations we have to check for the sonic condition, as the numerical schemes for each type of PDE are different.Till date no scheme has been developed for the steady state solution which can work well for all these types of PDEs. So there need to be completely separate modules to be developed for subsonic and supersonic flows, while for transonic flows it would be even more difficult to obtain solutions since the domain will contain all three types of PDEs.

However, by retaining the time derivative term in this system of equations makes it hyperbolic / parabolic independent of the speed of flow. Therefore, even if we are interested in only in the steady state solution, it is best to solve the transient set of equations to reach steady state. This is called the false-transient approach and is used in this work.

3.2 Matrix Form

These system of equations can be solved one by one or all at once. In the previous version of the ANUPRAVAHA solver for explicit Euler Navier Stokes equations, each equation was solved one at a time. It was found that the time taken for each time step is almost three to four times slower than it should be when the equations are solved at once. So in order to speed up the code the system of equations are changed to matrix form which means the equations are solved simultaneously in vector / matrix from and thereby reduce the time taken for each iteration.

The Euler equations which describes the inviscid compressible fluid motion can be presented in conservation form as

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} = 0 \tag{3.6}$$

$$\frac{\partial \rho u}{\partial x} + \frac{\partial (\rho u^2 + P)}{\partial x} + \frac{\partial (\rho uv)}{\partial y} + \frac{\partial (\rho uw)}{\partial z} = 0 \tag{3.7}$$

$$\frac{\partial (\rho v)}{\partial t} + \frac{\partial (\rho uv)}{\partial x} + \frac{\partial (\rho v^2 + P)}{\partial y} + \frac{\partial (\rho vw)}{\partial z} = 0 \tag{3.8}$$

$$\frac{\partial (\rho w)}{\partial t} + \frac{\partial (\rho wu)}{\partial x} + \frac{\partial (\rho wv)}{\partial y} + \frac{\partial (\rho w^2 + P)}{\partial z} = 0 \tag{3.9}$$

$$\frac{\partial (\rho E)}{\partial t} + \frac{\partial (\rho uH)}{\partial x} + \frac{\partial (\rho vH)}{\partial y} + \frac{\partial (\rho wH)}{\partial z} = 0 \tag{3.10}$$

where $P = \rho RT$,$E = C_v T + \frac{u^2+v^2+w^2}{2}$,$H = E + \frac{P}{\rho}$

Now for changing to matrix form consider

$$W \equiv \left\{ \begin{array}{c} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{array} \right\}$$

$$F_x \equiv \left\{ \begin{array}{c} \rho u \\ \rho u^2 + P \\ \rho uv \\ \rho uw \\ \rho uH \end{array} \right\}, F_y \equiv \left\{ \begin{array}{c} \rho v \\ \rho uv \\ \rho v^2 + P \\ \rho vw \\ \rho vH \end{array} \right\}, F_z \equiv \left\{ \begin{array}{c} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + P \\ \rho wH \end{array} \right\}$$

And so the equations in compact form become

$$\frac{\partial \{W_i\}}{\partial t} + \frac{\partial \{F_{xi}\}}{\partial x} + \frac{\partial \{F_{yi}\}}{\partial y} + \frac{\partial \{F_{zi}\}}{\partial z} = 0 \tag{3.11}$$

Now solving this equation with the matrices of conservative variables and fluxes in x,y,z directions, the time taken for each timestep is reduced by a significant amount. The results of time comparision will be discussed in the results section.

3.3 Boundary Conditions

Specification of boundary conditions are different for hyperbolic problems compared to that of parabolic and elliptic problems, and the flow of characteristics into or out of the computational domain affects the specification of the boundary conditions.

The key to understand the issue of number of boundary conditions that are needed at the boundary is that characteristics convey information in the $x - t$ space formed by the local normal direction and time. When information is introduced from outside into the computational domain, this information has to enter through a boundary condition; it can be shown that this occurs when the eigenvalue λ of the matrix of fluxes is positive at the boundary, and a physical

boundary condition has to be imposed [11]. On the other hand, when the eigenvalue λ is negative and the propagation occurs from the interior of the domain outwards from the boundary, this means that a boundary condition cannot be imposed from the outside. Such variable will be handled through "numerical boundary conditions", by extrapolating interior information to the boundary.

In summary, the number of physical conditions to be imposed at a boundary with inward normal vector $\vec{n}$, pointing into the computational domain, is defined by the number of characteristics entering the domain. For subsonic domain one of the eigenvalue will be less than zero and so one boundary condtion will be numerical and all others will be physical boundary conditions.

3.4 Closure

In this chapter Euler and Navier-stokes equations are discussed. The advantage of the matrix form of governing equation over the non-matrix form is mentioned and lastly how to specify boundary conditions is discussed.

Chapter 4

Implicit Approach to Solve Navier Stokes Equation

Real flow includes rotational, non-isentropic, and non-isothermal effects. Compressible inviscid flow including such effects requires simultaneous solution of continuity, momentum, and energy equations. Special computational schemes are required to resolve the shock discontinuities encountered in transonic flow. Another basic requirement for the solution of these equations is to ensure that solution schemes provide an adequate amount of artificial viscosity required for correct and rapid convergence towards a solution. In the present work, the Implicit MacCormack scheme has been chosen to solve the Navier stokes equations, since it is a very robust and tested scheme.

4.1 Governing Equations

The equations which describes the compressible fluid motion can be presented in conservation form as,

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} = 0 \tag{4.1}$$

$$\frac{\partial \rho u}{\partial x} + \frac{\partial (\rho u^2 + P - \tau_{xx})}{\partial x} + \frac{\partial (\rho uv - \tau_{xy})}{\partial y} + \frac{\partial (\rho uw - \tau_{xz})}{\partial z} = 0 \tag{4.2}$$

$$\frac{\partial (\rho v)}{\partial t} + \frac{\partial (\rho uv - \tau_{yx})}{\partial x} + \frac{\partial (\rho v^2 + P - \tau_{yy})}{\partial y} + \frac{\partial (\rho vw - \tau_{yz})}{\partial z} = 0 \tag{4.3}$$

$$\frac{\partial (\rho w)}{\partial t} + \frac{\partial (\rho wu - \tau_{zx})}{\partial x} + \frac{\partial (\rho wv - \tau_{zy})}{\partial y} + \frac{\partial (\rho w^2 + P - \tau_{zz})}{\partial z} = 0 \tag{4.4}$$

$$\begin{aligned} \frac{\partial (E_t)}{\partial t} + \frac{\partial ((E_t + p)u + q_x - u\tau_{xx} - v\tau_{xy} - w\tau_{xz})}{\partial x} + \\ \frac{\partial ((E_t + p)v + q_y - u\tau_{yx} - v\tau_{yy} - w\tau_{yz})}{\partial y} + \\ \frac{\partial ((E_t + p)w + q_z - u\tau_{zx} - v\tau_{zy} - w\tau_{zz})}{\partial z} = 0 \end{aligned} \tag{4.5}$$

where $P = \rho RT$, $E_t = \rho(C_v T + \frac{u^2+v^2+w^2}{2})$, $H = E + \frac{P}{\rho}$

4.2 Discretization of Governing Equation

The equations can be written in compact form as

$$\frac{\partial \{W\}}{\partial t} + \frac{\partial \{F_x\}}{\partial x} + \frac{\partial \{F_y\}}{\partial y} + \frac{\partial \{F_z\}}{\partial z} = 0 \tag{4.6}$$

$$\{W\} \equiv \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{Bmatrix}$$

$$\{F_x\} \equiv \begin{Bmatrix} \rho u \\ \rho u^2 + P - \tau_{xx} \\ \rho uv - \tau_{xy} \\ \rho uw - \tau_{xz} \\ (E_t + p)u + q_x - u\tau_{xx} - v\tau_{xy} - w\tau_{xz} \end{Bmatrix},$$

$$\{F_y\} \equiv \begin{Bmatrix} \rho v \\ \rho uv - \tau_{yx} \\ \rho v^2 + P - \tau_{yy} \\ \rho vw - \tau_{yz} \\ (E_t + p)v + q_y - u\tau_{yx} - v\tau_{yy} - w\tau_{yz} \end{Bmatrix},$$

$$\{F_z\} \equiv \begin{Bmatrix} \rho w \\ \rho uw - \tau_{zx} \\ \rho vw - \tau_{zy} \\ \rho w^2 + P - \tau_{zz} \\ (E_t + p)w + q_z - u\tau_{zx} - v\tau_{zy} - w\tau_{zz} \end{Bmatrix}$$

Note that the $\{F_x\}$, $\{F_y\}$, $\{F_z\}$ column vectors are used just for notational convenience. where W, Fx, Fy, Fz will represent the values of these column vectors for a given row. It is to be noted Fx, Fy, Fz can be treated as components of a physical vector $\overrightarrow{F}$ we can write

$$\frac{\partial W}{\partial t} + \nabla \cdot \mathbf{F} = 0 \tag{4.7}$$

which applies to each row of the equation(4.2).

The finite volume method uses the integral form of the equations while the governing equation above is in differential form. The corresponding integral form of the equation can be obtained by taking the integral of the equation over a control volume.

$$\oint_V \left(\frac{\partial W}{\partial t} + \nabla \cdot \mathbf{F} \right) \mathrm{d}V = 0$$

where V is the fluid domain under analysis. Using the divergence theorem, $\oint_V \nabla \cdot \overrightarrow{v} \mathrm{d}V = \oint_S \overrightarrow{v} \cdot \mathrm{d}\overrightarrow{S}$ we get

$$\oint_V \frac{\partial W}{\partial t} \mathrm{d}V + \oint_S \mathbf{F} \cdot \mathrm{d}\overrightarrow{S} \;=\; 0$$

Assuming the control volume is not changing with time, the equation can be written as,

$$\frac{\partial}{\partial t}\oint_V W \mathrm{d}V + \oint_S \mathbf{F}\cdot \mathrm{d}\overrightarrow{S} = 0$$

The equation can be divided into the temporal and convective parts, as shown, and we will now do the finite volume discretization of each part to get the full discretized equation.

$$\underbrace{\frac{\partial}{\partial t}\oint_V W \mathrm{d}V}_{Temporal\ Part} + \underbrace{\oint_S \mathbf{F}\cdot \mathrm{d}\overrightarrow{S}}_{Convective\ part} = 0$$

Temporal term:

The volume averaged value of conservative variable can be written for the pth cell as:

$$\oint_{V_p} W \mathrm{d}V = V_p W_p$$

where, V_p is the volume of the pth cell, and W_p is the value of its cell center.

Using this volume averaged value we can get the discretized form of the temporal term as:

$$\int dV \frac{\partial W}{\partial t} = V_p \frac{W_p^{n+1} - W_p^n}{\Delta t}$$

Convective term:

There are two methods to calculate the value of convective part at the new time level depending upon time value of the flux as,

1. Implicit: where the flux variable are taken to be at the new (unknown) time-level.

2. Explicit: where the flux variable are taken to be at the old (known) time-level.

In this study the implicit method is applied to discretize the convective part.

Equation 4.7 is integrated in time by using implicit method and written as,

$$V \frac{W^{n+1} - W^n}{\Delta t} + \int_V dV \bigtriangledown.(\mathbf{F}^{n+1}) = 0 \tag{4.8}$$

with a time step of size Δt. The superscript n refers to current time level and the result is a nonlinear system of algebraic equations, which calls for nonlinear iterations in each time step. But nonlinear iterations are computationally expensive and have poor convergence. To overcome this problem, we assumes sufficient smoothness and linearizes the equations around the current solution W^n by a Taylor series expansion of the fluxes

$$\mathbf{F}^{n+1} = \mathbf{F}^n + \left(\frac{\partial \mathbf{F}}{\partial W}\right)^n (W^{n+1} - W^n) + \bigcirc(\left\|W^{n+1} - W^n\right\|^2) \tag{4.9}$$

Substitution of equation 4.9 into the nonlinear equations 4.8 leads to a linear

algebraic system

$$V_P \frac{W^{n+1} - W^n}{\Delta t} + \int_V dv \bigtriangledown . \left(\mathbf{F}^n + \left(\frac{\partial \mathbf{F}}{\partial W} \right)^n \left(W^{n+1} - W^n \right) \right) = 0$$

$$or \; V_P \frac{W^{n+1} - W^n}{\Delta t} + \int_V dv \bigtriangledown . \left(\frac{\partial \mathbf{F}}{\partial W} \right)^n \left(W^{n+1} - W^n \right) = \int_V dv \left(-\bigtriangledown . \mathbf{F}^n \right)$$

Consider

$$\delta W^{n+1} \equiv W^{n+1} - W^n$$

Substituting δW^{n+1} in main equation,

$$V_P \frac{\delta W^{n+1}}{\Delta t} + \int_V dv \bigtriangledown . \left(\frac{\partial \mathbf{F}}{\partial W} \right)^n \delta W^{n+1} = \int_V dv (-\bigtriangledown . \mathbf{F}^n)$$

$$\left[I + \frac{\Delta t}{V} \int_V dv \bigtriangledown . \left(\frac{\partial \mathbf{F}}{\partial W} \right)^n \right] \delta W^{n+1} = -\frac{\Delta t}{V} \oint_S \mathbf{F}^n \cdot \mathrm{d}S$$

where $\frac{\partial \mathbf{F}}{\partial W_j}$ is the Jacobian of flux $\mathbf{F}$.

In the convective term, the explicit term $\bigtriangledown . \mathbf{F}^n$ the integral is carried out over the full surface of the control volume, without any approximation it can be divided into six parts over the east(e), west(w), north(n), south(s), top(t) and bottom(b) faces as follows:

$$\oint_{Sf} \mathbf{F} \cdot \mathrm{d}\overrightarrow{S} = \oint_e F_e \cdot \mathrm{d}S_e + \oint_w F_w \cdot \mathrm{d}S_w + \oint_n F_n \cdot \mathrm{d}S_n + \oint_s F_s \cdot \mathrm{d}S_s + \oint_t F_t \cdot \mathrm{d}S_t + \oint_b F_b \cdot \mathrm{d}S_b$$

where each face integral can be divided, without approximation, into 3 scalar parts:

$$\oint_{Sf} \mathbf{F} \cdot \mathrm{d}S_f = \oint_S F_x \mathrm{d}S_x + \oint_S F_y \mathrm{d}S_y + \oint_S F_{iz} \mathrm{d}S_z$$

The value of flux variable may change over the surface. For each scalar component, we now approximate the surface averaged value of the variable by its face-centroid value F_{if}:

$$\frac{1}{S_f}\oint_{Sf} F_i \mathrm{d}\overrightarrow{S}_f \;=\; F_{if}$$

Therefore we can write,

$$\oint_{Sf} \mathbf{F}\cdot \mathrm{d}\overrightarrow{S}_f = F_x S_{fx} + F_y S_{fy} + F_z S_{fz}$$

where S_{fi} is the i^{th} component of face vector $\overrightarrow{S}_f$. Repeating the procedure for each of the faces we can write

$$\begin{aligned}\oint_{Sf} \overrightarrow{F}\cdot \mathrm{d}\overrightarrow{S}_f = {} & F_{ex}S_{ex} + F_{ey}S_{ey} + F_{ez}S_{ez} + F_{wx}S_{wx} + F_{wy}S_{wy} + F_{wz}S_{wz} \\ & + F_{nx}S_{nx} + F_{ny}S_{ny} + F_{nz}S_{nz} + F_{sx}S_{sx} + F_{sy}S_{sy} + F_{sz}S_{sz} \\ & + F_{tx}S_{tx} + F_{ty}S_{ty} + F_{tz}S_{tz} + F_{bx}S_{bx} + F_{by}S_{by} + F_{bz}S_{bz}\end{aligned}$$

Now, putting the discretized convective terms together, the explicit term can be written in discretized form as:

$$\Delta F^n = -\sum_f \left(F_{fx}S_{fx} + F_{fy}S_{fy} + F_{fz}S_{fz}\right) \tag{4.10}$$

MacCormack [22] proposed a two-step approach to solve the wave equation, with a finite-difference method. It is known to be a robust scheme that gives stable results with good accuracy when provided with some artificial dissipation. As the scheme is a finite-difference method, we need to modify it for the finite-

volume method, which shall be done below. First, however, we will introduce the MacCormack finite-difference scheme for the wave equation, and then extend it to the finite-volume method for full Navier-Stokes equations in the later sections.

4.3 Explicit MacCormack Finite Difference Scheme

MacCormack′s scheme solves hyperbolic problems in two steps, popularly known as the predictor-corrector approach. It falls in the category of multi-step central schemes.

Consider a simple one dimensional model initial value problem in 1D:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \nu\frac{\partial u^2}{\partial^2 x} \tag{4.11}$$

with an initial condition u(x,0) $= u_0(x)$ The explicit MacCormack scheme is realized in two steps:

Predictor:

$$\begin{aligned} \triangle u_i^n &= -\frac{c\triangle t}{\triangle x}\left(u_{i+1}^n - u_i^n\right) + \frac{\Delta t\nu}{\Delta x^2}\left(u_{i+1}^n + 2u_i^n + u_{i-1}^n\right) \\ u_i^{\overline{n+1}} &= u_i^n + \triangle u_i^n \end{aligned} \tag{4.12}$$

where $u_i^{\overline{n+1}}$ is the so-called "predicted" value of the solution at the $n+1$ time-level, obtained explicitly in step 1 and $\triangle u_i^n \equiv u_i^{\overline{n+1}} - u_i^n$,

Corrector:

$$\begin{aligned} \triangle u_i^{\overline{n+1}} &= -\frac{c\triangle t}{\triangle x}\left(u_i^{\overline{n+1}} - u_{i-1}^{\overline{n+1}}\right) + \frac{\Delta t\nu}{\Delta x^2}\left(u_{i+1}^{\overline{n+1}} + 2u_i^{\overline{n+1}} + u_{i-1}^{\overline{n+1}}\right) \\ u_i^{n+1} &= \frac{1}{2}\left(u_i^n + u_i^{\overline{n+1}} + \triangle u_i^{\overline{n+1}}\right) \end{aligned} \tag{4.13}$$

The explicit scheme is stable under the CFL condition:

$$\triangle t \leq \frac{1}{(c/\triangle x) + 2\nu/\triangle x^2}$$

4.4 The Implicit Scheme

MacCormack also propesed an implicit version of the above scheme, that is not so commonly used. The implicit scheme is obtained by replacing one-sided difference in the convective terms

$$u_i^{n+1} = u_i^n - (1-\alpha)\frac{c\triangle t}{\triangle x}\left(u_{i+1}^n - u_i^n\right) + \alpha\frac{c\triangle t}{\triangle x}\left(u_{i+1}^{n+1} - u_i^{n+1}\right) + \frac{\nu\Delta t}{\Delta x^2}(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1})$$

or

$$\left(1 + \frac{\lambda\triangle t}{\triangle x}\right)\delta u_{i+1}^{n+1} = \frac{c\triangle t}{\triangle x}\Delta_+ u_i^n + \frac{\lambda\triangle t}{\triangle x}\delta u_i^{n+1} + \frac{\nu\Delta t}{\Delta x^2}(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1})$$

where

$$\delta u_i^{n+1} \equiv u_i^{n+1} - u_i^n, \qquad \Delta_+ u_i^n = u_{i+1}^n - u_i^n, \qquad \lambda = \alpha\,|c|$$

where α is the implicit blending parameter which is greater than 0.5.

Considering ν =0 for simplicity, the predictor and corrector steps of the implicit scheme are:

Predictor:

$$\begin{aligned} \triangle u_i^n &= -\frac{a\triangle t}{\triangle x}\left(u_{i+1}^n - u_i^n\right) \\ \left(1 + \lambda\frac{\triangle t}{\triangle x}\right)\delta u_i^{\overline{n+1}} &= \triangle u_i^n + \lambda\frac{\triangle t}{\triangle x}\delta u_{i+1}^{\overline{n+1}} \\ u_i^{\overline{n+1}} &= u_i^n + \delta u_i^{\overline{n+1}} \end{aligned} \tag{4.14}$$

Corrector:

$$\begin{aligned} \triangle \overline{u_i^{n+1}} &= -\frac{a\triangle t}{\triangle x}\left(\overline{u_i^{n+1}} - \overline{u_{i-1}^{n+1}}\right) \\ \left(1+\lambda\frac{\triangle t}{\triangle x}\right)\delta u_i^{n+1} &= \triangle \overline{u_i^{n+1}} + \lambda\frac{\triangle t}{\triangle x}\delta u_{i-1}^{n+1} \\ u_i^{n+1} &= \frac{1}{2}\left(u_i^n + \overline{u_i^{n+1}} + \delta u_i^{n+1}\right) \end{aligned} \tag{4.15}$$

The predictor step is evaluated starting at the greatest index i using an appropriate boundary condition and going to the lowest index. The corrector step is evaluated in the similar manner starting with boundary condition for lowest index and going to greatest one.

The linear scheme is unconditionally stable provided that the implicit blending parameter λ is chosen such that

$$\lambda \geq \frac{1}{2}\, max\left(|c| - \frac{\triangle x}{\triangle t}, 0\right) \tag{4.16}$$

All three steps in predictor can be evaluated together during one backward sweep through the mesh, i.e. it is not necessary to solve any system of linear equations. The same is valid for the corrector, which can be again realized by one forward sweep.

4.5 Implicit MacCormack scheme in FVM

In this section we will see how to apply the MacCormack scheme in the finite volume methodology. Since the MacCormack scheme is second order accurate in space and time, oscillations are observed in solution having abrupt step-changes in value.

The implicit MacCormack scheme in finite volume formulation is

Predictor:

$$\triangle W^n_{i,j,k} \;=\; -\Delta t\left(\frac{\Delta_+ F^n_{x_{i,j,k}}}{\Delta x}+\frac{\Delta_+ F^n_{y_{i,j,k}}}{\Delta y}+\frac{\Delta_+ F^n_{z_{i,j,k}}}{\Delta z}\right) \tag{4.17}$$

$$\left[I-\frac{\triangle t}{\triangle x}D^+_I|A|^n_{i,j,k}\right]\left[I-\frac{\triangle t}{\triangle y}D^+_J|B|^n_{i,j,k}\right]\left[I-\frac{\triangle t}{\triangle z}D^+_K|C|^n_{i,j,k}\right]\delta W^n_{i,j,k}=\triangle W^n_{i,j,k} \tag{4.18}$$

$$W^{\overline{n+1}}_{i,j,k}=W^n_{i,j,k}+\delta W^{\overline{n+1}}_{i,j,k} \tag{4.19}$$

Corrector:

$$\triangle W^{\overline{n+1}}_{i,j,k} \;=\; -\Delta t\left(\frac{\Delta_- F^{\overline{n+1}}_{x_{i,j,k}}}{\Delta x}+\frac{\Delta_- F^{\overline{n+1}}_{y_{i,j,k}}}{\Delta y}+\frac{\Delta_- F^{\overline{n+1}}_{z_{i,j,k}}}{\Delta z}\right) \tag{4.20}$$

$$\left[I+\frac{\triangle t}{\triangle x}D^-_I|A|^{\overline{n+1}}_{i,j,k}\right]\left[I+\frac{\triangle t}{\triangle x}D^-_J|B|^{\overline{n+1}}_{i,j,k}\right]\left[I+\frac{\triangle t}{\triangle z}D^-_K|C|^{\overline{n+1}}_{i,j,k}\right]\delta W^{\overline{n+1}}_{i,j,k}=\triangle W^{\overline{n+1}}_{i,j,k} \tag{4.21}$$

$$W^{n+1}_{i,j,k}=(W^n_{i,j,k}+W^{\overline{n+1}}_{i,j,k}+\delta W^{\overline{n+1}}_{i,j,k})/2 \tag{4.22}$$

Where the operators δ and Δ denote the implicit and explicit temporal difference operators, respectively. The first steps of predictor and corrector steps are equivalent to equation 4.10 which is FVM formulation of the fluxes.

Operators $D^+_I,\ D^-_I,\ D^+_J,\ D^-_J,\ D^+_K,\ D^-_K$ are one-sided forward and backward differences in each index dimension. $|A|\ ,\ |B|\ and\ |C|$ are diagonalized jacobian matrices. All these operators are explained later.

The values of $\frac{\Delta_+}{\Delta x}$, $\frac{\Delta_-}{\Delta x}$, $\frac{\Delta_+}{\Delta y}$, $\frac{\Delta_-}{\Delta y}$, $\frac{\Delta_+}{\Delta z}$, $\frac{\Delta_-}{\Delta z}$ operators are:

$$\begin{aligned}
\Delta_+ F_{x_{i,j,k}} &= F_{x_{i+1,j,k}} - F_{x_{i,j,k}} \\
\Delta_+ F_{x_{i,j,k}} &= F_{x_{i,j,k}} - F_{x_{i-1,j,k}} \\
\Delta_+ F_{x_{i,j,k}} &= F_{x_{i,j+1,k}} - F_{x_{i,j,k}} \\
\Delta_+ F_{x_{i,j,k}} &= F_{x_{i,j,k}} - F_{x_{i,j-1,k}} \\
\Delta_+ F_{x_{i,j,k}} &= F_{x_{i,j,k+1}} - F_{x_{i,j,k}} \\
\Delta_+ F_{x_{i,j,k}} &= F_{x_{i,j,k}} - F_{x_{i,j,k-1}}
\end{aligned}$$

The values of D_I^+, D_I^-, D_J+, D_J^-, D_K^+, D_K^- operators are:

$$\begin{aligned}
D_I^+ A_{i,j,k} &= \frac{|A|_{i+1,j,k} - |A|_{i,j,k}}{\Delta x} \\
D_I^- A_{i,j,k} &= \frac{|A|_{i,j,k} - |A|_{i-1,j,k}}{\Delta x} \\
D_J^+ B_{i,j,k} &= \frac{|B|_{i,j+1,k} - |B|_{i,j,k}}{\Delta y} \\
D_J^- B_{i,j,k} &= \frac{|B|_{i,j,k} - |B|_{i,j-1,k}}{\Delta y} \\
D_K^+ C_{i,j,k} &= \frac{|C|_{i,j,k+1} - |C|_{i,j,k}}{\Delta z} \\
D_K^- C_{i,j,k} &= \frac{|C|_{i,j,k} - |C|_{i,j,k-1}}{\Delta z}
\end{aligned}$$

Here the Jacobians of flux $\boldsymbol{F}$ is written as matrices A, B, C so that

$$\frac{\partial F_x}{\partial W} = A \qquad \frac{\partial F_y}{\partial W} = B \qquad \frac{\partial F_z}{\partial W} = C$$

Matrices $|A|$, $|B|$ *and* $|C|$ have positive eigenvalues and are related to the Jacobians A, B and C in a manner that will be explained below.

The inviscid jacobians A, B and C can be diagonalize by S_x, S_y, S_z. The matrices A, B, C can be witten as,

$$A = S_x^{-1}\Lambda_A S_x \qquad B = S_y^{-1}\Lambda_B S_y \qquad C = S_z^{-1}\Lambda_C S_z$$

The matrices S_x, S_y *and* S_z are each expressed as the product of two matrices. They can be written as,

$$S_x = \begin{pmatrix} 1 & 0 & 0 & 0 & \frac{-1}{c^2} \\ 0 & \rho c & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -\rho c & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{-u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 \\ \frac{-v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ \frac{-w}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ \alpha\beta & -u\beta & -v\beta & -w\beta & \beta \end{pmatrix} \tag{4.23}$$

$$S_y = \begin{pmatrix} 1 & 0 & 0 & 0 & \frac{-1}{c^2} \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & \rho c & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\rho c & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{-u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 \\ \frac{-v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ \frac{-w}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ \alpha\beta & -u\beta & -v\beta & -w\beta & \beta \end{pmatrix} \tag{4.24}$$

$$S_z = \begin{pmatrix} 1 & 0 & 0 & 0 & \frac{-1}{c^2} \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \rho c & 0 \\ 0 & 0 & 0 & -\rho c & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{-u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 \\ \frac{-v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ \frac{-w}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ \alpha\beta & -u\beta & -v\beta & -w\beta & \beta \end{pmatrix} \tag{4.25}$$

$$\Lambda_A = \begin{pmatrix} u & 0 & 0 & 0 & 0 \\ 0 & u+c & 0 & 0 & 0 \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & 0 & 0 & 0 & u-c \end{pmatrix}, \quad \Lambda_B = \begin{pmatrix} v & 0 & 0 & 0 & 0 \\ 0 & u & 0 & 0 & 0 \\ 0 & 0 & v+c & 0 & 0 \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & 0 & 0 & v-c \end{pmatrix} \tag{4.26}$$

$$\Lambda_C = \begin{pmatrix} w & 0 & 0 & 0 & 0 \\ 0 & w & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 \\ 0 & 0 & 0 & w+c & 0 \\ 0 & 0 & 0 & 0 & w-c \end{pmatrix} \tag{4.27}$$

and where $c = \sqrt{\gamma p/\rho}$ is the speed of sound, $\alpha = \frac{1}{2}(u^2+v^2+w^2)$ $and \beta = \gamma - 1$.

The inverses $S_x^{-1},\ S_y^{-1}\ and\ S_z^{-1}$ are simply the inverse matrix of $S_x,\ S_y\ and\ S_z$ respectively.

The matrices $|A|\ and\ |B|$ are defined by

$$|A| \ = \ S_x^{-1} D_A S_x \qquad |B| \ = \ S_y^{-1} D_B S_y \qquad |C| \ = \ S_z^{-1} D_C S_z$$

where $D_A,\ D_B\ and\ D_C$ are diagonal matrices defined by

$$D_A \ = \ \begin{pmatrix} \lambda_{A1} & 0 & 0 & 0 & 0 \\ 0 & \lambda_{A2} & 0 & 0 & 0 \\ 0 & 0 & \lambda_{A3} & 0 & 0 \\ 0 & 0 & 0 & \lambda_{A4} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{A5} \end{pmatrix} \tag{4.28}$$

$$D_B \ = \ \begin{pmatrix} \lambda_{B1} & 0 & 0 & 0 & 0 \\ 0 & \lambda_{B2} & 0 & 0 & 0 \\ 0 & 0 & \lambda_{B3} & 0 & 0 \\ 0 & 0 & 0 & \lambda_{B4} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{B5} \end{pmatrix} \tag{4.29}$$

$$D_C \ = \ \begin{pmatrix} \lambda_{C1} & 0 & 0 & 0 & 0 \\ 0 & \lambda_{C2} & 0 & 0 & 0 \\ 0 & 0 & \lambda_{C3} & 0 & 0 \\ 0 & 0 & 0 & \lambda_{C4} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{C5} \end{pmatrix} \tag{4.30}$$

and

$$\begin{aligned} \lambda_{A1} \ &= \ max\left\{|u| + \frac{2\nu}{\rho\Delta x} - \frac{1}{2}\frac{\Delta x}{\Delta t},\ 0\right\} \\ \lambda_{A2} \ &= \ max\left\{|u+c| + \frac{2\nu}{\rho\Delta x} - \frac{1}{2}\frac{\Delta x}{\Delta t},\ 0\right\} \\ \lambda_{A3} \ &= \ max\left\{|u| + \frac{2\nu}{\rho\Delta x} - \frac{1}{2}\frac{\Delta x}{\Delta t},\ 0\right\} \\ \lambda_{A4} \ &= \ max\left\{|u| + \frac{2\nu}{\rho\Delta x} - \frac{1}{2}\frac{\Delta x}{\Delta t},\ 0\right\} \\ \lambda_{A5} \ &= \ max\left\{|u-c| + \frac{2\nu}{\rho\Delta x} - \frac{1}{2}\frac{\Delta x}{\Delta t},\ 0\right\} \end{aligned}$$

$$\begin{aligned}
\lambda_{B1} &= max\left\{|v| + \frac{2\nu}{\rho\Delta y} - \frac{1}{2}\frac{\Delta y}{\Delta t},\, 0\right\} \\
\lambda_{B2} &= max\left\{|v| + \frac{2\nu}{\rho\Delta y} - \frac{1}{2}\frac{\Delta y}{\Delta t},\, 0\right\} \\
\lambda_{B3} &= max\left\{|v + c| + \frac{2\nu}{\rho\Delta y} - \frac{1}{2}\frac{\Delta y}{\Delta t},\, 0\right\} \\
\lambda_{B4} &= max\left\{|v| + \frac{2\nu}{\rho\Delta y} - \frac{1}{2}\frac{\Delta y}{\Delta t},\, 0\right\} \\
\lambda_{B5} &= max\left\{|v - c| + \frac{2\nu}{\rho\Delta y} - \frac{1}{2}\frac{\Delta y}{\Delta t},\, 0\right\}
\end{aligned}$$

$$\begin{aligned}
\lambda_{C1} &= max\left\{|w| + \frac{2\nu}{\rho\Delta z} - \frac{1}{2}\frac{\Delta z}{\Delta t},\, 0\right\} \\
\lambda_{C2} &= max\left\{|w| + \frac{2\nu}{\rho\Delta z} - \frac{1}{2}\frac{\Delta z}{\Delta t},\, 0\right\} \\
\lambda_{C3} &= max\left\{|w| + \frac{2\nu}{\rho\Delta z} - \frac{1}{2}\frac{\Delta z}{\Delta t},\, 0\right\} \\
\lambda_{C4} &= max\left\{|w + c| + \frac{2\nu}{\rho\Delta z} - \frac{1}{2}\frac{\Delta z}{\Delta t},\, 0\right\} \\
\lambda_{C5} &= max\left\{|w - c| + \frac{2\nu}{\rho\Delta z} - \frac{1}{2}\frac{\Delta z}{\Delta t},\, 0\right\}
\end{aligned}$$

$$\nu = max\left\{\mu, \lambda + 2\mu, \frac{\gamma\mu}{PrandtlNumber}\right\}$$

The Jacobian matrix formulation is explained in Appendix A.

For regions of the flow in which Δt satisfies the following explicit stability conditions

$$\Delta t \leq \frac{1}{2}\frac{\Delta x}{\left(|u| + c + \frac{2\nu}{\rho\Delta x}\right)} \qquad \Delta t \leq \frac{1}{2}\frac{\Delta y}{\left(|v| + c + \frac{2\nu}{\rho\Delta y}\right)} \qquad \Delta t \leq \frac{1}{2}\frac{\Delta z}{\left(|w| + c + \frac{2\nu}{\rho\Delta z}\right)} \tag{4.31}$$

all λ_A, λ_B *and* λ_C vanish and the set of Implicit equations reduces to the explicit equations with simple solution. For other regions in which neither relation is satisfied, the resulting difference equations are either upper or lower block bidiagonal equations with fairly straightforward solutions.

4.5.1 Artificial Viscosity

The MacCormack method operates satisfactorily in the regions where the variations of properties is smooth. But there is oscillations occurring around discontinuities, i.e., around a shock wave or in the boundary layer. So, *artificial smoothing* terms must be introduced, to damp these oscillations.

From the basic CFD theory we know that modified equation of a PDE gives us some information on the behaviour to be expected of the numerical solution of the difference equation. The modified equation of first order upwind scheme for the one-dimensional wave equation

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{4.32}$$

is shown below

$$\begin{aligned}\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \frac{a\Delta x}{2}(1-\nu)\frac{\partial^2 u}{\partial x^2} + \frac{a(\Delta x)_2}{6}(3\nu - 2\nu_2 - 1)\frac{\partial^3 u}{\partial x^3} \\ +O[(\Delta t)^3, (\Delta t)^2(\Delta x), (\Delta t)(\Delta x)^2, (\Delta x)^3]\end{aligned} \tag{4.33}$$

The dissipative term in the above equation, i.e., even-order derivative terms $\frac{\partial^2 u}{\partial x^2}$ is actually the artificial viscosity term implicitly embedded in the numerical scheme. It prevents the solution from going unstable due to the oscillations caused by the dispersive terms i.e. odd-order derivative terms $\frac{\partial^3 u}{\partial x^3}$. But for variable velocity problems, the MacCormack scheme often does not have enough artificial viscosity implicitly in the algorithm, and the solution will become unstable unless more artificial viscosity is added *explicitly* to the calculation, which makes the solution more inaccurate. Therefore, there is a trade off involved. The artificial viscosity formulation is explained in Appendix B.

4.6 Solution Procedure

Predictor: To solve first the predictor step of Eqs. 4.17 and 4.18 assuming Δt satisfies neither of Eqs. 4.31, we do the following:

$\triangle W^n_{i,j,k}$ value can be found explicitly using 4.10.

$$\begin{aligned}\triangle W^n_{i,j,k} &= F^n_{ex}S_{ex} + F^n_{ey}S_{ey} + F^n_{ez}S_{ez} + F^n_{wx}S_{wx} + F^n_{wy}S_{wy} + F^n_{wz}S_{wz} \\ &+ F^n_{nx}S_{nx} + F^n_{ny}S_{ny} + F^n_{nz}S_{nz} + F^n_{sx}S_{sx} + F^n_{sy}S_{sy} + F^n_{sz}S_{sz} \\ &+ F^n_{tx}S_{tx} + F^n_{ty}S_{ty} + F^n_{tz}S_{tz} + F^n_{bx}S_{bx} + F^n_{by}S_{by} + F^n_{bz}S_{bz}\end{aligned}$$

To solve the equation 4.18 consider

$$\delta W^*_{i,j,k} = \left(I - \frac{\Delta t}{\Delta y} D^+_J |B|_{i,j,k}\right)\left(I - \frac{\Delta t}{\Delta z} D^+_K |C|_{i,j,k}\right) \delta \overline{W^{n+1}_{i,j,k}} \tag{4.34}$$

Substituting eqn. 4.34 value to 4.18 we get,

$$\left[I - \frac{\triangle t}{\triangle x} D^+_I |A|^n_{i,j,k}\right] \delta W^*_{i,j,k} = \triangle W^n_{i,j,k} \tag{4.35}$$

Now substituting D^+_I value in this equation we get

$$\left[I - \frac{\triangle t}{\triangle x}\left(|A|^n_{i+1,j,k} - |A|^n_{i,j,k}\right)\right] \delta W^*_{i,j,k} = \triangle W^n_{i,j,k} \tag{4.36}$$

Rearranging equation 4.36 we can write

$$\left(I + \frac{\triangle t}{\triangle x}|A|^n_{i,j,k}\right) \delta W^*_{i,j,k} = \triangle W^n_{i,j,k} + + \frac{\triangle t}{\triangle x}|A|^n_{i+1,j,k}\, \delta W^*_{i+1,j,k} \tag{4.37}$$

It is an upper bidiagonal equation. The solution for $\delta W^*_{i,j,k}$ can be obtained for each j and k by sweeping in the decreasing i direction.

After obtaining $\delta W^*_{i,j,k}$ for all i, j, k then substituting this value in Eqn. 4.34 and we get

$$\left(I - \frac{\Delta t}{\Delta y} D^+_J |B|_{i,j,k}\right)\left(I - \frac{\Delta t}{\Delta z} D^+_K |C|_{i,j,k}\right) \delta \overline{W^{n+1}_{i,j,k}} = \delta W^*_{i,j,k} \tag{4.38}$$

Let us consider

$$\delta W^{**}_{i,j,k} \;=\; \left(I \;-\; \frac{\Delta t}{\Delta z} D^{+}_{K} \left|C\right|_{i,j,k}\right) \delta W^{\overline{n+1}}_{i,j,k} \tag{4.39}$$

Substituting this value into Eqn. 4.38 we get,

$$\left(I \;-\; \frac{\Delta t}{\Delta y} D^{+}_{J} \left|B\right|_{i,j,k}\right) \delta W^{**}_{i,j,k} \;=\; \delta W^{*}_{i,j,k} \tag{4.40}$$

Substituting D^{+}_{J} value in this equation we get

$$\left[I - \frac{\triangle t}{\triangle y}\left(|B|^{n}_{i,j+1,k} \;-\; |B|^{n}_{i,j,k}\right)\right] \delta W^{**}_{i,j,k} = \triangle W^{*}_{i,j,k} \tag{4.41}$$

Rearranging equation 4.41 we can write

$$\left(I + \frac{\triangle t}{\triangle y}|B|^{n}_{i,j,k}\right) \delta W^{**}_{i,j,k} = \triangle W^{*}_{i,j,k} \;+\; + \frac{\triangle t}{\triangle y}|B|^{n}_{i,j+1,k}\, \delta W^{**}_{i,j+1,k} \tag{4.42}$$

We can get the solution for $\delta W^{**}_{i,j,k}$ for each i and k by sweeping in the decreasing j direction.

After obtaining $\delta W^{**}_{i,j,k}$ for all i, j, k then substituting this value in Eqn. 4.39 and we get

$$\left(I \;-\; \frac{\Delta t}{\Delta z} D^{+}_{K} \left|C\right|_{i,j,k}\right) \delta W^{\overline{n+1}}_{i,j,k} \;=\; \delta W^{**}_{i,j,k} \tag{4.43}$$

Substituting D^{+}_{K} value in this equation we get

$$\left[I - \frac{\triangle t}{\triangle z}\left(|C|^{n}_{i,j,k+1} \;-\; |C|^{n}_{i,j,k}\right)\right] \delta W^{\overline{n+1}}_{i,j,k} \;=\; \triangle W^{**}_{i,j,k} \tag{4.44}$$

Rearranging equation 4.44 we can write

$$\left(I + \frac{\triangle t}{\triangle z}|C|^{n}_{i,j,k}\right) \delta W^{\overline{n+1}}_{i,j,k} = \triangle W^{**}_{i,j,k} \;+\; + \frac{\triangle t}{\triangle z}|C|^{n}_{i,j,k+1}\, \delta W^{\overline{n+1}}_{i,j,k+1} \tag{4.45}$$

We can get the solution for $\delta W^{\overline{n+1}}_{i,j,k}$ for each i and j by sweeping in the decreasing k direction.

Then we can go to the third step and calculate

$$W_{i,j}^{\overline{n+1}} = W_{i,j}^{n} + \delta W_{i,j}^{\overline{n+1}}$$

In the above procedure, the solution of the block bi-diagonal systems is carried out making use of the known decomposition of $|A|$, $|B|$, $|C|$ which reduces the computation in the inversion of the block matrices. For example, to solve Eq. 4.37 in the predictor, the equation is rewritten as

$$Sx_{i,j,k}^{-1}\left(I + \frac{\triangle t}{\triangle x}|D_A|_{i,j,k}^{n}\right) Sx_{i,j,k}\ \delta W_{i,j,k}^{*} = \triangle W_{i,j,k}^{n} + \frac{\triangle t}{\triangle x}|A|_{i+1,j,k}^{n}\,\delta W_{i+1,j,k}^{*}$$

and can be easily solved as

$$\delta W_{i,j,k}^{*} = Sx_{i,j,k}^{-1}\left(I + \frac{\triangle t}{\triangle x}|D_A|_{i,j,k}^{n}\right)^{-1} Sx_{i,j,k}\left[\triangle W_{i,j,k}^{n} + \frac{\triangle t}{\triangle x}|A|_{i+1,j,k}^{n}\,\delta W_{i+1,j,k}^{*}\right]$$

Note that the block matrix inversion is trivial because $Sx_{i,j,k}^{-1} and Sx_{i,j,k}$ are known and $\left(I + \frac{\triangle t}{\triangle x}|D_A|_{i,j,k}^{n}\right)$ is diagonal. This in fact means that a block bidiagonal matrix inversion is reduced to a scalar bidiagonal matrix inversion.

The procedure to solve this equation 4.37 is as follows:

For each j, k and for i = I, I-1, I-2..... 2, 1

1. $W = \triangle W_{i,j,k}^{n} + \frac{\triangle t}{\triangle x}|A|_{i+1,j,k}^{n}\,\delta W_{i+1,j,k}^{*}$

2. $X = Sx\,W$

3. D_A is calculated using 4.28

4. $Y = \left(I + \frac{\triangle t}{\triangle x}|D_A|_{i,j,k}^{n}\right)^{-1} X$

5. $\delta W^*_{i,j,k} = Sx^{-1}\, Y$

6. $Z = D_A\, Y$

7. $|A|_{i,j,k}\delta W^*_{i,j,k} = Sx^{-1}\, Z$

where X, Y, Z are the vectors assumed for the sake of simplicity to solve the equation in steps 2,4 and 6.

Since this W is a matrix, all the variables are calculated with one full sweep of i,j,k. Each of the above seven steps requires to calculate $\delta W^*_{i,j,k}$ for each i, j, k. The matrix inversion of step 4 is trivial because the matrix is diagonal. So we use the inversion of a diagonal matrix formula. Let D is a diagonal matrix and

$$D = \begin{pmatrix} a_{11} & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 \\ 0 & 0 & a_{33} & 0 & 0 \\ 0 & 0 & 0 & a_{44} & 0 \\ 0 & 0 & 0 & 0 & a_{55} \end{pmatrix}$$

Then according to this formula its inverse is given by:

$$D^{-1} = \begin{pmatrix} \frac{1}{a_{11}} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{a_{22}} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{a_{33}} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{a_{44}} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{a_{55}} \end{pmatrix}$$

Note that the solution $\delta W^*_{i,j,k}$ at grid point i,j,k is obtained at step 5. The flux $|A|_{i,j,k}\delta W^*_{i,j,k}$ to be used in the calculation at grid point $i-1$, j, k is obtained at step 7.

4.7 Boundary Conditions

4.7.1 Wall Boundary Condition

The boundary condition for the wall follows no slip condition unlike the invisid flow i.e., u=0, v=0, w=0. In the solver, no slip condition is given to the boundaries which are given as walls. For example, if j=0 is wall, then $\|B\|\,\delta u_i, 0, k$, $\|B\|\,\delta v_i, 0, k$, $\|B\|\,\delta w_i, 0, k$ are given no slip condition. The computed end flux terms (in this case) $\|B\|\,\delta U_i, 2, k$ are to be used as a boundary condition for the corrector step that sweeps away from this boundary in the increasing j direction and for the predictor step the value of the conservative variables W is given as the input to start the sweep.

4.7.2 Inflow Boundary Condition

The boundary conditions of all primitive variable for flow coming into the domain should physical if the flow is supersonic. If the flow is subsonic, one variable is given numerical boundary condition and others are given physical boundary conditions.

$$\frac{\partial u}{\partial t} = 0, \frac{\partial v}{\partial t} = 0, \frac{\partial w}{\partial t} = 0$$

4.7.3 Outflow Boundary Condition

We can get δW values by interpolating between the outflow plane and the first interior points.

$$\delta W_{boundary} = W_{interior}$$

4.7.4 Symmetry Boundary Condition

When the operator passes information away from the wall incoming values of δW are set equal to zero. When the operator passes information towards the wall, the outgoing flux is mirrored about the wall plane and propagated back into the flow using inward operator.

4.8 Closure

In this chapter we have seen the detailed formulation of the implicit MacCormack scheme which can be used for the study of compressible flows.

Chapter 5

Results and Discussion

5.1 Time comparision

In this section we compare time taken for explicit MacCormack (matrix form) and implicit MacCormack schemes for Euler equations for the following test cases.

The test cases are:

- Shocktube Problem
- Supersonic flow over a wedge
- Subsonic flow over a circular bump
- Subsonic flow over a airfoil
- Flow over re-entry capsule

The following tables compares the results of implicit MacCormack and explicit MacCormack Schemes in terms of time, CFL number and their validation.

Shock Tube:

This problem ([5], (pg-352),) comprises of a tube initially containing two regions of a stationary gas at different pressures, separated by a diaphragm. At t = 0, the diaphragm is removed instantaneously so that the pressure imbalance causes a unsteady flow containing a moving expansion fan, shock and contact discontinuity. The problem can be solved analytically as a 1-D case [25]. However, we solve the computational problem as a 2-D case, and compare it with the 1-D analytical solution.The computational results were obtained on a uniform grid of Δx= 0.1m . A Courant number of 1.1 has been used.

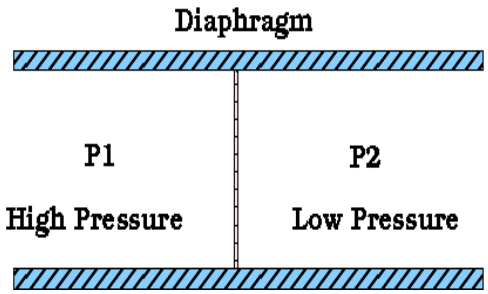


Figure 5.1: Shocktube

It contains two zones, first zone supports high pressure fluid and second zone supports low pressure fluid. The details of the geometry are:

- four slip walls
- two symmetric boundary surface

IC	Part 1	Part 2
Pressure	100000 Pa	10000 Pa
Temperature	300K	300K
u velocity	0	0
v velocity	0	0
w velocity	0	0

Initial Condition

Boundary Conditions

All boundaries are (slip) walls, while symmetry boundary condition are implemented on surfaces on the z-plane.

The calculation was done to compare with analytical results previously derived for the shocktube problem [25]. The analytical solution to the shock-tube problem at t = 0.0061s is compared to the computational result at the centerline of the tube (see Fig. 5.1). The explicit and Implicit MacCormack schemes are compared. The advantage of using an implicit scheme compared to a explicit scheme is the computation time. The expansion shock occurring on the left has been captured accurately as in explicit one with less computation time. The results from Implicit MacCormack and Explicit MacCormack scheme (with artificial viscosity) are presented.

Results

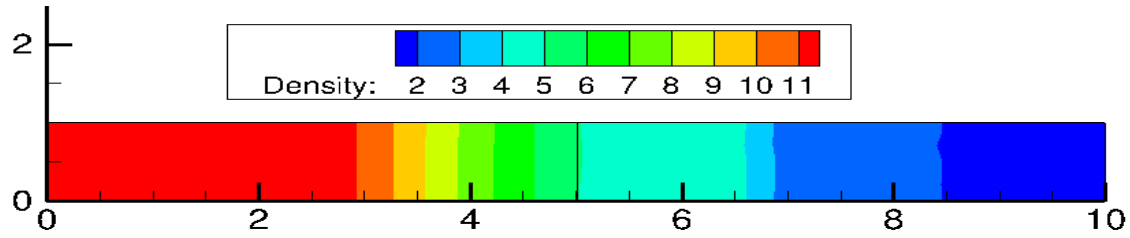


Figure 5.2: Density Contour at 6.1 ms with pressure ratio of 10 with constant Courant No = 1.1

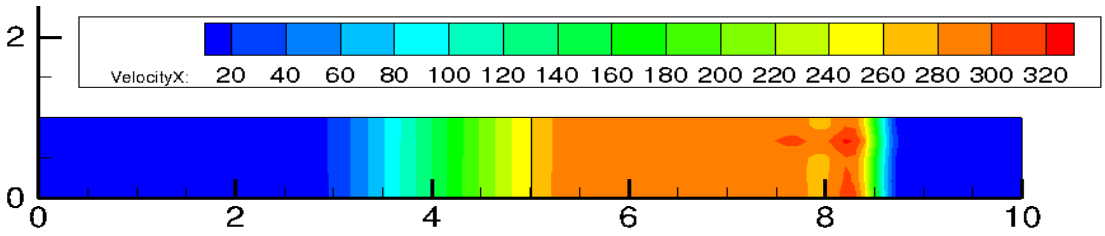


Figure 5.3: Velocity Contour at 6.1 ms with pressure ratio of 10 with constant Courant No = 1.1

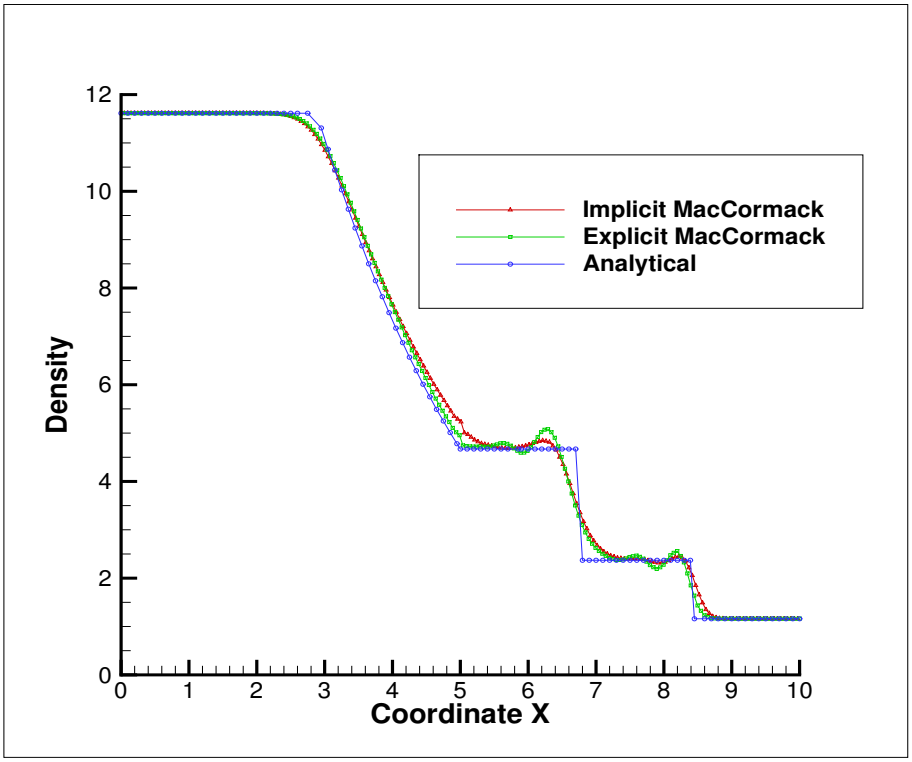


Figure 5.4: Density Plot at 6.1 ms

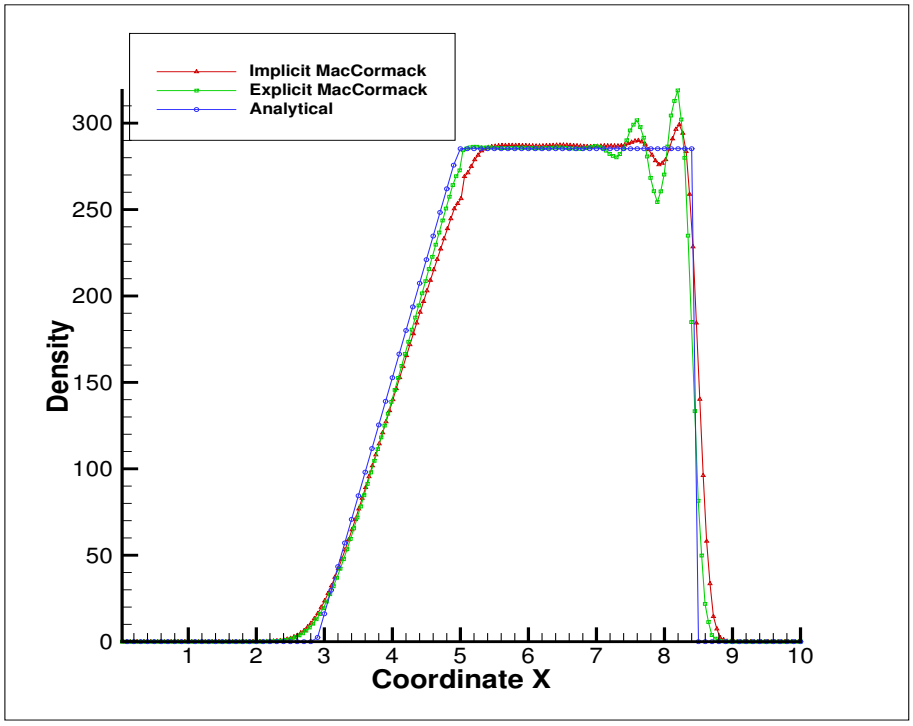


Figure 5.5: U Velocity Plot at 6.1 ms

Method	CFL	Total CPU Time	Time Steps	End time
Explicit MacCormack	0.75	13.5 s(before 24 s)*	121	0.0061
Implicit MacCormack	1.1	15.28 s	83	0.0061

Table 5.1: Computational time comparison

(* Computational time taken before implementing matrix form)

15^o **Wedge:**

We now consider the supersonic flow over a 2-D wedge with wedge angle 15° , as shown in figure. The inflow conditions are summarized in following table 5.2 and the present results have been compared with the analytical solution obtained from the standard $(\theta-\beta-M)$ chart and the analytical oblique shock relationships. Courant number of 0.3 and 1.1 are used for explicit and implicit MacCormack, respectively.

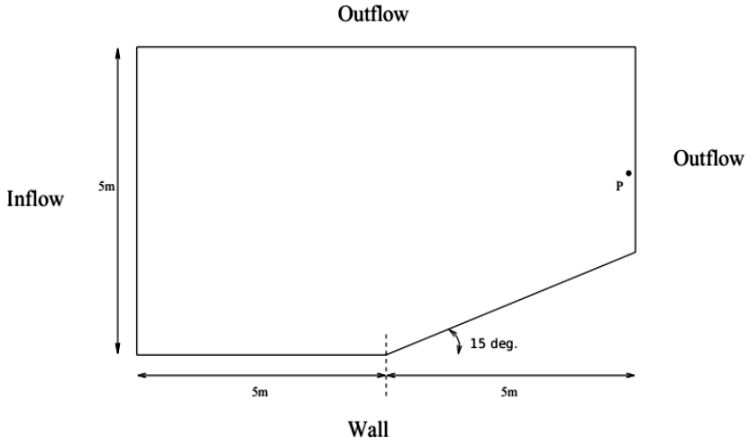


Figure 5.6: Computational domain

*Note: The solver explicitly asks for an outflow pressure but imposes this condition if and only if the flow is subsonic there.

Quantity	Inflow	Outflow
Pressure	101353 Pa	101353 Pa*
Temperature	288.9 K	-
U velocity	2.5 Mach	-
V velocity	0	-
W velocity	0	-

Table 5.2: Boundary conditions for supersonic wedge

Boundary Conditions

The steady-state contours of Mach number and static pressure using implicit MacCormack Scheme shown in figures below. Under the same flow condition the contours obtained by numerical computation done in Hirsch's book [6] is also shown in Fig. 5.4. The results downstream of the shock has been tabulated in Table 5.3 where, P_2/P_1 corresponds to the downstream and upstream pressure ratio. Point P refers to the point (1.495, 0.3) on the outflow plane. The analytical results are also presented. Pressure, density, mach and temperature values are extracted along x = 1.2 line for both the schemes and plotted along y axis. The plots are compared for explicit and implicit MacCormack schemes in figures 5.12, 5.1, 5.14, 5.15. Both the Explicit MacCorMack and Implicit MacCormack gives similar results and both have high accuracy, corresponding with the analytical results. Implicit MacCormack scheme however gives a solution within less computational time.

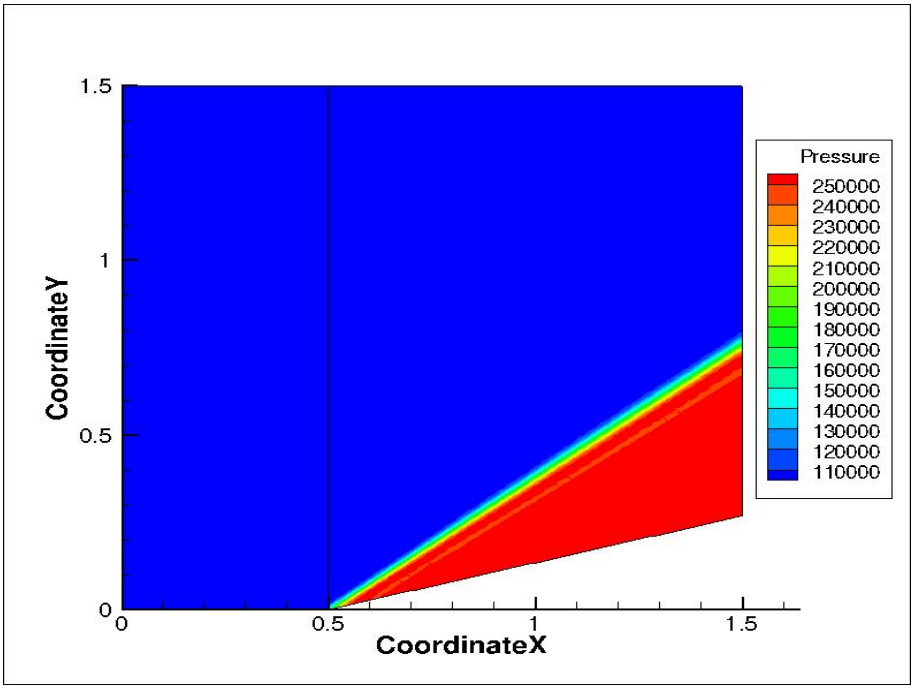


Figure 5.7: Pressure Contour with Implicit MacCormack Scheme

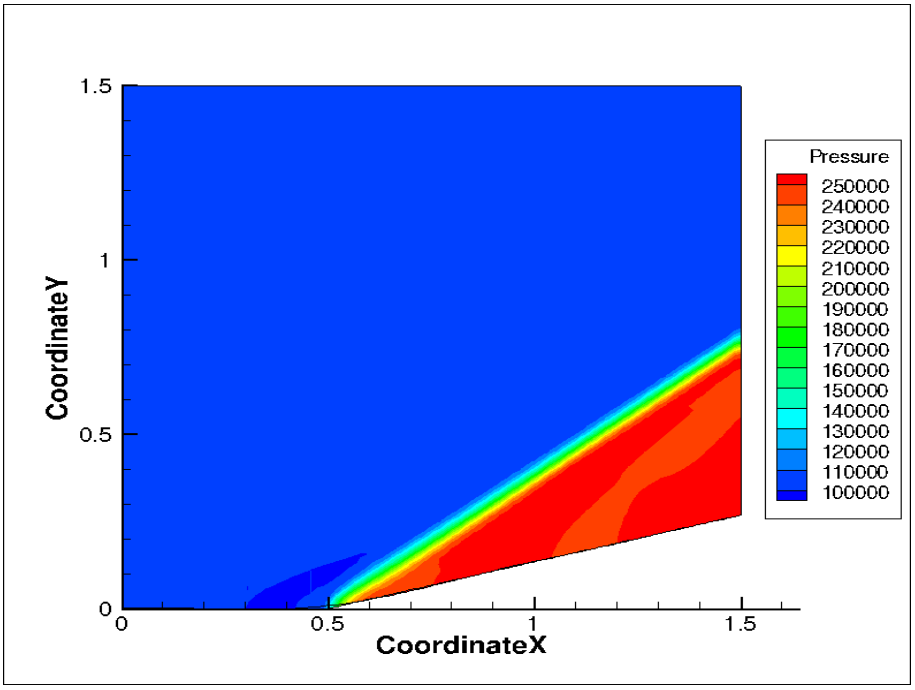


Figure 5.8: Pressure Contour with Explicit MacCormack Scheme

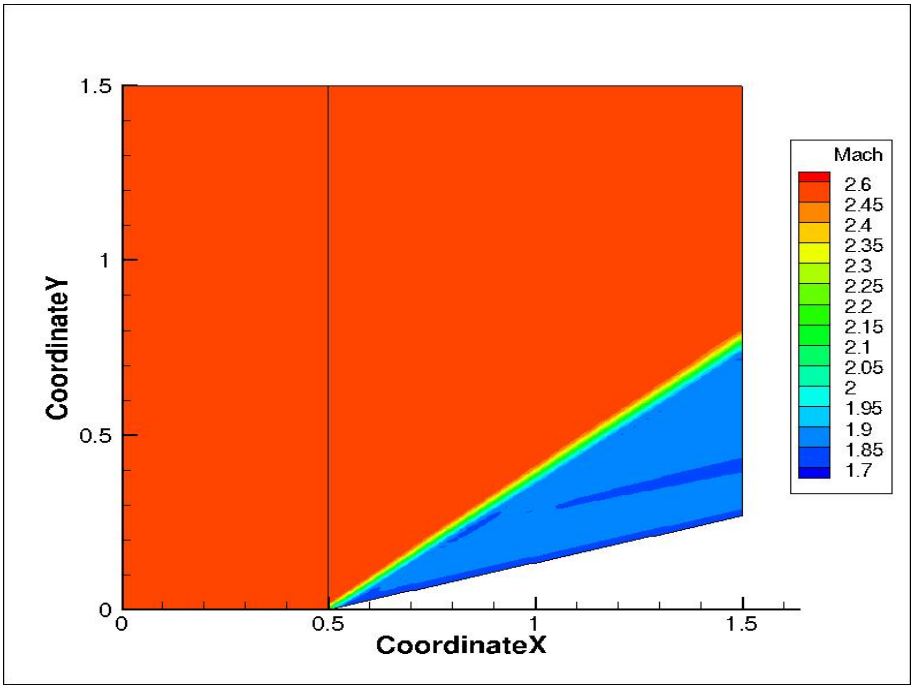


Figure 5.9: Mach Contour with Implicit MacCormack Scheme

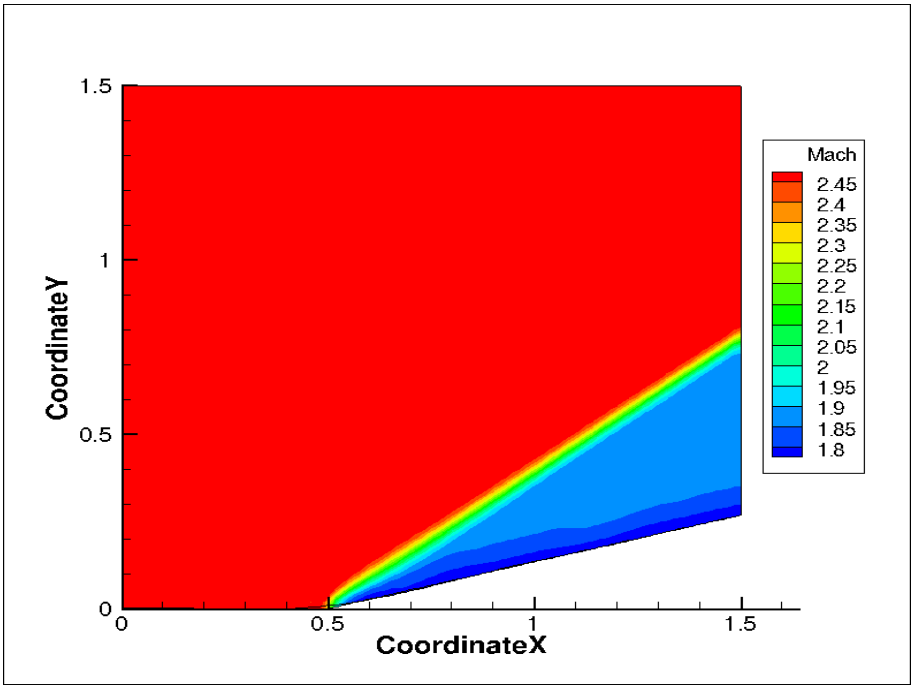


Figure 5.10: Mach Contour with Explicit MacCormack Scheme

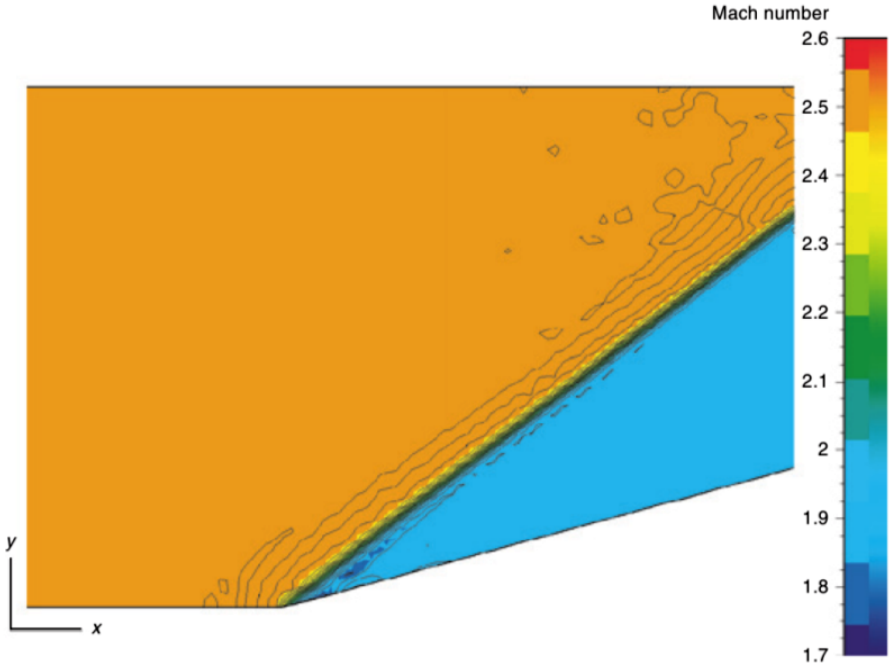


Figure 5.11: Mach Contour from Reference Hirsch [6]

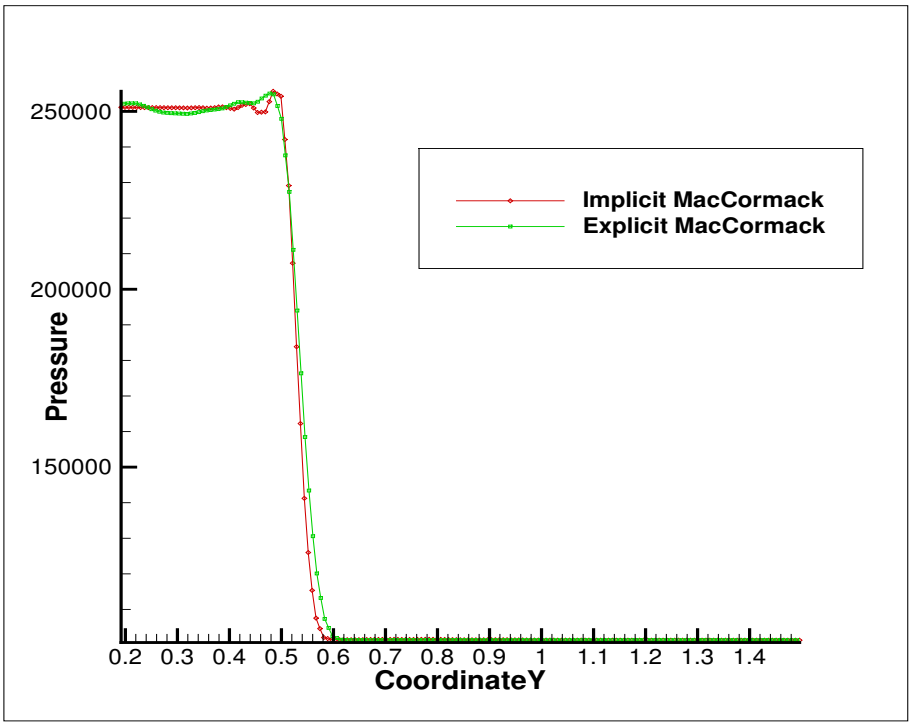


Figure 5.12: Variation of pressure along y co-ordinate

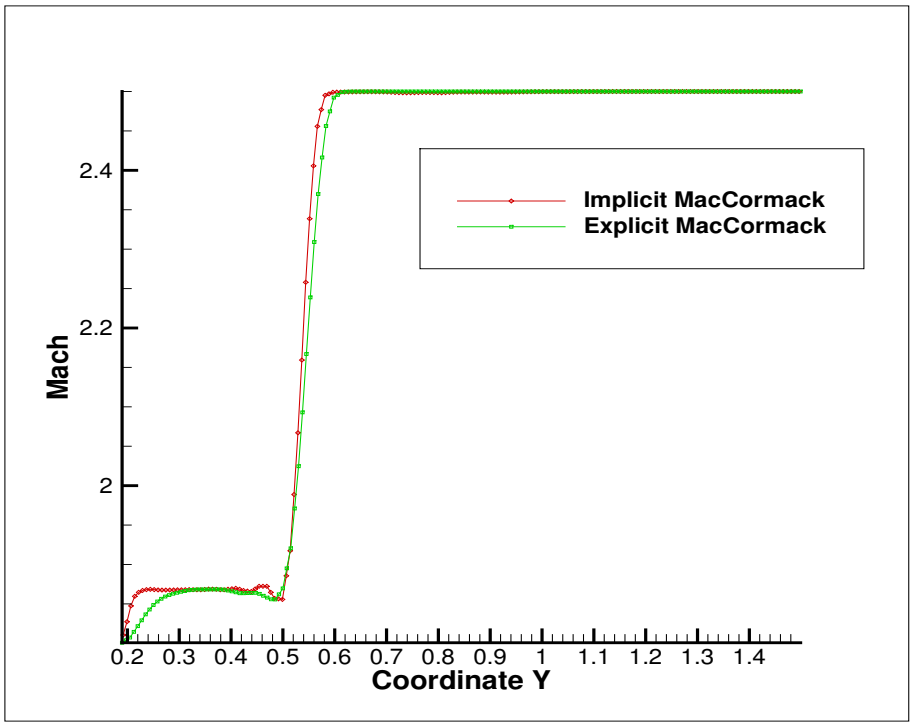


Figure 5.13: Variation of Mach Number along y co-ordinate

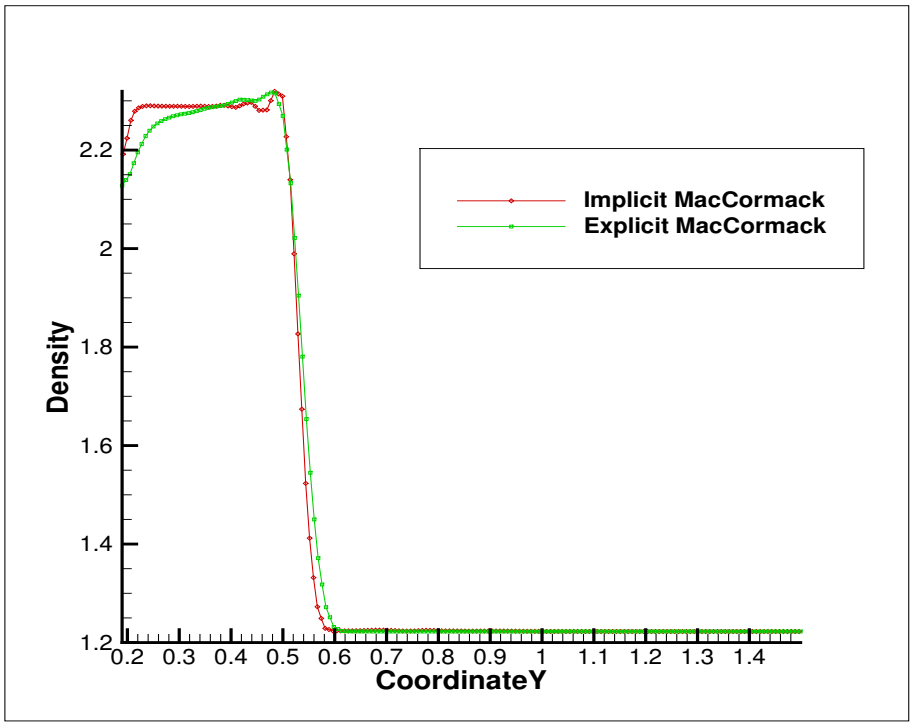


Figure 5.14: Variation of Density along y co-ordinate

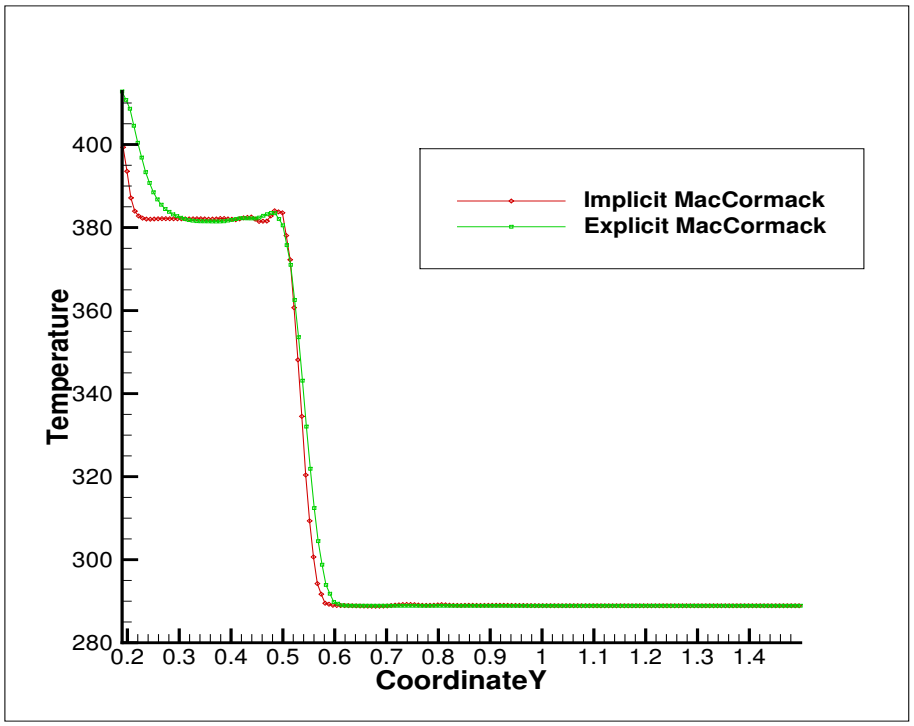


Figure 5.15: Variation of Temperature along Y co-ordinate

Validation:

Ratio's	**Analytical**	**Implicit MacCormack**	**Explicit MacCormack**
P_2/P_1	2.468	2.477	2.468
T_2/T_1	1.322	1.325	1.322
ρ_2/ρ_1	1.867	1.869	1.866
Mach	1.874	1.864	1.873
Shock angle(in degree)	36.945	37.954	38.66

Table 5.3: Validation with analytical solution

Computational Time comparison:

Method	CFL	Total CPU Time	Time Steps
Explicit MacCormack	0.3	330764 s 91.9(before 154 hrs)*	632573
Implicit MacCormack	1.1	997.3 s (17 min)	1075

Table 5.4: Computational time comparison

(* Computational time taken before implementing matrix form)

Circular Bump:

We now take a case considering internal flow. It consists of a channel of height L and length 3L, with a circular arc of length L and thickness equal to 0.1L, along the bottom wall, as shown in Fig. 5.18. For the subsonic case we use a pressure-driven inlet boundary condition. For initializing the flow-field, we have used free-stream conditions. The inlet x-velocity is calculated by numerical-extrapolation from the interior domain. Its specification in the problem below is indicative for Mach Number of the flow at the inlet and is used in the numerical algorithm. We used a Courant number of 1.1 for cases below. The implicit MacCormack has convergence difficulty for the subsonic case if we want to use residual which is less than 10^{-8}. So we are using the convergence criteria upto 10^{-6}.

5.1.1 Subsonic Case

The inlet Mach number is chosen equal to 0.5. We provide total pressure and total temperature at inlet with respect to the static condition, so as to get inlet Mach Number equal to 0.5. At outflow we use the free-stream condition (static condition). The boundary conditions are summarized in Table 5.5 and the solver results have been compared with the study done by Rincon and Elder et al. [20].

Boundary Conditions

The comparison for Mach contours for Explicit MacCormack, Implicit MacCormack and the reference is shown. Comparison of computational times are shown.

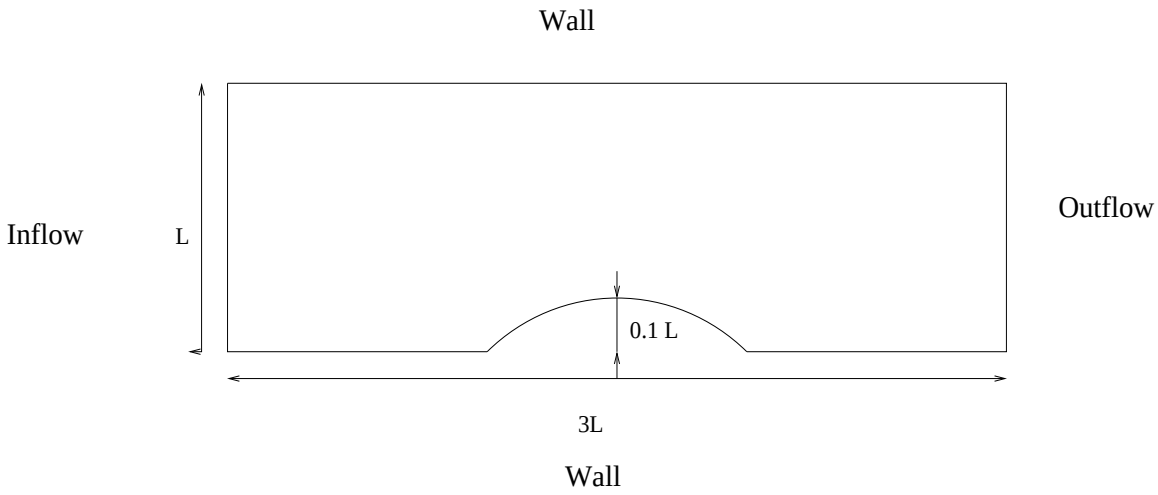


Figure 5.16: Computational domain [2]

Quantity	Inflow	Outflow
Pressure	120141.8 Pa	101300 Pa
Temperature	302.4 K	288 K
U velocity	174.287	-
V velocity	0	-
W velocity	0	-

Table 5.5: Boundary Condition for subsonic bump

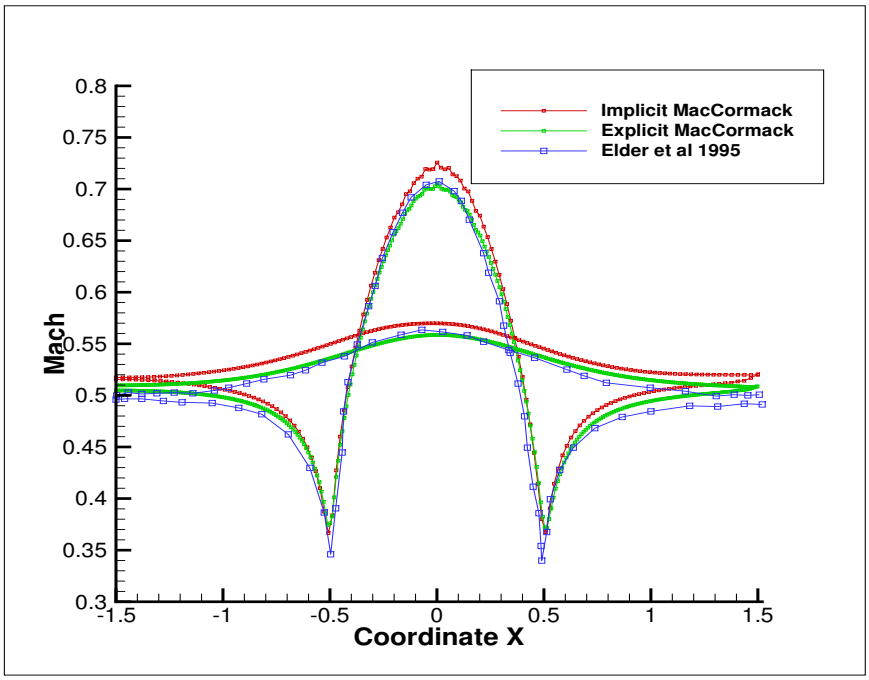


Figure 5.17: Variation of Mach number along lower and upper wall (M = 0.5)

Method	CFL	Total CPU Time	Time Steps	Residual
Explicit MacCormack	0.4	197384 s 54.8(before 87.03 hrs)*	396418	$3.3368e^{-6}$
Implicit MacCormack	1.1	16750.75 s (4.6 hrs)	31518	$3.3368e^{-6}$

Table 5.6: Computational time comparison

(* Computational time taken before implementing matrix form)

Aerofoil To validate the code for complex geometry, we have taken the case of NACA 0012 airfoil. We study the external flow at Mach number of 0.5. The computational domain for the NACA Aerofoil considered is shown below.

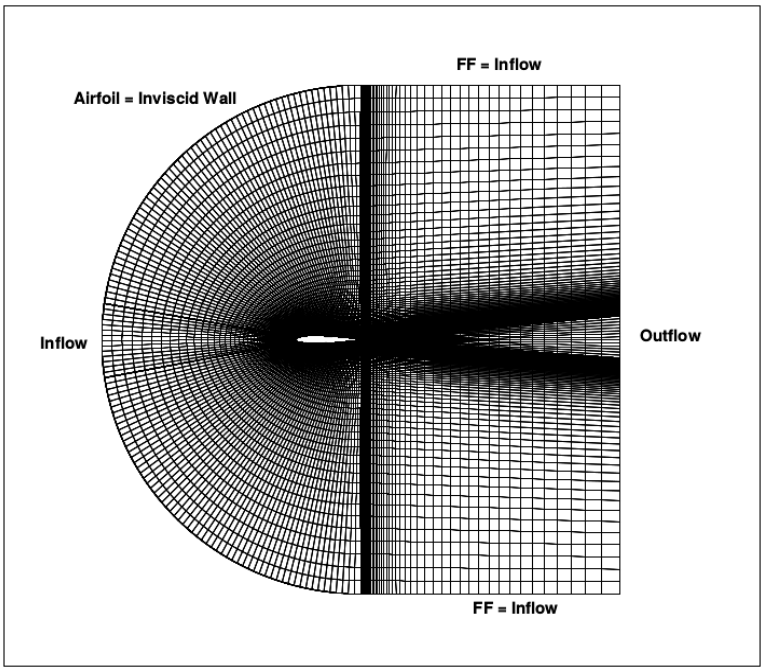


Figure 5.18: Computational domain [2]

5.1.2 Subsonic Case:Mach 0.5, Angle of Attack ($\alpha = 0^\circ$)

This is a subsonic case involving external flow. We have used velocity-driven boundary condition for inlet and far-field. The boundary conditions are tabulated in Table 5.7

Boundary Conditions

The comparison for Mach and pressure contours for Explicit MacCormack, Implicit MacCormack and the reference is shown in Fig. **??** and **??** respectively.

Quantity	Inflow	Outflow
Pressure	100000 Pa	100000 Pa
Temperature	300 K	288 K
U velocity	173.594	-
V velocity	0	-
W velocity	0	-

Table 5.7: Boundary conditions for NACA 0012 M = 0.5, $\alpha = 0°$

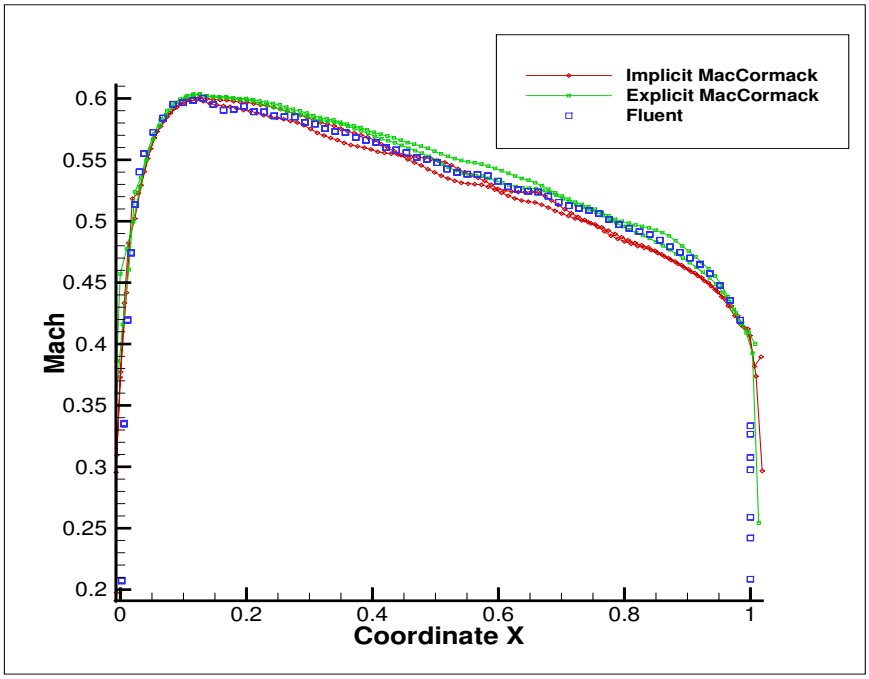


Figure 5.19: Variation of Mach number along airfoil wall (M = 0.5)

Aerofoil:

Method	CFL	Total Time	Time Steps
Explicit MacCormack	0.3	263336 s (73.1 hrs)(before 114.9 hr)*	597819
Implicit MacCormack	1.1	21522.67 s (5.7 hrs)	41728

Table 5.8: Computational time comparison

(* Computational time taken before implementing matrix form)

Capsule:

A ballistic reentry capsule has been considered to validate the solver for a complex geometry. The vehicle consists of a blunt bicone with 20/25 degree cone angles. All the dimensions are shown in Fig. 5.20. Inlet, outlet and inviscid wall has been shown through red, green and blue colour respectively. The free-stream pressure and temperature are 833Pa and 63K, respectively. Free-stream Mach number is taken as 5.0 with angle of attack of 4.66. We specify free-stream pressure at outflow, which actually has no role to play for a supersonic exit. The boundary conditions based on these are summarized in Table 5.9. We validate the result with the study done by [30]. In this study, the wind tunnel data [28] has been used for validation. We have also compare the two MacCormak scheme results.

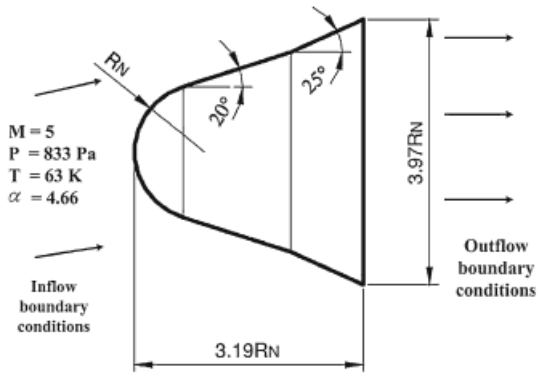


Figure 5.20: Re-entry vehicle model dimensions

Boundary Conditions

The plot of Cp distribution along the capsule wall is shown in Fig. 5.21.

Quantity	Inflow	Outflow
Pressure	833 Pa	833 Pa
Temperature	63 K	-
U velocity	792.88	-
V velocity	64.63	-
W velocity	0	-

Table 5.9: Boundary Condition

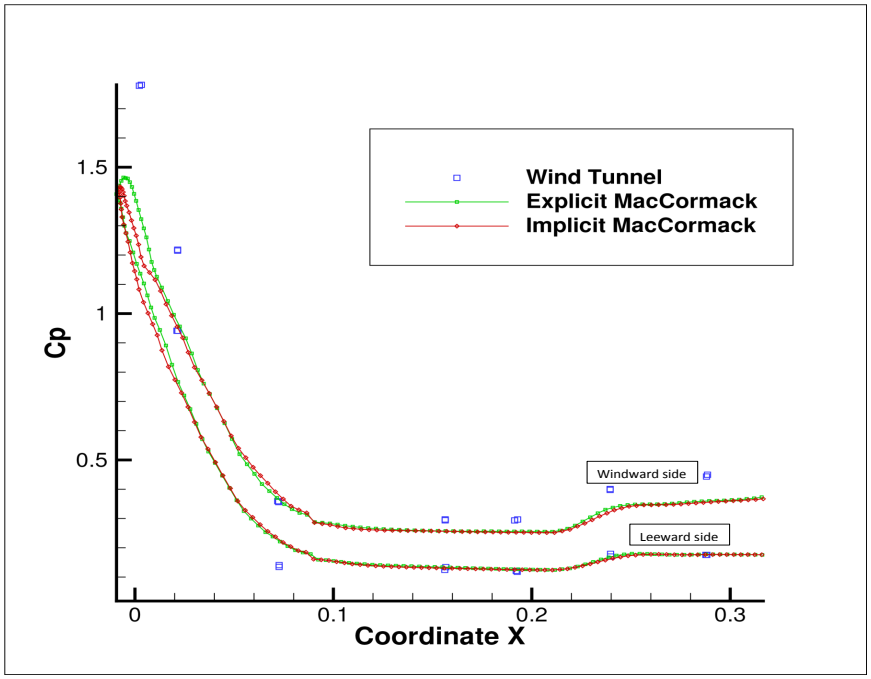


Figure 5.21: Variation of coefficient of pressure along the capsule wall

Method	CFL	Total Time	Time Steps	End Time
Explicit MacCormack	0.4	199441 s (55 hrs)(before 87hrs)*	396418	1 s
Implicit MacCormack	1.1	10038.25 s (2.8 hrs)	26617	0.003934 s

Table 5.10: Computational time comparison

5.2 Results of Implicit Scheme

The implicit scheme for Navier-Stokes equations are validated for flow over a flat plate.

5.2.1 Subsonic Flow over a Flat Plate Ma=0.5

We now consider the subsonic flow over a 2-D flatplate. Present results have been compared with analytical results. Courant number of 1.1 is used. The inflow conditions are summarized in the above table.

Computational Domain:

$$L_H = 1m$$

$$\delta = \frac{5L_H}{\sqrt{Re_L}}$$

$$L_v = 5\delta$$

$$L_v = 0.25$$

$$Re_L = \frac{\rho u L}{\mu}$$

$$\mu = \frac{\rho u L}{Re_L}$$

$$\mu = 0.0208445651 kg/m - sec$$

$$Pr = 0.71$$

$$k = \frac{\mu C_p}{Pr}$$

Boundary Conditions:

Wall:

Initial Conditions	Part
Pressure	101325 pa
Temperature	288.16 K
U Velocity	170.134 m/s
V Velocity	0 m/s
W Velocity	0 m/s

No slip: u=0 , v=0 , w=0 , $\frac{\partial T}{\partial n} = 0$ (adiabatic wall)

Non-Dimensional Y-distance:

$$\bar{y} = \frac{y}{x}\sqrt{Re_x}$$

Input and Farfield Boundary Conditions:

Input and farfield conditions are same as intial conditions in this case

For outlet or trailing edge x=L and thus $\bar{y}$ becomes

$$\bar{y} = \frac{y}{L}\sqrt{Re_L}$$

Flat Plate results are validated with the standard results from [32]. Here the plot for non-dimesional u velocity and non-dimensional $\bar{y}$ is shown.

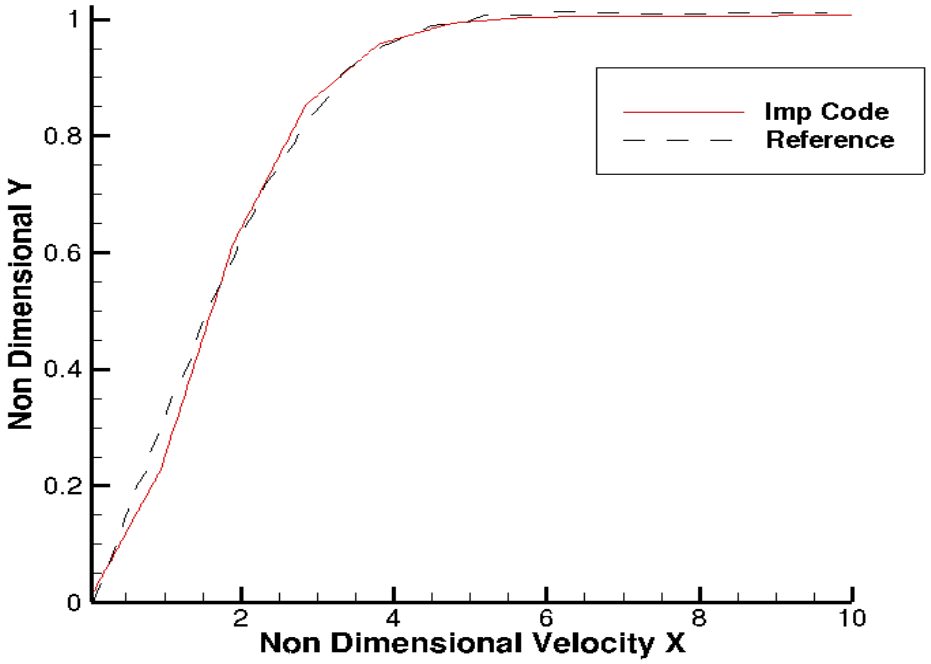


Figure 5.22: Plot for velocity x wrt y co-ordinate

5.2.2 Supersonic Flow over a Flat plate Ma=2.06

We now consider the supersonic flow over a 2-D flatplate. Present results have been compared with analytical results. Courant number of 1.1 is used. The inflow conditions are summarized in the table below.

Initial Conditions	Part
Pressure	101325 pa
Temperature	288.16 K
U Velocity	700.953 m/s
V Velocity	0 m/s
W Velocity	0 m/s

Computational Domain:

$$L_H = 1m$$

$$\delta = \frac{5L_H}{\sqrt{Re_L}}$$

$$L_v = 5\delta$$

$$Re_L = \frac{\rho u L}{\mu}$$

$$\mu = \frac{\rho u L}{Re_L}$$

$$\mu = 0.002202041 kg/m - sec$$

$$Pr = 0.71$$

$$k = \frac{\mu C_p}{Pr}$$

Boundary Conditions:

Wall:

No slip: u=0 , v=0 , w=0 , $\frac{\partial T}{\partial n} = 0$ (adiabatic wall)

Non-Dimensional Y-distance:

$$\bar{y} = \frac{y}{x}\sqrt{Re_x}$$

Input and Farfield Boundary Conditions:

Input and farfield conditions are same as intial conditions in this case

For outlet or trailing edge x=L and thus $\bar{y}$ becomes

$$\bar{y} = \frac{y}{L}\sqrt{Re_L}$$

Flat plate results are validated with the standard results from [32]. Here the plot for non-dimesional u velocity and non-dimensional $\bar{y}$ is shown.

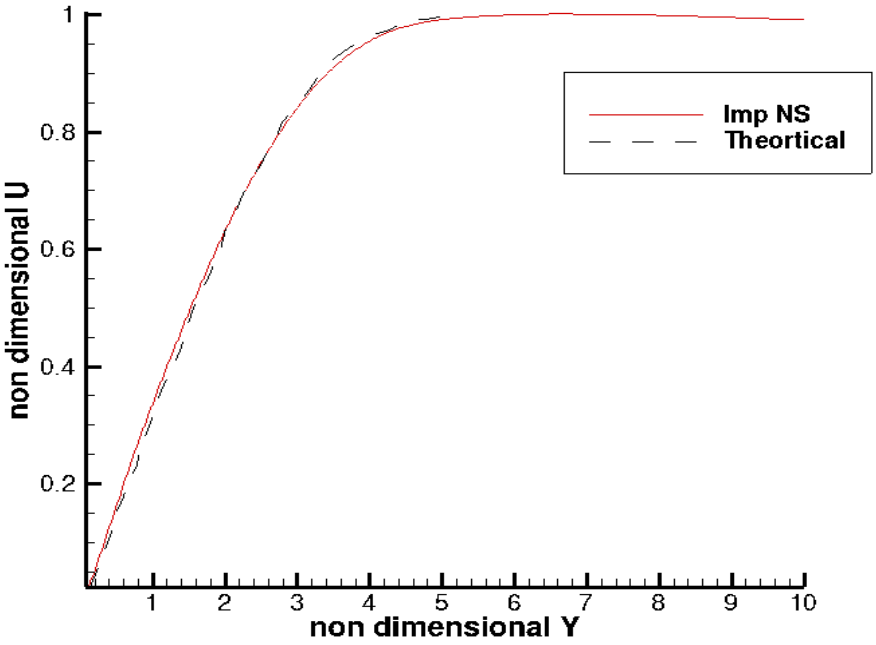


Figure 5.23: Plot for velocity x wrt y co-ordinate

5.3 Closure

In this chapter, time comparison between explicit scheme and implicit scheme are shown and implicit Navier-Stokes solver is validated with flow over flat plate for subsonic and supersonic flows.

Chapter 6

Conclusions and future work

Building on the earlier work of Sutrisha [21] and Rakesh [2], we have created a stand-alone version of the AnuPravaha Solver for computation of invisid and viscous flows using implicit schemes. An compressible flow module was created for the general-purpose CFD solver ANUPRAVHA, which uses the implicit MacCormack scheme with artificial viscosity in finite-volume form to solve the Euler and Navier-Stokes system (continuity, momentum and energy) of equations on a structured non-orthogonal multi-block grid.

The above method is stable for CFL of 1.1 , and is second order accurate in both space and time. In addition to this, the following features of this scheme should be pointed out.

a) For regions of the flow satisfying explicit stability criteria, the implicit method reduces to the corresponding explicit method and therefore no more computing time than the explicit scheme is needed in these regions. Due to this feature, the implicit MacCormack scheme is also called explicit-implicit or hybrid in some literature.

b) Viscous effects are included in the implicit operator in an approximate and very simple way to enhance the stability for viscous flows. Therefore the computation of the implicit operator and its inversion can be done with the help of the knowledge of the inviscid Jacobians. Two block bi-diagonal matrix inversions are reduced to two scalar bi-diagonal matrix inversions, a fact which greatly reduces the computation.

c) Although the scheme is unconditionally stable in von Neumann's sense, $\triangle t$ is still limited in practical computation, which is considered to be mainly due to the error created by approximate factorization taken in the procedure and the approximate linearization.

d) An intrinsic property of the two- step MacCormack type schemes, explicit or implicit, is the time step dependence of the steady state solution. Thus, convergent steady state solutions may only be reliable with sufficiently small $\triangle t$. Therefore one measure to achieve spatial accuracy is to reduce time step towards the end of the marching until variation of the solution with this reduction diminishes. This is obviously a disadvantage of the scheme for steady state solutions.

Future work can be in the direction of

1. Further validation of the present code can be made on complex geometries and for subsonic and transonic flow regimes.

2. Turbulence models like Spalart Allmaras, k-ϵ and k-ω can be implemented.

Appendix A

Jacobian Matrix Formulation

The inviscid Jacobians $|A|$, $|B|$ *and* $|C|$ can be diagonized by Sx, Sy and Sz respectively. i.e.

$$A = S_x^{-1} D_A S_x \qquad B = S_y^{-1} D_B S_y \qquad C = S_z^{-1} D_C S_z$$

$$S_x = \begin{pmatrix} 1 & 0 & 0 & 0 & \frac{-1}{c^2} \\ 0 & \rho c & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -\rho c & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{-u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 \\ \frac{-v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ \frac{-w}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 \\ \alpha\beta & -u\beta & -v\beta & -w\beta & \beta \end{pmatrix}$$

We use Mathematica to calculate the multiplication and inverse of the matricies.

So,

$$S_x = \begin{bmatrix} \left(1 - \frac{\alpha\beta}{c^2}\right) & \frac{u\beta}{c^2} & \frac{v\beta}{c^2} & \frac{w\beta}{c^2} & \frac{-\beta}{c^2} \\ -uc + \alpha\beta & c - u\beta & -v\beta & -w\beta & \beta \\ \frac{-v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ \frac{-w}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 \\ uc + \alpha\beta & -c - u\beta & -v\beta & -w\beta & \beta \end{bmatrix}$$

and

$$S_x^{-1} = \begin{bmatrix} 1 & \frac{1}{2c^2} & 0 & 0 & \frac{1}{2c^2} \\ u & \frac{c+u}{2c^2} & 0 & 0 & \frac{u-c}{2c^2} \\ v & \frac{v}{2c^2} & 0 & 0 & \frac{v}{2c^2} \\ w & \frac{w}{2c^2} & 0 & \rho & \frac{w}{2c^2} \\ a_{51} & a_{52} & v\rho & w\rho & a_{55} \end{bmatrix}$$

where

$$
\begin{aligned}
a_{51} &= u^2 + v^2 + w^2 - \alpha, \\
a_{52} &= \frac{1}{2\beta} + \frac{u}{2c} + \frac{u^2 + v^2 + w^2 - \alpha}{2c^2} \\
a_{55} &= \frac{1}{2\beta} - \frac{u}{2c} + \frac{u^2 + v^2 + w^2 - \alpha}{2c^2}
\end{aligned}
$$

$$
\begin{aligned}
S_y &= \begin{pmatrix} 1 & 0 & 0 & 0 & \frac{-1}{c^2} \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & \rho c & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\rho c & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{-u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 \\ \frac{-v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ \frac{-w}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ \alpha\beta & -u\beta & -v\beta & -w\beta & \beta \end{pmatrix} \\
&= \begin{bmatrix} \left(1 - \frac{\alpha\beta}{c^2}\right) & \frac{u\beta}{c^2} & \frac{v\beta}{c^2} & \frac{w\beta}{c^2} & \frac{-\beta}{c^2} \\ \frac{-u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 \\ -vc + \alpha\beta & -u\beta & c - v\beta & -w\beta & \beta \\ \frac{-w}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 \\ vc + \alpha\beta & -u\beta & -c - v\beta & -w\beta & \beta \end{bmatrix}
\end{aligned}
$$

and

$$
S_y^{-1} = \begin{bmatrix} 1 & 0 & \frac{1}{2c^2} & 0 & \frac{1}{2c^2} \\ u & \rho & \frac{u}{2c^2} & 0 & \frac{u}{2c^2} \\ v & 0 & \frac{v+c}{2c^2} & 0 & \frac{v-c}{2c^2} \\ w & 0 & \frac{w}{2c^2} & \rho & \frac{w}{2c^2} \\ a_{51} & u\rho & a_{53} & w\rho & a_{55} \end{bmatrix}
$$

where

$$
\begin{aligned}
a_{51} &= u^2 + v^2 + w^2 - \alpha, \\
a_{52} &= \frac{1}{2\beta} + \frac{v}{2c} + \frac{u^2 + v^2 + w^2 - \alpha}{2c^2} \\
a_{55} &= \frac{1}{2\beta} - \frac{v}{2c} + \frac{u^2 + v^2 + w^2 - \alpha}{2c^2}
\end{aligned}
$$

$$S_z = \begin{pmatrix} 1 & 0 & 0 & 0 & \frac{-1}{c^2} \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \rho c & 0 \\ 0 & 0 & 0 & -\rho c & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{-u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 \\ \frac{-v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ \frac{-w}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ \alpha\beta & -u\beta & -v\beta & -w\beta & \beta \end{pmatrix}$$

$$= \begin{bmatrix} \left(1 - \frac{\alpha\beta}{c^2}\right) & \frac{u\beta}{c^2} & \frac{v\beta}{c^2} & \frac{w\beta}{c^2} & \frac{-\beta}{c^2} \\ \frac{-u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 \\ \frac{-v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ -wc + \alpha\beta & -u\beta & -v\beta & c - w\beta & \beta \\ vc + \alpha\beta & -u\beta & -v\beta & -c - w\beta & \beta \end{bmatrix}$$

and

$$S_z^{-1} = \begin{bmatrix} 1 & 0 & 0 & \frac{1}{2c^2} & \frac{1}{2c^2} \\ u & \rho & 0 & \frac{u}{2c^2} & \frac{u}{2c^2} \\ v & 0 & \rho & \frac{v}{2c^2} & \frac{v}{2c^2} \\ w & 0 & 0 & \frac{w+c}{2c^2} & \frac{w-c}{2c^2} \\ a_{51} & u\rho & v\rho & a_{54} & a_{55} \end{bmatrix}$$

where

$$a_{51} = u^2 + v^2 + w^2 - \alpha,$$

$$a_{52} = \frac{1}{2\beta} + \frac{w}{2c} + \frac{u^2 + v^2 + w^2 - \alpha}{2c^2}$$

$$a_{55} = \frac{1}{2\beta} - \frac{w}{2c} + \frac{u^2 + v^2 + w^2 - \alpha}{2c^2}$$

Appendix B

Artificial Viscosity Formulation

The following explains the artificial viscosity formulation which has been frequently used in connection with the MacCormack technique. We show here the formulation for an unsteady, two-dimensional equation.

$$\frac{\partial U}{\partial t} = -\frac{\partial U}{\partial x} - \frac{G}{y} + J \tag{B.1}$$

where U is the solution vector, $U = \left[\begin{array}{cccc} \rho & \rho u & \rho v & \rho(e + V^2/2) \end{array} \right]$.

At each step of the time-marching solution, a small amount of artificial viscosity can be added in the following form:

$$\begin{aligned} S^t_{i,j} = C_x \frac{\left| p^t_{i+1,j} - 2p^t_{i,j} + p^t_{i-1,j} \right|}{p^t_{i+1,j} - 2p^t_{i,j} + p^t_{i-1,j}} (U^t_{i+1,j} - 2U^t_{i,j} + U^t_{i-1,j}) \\ + C_y \frac{\left| p^t_{i,j+1} - 2p^t_{i,j} + p^t_{i,j-1} \right|}{p^t_{i,j+1} - 2p^t_{i,j} + p^t_{i,j-1}} (U^t_{i,j+1} - 2U^t_{i,j} + U^t_{i,j-1}) \\ \text{where we have taken, } C_x = C_y = C_z = 0.12 \end{aligned} \tag{B.2}$$

Eq. B.2 is a fourth order numerical dissipation expression. On the predictor step $S^t_{i,j}$ is evaluated based on the known quantities at time t. On the corrector step, the corresponding value of $S^t_{i,j}$ is obtained by using the predicted (barred) quantities as $\bar{S}^t_{i,j}$.

$$\begin{aligned}\bar{S}^t_{i,j} = C_x \frac{\left|\bar{p}^t_{i+1,j} - 2\bar{p}^t_{i,j} + \bar{p}^t_{i-1,j}\right|}{\bar{p}^t_{i+1,j} - 2\bar{p}^t_{i,j} + \bar{p}^t_{i-1,j}}(\bar{U}^t_{i+1,j} - 2\bar{U}^t_{i,j} + \bar{U}^t_{i-1,j}) \\ + C_y \frac{\left|\bar{p}^t_{i,j+1} - 2\bar{p}^t_{i,j} + \bar{p}^t_{i,j-1}\right|}{\bar{p}^t_{i,j+1} - 2\bar{p}^t_{i,j} + p^t_{i,j-1}}(\bar{U}^t_{i,j+1} - 2\bar{U}^t_{i,j} + \bar{U}^t_{i,j-1})\end{aligned} \tag{B.3}$$

where we have taken, $C_x = C_y = C_z = 0.12$

The value of $S^t_{i,j}$ and $\bar{S}^t_{i,j}$ are added at various stages of MacCormack scheme as shown below with the help of calculation of density from the continuity equation. For this $U = \rho$.

On the predictor step,

$$\bar{\rho}^{t+\Delta t}_{i,j} = \rho^t_{i,j} + \left(\frac{\partial \rho}{\partial t}\right)^t_{i,j} \Delta t + S^t_{i,j} \tag{B.4}$$

On the corrector step,

$$\rho^{t+\Delta t}_{i,j} = \rho^t_{i,j} + \left(\frac{\partial \rho}{\partial t}\right)_a v\Delta t + \bar{S}^{t+\Delta t}_{i,j} \tag{B.5}$$

Bibliography

[1] Amit Shivaji Dighe, "Numerical Computation of Compressible Fluid Flows", Thesis for the Degree of Master of Technology, *Department of Mechanical Engineering*, Indian Institute of Technology Hyderabad, 2012.

[2] Ashwani Assam, "Development of a General-Purpose Compressible Flow AnuPravaha Based Solver", Thesis for the Degree of Master of Technology, *Department of Mechanical Engineering*, Indian Institute of Technology Hyderabad, 2014.

[3] Axel Rohde, "Eigenvalues and Eigenvectors of the Euler Equations in General Geometries", *AIAA journal*, Vol. 20, number 9 pp. 2001-2609.

[4] B. Engquist and A. Majda, "Radiation boundary conditions for acoustic and elastic wave calculations.", *Communications on Pure and Applied Mathematics 32*, pp- 313–357, 1979.

[5] C. B. Laney, " Computational gasdynamics", *Univ. Press, Cambridge u.a*, 1998

[6] C. Hirsch., "Numerical computation of internal and external flows fundamentals of computational fluid dynamics.", *Elsevier, Amsterdam*,2007.

[7] D. H. Rudy and J. C. Strikwerda, "A nonreflecting outflow boundary condition for subsonic navier-stokes calculations", *Journal of Computational Physics 36*, pp- 55–70. Cited by 0234, 1980.

[8] D. L. Whitfield and J. M. Janus, "Three-dimensional unsteady Euler equations solution using flux vector splitting", *00121*, 1984.

[9] Eswaran V. and Prakash S., "A finite volume method for Navier Stokes equations", *Proc. Third Asian CFD Conference*,Vol. 1, pp. 127-136, Bangalore, India, July 1998.

[10] Eswaran V. et al., "Development of a General Purpose Robust CFD Solver", Project Report No. 1, *Department of Mechanical Engineering*, Indian Institute of Technology Kanpur, December, 2005.

[11] G. Hedstrom, "Nonreflecting boundary conditions for nonlinear hyperbolic systems", *Journal of Computational Physics 30*, pp- 222–237, 1979.

[12] Gurris, Marcel and Kuzmin, Dmitri and Turek, Stefan, "Implicit finite element schemes for the stationary compressible Euler equations", *International Journal for Numerical Methods in Fluids*, vol-69, No-1, pp=1-28, Wiley Online Library, 2012

[13] Harten Ami, "Numerical Solution of the Navier–Stokes Equations by Semi–Implicit Schemes", *Journal of Computational Physics*, vol. 49, pp. 357–393

[14] Hozman, J, "High resolution schemes for hyperbolic conservation laws", *Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic. Wds*, vol. 6, pp. 59-64

[15] Feistauer, Miloslav and CESENEK, JAN, "ON NUMERICAL SIMULATION OF AIRFOIL VIBRATIONS INDUCED BY COMPRESSIBLE FLOW", Thesis for the Degree of Master of Technology, *Proceedings of ALGORITMY*, pp. 22-31, 2012.

[16] Fürst, J and Furmánek, P, "An implicit MacCormack scheme for unsteady flow calculations", *Computers & Fluids*, Vol. 46,number 1, pp. 231–236, 2011.

[17] J. Blazek., "Computational fluid dynamics principles and applications.", *Elsevier, Amsterdam; San Diego*, 2005.

[18] J. D. Anderson., "Computational fluid dynamics: the basics with applications", *McGraw-Hill, Boston, Mass*, 2003.

[19] J. D. Anderson., "Modern compressible flow: with historical perspective", *McGrawHill, New York*, 1995.

[20] J. Rincon and R. Elder, "A high-resolution pressure-based method for compressible flows", *Computers & fluids 26*, pp-217–231, 1997.

[21] Kalkote Nikhil Narayan, "Development of Compressible Flow Module In AnuPravaha Solver", Thesis for the Degree of Master of Technology, *Department of Mechanical Engineering*, Indian Institute of Technology Hyderabad, 2013.

[22] MacCormack, Robert William, "A numerical method for solving the equations of compressible viscous flow", *AIAA journal*, Vol. 20, number 9 pp. 1275-1281, 1982.

[23] MacCormack, K. R. Kneile, "Implicit solution of the 3-D compressible Navier-Stokes equations for internal flows", *Springer-Verlag*, Ninth International Conference on Numerical Methods in Fluid Dynamics, 1985.

[24] N. S. Madhavan, V. Swaminathan, "Implicit numerical solution of unsteady Euler equations for Transonic flows", *Indian J. pure appl. Math*, vol. 17(9),pp: 1164-1173,September 1986.

[25] Maurice J Zucrow and Joe D Hoffmann, "Gas dynamics", *Vol 1*, 1796.

[26] Rhie C.M. and Chow W.L., "Three-dimensional unsteady Euler equations solution using flux vector splitting", *00121*,1983.

[27] Richtmyer, R.D. and Morton, K.W., "Difference methods for initial-value problems", *Interscience Publishers*, 1969.

[28] R. Kalimuthu., "Surface Pressure Measurement Results on the SRE (biconic) Configuration at Mach = 5.",

[29] Sommer, Tomáš and Helmich, Martin, "Mathematical modeling of fluid flow using the numerical scheme with artificial viscosity",

[30] S. Nagdewe, G. Shevare, and H.-D. Kim., "Study on the numerical schemes for hypersonic flow simulation.", *Shock Waves 19,*, pp. 433–442, 2009.

[31] Qin, Ning, "Towards numerical simulation of hypersonic flow around space-plane shapes", *PhD thesis*, The Univrsity of Glasgow, 1987.

[32] A. H. Shapiro., "The dynamics and thermodynamics of compressible fluid flow, Vol. 2.",