

Development of a General-Purpose Density Based AnuPravaha Solver

Kuchana Rakesh

A Dissertation Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology

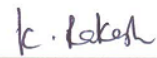


Department of Mechanical & Aerospace Engineering

June 2015

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.



(Signature)

(Kuchana Rakesh)

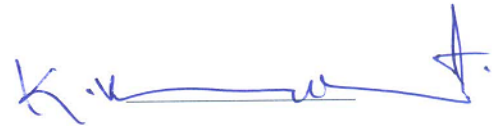
(ME13M1031)

Approval Sheet

This Thesis entitled Development of a General-Purpose Density Based AnuPravaha Solver by Kuchana Rakesh is approved for the degree of Master of Technology from IIT Hyderabad



(Dr. Mangadoddy Narasimha) Examiner
Dept. of Chemical Engineering
IITH



(Dr. K. Venkatasubbaiah) Examiner
Dept. of Mechanical & Aerospace Engineering
IITH



(Prof. Vinayak Eswaran) Adviser
Dept. of Mechanical & Aerospace Engineering
IITH

Acknowledgements

I feel honored to express my deepest and most sincere gratitude to my thesis supervisor, Prof. Vinayak Eswaran, for his invaluable guidance, kind suggestion and encouragement throughout the progress of this work. I am highly indebted to him for providing me ample freedom to work my way and simultaneously giving me a taste of research work.

I would like to thank Narendra Gajbhiye, Praveen Throvagunta and Ashwani Assam for their valuable suggestion at various point 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 make a special mention of the computational facilities provided by my thesis supervisor.

Finally I would like to thank my parents, my sister's for their constant encouragement and support.

Abstract

With India focusing 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 the research of Aerospace problems. The present work aims to extend the Euler Solver developed by previous post-graduate students as part of the general purpose AnuPravaha Solver to Navier-Stokes solver and develop it as a stand-alone compressible flow solver. The solver has been extended for laminar flows, and in this present work viscous terms have been implemented successfully. Boundary conditions are also implemented successfully for viscous flows. The Navier-Stokes solver is tested successfully for subsonic and supersonic cases for flow over a flat-plate and other problems and compared with analytical or numerical results from previous researchers.

Contents

Declaration	ii
Approval Sheet	iii
Acknowledgements	iv
Abstract	v
1 Introduction	1
1.1 Literature review	2
1.2 Objective of present work	3
2 Mathematical Nature Of Equations And Boundary Conditions	4
2.1 Domain of dependance and zone of influence	5
2.1.1 Scalar Conservation Law	5
2.1.2 System of PDEs	6
2.2 System of partial differential equation in multi-dimensions	8
2.2.1 System of First Order Steady-State PDEs	9
2.2.2 Characteristic and Characteristic Surface in Multi-dimensions	10
2.3 Advantage of Conservation form over the non-conservation form	14
2.4 Boundary Condition Specification in Hyperbolic System	14
2.5 Closure	16
3 Navier Stokes Equations, Boundary Conditions and CFL Criterion	17
3.1 A Short Description of Navier-Stokes Equations	17
3.2 Boundary Condition Treatment In Terms of Primitive Variables	19
3.2.1 Implementation of Boundary Conditions	19
3.2.2 Inflow BC	21
3.2.3 Outflow BC	21
3.2.4 Wall (or Solid) Boundary	22
3.2.5 Symmetry Boundary Conditions	22
3.3 CFL Condition	22
3.4 Closure	23

4	Numerical Methodology For Navier-Stokes Equations	24
4.1	Numerical Schemes	24
4.2	Governing Equations	25
4.2.1	Viscous Terms for X-Momentum:	26
4.2.2	Viscous Dissipation Term in Energy equation	27
4.3	Discretization of Governing Equation	28
4.4	MacCormack Scheme in FVM	32
4.4.1	Artificial Viscosity	33
4.4.2	MacCormack Algorithm	34
4.5	Advection Upstream Splitting Method (AUSM)	34
4.5.1	AUSM Algorithm	37
4.6	Diffusion term:	38
4.6.1	Diffusion Fluxes:	39
4.6.2	Computation of Spatial Derivatives at Cell-Center of a Non-orthogonal Grid (Least Square Method) :	43
4.7	MacCormack Scheme Discretization for 2D Rectangular Grids	43
4.7.1	MacCormack Finite Difference Discretization in 2D	44
4.7.2	Discretization of Viscous Terms with Central Difference	45
4.7.3	2d-MacCormack Finite Volume Discretization for Rectangular Grids	46
4.8	Closure	47
5	Results And Discussion	49
5.1	Preliminary Results from 2D-Rectangular Codes for Supersonic Flow Over a Flat Plate	49
5.1.1	MacCormack Finite Difference Method:	51
5.1.2	MacCormack Discretization vs Central Discretization of Viscous Terms	54
5.1.3	Finite Difference Method vs Finite Volume Method	56
5.2	Supersonic flow Over a Flat Plate $Ma=2.06$, $Re=39,0000$	58
5.3	Subsonic Flow Over a Flat Plate $Ma=0.5$, $Re=10,000$	63
5.4	Supersonic Flow Over a Wedge	67
6	Conclusions and Future work	72
	Appendices	74
	A Artificial Viscosity Formulation	75
	References	77

Chapter 1

Introduction

Compressible flow effects are encountered in numerous engineering applications involving high speed flows, e.g. gas turbines, steam turbines, internal combustion engines, rocket engines, high-speed aerodynamics, high speed propellers, gas pipe flows, etc. Special phenomena such as compression shocks, entropy layers, expansion fans, flow induced noise etc. are of fundamental scientific importance and directly affect the performance and endurance of these engineering applications. Taking this as the motivation, the task of incorporating a density-based solver in the Anupravaha - General purpose CFD solver was undertaken two years ago. This solver, which was initially only for incompressible flows, has been under continuous development since 2004 in the research group of Prof. Vinayak Eswaran. It already has the Pressure-based solver having modules for turbulence, multiphase flows, radiation problems, MHD, Solidifications and Melting, etc. To cater exclusively to aerospace applications and for the ease of management of the solver, we have now tried to separate out the density-based solver in a stand-alone code.

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 at Mach number of 0.3 and greater.

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 discontinuous solutions in certain cases e.g vortex sheets, contact discontinuities or shock waves. So, for compressible flows special attention is required for solution methods which will accurately capture these discontinuities.

In compressible fluid flows, properties are not only transported by the flow, but also by the propagation of waves. This requires the construction of flux interpolations, that take into account that transports can occur in any direction.

Thus, a major difference between solution methods for compressible flow and incompressible flow lies in the boundary conditions implementation. In compressible flows, 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.

1.1 Literature review

Hirsch [1], 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. Euler equations are solved in conservative form but require imposition of boundary condition in primitive form; in Chapter 19 [2] discusses the implementation of boundary conditions (both physical and numerical) from characteristic extrapolation for conservative and primitive variables, along with different extrapolation methods.

All the boundary conditions have been implemented to incorporate curved surface, based on the book of Blazek [3], and the work done in [4].

The second focus of this thesis has been the incorporation of High Resolution schemes into the solver. The second volume of [2] discusses almost all basic numerical schemes pertaining to Euler and Navier-Stokes equations.

Discretization of the Navier-Stokes equation using the MacCormack method can be found in previous work [5] but it was using a finite-difference scheme, with discretization of viscous fluxes which are forward differenced in the predictor step and backward differenced in the corrector step. In this present work, a finite-volume method is used with central discretization of viscous terms in both the predictor and corrector steps, compared with the standard MacCormack discretization. The final code is a 3D structured solver using finite-volume method.

In discretization of viscous fluxes by the old method we need gradients at the faces, which are obtained by constructing an auxiliary control volume centered at the face [3]. An alternative approach would be that the gradients are first evaluated for each cell (similar

to the convective fluxes) and then averaged to obtain the gradient at the cell faces. Although this approach may appear more attractive than the current methodology, it is not recommended since it leads to strong odd-even decoupling.

Chandrasekhar [6] in his thesis explained the finite-volume discretization of viscous terms for the momentum equations, which was also used in this work. We have discretized the energy equation in a similar manner. In this formulation there is no odd-even decoupling.

Finding the gradients at the center can be done either by Green-Gauss or least square cell method [3], but as the least-square cell method is more accurate than the Green-Gauss method, this present work uses the least square method.

However, for better stability and accuracy, Assam [7] implemented the upwind-based scheme AUSM+[8] with second order accuracy. This scheme was first presented by Liou in 1993 as Advection Upstream Splitting Method (AUSM) [9]. The scheme has went through several revision since then, which can be found in [10].

1.2 Objective of present work

- Expanding the inviscid Euler solver to viscous laminar Navier-Stokes solver.
- Incorporating the MacCormack and AUSM+ algorithms for this solver and expanding the boundary conditions for viscous flows in the Compressible Anupravaha solver.
- To validate this solver for different regimes of flow i.e. subsonic, transonic and in supersonic regime.

Chapter 2

Mathematical Nature Of Equations And Boundary Conditions

Let us consider the system of governing differential equations for inviscid compressible flow in primitive variable form, also known as the Euler equations, describing the conservation of mass, momentum and energy.

$$\begin{aligned}u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} + \rho \frac{\partial u}{\partial x} + \rho \frac{\partial v}{\partial y} &= 0 \\u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= -\frac{1}{\rho} \frac{\partial p}{\partial x} \\u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} &= -\frac{1}{\rho} \frac{\partial p}{\partial y}\end{aligned}\tag{2.1}$$

However, for the special case of irrotational compressible flow we can equivalently write the potential flow equation as,

$$\left(1 - \frac{u^2}{c^2}\right) \frac{\partial^2 \phi}{\partial x^2} - \frac{2uv}{c^2} \frac{\partial^2 \phi}{\partial x \partial y} + \left(1 - \frac{v^2}{c^2}\right) \frac{\partial^2 \phi}{\partial y^2} = 0\tag{2.2}$$

where ϕ is the potential function defined by,

$$u = \frac{\partial \phi}{\partial x} \quad ; \quad v = \frac{\partial \phi}{\partial y}\tag{2.3}$$

These two sets of PDE's describe exactly the same physics for steady irrotational flow. But looking at the forms of the equations, they appear to be quite different, with the first set of equations (2.1) seeming to be convection dominated, whereas the second Eq. (2.2) seemingly diffusion dominated. Convection and diffusion are two important phenomena in fluid mechanics, and relate to very different physical behavior. But as we know, both

Eq. (2.1) and Eq. (2.2) describe the same physics.

So, it is clear from the above discussion that one cannot predict the behavior of the solutions of differential equation by just looking at the form of the equations. It is not the form of the governing equation which decides the behavior of solutions; rather, it is the eigenvalue matrix (discussed in the next section) which determines this, and provides a tool to analyze the nature of governing equations, independent of physical form, to tell us something about the nature of their solutions. Thus, in these chapter we try to learn these tools and others to develop our understanding of hyperbolic equations.

2.1 Domain of dependance and zone of influence

2.1.1 Scalar Conservation Law

To understand the propagation of information in the solutions of governing equations, consider a single equation describing a conservation law:

$$\frac{\partial w}{\partial t} + \frac{\partial F}{\partial x} = 0 \quad w(x, 0) = w_0(x) \quad (2.4)$$

$$\frac{\partial w}{\partial t} + \frac{dF}{dw} \frac{\partial w}{\partial x} = 0 \quad (2.5)$$

Introducing wave speed as, $\frac{dF}{dw} = \lambda$ in the above Eq. (2.5) can also be written as,

$$\frac{\partial w}{\partial t} + \lambda \frac{\partial w}{\partial x} = 0 \quad (2.6)$$

An analytical solution to above equation exists and can be found using the method of separation of variables, and can be written as,

$$w(x, t) = w_0(x - \lambda t) \quad (2.7)$$

The analytical solution of this equations for a constant and positive wave-speed shows that solution is constant along the line satisfying satisfying the condition $\frac{dx}{dt} = \lambda$; this line is called the *characteristic*. For linear PDEs the solution is constant along the characteristic, while for non-linear PDEs the solution may vary along characteristic but it will be purely a function of the curvilinear co-ordinate describing the characteristic. Now from Eq. (2.7), the solution at a point for the hyperbolic systems is dependent on the previous solutions at points lying on the same characteristic, and ultimately on the initial condition at that characteristic at $t=0$. So, for a single first-order hyperbolic equation all the points on a particular characteristic line form the domain of dependence and zone of influence of any point on that characteristic.

2.1.2 System of PDEs

Now, moving one step forward, consider a general 1-D unsteady system of first-order PDEs,

$$\frac{\partial w_i}{\partial t} + \frac{\partial F_i}{\partial x} = 0 \quad (2.8)$$

where w_i is a vector containing conservative variables

F_i is a vector containing flux associated with w_i

The Eq. (2.8) can be written as,

$$\frac{\partial w_i}{\partial t} + [A] \frac{\partial w_i}{\partial x} = 0 \quad (2.9)$$

where $[A]$ is called the *Jacobian* (matrix).

The Jacobian in Eq. (2.9) determines the behavior of the solutions, based on the nature of its eigenvalues. Diagonalizing the Jacobian in Eq. (2.9), we can rewrite the equation as:

$$\frac{\partial w_i}{\partial t} + [Q_r] [\lambda_i] [Q_l] \frac{\partial w_i}{\partial x} = 0 \quad (2.10)$$

where Q_l , Q_r are the left and right eigenvector matrices explained through the following relations.

For any diagonalizable matrix A , Q_r is a matrix whose columns \mathbf{r}_i are *right characteristics vectors* or *right eigenvectors* of A , and Q_l is a matrix whose rows \mathbf{l}_i are *left characteristics vectors* or *left eigenvectors* of A .

The right characteristic vectors are defined as follows:

$$A\mathbf{r}_i = \lambda_i\mathbf{r}_i \quad (2.11)$$

While less familiar, left characteristic vectors are defined in almost the same way as right characteristic vectors, except that left characteristic vectors multiply A on the left than on the right.

$$\mathbf{l}_i A = \lambda_i \mathbf{l}_i \quad (2.12)$$

Also,

$$[\lambda] = [Q_l][A][Q_r]$$

$$[Q_l] = [Q_r]^{-1}$$

As mentioned earlier, the nature of λ_i determines the behavior of the solution. If a full set of real non-zero eigenvalues exists then the system of equations is called *Hyperbolic*, if a

full set of real eigenvalues does not exist then the system is called *Parabolic* and if some of eigenvalues are complex then the system is called *Elliptic*.

Let us now consider a hyperbolic system. By multiplying Eq. (2.10) by $[Q_l]$ and rewriting equation(2.10) we get

$$[Q_l] \frac{\partial w_i}{\partial t} + [Q_l][Q_r][\lambda_i][Q_l] \frac{\partial w_i}{\partial x} = 0 \quad (2.13)$$

As we know,

$$[Q_l] = [Q_r]^{-1} \quad (2.14)$$

$$[Q_l] \frac{\partial w_i}{\partial t} + [\lambda_i][Q_l] \frac{\partial w_i}{\partial x} = 0 \quad (2.15)$$

Introducing new set of variables as,

$$\delta v_i = [Q_l] \delta w_i \quad (2.16)$$

Eq. (2.15) can be written as,

$$\frac{\partial v_i}{\partial t} + [\lambda_i] \frac{\partial v_i}{\partial x} = 0 \quad (2.17)$$

where $[\lambda_i]$ is a diagonal matrix of eigenvalues.

Now, note the simplicity of Eq. (2.17) in comparison to Eq. (2.8); unlike in the latter equation, the equations in system (2.17) are decoupled, i.e., the solution of any one of them is independent of the solution of others. Thus the component equations of (2.17) can thus be solved separately and easily. Then, by inverting Eq. (2.16) we can obtain the solution to system of Eq. (2.8).

As shown in the figure for a system of 3 equations, the information flowing along corresponding characteristic lines passing through a point P determine the solution at P by the superposition of the characteristic information. From this, it is not difficult to show that solution at point P depends only on the solution in region APCBA and it has nothing to do with the solution outside this region. Hence this region is termed as the *domain of dependence*. By extending the characteristic lines beyond point P, we can say that solution at this point is going to affect the solution in the region FPDE, hence this region is termed the *zone of influence*. The point to note here is that a numerical scheme determining the solution (for a time-step) at point P must include only points from the domain of dependence to capture the solution correctly — failing to do this causes serious issues, mostly resulting in the blowing up of the solution. This condition is called *Courant-Friedrichs-Lewy* condition and will be discussed in the next chapter.

Depending upon the nature (positive or negative valued) of the eigenvalues, character-

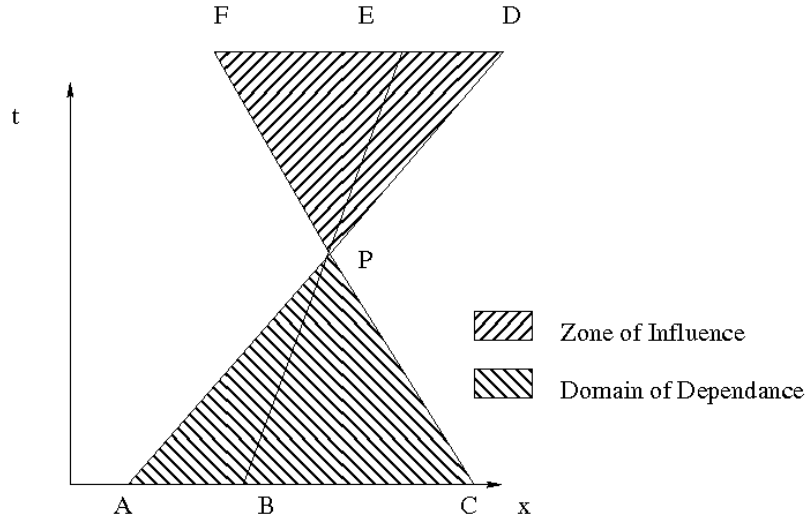


Figure 2.1: Characteristics in 3 equation system

istic information may flow from left to right or right to left (Fig. 2.1). So, the problem to well-posed boundary conditions must be handled carefully. We will discuss this in the next few sections.

Further discussion over characteristics can be found in the ([11], Section 2). Here, in the following section we shall present the method to find the nature of the governing equations for the multi-dimensional cases.

2.2 System of partial differential equation in multi-dimensions

By direct matrix manipulation we can find out the nature of governing equation for 1-D case. We discuss here the general method to find out the nature of governing equations applicable for all the cases. Governing equations defining conservation laws in multi-dimensions can be written as:

Steady state:

$$[A] \frac{\partial w_i}{\partial x} + [B] \frac{\partial w_i}{\partial y} + [C] \frac{\partial w_i}{\partial z} = 0 \quad (2.18)$$

Unsteady state:

$$[I] \frac{\partial w_i}{\partial t} + [A] \frac{\partial w_i}{\partial x} + [B] \frac{\partial w_i}{\partial y} + [C] \frac{\partial w_i}{\partial z} = 0 \quad (2.19)$$

Diagonalizing the Jacobian matrices, we get:

$$I \frac{\partial w_i}{\partial t} + [Q_a][\lambda_a][Q_a]^{-1} \frac{\partial w_i}{\partial x} + [Q_b][\lambda_b][Q_b]^{-1} \frac{\partial w_i}{\partial y} + [Q_c][\lambda_c][Q_c]^{-1} \frac{\partial w_i}{\partial z} = 0 \quad (2.20)$$

Since, $[Q_a] \neq [Q_b] \neq [Q_c]$ we cannot replace, as before the conservative variables with another set such that the equations in the system (2.18) or (2.19) gets decoupled. Therefore a deeper analysis needs to be done to get eigenvalues and characteristic variables.

2.2.1 System of First Order Steady-State PDEs

The following steps define the procedure to identify the nature of a mathematical system. These are taken from [1]:

Step 1: Write the system of PDEs describing the mathematical model as a system of first order PDEs

Suppose we have n unknown variables w_j , in $(m + 1)$ -dimensional space x_J , we can group all the variables w_j in an $(n \times 1)$ vector column w and write the system of first order PDEs under the general form:

$$\sum_j A_j \frac{\partial w}{\partial x_j} = T \quad j = 1, 2, 3, \dots, m + 1$$

$$w = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \cdot \\ \cdot \\ \cdot \\ w_n \end{pmatrix} \quad (2.21)$$

where A_j are $(n \times n)$ matrices and T is a column vector of the non-homogeneous source terms. The matrices A_j and T can depend on x_j and w , but not on the derivatives of w .

Step 2: Consider a plane wave solution of amplitude \hat{U} in the space of the independent variables x with components $x_j (j = 1, \dots, m + 1)$, defined by

$$w = \hat{U} e^{i(\vec{n} \cdot \vec{x})} \quad (2.22)$$

where $i = \sqrt{-1}$, \vec{n} is a vector in the m -dimensional space of the independent variables x_j and \hat{U} is an $(n \times 1)$ column vector.

Step 3: Introduce this solution in the homogeneous part of the system (2.21) and find the values of n satisfying the resulting equation.

The homogenous part of Eq. (2.21) is written as

$$\Sigma_j A_j \frac{\partial w}{\partial x_j} = 0 \quad j = 1, 2, 3 \dots m + 1 \quad (2.23)$$

and the function (2.22) is a solution of this system of equations if the homogeneous algebraic system of equations:

$$[\Sigma_j A_j n_j] \hat{U} = 0 \quad (2.24)$$

has non-vanishing solutions for the amplitude \hat{U} . This will be the case if and only if the determinant of the matrix $\Sigma_j A_j n_j$ vanishes.

Step 4: Find the n solutions of the equation

$$\det [\Sigma_j A_j n_j] \quad (2.25)$$

Eq. (2.25) defines a condition on the normals \vec{n} . This equation can have at most n solutions, and for each of these normals n_i , the system (2.25) has a non-trivial solution.

The system is said to be hyperbolic if all the n characteristic normals n_i are real and if the solutions of the n associated systems of equations (2.25) are linearly independent. If all the characteristics are complex, the system is said to be elliptic. If some are real and other complex the system is classed as hybrid. If the matrix $\Sigma_j [A_j n_j]$ is not of rank n , i.e. there are less than n real characteristic normals then the system is said to be parabolic.

The last case will occur, for instance, when at least one of the variables, say w_1 has derivatives with respect to one coordinate, say x_1 , missing. This implies that the components $A_1 = 0$ for all equations i .

2.2.2 Characteristic and Characteristic Surface in Multi-dimensions

Parabolic and hyperbolic equations play an important role in CFD, due to their association to diffusion and convection phenomena. They are recognized by the existence of real characteristic normals, solutions of Eq. (2.25). Each of these normals n_i defines therefore normal to the surface, which is called the characteristic surface. We will show here the very important consequences of these properties, as they have a significant effect on the whole process of discretization in CFD.

If we define a surface $S(x_j) = 0$, in the $(m + 1)$ -dimensional space of the independent variables x_j , the normal to this surface is defined by the gradient of the function $S(x_j)$, as

$$\vec{n} = \frac{\nabla S}{\|\nabla S\|} \quad (2.26)$$

(Henceforth, the normalizing $\|\nabla S\|$ is to be absorbed into the function S).

What is the significance of this characteristic surface in terms of wave propagation, referring to the plane wave solution Eq. (2.22)?

If Eq. (2.26) is introduced in the plane wave Eq. (2.22), a general representation is defined as,

$$w = \hat{U} e^{i(\vec{x} \cdot \nabla S)} = \hat{U} e^{i(x_j S_j)} \quad \text{with } S_j \equiv \frac{\partial S}{\partial x_j} \quad (2.27)$$

If we consider the tangent plane to the surface $S(x_j) = 0$, defined by

$$S(x_j) = S(0) + \vec{x} \cdot \nabla S = S(0) + x_j \frac{\partial S}{\partial x_j} = S(0) + x_j n_j \quad (2.28)$$

we observe that along the constant values of the phase of the wave $\phi = \vec{x} \cdot \nabla S$, the quantity w is constant.

Hence, we can consider that, the quantity U is propagating at a constant value in the direction of the normal \vec{n} .

The surface S is called a wave-front surface, defined as the surface separating the space domain already influenced by the propagating quantity w from the points not yet reached by the wave.

Observe that in the general case of n unknown flow quantities u_i , we have n characteristic surfaces, for a pure hyperbolic problem.

In a two-dimensional space the characteristic surface reduces to a characteristic line. The properties w are transported along the line $S(x, y) = 0$ and the vectors tangent to the characteristic line are obtained by expressing that along the wavefront:

$$dS = \nabla S \cdot dx = \frac{\partial S}{\partial x} dx + \frac{\partial S}{\partial y} dy = 0 \quad (2.29)$$

Hence, the direction of the characteristic line in two dimensions is given by

$$\frac{dy}{dx} = -\frac{S_x}{S_y} = -\frac{n_x}{n_y} \quad (2.30)$$

In two dimensions, there are two characteristic directions for a hyperbolic equation. Hence out of each point in the (x, y) domain, two characteristics can be defined, along which two quantities propagate. As we have as many unknowns, at each point the solution can be obtained from the characteristic-related quantities that have propagated from the

boundary or initial condition to that point.

To get a physical understanding of the discussion so far we can consider unsteady inviscid flow. The unsteady inviscid flow equation or the unsteady Euler Equation in non-conservation form is written as

$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} + w \frac{\partial \rho}{\partial z} = 0 \quad (2.31)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial x} \quad (2.32)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial y} \quad (2.33)$$

$$\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial z} \quad (2.34)$$

$$\frac{\partial E}{\partial t} + u \frac{\partial E}{\partial x} + v \frac{\partial E}{\partial y} + w \frac{\partial E}{\partial z} = -\frac{1}{\rho} \left(\frac{\partial p u}{\partial x} + \frac{\partial p v}{\partial y} + \frac{\partial p w}{\partial z} \right) \quad (2.35)$$

By following the analysis given in Sec. 2.2, we can find that the above governing equations is *hyperbolic*, no matter whether the flow is locally subsonic or supersonic. More precisely, we say the flows are hyperbolic with respect to time. (The classification of the unsteady Euler Equations as hyperbolic with respect to time is derived in Sec 11.2.1 of [5].) This implies that in such unsteady flows, no matter whether we have one, two, or three spatial directions, the marching direction is always the *time* direction. Let us examine this more closely to understand the marching behavior discussed before for hyperbolic partial differential equations. For one dimensional flow, consider a point P in the xt plane shown in Fig. 2.2. The region influenced by P is the shaded area between the two advancing characteristics through P . The x -axis ($t = 0$) is the initial data line. The interval ab is the only portion of the initial data along the x axis which the solution at P depends. Extending these thoughts for two-dimensional unsteady flow, consider point P in the xyt space as shown in Fig. 2.3. The region influenced by P and the portion of the boundary in the xy plane upon which the solution at P depends are shown in this figure. Starting with known initial data in the xy plane, the solution “marches” forward in time. The same extension can be applied to the 3D case.

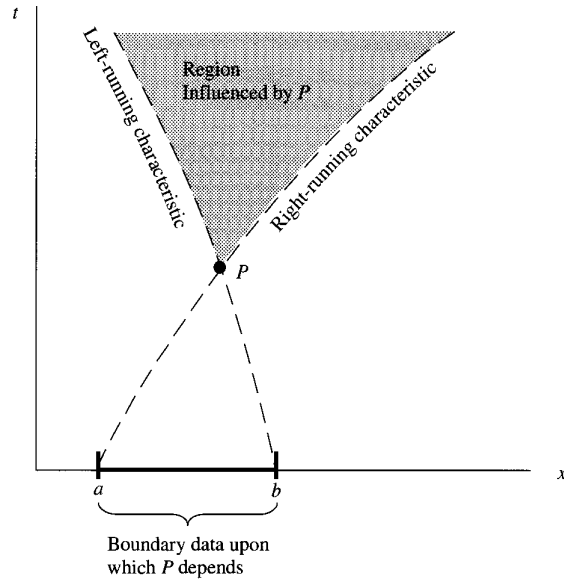


Figure 2.2: Domain and boundaries for the solution of hyperbolic equations. One dimensional unsteady flow. [5]

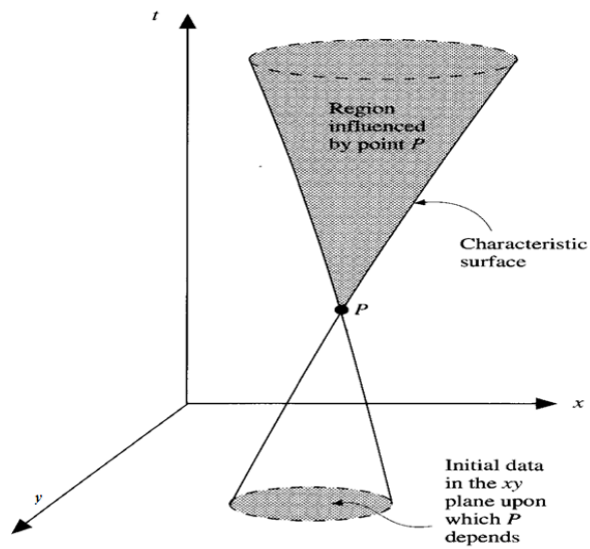


Figure 2.3: Domain and boundaries for the solution of hyperbolic equations. Two dimensional unsteady flow. [5]

2.3 Advantage of Conservation form over the non-conservation form

The conservation and non-conservation form of the continuity equation is shown below.

Conservation form:

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{V}) = 0$$

Non-Conservation form:

$$\frac{D\rho}{Dt} + \rho \vec{\nabla} \cdot (\vec{V}) = 0 \text{ with } \frac{D}{Dt} \equiv \frac{\partial}{\partial t} + \vec{V} \cdot \vec{\nabla}$$

The labelling of the governing equations as either conservation or non-conservation form grew out of modern CFD, as well as concern for which method has to be preferred for a given CFD applications. We shall state here the two perspectives for the advantage of conservation form over the non-conservation form. The detail understanding of these can be found in [5].

1. The conservation form of the governing equations allows to write the system of equation in a general form. Thus, it provides an ease and better organization for numerical and computer programming.
2. Experience has shown that the conservation form of equation is better for shock-capturing method (used in this thesis). For the non-conservation form, the computed flow-field has unsatisfactory results. The reason for this is, the conservation form uses flux variables as the dependent variable and because the changes in these flux variables are either zero or small across a shock wave, the numerical quality of the shock-capturing method will be enhanced. Whereas, the non-conservation form uses the primitive variables as dependent variable, and one would see a large discontinuity in them.

2.4 Boundary Condition Specification in Hyperbolic System

Boundary condition specification is an important part of any CFD problem statement and has to be compatible with physical and numerical properties of problem.

We have already seen that information in a hyperbolic problem propagates in a specific characteristic direction, the eigenvalue spectrum of the Jacobian matrix defines how information is going to propagate. Hence for a hyperbolic problem to be well-posed we cannot specify general boundary conditions on all boundaries. Rather, the following questions have to be answered:

1. How many boundary conditions have to be imposed at a given boundary?
2. What are the boundary conditions that have to be imposed at the boundary?
3. How are the remaining variables (i.e., those without BCs) to be handled at the boundary?

In this section we will discuss only the answer to the first question pertaining to the Euler problem. The five eigenvalues of the system, which correspond to the speed of propagation of five characteristic quantities, are given by,

$$\frac{\vec{u} \cdot \vec{k}}{k}, \quad \frac{\vec{u} \cdot \vec{k}}{k}, \quad \frac{\vec{u} \cdot \vec{k}}{k}, \quad \frac{\vec{u} \cdot \vec{k}}{k} + c, \quad \frac{\vec{u} \cdot \vec{k}}{k} - c$$

where c is the local sonic speed. [1]

The derivation for the above eigenvalues can be found in the Section of 11.2.1 of [5]. Since, the transport properties at a surface are determined by the normal components of the fluxes, the number and type of conditions at a boundary of a multi-dimensional domain will be determined by the propagation of waves with the following speeds:

$$\lambda_1 = \vec{u} \cdot \hat{e}_n = v_n$$

$$\lambda_2 = \vec{u} \cdot \hat{e}_n = v_n$$

$$\lambda_3 = \vec{u} \cdot \hat{e}_n = v_n$$

$$\lambda_4 = \vec{u} \cdot \hat{e}_n + c = v_n + c$$

$$\lambda_5 = \vec{u} \cdot \hat{e}_n - c = v_n - c$$

where v_n is the *inward* normal velocity component at the considered surface, coming into the computational domain. The first three eigenvalues correspond to the entropy and vorticity waves, while the two remaining eigenvalues, are associated with acoustic waves. This defines a locally quasi-one-dimensional propagation of information and we can therefore look at how the propagation behaves at a boundary, from the the sign of these eigenvalues at the boundary.

The key to the understanding of the issue of the number of boundary conditions that are needed at the boundary is that characteristics convey information in the $(n - t)$ space formed by the local normal direction and time. When information is propagated from outside into the computational domain, it means that this information has to be obtained by a

boundary condition; this occurs when the eigenvalue λ is positive, and a physical boundary condition has to be imposed. On the other hand, when the eigenvalue λ is negative and the propagation occurs from the interior of the domain towards 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.

2.5 Closure

In this chapter we saw very basic properties of system of partial differential equations with emphasis on the hyperbolic type. These properties must be understood before implementing boundary conditions, to avoid ill-posedness of system.

We saw how information flows along with characteristic in hyperbolic systems and we used this information to determine the number of variables to be assigned at the boundary, based on direction of characteristic waves, i.e. whether the characteristic is flowing into the domain or out of the domain. Also, we saw the advantage of the conservation form of governing equation over the non-conservation form for hyperbolic system.

Chapter 3

Navier Stokes Equations, Boundary Conditions and CFL Criterion

3.1 A Short Description of Navier-Stokes Equations

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 Eq. 3.8, and the total energy as a function of the temperature Eq. 3.9. 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.

System of Navier-Stokes Equations in Conservative Form:

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

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

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

$$\frac{\partial}{\partial t}(\rho w) + \frac{\partial}{\partial x}(\rho uw - \tau_{yz}) + \frac{\partial}{\partial y}(\rho vw - \tau_{yy}) + \frac{\partial}{\partial z}(\rho w^2 + p - \tau_{zz}) = 0 \quad (3.4)$$

$$\frac{\partial}{\partial t}(E_t) + \frac{\partial}{\partial x}((E_t + p)u + q_x - u\tau_{xx} - v\tau_{xy} - w\tau_{xz}) \quad (3.5)$$

$$+ \frac{\partial}{\partial y}((E_t + p)v + q_y - u\tau_{yx} - v\tau_{yy} - w\tau_{yz}) \quad (3.6)$$

$$+ \frac{\partial}{\partial z}((E_t + p)w + q_z - u\tau_{zx} - v\tau_{zy} - w\tau_{zz}) = 0 \quad (3.7)$$

where,

$$P = \rho RT \quad (3.8)$$

$$E_t = \rho e = \rho \left(C_v T + \frac{u^2 + v^2 + w^2}{2} \right) \quad (3.9)$$

The mathematical nature of the steady and unsteady Navier-Stokes equations is 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 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.

Thus, by *just the addition of the temporal dimension 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 steady state. This is called the false-transient approach and is used in this work.

3.2 Boundary Condition Treatment In Terms of Primitive Variables

In the preceding chapter we have seen how boundary condition specification is different for hyperbolic problems compared to that of parabolic and elliptic problems, and have seen how the flow of characteristics into or out of the computational domain affects the specification of the boundary conditions.

Extending the thoughts developed in Sec. 2.4 and referring the literature [1, 2] we can present the following table and implementation of boundary condition for Navier-Stokes equations. This way we answer all the three questions required for the specification of boundary conditions. Namely,

1. How many boundary conditions should be specified?
2. What boundary conditions should be specified?
3. What boundary conditions will have a numerical boundary condition?¹

The answer to the first question depends upon on the number of characteristics that enter into the domain at a boundary. The following table summarises the number of physical/numerical B.C. specification in 3D Navier-Stokes equations.

The second question will be answered in the following subsection.

3.2.1 Implementation of Boundary Conditions

For implementing the boundary condition for the structured grid arrangement, we use the fictitious cell with zero-volume approach. The value of the fictitious cell is updated using the value calculated at the boundary directly. But this, method has to be reviewed, for inhomogeneous Neumann conditions at the boundary. Under such condition, for non-orthogonal grid, taking fictitious cell-center at the face center will lead to complexity. One has to take into account the cross-diffusion terms also.

¹Which not an actual boundary conditions, but is prescribed for variables, to obtain the numerical solution.

Type	Sub-sonic		
	No of +ve Eigen values	No of physical BC	No of Numerical B.C
Inflow	Five	Five	Zero
Outflow	One	One	Four
	Super-sonic		
	No of +ve Eigen values	No of physical BC	No of Numerical B.C
Inflow	Five	Five	Zero
Outflow	Zero	Zero	Five
	Wall		
	No of +ve Eigen values	No of physical BC	No of Numerical B.C
	Four	Four	One

Table 3.1: No of boundary condition to be fixed on boundary in the Navier-stokes system of equations.

The characteristic variables has to be defined in terms of the primitive variables and using them we have to specify the boundary conditions. The detail is very interesting and can be found in [4]. The boundary condition used in this thesis is based on these concepts.

The two basic flow situations at the boundary is sketched in the Fig. 3.1.

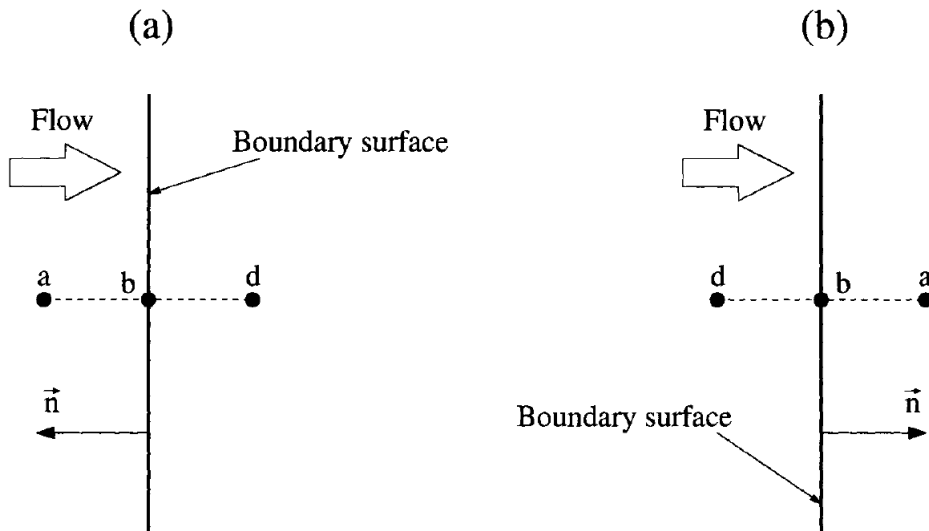


Figure 3.1: Flow Situation at boundary: inflow (a) and outflow (b) situation. Position a is outside, b on the boundary, and the position d is inside the physical domain. The unit normal vector $\vec{n} = [n_x, n_y, n_z]^T$ points out of the domain.[3]

3.2.2 Inflow BC

Subsonic and Supersonic Inflow

When the flow is either supersonic or subsonic, all boundary conditions are physical. The conservative variables on the boundary (point b in Fig. 3.1) are determined by free-stream values only.

3.2.3 Outflow BC

Subsonic Outflow

It requires only one physical boundary conditions, the others have to be numerical boundary conditions. The most appropriate physical condition, particularly for internal flows and corresponding to most experimental situations, consists in fixing the downstream static pressure. This can also be applied for external flow problems. The following numerical formulation is used:

$$\begin{aligned}
 p_b &= p_a \\
 \rho_b &= \rho_d + \frac{(p_b - p_a)}{c_o^2} \\
 u_b &= u_d + n_x \frac{(p_d - p_b)}{\rho_o c_o} \\
 v_b &= v_d + n_y \frac{(p_d - p_b)}{\rho_o c_o} \\
 w_b &= w_d + n_z \frac{(p_d - p_b)}{\rho_o c_o}
 \end{aligned} \tag{3.10}$$

with p_a being the prescribed static pressure.

A point to be considered is that when imposing a constant pressure at a subsonic exit section, one actually allows perturbation waves to be reflected at the boundaries. The non-reflecting boundary condition [12, 13] expresses the physical boundary condition as the requirement that the local perturbations propagated along incoming characteristics be made to vanish. We use the work of Rudy [14] to implement the non-reflecting boundary condition. It has the following form.

$$\frac{\partial u}{\partial t} - \frac{1}{\rho_b a_b} \frac{p_b^{n+1} - p_b^n}{\Delta t} - \frac{\alpha}{\rho_a} (p_b^{n+1} - p_b^*) = 0 \tag{3.11}$$

where,

$$\alpha = \begin{cases} 0.25 & \text{for } M \geq 0.7 \\ 0.6 & \text{for } 0.5 \leq M \leq 0.7 \\ 1 & \text{otherwise} \end{cases}$$

,

p_b^* is the constant pressure imposed at the subsonic exit section and a_b is the sonic speed.

Supersonic Outflow

When the flow is supersonic at outflow, all the conservative variables at the boundary must be determined from the solution inside the boundary.

3.2.4 Wall (or Solid) Boundary

For a viscous fluid which passes a solid wall, the relative velocity between surface and fluid is zero. In the case of stationary wall all the velocity components are zero. i.e., $u=0;v=0;w=0$;

3.2.5 Symmetry Boundary Conditions

We apply the Neumann boundary condition on characteristic variables to update value of the boundary slabs corresponding to symmetry boundaries.

Numerical boundary conditions: We end this section with the discussion on how to implement the numerical boundary conditions. This is particularly important for solid walls, where we want to determine the pressure variations. The simplest way is to take the value at the cell centre of the associated cell. This is a zero-order extrapolations. We can also apply volume extrapolation using the inside two cells. This is first-order and thus supposedly more accurate. *Through numerical experiments we have found that the linear extrapolation esp. at wall and symmetry plane can lead to divergence in the solution.*

3.3 CFL Condition

For viscous flows, the spectral radii of the viscous flux Jacobians have to be included in the computation of Δt . They can severely limit the maximum time step in boundary layers. The time step[3] ,

$$\Delta t_i = CN \frac{\Omega}{(\hat{\Lambda}_c^I + \hat{\Lambda}_c^J + \hat{\Lambda}_c^K) + C1(\hat{\Lambda}_v^I + \hat{\Lambda}_v^J + \hat{\Lambda}_v^K)}$$

Δt_i is calculated at the cell center.

The spectral radii of the convective flux Jacobians for the three grid directions,

$$\begin{aligned}\hat{\Lambda}_c^I &= (|\vec{v} \cdot \vec{n}^I| + c) \Delta S^I \\ \hat{\Lambda}_c^J &= (|\vec{v} \cdot \vec{n}^J| + c) \Delta S^J \\ \hat{\Lambda}_c^K &= (|\vec{v} \cdot \vec{n}^K| + c) \Delta S^K\end{aligned}$$

If we assume that an eddy-viscosity turbulence model is employed, the viscous spectral radii are given by,

$$\hat{\Lambda}_v^I = \max\left(\frac{4}{3\rho}, \frac{\nu}{\rho}\right) \left(\frac{\mu_L}{Pr_L} + \frac{\mu_T}{Pr_T}\right) \frac{(\Delta S^I)^2}{\Omega}$$

and similarly for the other directions.

The constant which multiplies the viscous spectral radii is usually set as $C1 = 4$ for central spatial discretization, $C1 = 2$ for first-order upwind and $C1 = 1$ for second-order upwind discretization.

The symbol μ_L denotes the laminar and μ_T the turbulent dynamic viscosity coefficient, respectively. Furthermore, Pr_L and Pr_T are the laminar and the turbulent Prandtl numbers.

If the flow is laminar we can neglect turbulent properties,

$$\hat{\Lambda}_v^I = \max\left(\frac{4}{3\rho}, \frac{\nu}{\rho}\right) \left(\frac{\mu_L}{Pr_L}\right) \frac{(\Delta S^I)^2}{\Omega}$$

$$\Delta t = \min(\Delta t_i)$$

3.4 Closure

In this chapter we have set up the basic governing equations we use for the study of viscous compressible flows. The boundary conditions described in the previous chapter have been implemented. Finally, we have discussed the criteria for the calculation of the time-step, based on the CFL criterion.

Chapter 4

Numerical Methodology For Navier-Stokes Equations

4.1 Numerical Schemes.

Diffusive processes affecting flow due to viscosity and thermal conductivity are included by the N-S system of equations. As the Reynolds number increases, boundary layers are formed and the laminar flow undergoes a transition toward turbulence. In high Reynolds number and high Mach number flows, shock waves and turbulent boundary layer interactions are most likely to occur.

In the present work, The MacCormack scheme has been chosen to solve Navier-Stokes equations, since it is a very robust and tested scheme. We have also implemented the upwind method for the convective term in the N-S equations, by incorporating the AUSM+ scheme in the solver. These convective schemes will be discussed in this chapter later on.

Important techniques used in the solvers are listed below. These will be explained in detail in this section.

1. Discretization of convective terms
 - (a) MacCormack scheme
 - (b) AUSM+ -up
2. Discretization of Diffusive terms
 - (a) Diffusive formulation
3. Gradient discretization at the cell center
 - (a) Least square's method

4.2 Governing Equations

The system of Navier-Stokes and Energy equations which describes the viscid compressible fluid motion can be presented in conservation form as,

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

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

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

$$\frac{\partial}{\partial t}(\rho w) + \frac{\partial}{\partial x}(\rho uw - \tau_{zx}) + \frac{\partial}{\partial y}(\rho vw - \tau_{zy}) + \frac{\partial}{\partial z}(\rho w^2 + p - \tau_{zz}) = 0 \quad (4.4)$$

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

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

$$\tau_{xx} = 2\mu \frac{\partial u}{\partial x} - 2\frac{\mu}{3}(\nabla \cdot V), \tau_{yy} = 2\mu \frac{\partial v}{\partial y} - 2\frac{\mu}{3}(\nabla \cdot V), \tau_{zz} = 2\mu \frac{\partial w}{\partial z} - 2\frac{\mu}{3}(\nabla \cdot V)$$

$$\nabla \cdot V = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}$$

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

$$\tau_{yz} = \mu \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) = \tau_{zy}$$

$$\tau_{zx} = \mu \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) = \tau_{xz}$$

$$q_x = -k \frac{\partial T}{\partial x}, q_y = -k \frac{\partial T}{\partial y}, q_z = -k \frac{\partial T}{\partial z}$$

4.2.1 Viscous Terms for X-Momentum:

$$\begin{aligned}
& \frac{\partial}{\partial x}(\tau_{xx}) + \frac{\partial}{\partial y}(\tau_{xy}) + \frac{\partial}{\partial z}(\tau_{xz}) \\
&= \frac{\partial}{\partial x} \left[2\mu \frac{\partial u}{\partial x} + \lambda(\nabla \cdot V) \right] + \frac{\partial}{\partial y} \left[\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] + \frac{\partial}{\partial z} \left[\mu \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \right] \\
&= \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) + \frac{\partial}{\partial z} \left(\mu \frac{\partial u}{\partial z} \right) + \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial x} (\lambda \nabla \cdot V) + \frac{\partial}{\partial y} \left(\mu \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial z} \left(\mu \frac{\partial w}{\partial x} \right) \\
&= \nabla \cdot \mu \nabla u + \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial x} (\lambda \nabla \cdot V) + \frac{\partial \mu}{\partial y} \frac{\partial v}{\partial x} + \mu \frac{\partial^2 v}{\partial y \partial x} + \frac{\partial \mu}{\partial z} \frac{\partial w}{\partial x} + \mu \frac{\partial^2 w}{\partial z \partial x} \\
&= \nabla \cdot \mu \nabla u + \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial x} (\lambda \nabla \cdot V) + \frac{\partial \mu}{\partial y} \frac{\partial v}{\partial x} + \frac{\partial \mu}{\partial z} \frac{\partial w}{\partial x} + \frac{\partial}{\partial x} \left(\mu \frac{\partial v}{\partial y} \right) - \frac{\partial \mu}{\partial x} \frac{\partial v}{\partial y} + \frac{\partial}{\partial x} \left(\mu \frac{\partial w}{\partial z} \right) \\
&\quad - \frac{\partial \mu}{\partial x} \frac{\partial w}{\partial z} \\
&= \nabla \cdot \mu \nabla u + \frac{\partial}{\partial x} \left[\mu \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \right] + \frac{\partial}{\partial x} (\lambda \nabla \cdot V) + \frac{\partial \mu}{\partial y} \frac{\partial v}{\partial x} + \frac{\partial \mu}{\partial z} \frac{\partial w}{\partial x} - \frac{\partial \mu}{\partial x} \frac{\partial v}{\partial y} - \frac{\partial \mu}{\partial x} \frac{\partial w}{\partial z} \\
&= \nabla \cdot \mu \nabla u + \frac{\partial}{\partial x} [(\mu + \lambda) \nabla \cdot V] + \left[\frac{\partial \mu}{\partial y} \frac{\partial v}{\partial x} + \frac{\partial \mu}{\partial z} \frac{\partial w}{\partial x} - \frac{\partial \mu}{\partial x} \frac{\partial v}{\partial y} - \frac{\partial \mu}{\partial x} \frac{\partial w}{\partial z} \right] \\
&= \nabla \cdot \mu \nabla u + \frac{\partial}{\partial x} \left(\frac{\mu}{3} \nabla \cdot V \right) + \left[\frac{\partial \mu}{\partial y} \frac{\partial v}{\partial x} + \frac{\partial \mu}{\partial z} \frac{\partial w}{\partial x} - \frac{\partial \mu}{\partial x} \frac{\partial v}{\partial y} - \frac{\partial \mu}{\partial x} \frac{\partial w}{\partial z} \right]
\end{aligned}$$

Performing a similar expansion for Y and Z momentum equations, the Navier-Stokes equation can be written as,

U-Momentum

$$\begin{aligned}
\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u^2) + \frac{\partial}{\partial y}(\rho uv) + \frac{\partial}{\partial z}(\rho uw) &= -\frac{\partial p}{\partial x} + (\nabla \cdot \mu \nabla u) + \frac{\partial}{\partial x} \left(\frac{\mu}{3} \nabla \cdot V \right) \\
&\quad + \frac{\partial \mu}{\partial y} \frac{\partial v}{\partial x} + \frac{\partial \mu}{\partial z} \frac{\partial w}{\partial x} - \frac{\partial \mu}{\partial x} \frac{\partial v}{\partial y} - \frac{\partial \mu}{\partial x} \frac{\partial w}{\partial z}
\end{aligned}$$

V-Momentum

$$\begin{aligned}
\frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho uv) + \frac{\partial}{\partial y}(\rho v^2) + \frac{\partial}{\partial z}(\rho vw) &= -\frac{\partial p}{\partial y} + (\nabla \cdot \mu \nabla v) + \frac{\partial}{\partial y} \left(\frac{\mu}{3} \nabla \cdot V \right) \\
&\quad + \frac{\partial \mu}{\partial x} \frac{\partial u}{\partial y} + \frac{\partial \mu}{\partial z} \frac{\partial w}{\partial y} - \frac{\partial \mu}{\partial y} \frac{\partial u}{\partial x} - \frac{\partial \mu}{\partial y} \frac{\partial w}{\partial z}
\end{aligned}$$

W-Momentum

$$\begin{aligned}
\frac{\partial}{\partial t}(\rho w) + \frac{\partial}{\partial x}(\rho uw) + \frac{\partial}{\partial y}(\rho vw) + \frac{\partial}{\partial z}(\rho w^2) &= -\frac{\partial p}{\partial z} + (\nabla \cdot \mu \nabla w) + \frac{\partial}{\partial z} \left(\frac{\mu}{3} \nabla \cdot V \right) \\
&\quad + \frac{\partial \mu}{\partial x} \frac{\partial u}{\partial z} + \frac{\partial \mu}{\partial y} \frac{\partial v}{\partial z} - \frac{\partial \mu}{\partial z} \frac{\partial u}{\partial x} - \frac{\partial \mu}{\partial z} \frac{\partial v}{\partial y}
\end{aligned}$$

4.2.2 Viscous Dissipation Term in Energy equation

$$\begin{aligned}
& \frac{\partial}{\partial x}(u\tau_{xx} + v\tau_{xy} + w\tau_{xz}) + \frac{\partial}{\partial y}(u\tau_{yx} + v\tau_{yy} + w\tau_{yz}) + \frac{\partial}{\partial z}(u\tau_{zx} + v\tau_{zy} + w\tau_{zz}) \\
&= \frac{\partial u}{\partial x}\tau_{xx} + \frac{\partial v}{\partial x}\tau_{xy} + \frac{\partial w}{\partial x}\tau_{xz} + u\frac{\partial}{\partial x}(\tau_{xx}) + v\frac{\partial}{\partial x}(\tau_{xy}) + w\frac{\partial}{\partial x}(\tau_{xz}) \\
&+ \frac{\partial u}{\partial y}\tau_{yx} + \frac{\partial v}{\partial y}\tau_{yy} + \frac{\partial w}{\partial y}\tau_{yz} + u\frac{\partial}{\partial y}(\tau_{yx}) + v\frac{\partial}{\partial y}(\tau_{yy}) + w\frac{\partial}{\partial y}(\tau_{yz}) \\
&+ \frac{\partial u}{\partial z}\tau_{zx} + \frac{\partial v}{\partial z}\tau_{zy} + \frac{\partial w}{\partial z}\tau_{zz} + u\frac{\partial}{\partial z}(\tau_{zx}) + v\frac{\partial}{\partial z}(\tau_{zy}) + w\frac{\partial}{\partial z}(\tau_{zz}) \\
&= \frac{\partial u}{\partial x}\tau_{xx} + \frac{\partial v}{\partial x}\tau_{xy} + \frac{\partial w}{\partial x}\tau_{xz} + u\left(\frac{\partial}{\partial x}(\tau_{xx}) + \frac{\partial}{\partial y}(\tau_{yx}) + \frac{\partial}{\partial z}(\tau_{zx})\right) \\
&v\left(\frac{\partial}{\partial x}(\tau_{xy}) + \frac{\partial}{\partial y}(\tau_{yy}) + \frac{\partial}{\partial z}(\tau_{zy})\right) + w\left(\frac{\partial}{\partial x}(\tau_{xz}) + \frac{\partial}{\partial y}(\tau_{yz}) + \frac{\partial}{\partial z}(\tau_{zz})\right) \\
&+ \frac{\partial u}{\partial y}\tau_{yx} + \frac{\partial v}{\partial y}\tau_{yy} + \frac{\partial w}{\partial y}\tau_{yz} + \frac{\partial u}{\partial z}\tau_{zx} + \frac{\partial v}{\partial z}\tau_{zy} + \frac{\partial w}{\partial z}\tau_{zz}
\end{aligned}$$

=u(viscous part of u-momentum)+v(viscous part of v-momentum)+w(viscous part of w-momentum)+derivative terms.

Final governing equations:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0 \quad (4.6)$$

$$\begin{aligned}
\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + P)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} + \frac{\partial(\rho uw)}{\partial z} &= (\nabla \cdot \mu \nabla u) + \frac{\partial}{\partial x} \left(\frac{\mu}{3} \nabla \cdot V \right) \\
&+ \frac{\partial \mu}{\partial y} \frac{\partial v}{\partial x} + \frac{\partial \mu}{\partial z} \frac{\partial w}{\partial x} - \frac{\partial \mu}{\partial x} \frac{\partial v}{\partial y} - \frac{\partial \mu}{\partial x} \frac{\partial w}{\partial z}
\end{aligned} \quad (4.7)$$

$$\begin{aligned}
\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho vu)}{\partial x} + \frac{\partial(\rho v^2 + P)}{\partial y} + \frac{\partial(\rho vw)}{\partial z} &= (\nabla \cdot \mu \nabla v) + \frac{\partial}{\partial y} \left(\frac{\mu}{3} \nabla \cdot V \right) \\
&+ \frac{\partial \mu}{\partial x} \frac{\partial u}{\partial y} + \frac{\partial \mu}{\partial z} \frac{\partial w}{\partial y} - \frac{\partial \mu}{\partial y} \frac{\partial u}{\partial x} - \frac{\partial \mu}{\partial y} \frac{\partial w}{\partial z}
\end{aligned} \quad (4.8)$$

$$\begin{aligned}
\frac{\partial(\rho w)}{\partial t} + \frac{\partial(\rho wu)}{\partial x} + \frac{\partial(\rho wv)}{\partial y} + \frac{\partial(\rho w^2 + P)}{\partial z} &= (\nabla \cdot \mu \nabla w) + \frac{\partial}{\partial z} \left(\frac{\mu}{3} \nabla \cdot V \right) \\
&+ \frac{\partial \mu}{\partial x} \frac{\partial u}{\partial z} + \frac{\partial \mu}{\partial y} \frac{\partial v}{\partial z} - \frac{\partial \mu}{\partial z} \frac{\partial u}{\partial x} - \frac{\partial \mu}{\partial z} \frac{\partial v}{\partial y}
\end{aligned} \quad (4.9)$$

$$\begin{aligned} \frac{\partial(\rho E)}{\partial t} + \frac{\partial(\rho u H)}{\partial x} + \frac{\partial(\rho v H)}{\partial y} + \frac{\partial(\rho w H)}{\partial z} &= (\nabla \cdot K \nabla T) + u F_{du} + v F_{dv} + w F_{dw} \\ + \frac{\partial u}{\partial x} \tau_{xx} + \frac{\partial v}{\partial x} \tau_{xy} + \frac{\partial w}{\partial x} \tau_{xz} + \frac{\partial u}{\partial y} \tau_{yx} + \frac{\partial v}{\partial y} \tau_{yy} + \frac{\partial w}{\partial y} \tau_{yz} + \frac{\partial u}{\partial z} \tau_{zx} + \frac{\partial v}{\partial z} \tau_{zy} + \frac{\partial w}{\partial z} \tau_{zz} \end{aligned} \quad (4.10)$$

F_{du}, F_{dv}, F_{dw} are viscous terms in x, y and z momentum equations.

4.3 Discretization of Governing Equation

The equations Eq. 4.6 to Eq. 4.10 can be written in compact form as

$$\frac{\partial \{w\}}{\partial t} + \frac{\partial \{F_x\}}{\partial x} + \frac{\partial \{F_y\}}{\partial y} + \frac{\partial \{F_z\}}{\partial z} - F_d = 0 \quad (4.11)$$

F_x, F_y, F_z are the convective fluxes and F_d is the diffusive flux.

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

$$F_x \equiv \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ \rho uw \\ \rho u H \end{pmatrix}, F_y \equiv \begin{pmatrix} \rho v \\ \rho v^2 + P \\ \rho vw \\ \rho v w \\ \rho v H \end{pmatrix}, F_z \equiv \begin{pmatrix} \rho w \\ \rho w^2 + P \\ \rho vw \\ \rho w^2 + P \\ \rho w H \end{pmatrix}$$

$$F_d \equiv \begin{pmatrix} 0 \\ \text{R.H.S of the u-momentum (4.7)} \\ \text{R.H.S of the v-momentum(4.8)} \\ \text{R.H.S of the w-momentum(4.9)} \\ \text{R.H.S of the Energy(4.10)} \end{pmatrix}$$

Note that the F_x, F_y, F_z column vectors are used just for notational convenience. Now for an arbitrary component of w, w_i we can write,

$$\frac{\partial w_i}{\partial t} + \nabla \cdot \mathbf{F}_i - \mathbf{F}_{di} = 0 \quad (4.12)$$

where each row i respectively represents the governing continuity, momentum, energy equations.

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.

$$\oint_V \left(\frac{\partial w_i}{\partial t} + \nabla \cdot \mathbf{F}_i - \mathbf{F}_{di} \right) dV = 0 \quad (4.13)$$

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

$$\oint_V \frac{\partial w_i}{\partial t} dV + \oint_S \mathbf{F}_i \cdot d\vec{S} - \oint_V \mathbf{F}_{di} dV = 0$$

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

$$\frac{\partial}{\partial t} \oint_V w_i dV + \oint_S \mathbf{F}_i \cdot d\vec{S} - \oint_V \mathbf{F}_{di} dV = 0$$

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

$$\underbrace{\frac{\partial}{\partial t} \oint_V w_i dV}_{TemporalPart} + \underbrace{\oint_S \mathbf{F}_i \cdot d\vec{S}}_{ConvectivePart} - \underbrace{\oint_V \mathbf{F}_{di} dV}_{DiffusionPart} = 0$$

$$\text{Let, } \oint_V \mathbf{F}_{di} dV = F_D$$

The control volume V can be arbitrarily chosen so we use the above equation for each of the finite-volume cells of a chosen grid spanning the entire computational domain. The shape of the finite volume cells are the user's choice. The numerical results obtained from hexahedral elements are more accurate than that obtained from any other element like triangular prism, pyramid or polyhedral. And it is easy to create a structured grid from hexahedral elements rather than use of any other prismatic shaped element. Thus, we use non-orthogonal hexahedral elements, as shown in the figure, having east, west, north, south, top and bottom faces.

Temporal term: It is assumed the volume averaged value of conservative variable can be written for the p th cell as:

$$\frac{1}{V_p} \oint_{V_p} w dV = w_p$$

thus,

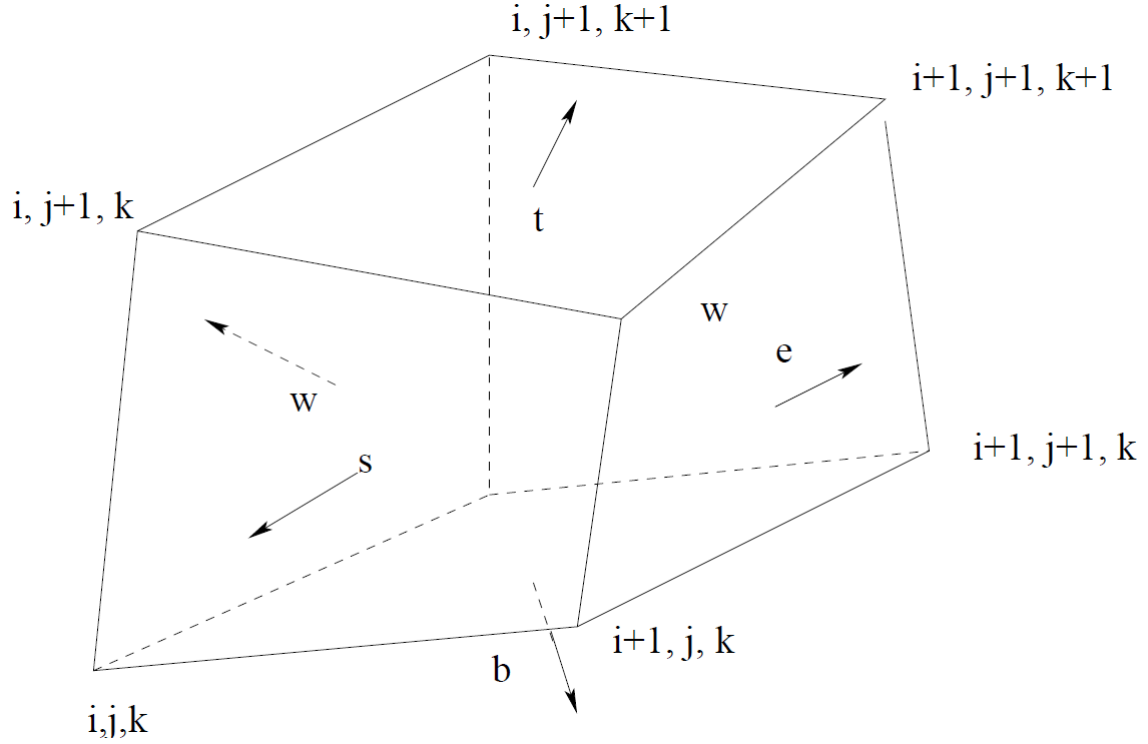


Figure 4.1: Finite volume cell

$$\oint_{V_p} w dV = V_p w_p$$

where, V_p is the volume of the p th cell, and w_p is the value of w at the cell-center(centroid).

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

$$\frac{\partial w_i}{\partial t} = V_p \frac{w_p^{n+1} - w_p^n}{\Delta t}$$

Convective term: In the convective term, 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:

$$\oint_S \vec{F}_i \cdot d\vec{S} = \oint_e \vec{F}_i \cdot d\vec{S}_e + \oint_w \vec{F}_i \cdot d\vec{S}_w + \oint_n \vec{F}_i \cdot d\vec{S}_n + \oint_s \vec{F}_i \cdot d\vec{S}_s + \oint_t \vec{F}_i \cdot d\vec{S}_t + \oint_b \vec{F}_i \cdot d\vec{S}_b$$

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

$$\oint_{S_f} \vec{F}_i \cdot d\vec{S}_f = \oint_S F_{ix} dS_x + \oint_S F_{iy} dS_y + \oint_S F_{iz} dS_z$$

The value of the 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} .

Therefore we can write,

$$\oint_{S_f} \vec{F}_i \cdot d\vec{S}_f = F_{ix} S_{fx} + F_{iy} S_{fy} + F_{iz} S_{fz}$$

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

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

Now, putting the discretized temporal, convective and diffusive terms together, the Navier-Stokes equations can be written in discretized form as:

$$V_p \frac{w_{ip}^{n+1} - w_{ip}^n}{\Delta t} = - \sum_f (F_{ifx} S_{fx} + F_{ify} S_{fy} + F_{ifz} S_{fz})^n + F_{iD}^n \quad (4.14)$$

So, while discretizing the equation the following approximations are considered:

1. The volume-average value of the variable is approximated by its cell-center value.
2. The surface-average value of the variable on a cell-face is approximated by its face-center value.
3. Second-order time variation is neglected.
4. The faces of the cell are taken to be flat.

The approach used here for space discretization (dividing the domain into small finite-volume cells) is the cell-centered approach because of its various advantages for a general-purpose finite-volume solver. Therefore, the data-structure used for the discretized equation stores the cell-values in its solution array. But the discretized equation also requires the surface averaged value at the six faces of each cell. To get the variable face values from cell center values we need some interpolation function. Since, we are dealing with a convective equation whose nature is quite different from a diffusion equation, the interpolation function

should give greater weight to upwind than to downwind values, in estimating the variable value at any given point.

4.4 MacCormack Scheme in FVM

The MacCormack scheme [5] in the finite volume methodology is described below. Rewriting

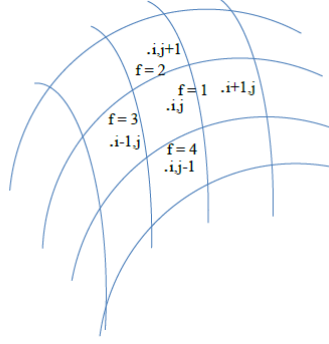


Figure 4.2: Finite volume grid

discretized equation (4.14)

$$V_p \frac{w_{ip}^{n+1} - w_{ip}^n}{\Delta t} = - \sum_f (F_{ifx} S_{fx} + F_{ify} S_{fy} + F_{ifz} S_{fz})^n + F_{iD}^n \quad (4.15)$$

and assuming step 1 as forward differenced and step 2 as backward differenced, the finite volume discretization can be written as,

Step 1: Predictor Step

$$\begin{aligned} w_{iP}^* = w_{ip}^n - \frac{\Delta t}{V_p} & (F_{iEx}^n S_{ex} + F_{iEy}^n S_{ey} + F_{iEz}^n S_{ez} + F_{iPx}^n S_{wx} + F_{iPy}^n S_{wy} + F_{iPz}^n S_{wz} \\ & + F_{iNx}^n S_{nx} + F_{iNy}^n S_{ny} + F_{iNz}^n S_{nz} + F_{iPx}^n S_{sx} + F_{iPy}^n S_{sy} + F_{iPz}^n S_{sz} \\ & + F_{iTx}^n S_{tx} + F_{iTy}^n S_{ty} + F_{iTz}^n S_{tz} + F_{iPx}^n S_{bx} + F_{iPy}^n S_{by} + F_{iPz}^n S_{bz}) \\ & + \frac{\Delta t}{V_p} (F_{iD}^n) \end{aligned} \quad (4.16)$$

Step 2: Corrector Step

$$\begin{aligned}
w_{ip}^{**} = w_{ip}^* - \frac{\Delta t}{V_p} & (F_{iPx}^* S_{ex} + F_{iPy}^* S_{ey} + F_{iPz}^* S_{ez} + F_{iWx}^* S_{wx} + F_{iWy}^* S_{wy} + F_{iWz}^* S_{wz} \\
& + F_{iPx}^* S_{nx} + F_{iPy}^* S_{ny} + F_{iPz}^* S_{nz} + F_{iSx}^* S_{sx} + F_{iSy}^* S_{sy} + F_{iSz}^* S_{sz} \\
& + F_{iPx}^* S_{tx} + F_{iPy}^* S_{ty} + F_{iPz}^* S_{tz} + F_{iSx}^* S_{bx} + F_{iSy}^* S_{by} + F_{iSz}^* S_{bz}) \\
& + \frac{\Delta t}{V_p} (F_{iD}^*)
\end{aligned} \tag{4.17}$$

and the new time-level value can be calculated as:

$$w_{iP}^{n+1} = \frac{w_{iP}^n + w_{iP}^{**}}{2} \tag{4.18}$$

We see above that we have applied forward difference step in the predictor step and backward difference in the corrector step for convective terms. We can apply the same in reverse also. The left running waves are better captured by the former one, whereas the right running waves are better captured by the second version. To avoid favoring either left- or right-running waves, the two versions are often combined, reversing the order of FTBS and FTFS after every time-steps ([15], Pg.361). But, the diffusion term is same in both predictor and corrector steps. Since the MacCormack scheme is second order accurate in space and time, oscillations are observed in solution having abrupt step-changes in value. The amplitude of oscillation depends upon the magnitude of the change; often this causes the variable to crosses the CFL condition and the solution gets blown to infinity. To avoid such oscillations and to get physically acceptable solutions, we need to add a small amount of diffusion in the governing equation which is discussed in the next section.

4.4.1 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 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 for the one-dimensional wave equation given by

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0 \tag{4.19}$$

is shown below

$$\begin{aligned} \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = & \frac{a\Delta x}{2}(1 - \nu) \frac{\partial^2 u}{\partial x^2} + \frac{a(\Delta x)_2}{6}(3\nu - 2\nu_2 - 1) \frac{\partial^3 u}{\partial x^3} \\ & + O[(\Delta t)^3, (\Delta t)^2(\Delta x), (\Delta t)(\Delta x)^2, (\Delta x)^3] \end{aligned} \quad (4.20)$$

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 A. Next we present in the following subsection Sec. 4.4.2 the MacCormack algorithm which has been implemented in this solver.

4.4.2 MacCormack Algorithm

1. Initialization of variables.
2. Predictor step for all five equations (Find conservative variables) using previous solution variables. Eq. 4.16.
3. Calculate the primitive variables (u,v,w,p,T) for predictor step, and applying the boundary conditions for primitive variables.
4. Corrector step for all five equations (Find conservative variables) Eq. 4.18 using predictor solution variables.
5. Calculate the primitive variables (u,v,w,p,T) for corrector step, and applying the boundary conditions for primitive variables.
6. Accepting corrector step variables as current step flow solution.
7. Checking the convergence criteria.

4.5 Advection Upstream Splitting Method (AUSM)

AUSM is developed as a numerical inviscid flux function for solving a general system of conservation equations and especially it is used to simulate hyperbolic conservation equations. It is based on the upwind concept and was motivated to provide an alternative approach to other upwind methods, such as the Godunov method, flux difference splitting methods by Roe, and Solomon and Osher, flux vector splitting methods by Van Leer, and Steger and Warming. It was first presented by Liou and Steffen [9].

We present here the AUSM⁺-up of Liou (2006) [8]. The token used “-up” will be clear after the scheme formulation is made. The scheme relies on splitting the convective flux vector \mathbf{F} into a convective component $\mathbf{F}^{(c)}$ and a pressure component $\mathbf{F}^{(p)}$. For the x-split three dimensional flux we have

$$\mathbf{F}(\mathbf{w}) = \begin{Bmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ \rho uw \\ \rho u(E + P) \end{Bmatrix} = \begin{Bmatrix} \rho u \\ \rho u^2 \\ \rho uv \\ \rho uw \\ \rho uH \end{Bmatrix} + \begin{Bmatrix} 0 \\ P \\ 0 \\ 0 \\ 0 \end{Bmatrix} \equiv \mathbf{F}^{(c)} + \mathbf{F}^{(P)} \quad (4.21)$$

where, w represents conservative variables.

By introducing the Mach number and enthalpy

$$M = \frac{u}{a}, \quad H = \frac{E+P}{\rho}$$

we write,

$$\mathbf{F}^{(c)} = M \begin{Bmatrix} \rho a \\ \rho au \\ \rho av \\ \rho aw \\ \rho aH \end{Bmatrix} \equiv M \hat{\mathbf{F}}^{(c)} \quad (4.22)$$

The face straddles two neighboring cells labeled by subscripts “**L**” and “**R**”, respectively, namely left and right of the interface. The values of various variable at the face is denoted by subscript “1/2”. In defining the flux at face (say $\mathbf{F}_{1/2}$), the scheme take

$$\mathbf{F}_{1/2} = \mathbf{F}_{1/2}^{(c)} + \mathbf{F}_{1/2}^{(p)} \quad (4.23)$$

where the *convective flux component* is given by

$$\mathbf{F}_{1/2}^{(c)} = M_{1/2} [\hat{\mathbf{F}}^{(c)}]_{1/2} \quad (4.24)$$

with definition

$$[\bullet]_{1/2} = \begin{cases} [\bullet]_L & \text{if } M_{1/2} \geq 0 \\ [\bullet]_R & \text{if } M_{1/2} < 0 \end{cases}$$

The flux vector in Eq. 4.24 is upwinded as per the sign of face Mach number $M_{1/2}$. The convective terms are upstream-biased using the velocity implied in the face Mach Number. For this reason Liou and Steffen call their scheme AUSM, which stands for Advection Upstream Splitting Method.

The face Mach number is given by the splitting

$$M_{1/2} = M_L^+ + M_R^- \quad (4.25)$$

with the positive and negative component yet to be defined. The splitting of the pressure flux components depends on the splitting of the pressure itself, namely

$$P_{1/2} = P_L^+ + P_R^- \quad (4.26)$$

For the splitting of Mach number and pressure we follow the formulation presented in Liou (2006) [8]. The objective of doing this was to make the earlier version of AUSM to be uniformly valid for all speed regimes.

The final algorithm for flux calculation is given as follows *AUSM⁺ - up*.

1. We define

$$M_L = \frac{V_L}{a_{1/2}}, \quad M_R = \frac{V_R}{a_{1/2}}$$

$$\text{where, } V_i = \vec{V}_i \cdot \hat{n}$$

The corresponding speed of sound at the interface is given by

$$a_{1/2} = \frac{a_L + a_R}{2} \text{ or, } a_{1/2} = \min(\hat{a}_i, \hat{a}_R)$$

$$\text{where, } \hat{a}_i = \frac{a_L^*}{\max(a_L^*, V_L)}, \quad \hat{a}_R = \frac{a_R^*}{\max(a_R^*, -V_R)}$$

where, a^* is defined as the speed of sound based on total enthalpy (H_t). And is given as $a^* = \sqrt{\frac{2(\gamma-1)}{\gamma+1} H_t}$.

2. $\bar{M}^2 = \frac{V_L^2 + V_R^2}{2a_{1/2}^2}$,
 $M_o^2 = \min(1, \max(\bar{M}^2, M_\infty^2)) \in [0, 1]$,

$$f_a(M_o) = M_o(2 - M_o) \in [0, 1],$$

A pressure diffusion term M_p , is introduced to enhance calculations of low Mach number flow. It is defined as

$$M_p = -\frac{K_p}{f_a} \max(1 - \sigma \bar{M}^2, 0) \frac{P_R - P_L}{(\rho a)_{1/2}},$$

$$\text{where, } 0 \leq K_p \leq 1, \sigma \leq 1 \text{ and } \rho_{1/2} = \frac{\rho_L + \rho_R}{2}$$

Thus,

$$M_{1/2} = M_{(4)}^+(M_L) + M_{(4)}^-(M_R) + M_p$$

$$\text{where, } M_{(4)}^\pm(M) = \begin{cases} M_{(1)}^\pm & \text{if } |M| \geq 1 \\ M_{(2)}^\pm (1 \mp 16\beta M_{(2)}^\mp) & \text{otherwise} \end{cases}$$

where,

$$M_{(1)}^\pm(M) = \frac{1}{2}(M \pm |M|)$$

$$M_{(2)}^\pm(M) = \pm \frac{1}{4}(M \pm 1)^2$$

$$3. P_{1/2} = P_{(5)}^+(M_L)P_i + P_{(5)}^-(M_R)P_R + P_u$$

$$\text{where, } P_{(5)}^\pm(M) = \begin{cases} \frac{1}{M}M_{(1)}^\pm & \text{if } |M| \geq 1 \\ M_{(2)}^\pm [(\pm 2 - M) \mp 16\alpha MM_{(2)}^\mp] & \text{otherwise} \end{cases}$$

where, P_u is the velocity diffusion term similar to interface Mach number defined in step 2. $P_u = -K_u P_{(5)}^+(M_i) P_{(5)}^-(M_R) (\rho_L + \rho_R) (f_a a_{1/2}) (u_R - u_L)$

where, $0 \leq K_u \leq 1$

$$4. \alpha = \frac{3}{16}(-4 + 5f_a^2) \in [-\frac{3}{8} + \frac{3}{16}],$$

$$\beta = \frac{1}{8}, K_p = 0.25, K_u 0.75 \text{ and } \sigma = 1.0$$

5. The mass flux at the interface has the following form

$$\dot{m}_{1/2} = a_{1/2} M_{1/2} \begin{cases} \rho_L & \text{if } M_{1/2} \geq 0 \\ \rho_R & \text{otherwise} \end{cases}$$

6. Finally, the flux vector is written as

$$\mathbf{F}_{1/2}(w) = \dot{m}_{1/2} \begin{cases} w_L & \text{if } w_{1/2} \geq 0 \\ w_R & \text{otherwise} \end{cases} + P_{1/2} \quad (4.27)$$

The value of the conservative variables w_L and w_R are the left and right states at the face. They are obtained using MUSCL extrapolation formula [2]. Considering non-uniform structured grid, we have used second order volume extrapolation. At the cells next to the boundary we use simple zero-order extrapolation, i.e. we have 1st order accuracy at the boundary. For the multi-block approach, we have maintained second-order accuracy at the block interface using center-difference approach. We have also tried with single and double upwind approach. But, the center-difference has been given best results in terms of accuracy and stability.

The scheme is called AUSM⁺-up. The suffix ‘‘u’’ is used to indicate the velocity diffusion term (P_u) included in the pressure split of flux vector and the suffix ‘‘p’’ is used to indicate the pressure diffusion term (M_p) included in the convective split part of the flux vector. Next we present in the following subsection Sec. 4.5.1 the AUSM algorithm which has been implemented in this solver.

4.5.1 AUSM Algorithm

1. Initialization of variables.
2. Calculating the convective fluxes by upwind AUSM scheme.
3. Calculating the viscous terms using previous solution variables.

4. Calculating the conservative variables using governing equations for current time step.
5. Updating the primitive variables from conservative variables, and applying the boundary conditions for primitive variables.
6. Checking the steady state convergence criteria.

Compressible Navier-Stokes equation in compact form:

$$\oint_V \left(\frac{\partial w_i}{\partial t} + \nabla \cdot \mathbf{F}_i - \mathbf{F}_{di} \right) dV = 0$$

$$\oint_V \frac{\partial w_i}{\partial t} dV + \oint_V \nabla \cdot \mathbf{F}_i dV - \oint_V \mathbf{F}_{di} dV = 0$$

$$V_P \left(\frac{w_i^{n+1} - w_i^n}{\Delta t} \right) + \oint_V \nabla \cdot \mathbf{F}_i dV - \oint_V \mathbf{F}_{di} dV = 0$$

$$w_i^{n+1} = w_i^n - \frac{\Delta t}{V_P} \left(\oint_V \nabla \cdot \mathbf{F}_i dV \right) + \frac{\Delta t}{V_P} \left(\oint_V \mathbf{F}_{di} dV \right)$$

$$w_i^{n+1} = w_i^n - \frac{\Delta t}{V_P} \underbrace{\left(\oint_V \nabla \cdot \mathbf{F}_i dV \right)^n}_{ConvectivePart} + \frac{\Delta t}{V_P} \underbrace{\left(\oint_V \mathbf{F}_{di} dV \right)^n}_{DiffusionPart}$$

Convective Part: Discretization and procedure to find convective terms by AUSM Scheme has been already discussed in the earlier section, based on work done by Ashwani [7]

Diffusive Part: Discretization and procedure to find diffusive terms will be discussed in detail, in the next section.

4.6 Diffusion term:

Viscous terms for u-momentum

$$F_d = (\nabla \cdot \mu \nabla u) + \frac{\partial}{\partial x} \left(\frac{\mu}{3} \nabla \cdot \mathbf{V} \right) + \frac{\partial \mu}{\partial y} \frac{\partial v}{\partial x} + \frac{\partial \mu}{\partial z} \frac{\partial w}{\partial x} - \frac{\partial \mu}{\partial x} \frac{\partial v}{\partial y} - \frac{\partial \mu}{\partial x} \frac{\partial w}{\partial z} \quad (4.28)$$

$$\oint_V \mathbf{F}_d dV = \oint_V \left[(\nabla \cdot \mu \nabla \mathbf{u}) + \frac{\partial}{\partial \mathbf{x}} \left(\frac{\mu}{3} \nabla \cdot \mathbf{V} \right) + \frac{\partial \mu}{\partial \mathbf{y}} \frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \frac{\partial \mu}{\partial \mathbf{z}} \frac{\partial \mathbf{w}}{\partial \mathbf{x}} - \frac{\partial \mu}{\partial \mathbf{x}} \frac{\partial \mathbf{v}}{\partial \mathbf{y}} - \frac{\partial \mu}{\partial \mathbf{x}} \frac{\partial \mathbf{w}}{\partial \mathbf{z}} \right] dV$$

$$\oint_V \mathbf{F}_d dV = \oint_V (\nabla \cdot \mu \nabla \mathbf{u}) dV + \oint_V \frac{\partial}{\partial \mathbf{x}} \left(\frac{\mu}{3} \nabla \cdot \mathbf{V} \right) dV + \oint_V \left[\frac{\partial \mu}{\partial \mathbf{y}} \frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \frac{\partial \mu}{\partial \mathbf{z}} \frac{\partial \mathbf{w}}{\partial \mathbf{x}} - \frac{\partial \mu}{\partial \mathbf{x}} \frac{\partial \mathbf{v}}{\partial \mathbf{y}} - \frac{\partial \mu}{\partial \mathbf{x}} \frac{\partial \mathbf{w}}{\partial \mathbf{z}} \right] dV$$

$$\text{Let, Div} = \left(\frac{\mu}{3} \nabla \cdot \mathbf{V} \right)$$

$$\begin{aligned} \oint_V \mathbf{F}_d dV &= \oint_V (\nabla \cdot \mu \nabla \mathbf{u}) dV + \oint_V \frac{\partial \text{Div}}{\partial \mathbf{x}} dV + \oint_V \left[\frac{\partial \mu}{\partial y} \frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \frac{\partial \mu}{\partial z} \frac{\partial \mathbf{w}}{\partial \mathbf{x}} - \frac{\partial \mu}{\partial \mathbf{x}} \frac{\partial \mathbf{v}}{\partial y} - \frac{\partial \mu}{\partial \mathbf{x}} \frac{\partial \mathbf{w}}{\partial z} \right] dV \\ \oint_V \mathbf{F}_d dV &= \oint_V (\nabla \cdot \mu \nabla \mathbf{u}) dV + \left[\frac{\partial \text{Div}}{\partial x} + \frac{\partial \mu}{\partial y} \frac{\partial v}{\partial x} + \frac{\partial \mu}{\partial z} \frac{\partial w}{\partial x} - \frac{\partial \mu}{\partial x} \frac{\partial v}{\partial y} - \frac{\partial \mu}{\partial x} \frac{\partial w}{\partial z} \right] V_P \\ \oint_V \mathbf{F}_d dV &= \underbrace{\oint_V (\nabla \cdot \mu \nabla \mathbf{u}) dV}_{\text{DiffusionFlux}} + \underbrace{\left[\frac{\partial \text{Div}}{\partial x} + \frac{\partial \mu}{\partial y} \frac{\partial v}{\partial x} + \frac{\partial \mu}{\partial z} \frac{\partial w}{\partial x} - \frac{\partial \mu}{\partial x} \frac{\partial v}{\partial y} - \frac{\partial \mu}{\partial x} \frac{\partial w}{\partial z} \right] V_P}_{\text{Gradients-at-the-center}} \quad (4.29) \end{aligned}$$

same as above is done for y and z momentum equations. We get two parts a diffusion part and a gradient part as we can see in Eq. 4.29

Viscous Terms For Energy equation:

$$\begin{aligned} F_d &= (\nabla \cdot K \nabla T) + uF_{du} + vF_{dv} + wF_{dw} + \frac{\partial u}{\partial x} \tau_{xx} + \frac{\partial v}{\partial x} \tau_{xy} + \frac{\partial w}{\partial x} \tau_{xz} \\ &\quad + \frac{\partial u}{\partial y} \tau_{yx} + \frac{\partial v}{\partial y} \tau_{yy} + \frac{\partial w}{\partial y} \tau_{yz} + \frac{\partial u}{\partial z} \tau_{zx} + \frac{\partial v}{\partial z} \tau_{zy} + \frac{\partial w}{\partial z} \tau_{zz} \quad (4.30) \end{aligned}$$

Let,

$$G = \frac{\partial u}{\partial x} \tau_{xx} + \frac{\partial v}{\partial x} \tau_{xy} + \frac{\partial w}{\partial x} \tau_{xz} + \frac{\partial u}{\partial y} \tau_{yx} + \frac{\partial v}{\partial y} \tau_{yy} + \frac{\partial w}{\partial y} \tau_{yz} + \frac{\partial u}{\partial z} \tau_{zx} + \frac{\partial v}{\partial z} \tau_{zy} + \frac{\partial w}{\partial z} \tau_{zz} \quad (4.31)$$

where, G = gradient terms at the cell center

$$\begin{aligned} \oint_V \mathbf{F}_d dV &= \oint_V [(\nabla \cdot K \nabla T) + uF_{du} + vF_{dv} + wF_{dw} + G] dV \\ \oint_V \mathbf{F}_d dV &= \oint_V (\nabla \cdot K \nabla T) dV + \oint_V (uF_{du} + vF_{dv} + wF_{dw}) dV + \oint_V G dV \\ \oint_V \mathbf{F}_d dV &= \underbrace{\oint_V (\nabla \cdot K \nabla T) dV}_{\text{DiffusionFlux}} + (uF_{du} + vF_{dv} + wF_{dw}) V_P + G V_P \end{aligned}$$

F_{du}, F_{dv}, F_{dw} are the viscous terms in u,v,w momentum equations and the calculation of these terms already discussed earlier.

4.6.1 Diffusion Fluxes:

$$\oint_V (\nabla \cdot \Gamma_\Phi \nabla \Phi) dV = \int_s \Gamma_\Phi \nabla \Phi \cdot dS \quad (4.32)$$

The surface integral over diffusion flux of variable Φ can be approximated as

$$\int_s \Gamma_\Phi \nabla \Phi \cdot dS = \sum_{j=e,w,n,s,t,b} (\Gamma_\Phi \nabla \Phi \cdot S)_j = \sum_j -F_j d \quad (4.33)$$

for any face we can write,

$$s_j = \alpha_1 n^1 + \alpha_2 n^2 + \alpha_3 n^3 \quad (4.34)$$

where n^1, n^2, n^3 are three linearly independent (not necessarily orthogonal) unit vectors. Therefore,

$$\nabla \Phi \cdot S_j = \nabla \Phi \cdot (\alpha_1 n^1 + \alpha_2 n^2 + \alpha_3 n^3) \quad (4.35)$$

$$= \alpha_1 \nabla \Phi \cdot n^1 + \alpha_2 \nabla \Phi \cdot n^2 + \alpha_3 \nabla \Phi \cdot n^3 \quad (4.36)$$

If $\Delta\Phi^1, \Delta\Phi^2, \Delta\Phi^3$ are the differences in Φ between the two ends of the line segments $\Delta x^1, \Delta x^2, \Delta x^3$ then,

$$\Delta\Phi^1 = \nabla \Phi \cdot \Delta x^1, \Delta\Phi^2 = \nabla \Phi \cdot \Delta x^2, \Delta\Phi^3 = \nabla \Phi \cdot \Delta x^3 \quad (4.37)$$

If $\Delta x^1, \Delta x^2, \Delta x^3$ are in the direction of $n^1, n^2,$ and n^3 respectively then equation, we can write

$$\frac{\Delta\Phi^1}{\Delta x^1} = \nabla \Phi \cdot n^1, \frac{\Delta\Phi^2}{\Delta x^2} = \nabla \Phi \cdot n^2, \frac{\Delta\Phi^3}{\Delta x^3} = \nabla \Phi \cdot n^3 \quad (4.38)$$

where Δx^1 is the magnitude of ΔX^1 etc. Therefore using equation and we can write,

$$\nabla \Phi \cdot S = \alpha_1 \frac{\Delta\Phi^1}{\Delta x^1} + \alpha_2 \frac{\Delta\Phi^2}{\Delta x^2} + \alpha_3 \frac{\Delta\Phi^3}{\Delta x^3} \quad (4.39)$$

To get $\alpha_1, \alpha_2, \alpha_3$, we have

$$n^1 = (n_{11} n_{12} n_{13})$$

$$n^2 = (n_{21} n_{22} n_{23})$$

$$n^3 = (n_{31} n_{32} n_{33})$$

Where $n_{11}, n_{12},$ and n_{13} are the Cartesian components of \mathbf{n} and which can be easily determined by $\frac{\Delta X_1^1}{\Delta x^1}, \frac{\Delta X_2^1}{\Delta x^2}, \frac{\Delta X_3^1}{\Delta x^3}$ where, $\Delta X_1^1, \Delta X_2^1, \Delta X_3^1$ are the components of vector ΔX^1 etc. The other values n_{11}, n_{12}, \dots etc. can be similarly determined. Therefore equation can be written as,

$$\begin{bmatrix} n_{11} & n_{21} & n_{31} \\ n_{12} & n_{22} & n_{32} \\ n_{13} & n_{23} & n_{33} \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{Bmatrix} = \begin{Bmatrix} S_{1j} \\ S_{2j} \\ S_{3j} \end{Bmatrix}$$

where, S_{1j}, S_{2j}, S_{3j} are the Cartesian components of the surface vectors S_j Using Cramer's rule, we get

$$\alpha_1 = \frac{D_1}{D}, \alpha_2 = \frac{D_2}{D}, \alpha_3 = \frac{D_3}{D} \quad (4.40)$$

Where D is the determinant of the coefficient matrix. The diffusion flux is made of two distinct parts: normal diffusion and cross derivative diffusion. The second part arises from the non-orthogonality of the grid. The normal derivative diffusion flux of Φ through any cell face involves the values of Φ at cell center whereas the cross-derivative diffusion flux takes into account the edge center value of Φ . The example of the east face is taken to illustrate the diffusion model and is shown in Fig. 4.3. We get the normal diffusion term $\frac{\Phi_E - \Phi_P}{|\Delta x_1|}$ involving cell center values and the cross diffusion terms $\frac{\Phi_{se} - \Phi_{ne}}{|\Delta x_2|}$ and $\frac{\Phi_{te} - \Phi_{be}}{|\Delta x_3|}$ involving the edge center values $\Phi_{te}, \Phi_{be}, \Phi_{se}, \Phi_{ne}$. We compute the diffusion flux by

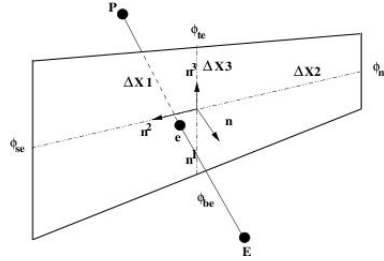


Figure 4.3: Face-Representation

$$F_e^d = -\Gamma_\phi \left(\alpha_1 \frac{\phi_E - \phi_P}{\Delta x^1} + \alpha_2 \frac{\phi_{se} - \phi_{ne}}{\Delta x^2} + \alpha_3 \frac{\phi_{te} - \phi_{be}}{\Delta x^3} \right) \quad (4.41)$$

$$\beta_1 = \frac{\alpha_1}{\Delta x^1}, \beta_2 = \frac{\alpha_2}{\Delta x^2}, \beta_3 = \frac{\alpha_3}{\Delta x^3}$$

$$F_e^d = -\Gamma_\phi (\beta_1 (\phi_E - \phi_P) + \beta_2 (\phi_{se} - \phi_{ne}) + \beta_3 (\phi_{te} - \phi_{be})) \quad (4.42)$$

To calculate the edge center values appearing in cross derivative diffusion flux, the following interpolation scheme is used.

$$\Phi_{te} = \frac{V_{TE}\Phi_P + V_P\Phi_{TE} + V_T\Phi_E + V_E\Phi_T}{V_{TE} + V_P + V_T + V_E} \quad (4.43)$$

$$\Phi_{be} = \frac{V_{BE}\Phi_P + V_P\Phi_{BE} + V_B\Phi_E + V_E\Phi_B}{V_{BE} + V_P + V_T + V_E} \quad (4.44)$$

$$\Phi_{ne} = \frac{V_{NE}\Phi_P + V_P\Phi_{NE} + V_N\Phi_E + V_E\Phi_N}{V_{NE} + V_P + V_N + V_E} \quad (4.45)$$

$$\Phi_{se} = \frac{V_{SE}\Phi_P + V_P\Phi_{SE} + V_S\Phi_E + V_E\Phi_S}{V_{SE} + V_P + V_S + V_E} \quad (4.46)$$

Where the notation used is that V_{TE} is the volume of the cell to the top-east of cell P and Φ_{TE} is its cell center value. Other edge center values can be determined by similar interpolation.

Special Treatment Of Diffusion Fluxes Of Corner Cells:

The solver is a multi-zone solver, and the corner cells of each zone may have more than two neighboring zones. Calculation of diffusion fluxes with the same existing diffusion formulation at corner cells of zones is difficult due to complexity involved in traversing the zones to find the corner cell values of the next zone. Therefore, a 3-Point diffusion formulation, suggested by Eswaran (private communication) has been used instead of the usual 4-Point diffusion formulation, used in the interior of domain. This formulation has been validated for different bench-mark problems and shows a good agreement with 4-Point formulation even for complex geometries. Fig. 4.4 shows implementation of the 3-Point formulation. The 3-Point formulation is used only at the corner cell edges, for corners of all the zones.

To calculate the edge center values appearing in cross derivative diffusion flux at the corners of the corner cells, the following 3-Point interpolation schemes is used.

$$\Phi_{te} = \frac{V_E\Phi_P + V_P\Phi_E}{V_E + V_P} + \frac{V_T\Phi_P + V_P\Phi_T}{V_T + V_P} - \Phi_P \quad (4.47)$$

$$\Phi_{be} = \frac{V_E\Phi_P + V_P\Phi_E}{V_E + V_P} + \frac{V_b\Phi_P + V_P\Phi_b}{V_b + V_P} - \Phi_P \quad (4.48)$$

$$\Phi_{ne} = \frac{V_E\Phi_P + V_P\Phi_E}{V_E + V_P} + \frac{V_N\Phi_P + V_P\Phi_N}{V_N + V_P} - \Phi_P \quad (4.49)$$

$$\Phi_{se} = \frac{V_E\Phi_P + V_P\Phi_E}{V_E + V_P} + \frac{V_S\Phi_P + V_P\Phi_S}{V_S + V_P} - \Phi_P \quad (4.50)$$

where the notation used is the same as in Eqn..

4.6.2 Computation of Spatial Derivatives at Cell-Center of a Non-orthogonal Grid (Least Square Method) :

In this section, The procedure for calculating the gradients of any variable Φ in x, y and z directions at centroid of the control volume in a non-orthogonal grid is described.

Let the difference vectors in the 3 index directions be,

$$\begin{aligned}\vec{\Delta}_i &= \Delta_{xi}\hat{i} + \Delta_{yi}\hat{j} + \Delta_{zi}\hat{k} \\ \vec{\Delta}_i &= (x_{i+1,j,k} - x_{i-1,j,k})\hat{i} + (y_{i+1,j,k} - y_{i-1,j,k})\hat{j} + (z_{i+1,j,k} - z_{i-1,j,k})\hat{k} \\ \vec{\Delta}_j &= \Delta_{xj}\hat{i} + \Delta_{yj}\hat{j} + \Delta_{zj}\hat{k} \\ \vec{\Delta}_j &= (x_{i,j+1,k} - x_{i,j-1,k})\hat{i} + (y_{i,j+1,k} - y_{i,j-1,k})\hat{j} + (z_{i,j+1,k} - z_{i,j-1,k})\hat{k} \\ \vec{\Delta}_k &= \Delta_{xk}\hat{i} + \Delta_{yk}\hat{j} + \Delta_{zk}\hat{k} \\ \vec{\Delta}_k &= (x_{i,j,k+1} - x_{i,j,k-1})\hat{i} + (y_{i,j,k+1} - y_{i,j,k-1})\hat{j} + (z_{i,j,k+1} - z_{i,j,k-1})\hat{k}\end{aligned}$$

now we know that for any scalar

$$\Delta\Phi_i = \Phi_{i+1,j,k} - \Phi_{i-1,j,k} = \frac{\partial\Phi}{\partial x}\Delta_{xi} + \frac{\partial\Phi}{\partial y}\Delta_{yi} + \frac{\partial\Phi}{\partial z}\Delta_{zi} + HOT \quad (4.51)$$

$$\Delta\Phi_j = \Phi_{i,j+1,k} - \Phi_{i,j-1,k} = \frac{\partial\Phi}{\partial x}\Delta_{xj} + \frac{\partial\Phi}{\partial y}\Delta_{yj} + \frac{\partial\Phi}{\partial z}\Delta_{zj} + HOT \quad (4.52)$$

$$\Delta\Phi_k = \Phi_{i+1,j,k} - \Phi_{i-1,j,k} = \frac{\partial\Phi}{\partial x}\Delta_{xk} + \frac{\partial\Phi}{\partial y}\Delta_{yk} + \frac{\partial\Phi}{\partial z}\Delta_{zk} + HOT \quad (4.53)$$

$$\begin{bmatrix} \Delta_{xi} & \Delta_{yi} & \Delta_{zi} \\ \Delta_{xj} & \Delta_{yj} & \Delta_{zj} \\ \Delta_{xk} & \Delta_{yk} & \Delta_{zk} \end{bmatrix} \begin{Bmatrix} \frac{\partial\Phi}{\partial x} \\ \frac{\partial\Phi}{\partial y} \\ \frac{\partial\Phi}{\partial z} \end{Bmatrix} = \begin{Bmatrix} \Delta\Phi_i \\ \Delta\Phi_j \\ \Delta\Phi_k \end{Bmatrix}$$

the left matrix can be easily inverted either during initialization or during computation,i.e.,

$$\begin{Bmatrix} \frac{\partial\Phi}{\partial x} \\ \frac{\partial\Phi}{\partial y} \\ \frac{\partial\Phi}{\partial z} \end{Bmatrix} = [\Delta^{-1}] \begin{Bmatrix} \Delta\Phi_i \\ \Delta\Phi_j \\ \Delta\Phi_k \end{Bmatrix}$$

4.7 MacCormack Scheme Discretization for 2D Rectangular Grids

Before the MacCormack and AUSM+ algorithm for N-S equations were implemented in the Anupravaha solver, the MacCormack algorithm was tested on a rectangular grid. We present the 2D algorithm below.

4.7.1 MacCormack Finite Difference Discretization in 2D

Governing equations of compressible fluid flow in 2D

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0$$

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E_t \end{pmatrix}$$

$$E = \begin{pmatrix} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{xy} \\ (E_t + p)u - u\tau_{xx} - v\tau_{xy} + q_x \end{pmatrix}$$

$$F = \begin{pmatrix} \rho v \\ \rho uv - \tau_{yx} \\ \rho v^2 + p - \tau_{yy} \\ (E_t + p)v - u\tau_{yx} - v\tau_{yy} + q_y \end{pmatrix}$$

Predictor Step:

$$\frac{U_{ij}^* - U_{ij}^n}{\Delta t} + \frac{E_{i+1j}^n - E_{ij}^n}{\Delta x} + \frac{F_{ij+1}^n - F_{ij}^n}{\Delta y} = 0 \quad (4.54)$$

Corrector Step:

$$\frac{U_{ij}^{**} - U_{ij}^*}{\Delta t} + \frac{E_{ij}^* - E_{i-1j}^*}{\Delta x} + \frac{F_{ij}^* - F_{ij-1}^*}{\Delta y} = 0 \quad (4.55)$$

Final Step:

$$U_{ij}^{n+1} = \frac{U_{ij}^n + U_{ij}^{**}}{2} \quad (4.56)$$

$$\tau_{xy} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right); \tau_{xx} = 2\mu \frac{\partial u}{\partial x} + \lambda(\nabla \cdot V)$$

$$\tau_{yy} = 2\mu \frac{\partial v}{\partial y} + \lambda(\nabla \cdot V)$$

$$\lambda = -\frac{2\mu}{3}; q_x = -k \frac{\partial T}{\partial x}; q_y = -k \frac{\partial T}{\partial y}; e = c_v T$$

MacCormack Discretization of Viscous Terms

To maintain second order accuracy, the x-derivative terms appearing in E are differenced in the opposite direction to that used for $\frac{\partial E}{\partial x}$, while the y-derivative terms are approximated with central differences. Likewise, the y-derivative terms appearing in F are differenced in the opposite direction to that used for $\frac{\partial F}{\partial y}$, while the x-derivative terms in F are approximated with central differences.

4.7.2 Discretization of Viscous Terms with Central Difference

Governing equations of compressible fluid flow in 2D

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial E_V}{\partial x} + \frac{\partial F_V}{\partial y} = 0$$

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E_t \end{pmatrix}$$

$$E = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E_t + p)u \end{pmatrix}$$

$$F = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E_t + p)v \end{pmatrix}$$

$$E_V = \begin{pmatrix} 0 \\ -\tau_{xx} \\ -\tau_{xy} \\ -u\tau_{xx} - v\tau_{xy} + q_x \end{pmatrix}$$

$$F_V = \begin{pmatrix} 0 \\ -\tau_{yx} \\ -\tau_{yy} \\ -u\tau_{yx} - v\tau_{yy} + q_y \end{pmatrix}$$

Predictor Step:

$$\frac{U_{ij}^* - U_{ij}^n}{\Delta t} + \frac{E_{i+1j}^n - E_{ij}^n}{\Delta x} + \frac{F_{ij+1}^n - F_{ij}^n}{\Delta y} + \frac{E_{Vi+1j}^n - E_{Vi-1j}^n}{2\Delta x} + \frac{F_{Vij+1}^n - F_{Vij-1}^n}{2\Delta y} = 0 \quad (4.57)$$

Corrector Step:

$$\frac{U_{ij}^{**} - U_{ij}^*}{\Delta t} + \frac{E_{ij}^* - E_{i-1j}^*}{\Delta x} + \frac{F_{ij}^* - F_{ij-1}^*}{\Delta y} + \frac{E_{Vi+1j}^* - E_{Vi-1j}^*}{2\Delta x} + \frac{F_{Vij+1}^* - F_{Vij-1}^*}{2\Delta y} = 0 \quad (4.58)$$

Final Step:

$$U_{ij}^{n+1} = \frac{U_{ij}^n + U_{ij}^{**}}{2} \quad (4.59)$$

$$\tau_{xy} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right); \tau_{xx} = 2\mu \frac{\partial u}{\partial x} + \lambda(\nabla \cdot V)$$

$$\tau_{yy} = 2\mu \frac{\partial v}{\partial y} + \lambda(\nabla \cdot V)$$

$$\lambda = -\frac{2\mu}{3}; q_x = -k \frac{\partial T}{\partial x}; q_y = -k \frac{\partial T}{\partial y}; e = c_v T$$

Irrespective of x-derivative terms or y-derivative terms, all viscous terms are approximated by central differences.

4.7.3 2d-MacCormack Finite Volume Discretization for Rectangular Grids

2d-equations

x-momentum:

$$\begin{aligned} \frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u^2) + \frac{\partial}{\partial y}(\rho uv) = & -\frac{\partial p}{\partial x} + (\nabla \cdot \mu \nabla u) + \frac{\partial}{\partial x} \left(\frac{\mu}{3} \nabla \cdot V \right) \\ & + \frac{\partial \mu}{\partial y} \frac{\partial v}{\partial x} - \frac{\partial \mu}{\partial x} \frac{\partial v}{\partial y} \end{aligned} \quad (4.60)$$

y-momentum:

$$\begin{aligned} \frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho uv) + \frac{\partial}{\partial y}(\rho v^2) = & -\frac{\partial p}{\partial y} + (\nabla \cdot \mu \nabla v) + \frac{\partial}{\partial y} \left(\frac{\mu}{3} \nabla \cdot V \right) \\ & + \frac{\partial \mu}{\partial x} \frac{\partial u}{\partial y} - \frac{\partial \mu}{\partial y} \frac{\partial u}{\partial x} \end{aligned} \quad (4.61)$$

discretization of convective terms uses same discretization as finite difference standard MacCormack discretization. But for diffusion terms we need gradients at the cell center and cell faces. next we discuss about these gradients.

Derivarive at the centre:

$$\frac{\partial \Phi}{\partial x} = \frac{1}{V_p} \left(\sum_f \Phi_f S_{fx} \right) = \frac{1}{V_p} (\Phi_e S_{ex} + \Phi_w S_{wx} + \Phi_n S_{nx} + \Phi_s S_{sx})$$

$$\frac{\partial \Phi}{\partial x} = \frac{\Phi_{i+1,j} - \Phi_{i-1,j}}{2\Delta x}$$

$$\frac{\partial \Phi}{\partial y} = \frac{\Phi_{i,j+1} - \Phi_{i,j-1}}{2\Delta y}$$

Derivative at the faces: for example derivative at the east face,

$$\left(\frac{\partial \Phi}{\partial x}\right)_e = \frac{\Phi_{i+1,j} - \Phi_{i,j}}{\Delta x}$$

$$\left(\frac{\partial \Phi}{\partial y}\right)_e = \frac{\Phi_{ne} - \Phi_{se}}{\Delta y}$$

and Φ_{ne}, Φ_{se} values can be found out by volume interpolating the four neighbouring cells.

$$\Phi_{ne} = \frac{\Phi_{i,j} + \Phi_{i+1,j} + \Phi_{i,j+1} + \Phi_{i+1,j+1}}{4}$$

$$\Phi_{se} = \frac{\Phi_{i,j} + \Phi_{i+1,j} + \Phi_{i,j-1} + \Phi_{i+1,j-1}}{4}$$

4.8 Closure

In this chapter we have seen the discretization of convective fluxes by central scheme such as MacCormack and the upwind based scheme AUSM⁺-up. Also, we saw the detail formulation and discretization of viscous terms of compressible flows. In the end of these chapter we added preliminary work done by different methods.

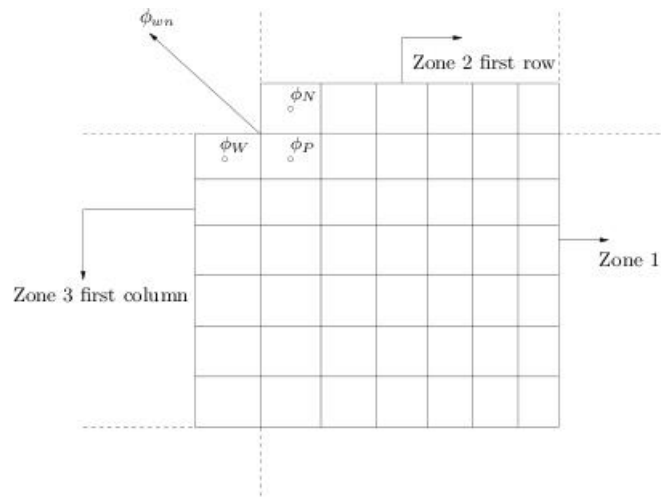


Figure 4.4: Three Point Formulation

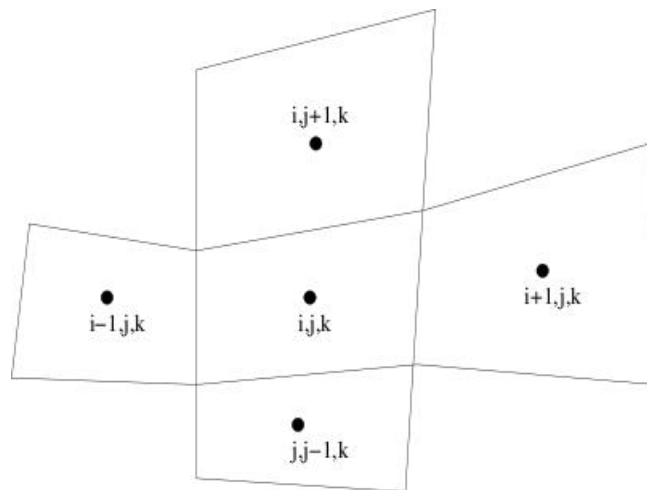


Figure 4.5: Neighbouring cell indices

Chapter 5

Results And Discussion

In this chapter we present the results obtained using the solver for different geometric configurations and for different regimes of flow. The results are all compared with standard benchmark and/or analytical results. The MacCormack scheme with artificial viscosity (referred as *MacCormack*) and AUSM⁺-up scheme (referred as *AUSM+*) have been tested. The aim is to study the performance of these schemes and validate the compressible flow solver module of Anupravaha. Since, the solver is 3D-based, for all the 2D test-cases we have given a minimum of 4 cells thickness in the z-direction and symmetry boundary conditions are applied on the two boundaries normal to the z-direction.

Before going to implement the viscous terms in the Anupravaha solver, we tried the finite difference method given by Anderson [5], and compared it with the results therein. The scheme presented by Anderson [5] uses the MacCormack for both convective and viscous terms, while the scheme presented in chapter 4 uses the discretization of viscous terms by central difference, while the convective discretization was done using the standard MacCormack scheme. So, in the first section Sec. 5.1 we present the results obtained by 2-D finite-difference scheme and 2D-finite volume scheme. In the next sections Sec. 5.2, 5.3 and 5.4, we present the results obtained by the Anupravaha solver i.e., 3-D finite volume structured code.

5.1 Preliminary Results from 2D-Rectangular Codes for Supersonic Flow Over a Flat Plate

Set-up

Consider the supersonic flow over a thin sharp flat plate [5] at zero incidence and of length L , as sketched in Fig. 5.1 A laminar boundary layer develops at the leading edge of the flat plate and remains laminar for the case of relatively low Reynolds number. The oncoming free stream no longer sees a sharp flat plate. Rather due to the presence of the viscous boundary layer, the plate possesses an effective curvature. Consequently, a curved induced

shock wave is generated at the leading edge. Now for the problem at hand, consider the computational domain, in this case a rectangular structured grid, as shown in the figure. The flow at the upstream boundary ($x=0.0$, $i=1$ or IMIN) is at Mach4 with pressure, temperature, and speed of sound is equal to their respective sea level values. The length of the plate is 0.00001 m. This is extremely small but large compared to the mean free path of the oncoming air molecules, and sufficient to capture the desired physics. The Reynolds number is about 1000. We want low Reynold number in order to keep computational running times relatively short compared to high Reynold number applications requiring considerably finer grids.

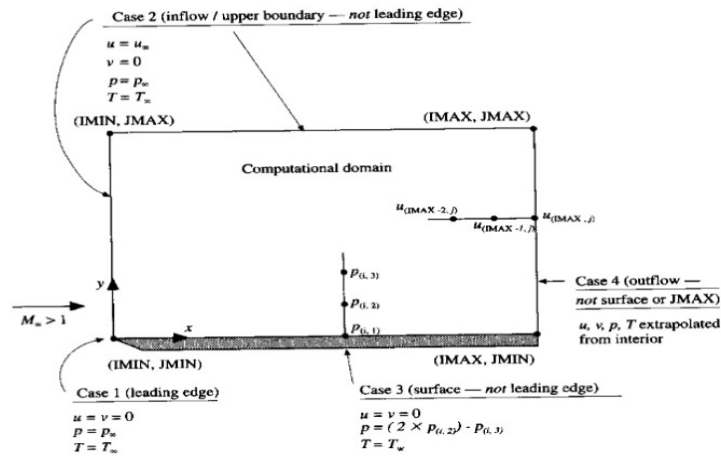


Figure 5.1: Flatplate [5]

$$L_H = 0.00001m$$

$$L_V = 5\delta$$

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

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

No.of grids 70×70

Constants for air:

Specific gas constant(R)= 287

ratio of specific heats(γ)= 1.4

Prandtl number(Pr)= 0.71

dynamic viscosity(μ)= $1.7894e-5$ kg/(m-sec)

Initial Condition

IC	Part
Pressure	101325 Pa
Temperature	288.16K
U velocity	4 Mach
V velocity	0
W velocity	0

Boundary Conditions

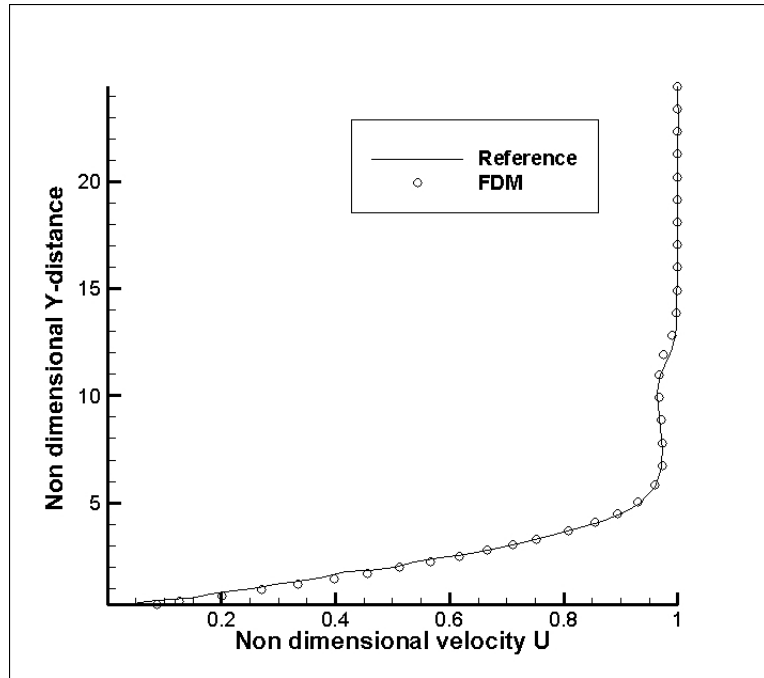
Quantity	Inflow	Outflow
Pressure	101325 Pa	-
Temperature	288.16 K	-
U velocity	4 Mach	-
V velocity	0	-
W velocity	0	-

Table 5.1: Boundary conditions for supersonic flatplate

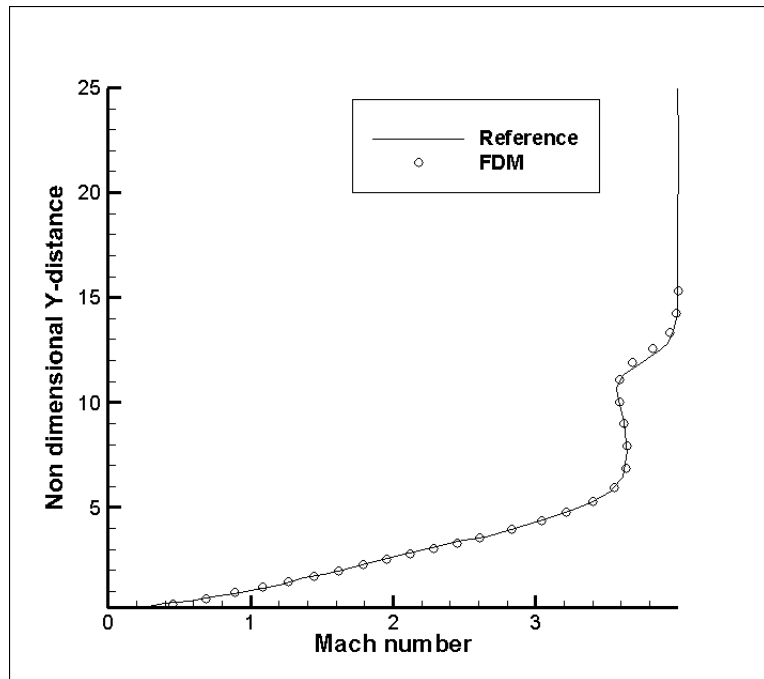
5.1.1 MacCormack Finite Difference Method:

Governing equations and Discretization has been discussed in Sec. 4.7.1. The steady-state results using MacCormack Scheme is shown in Fig. 5.2 and 5.3.

The finite-difference based MacCormack scheme gave similar and accurate results when compared with the reference [5].

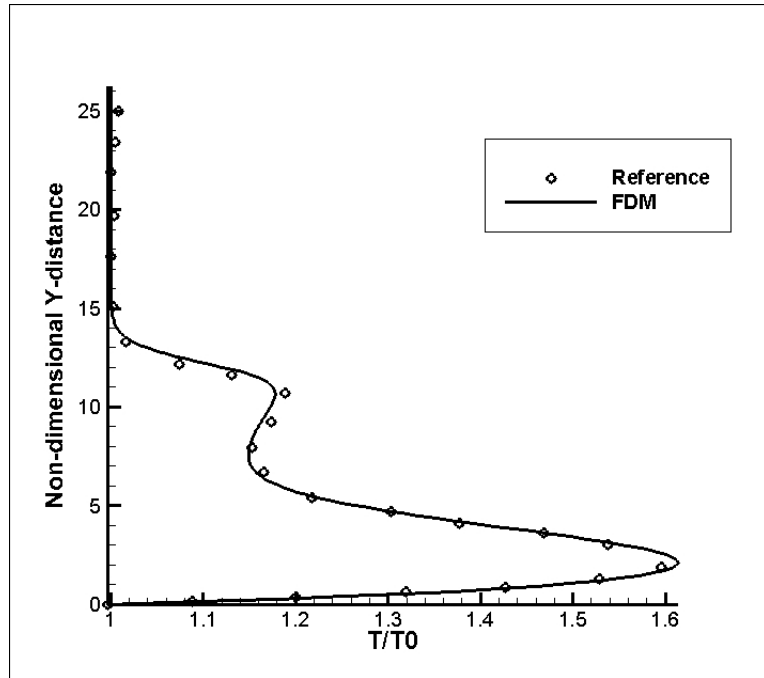


(a) Velocity-x

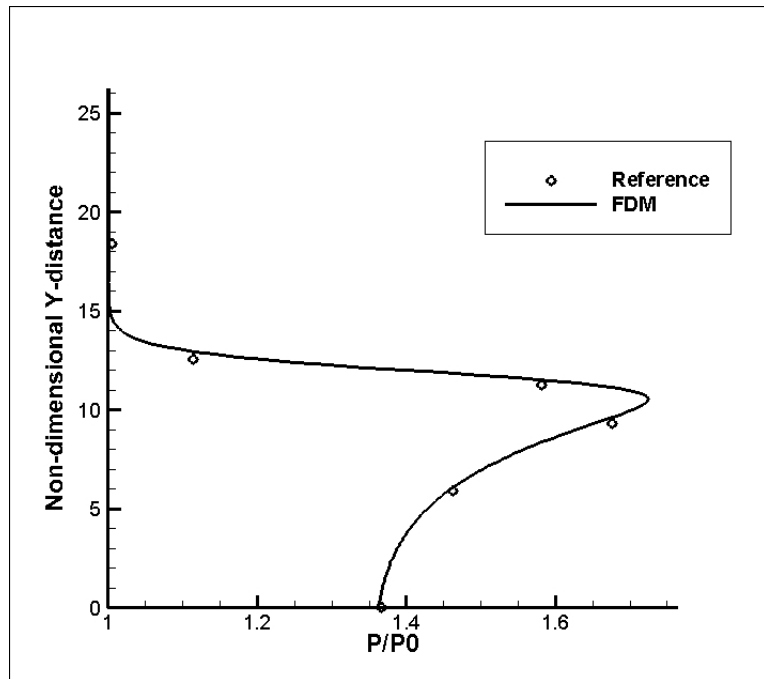


(b) Machnumber

Figure 5.2: Plot along outlet



(a) Temperature



(b) Pressure

Figure 5.3: Plot along outlet

5.1.2 MacCormack Discretization vs Central Discretization of Viscous Terms

Governing equations and discretization has been discussed in previous chapter Sec. 4.7.1. In this section we compare the MacCormack discretization with central discretization of viscous terms.

Both the MacCormack and central discretization of viscous terms give the similar results as we can see in Fig. 5.4 and 5.5.

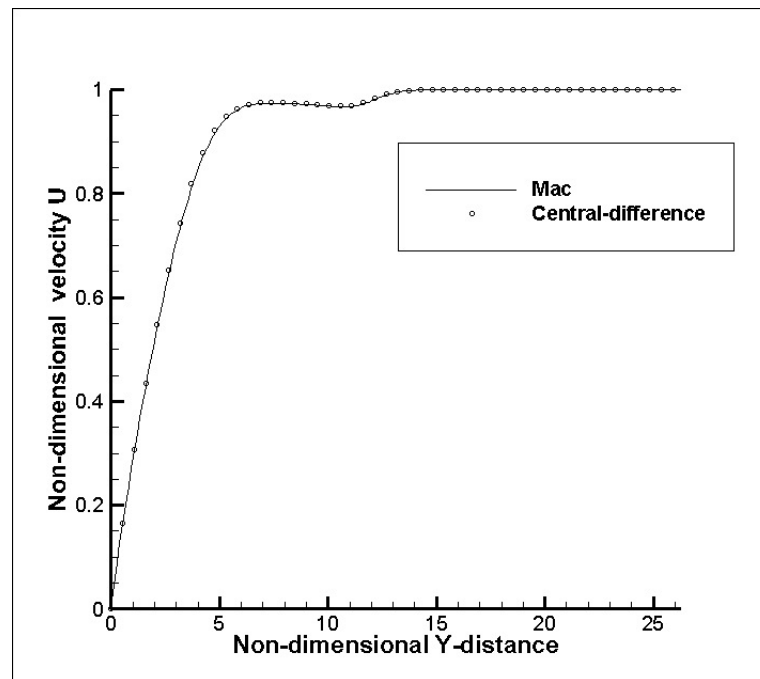


Figure 5.4: Plot for non-dimensioned velocity in X-direction w.r.t non-dimensioned Y

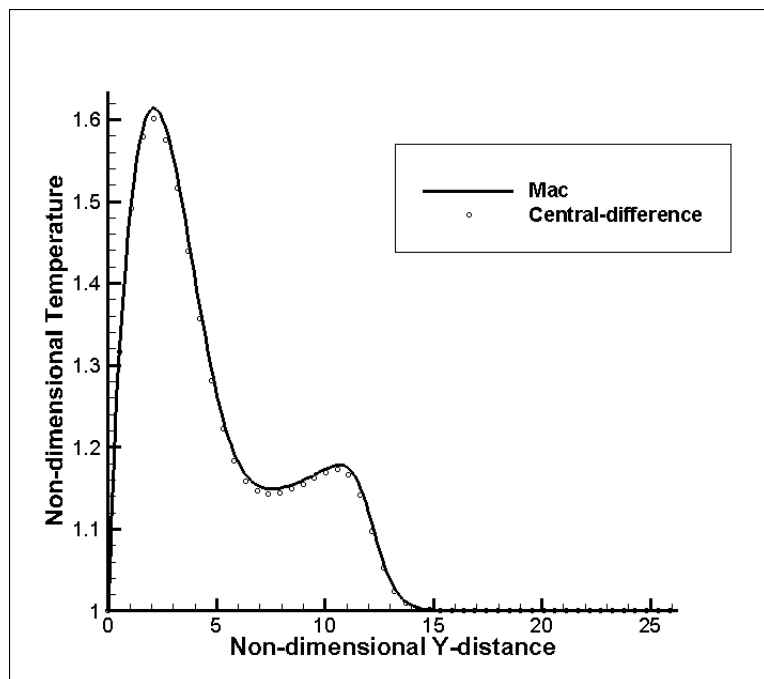


Figure 5.5: Plot for non-dimensioned temperature in X-direction w.r.t non-dimensioned Y

5.1.3 Finite Difference Method vs Finite Volume Method

In this section we compare finite difference discretization and finite volume discretization. Finite difference discretization uses the MacCormack scheme explained Sec 4.7.1. Finite volume discretization uses the same formulation as discussed in Sec. 4.7.3.

Both the finite-difference and finite-volume schemes give the same results as we can see Fig. in 5.6, 5.7 and 5.8.

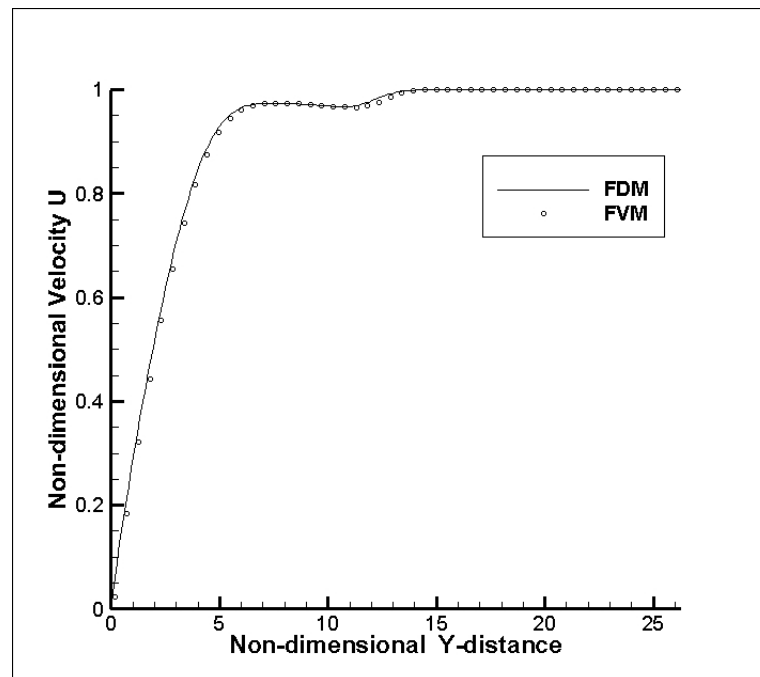


Figure 5.6: Plot for Velocity-x w.r.t non-dimensioned Y

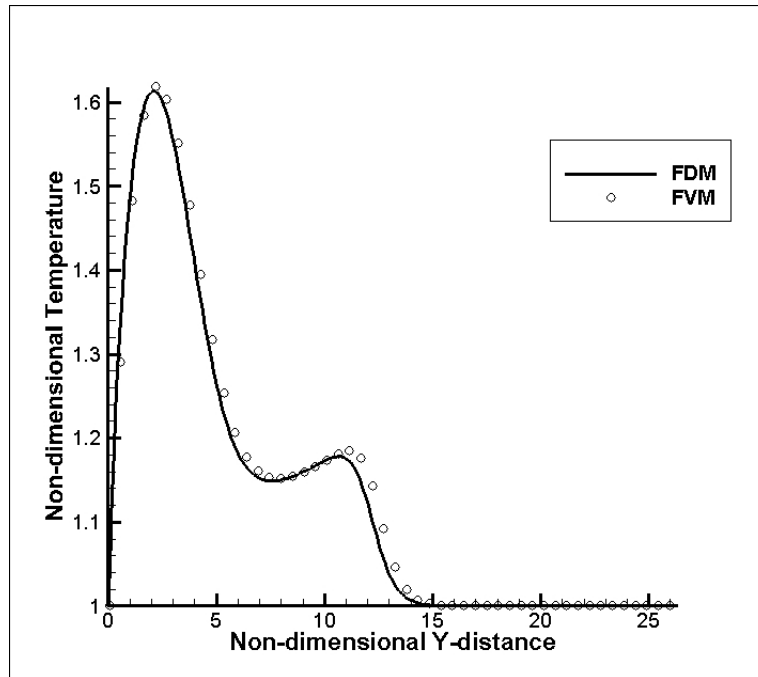


Figure 5.7: Plot for Temperature w.r.t non-dimensional Y

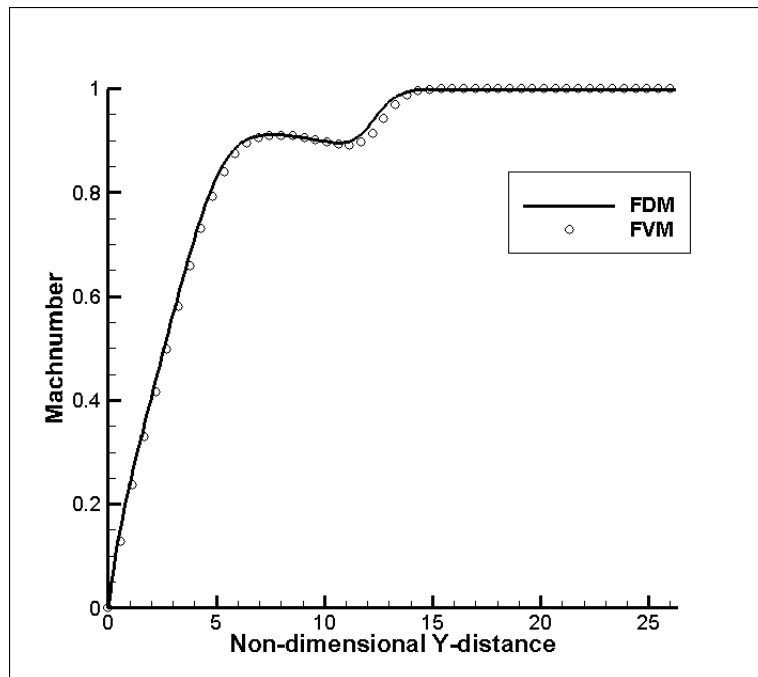


Figure 5.8: Plot for Mach number w.r.t non-dimensional Y

5.2 Supersonic flow Over a Flat Plate $Ma=2.06$, $Re=39,0000$

We now consider the supersonic flow over a 2-D flatplate. The inflow conditions are summarized in Table 5.2. Present results have been compared with analytical results [16]. A Courant number of 0.3 was used for both AUSM+ and MacCormack. AUSM+ scheme however gives a solution with less oscillations, as we can see in contours.

Computational Domain:

$$L_H = 1m$$

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

$$L_V = 5\delta$$

$$L_V = 0.04m$$

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

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

$$\mu = 0.002202041 \text{ kg/(m-sec)}$$

$$Pr = 0.71$$

$$k = \frac{\mu * Cp}{Pr}$$

Initial Condition:

IC	Part
Pressure	101325 Pa
Temperature	288.16K
U velocity	700.953 m/sec
V velocity	0
W velocity	0

Boundary Conditions:

Wall:

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

Non-dimensionalz y-distance:

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

Quantity	Inflow and Top	Outflow
Pressure	101325 Pa	-
Temperature	288.16 K	-
U velocity	700.953 m/sec	-
V velocity	0	-
W velocity	0	-

Table 5.2: Boundary conditions $M = 2.06$, $\alpha = 0^\circ$

For outlet or trailing edge i.e. $x=L$, \bar{y} becomes

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

The steady-state contours using MacCormack scheme is shown in Fig. 5.10 and the steady-state contours using AUSM+ is shown in Fig. 5.11. Comparison of MacCormack and AUSM+ shown in Fig. 5.9. Both the MacCorMack and AUSM+ gives similar results and both have high accuracy, corresponding to the analytical results [16] .

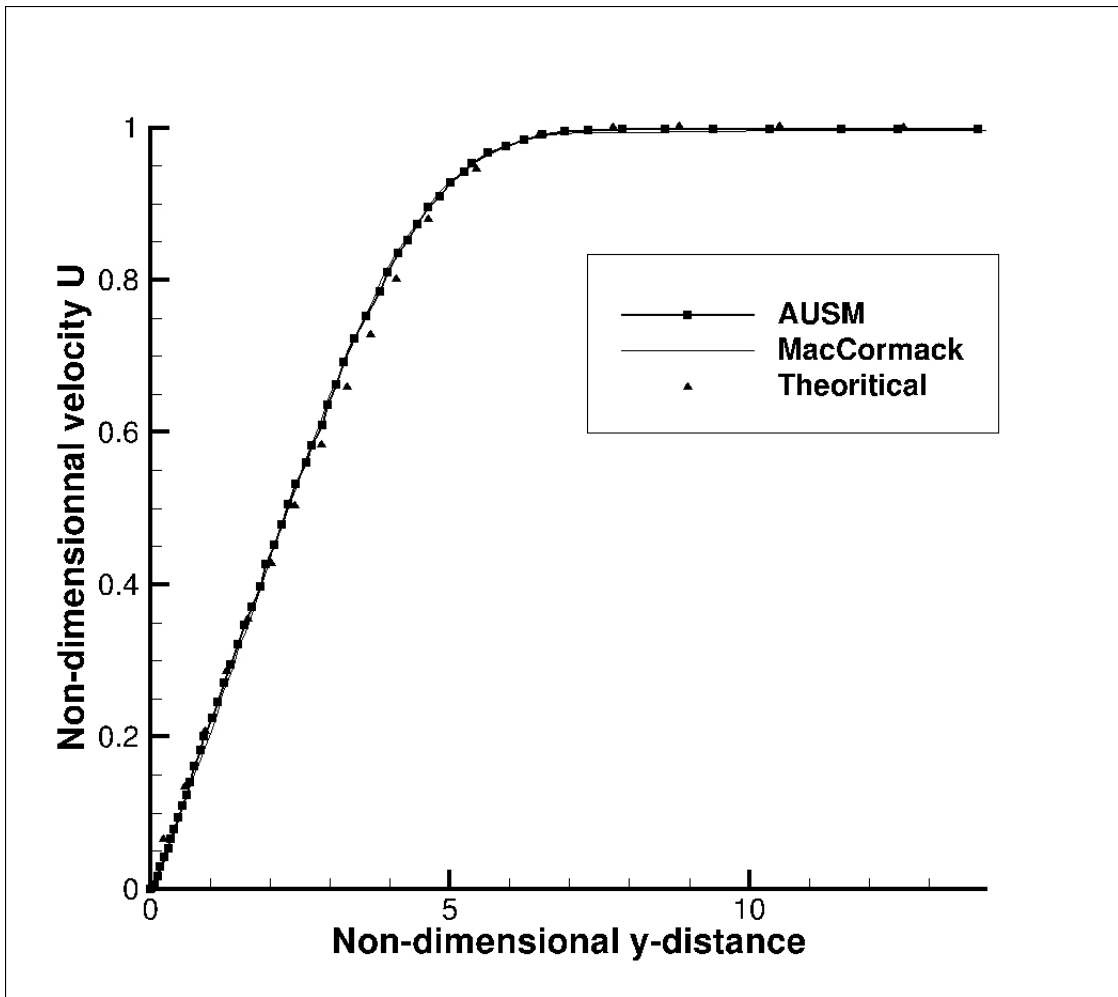
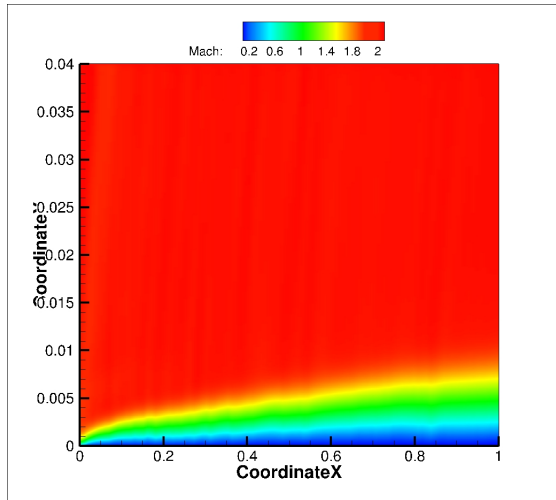
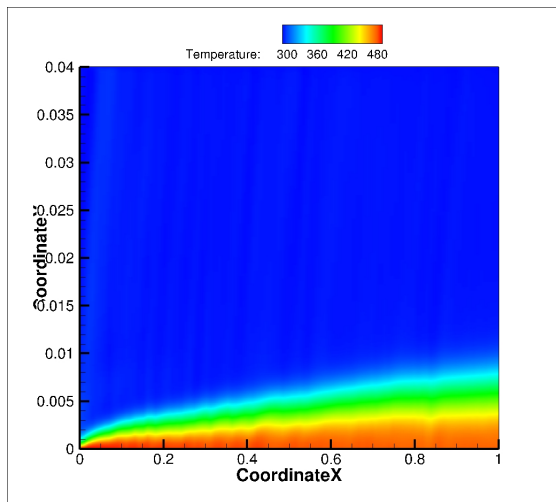


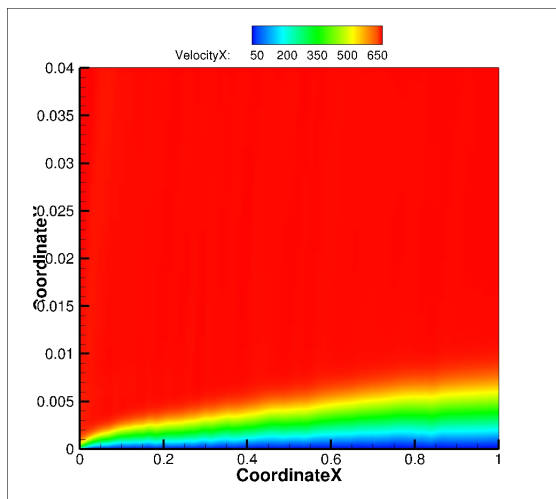
Figure 5.9: Plot for Velocity-x w.r.t non-dimensioned Y



(a) Machnumber

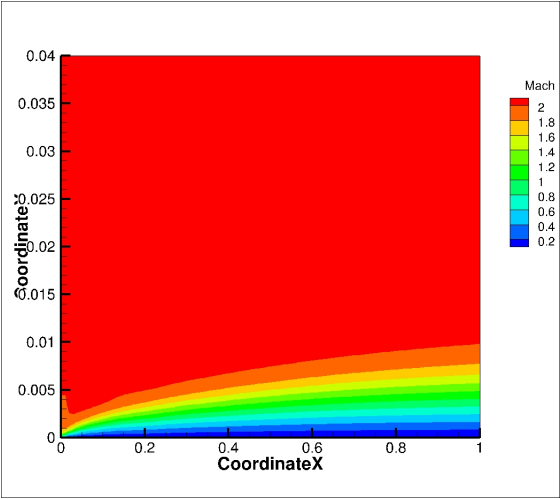


(b) Temperature

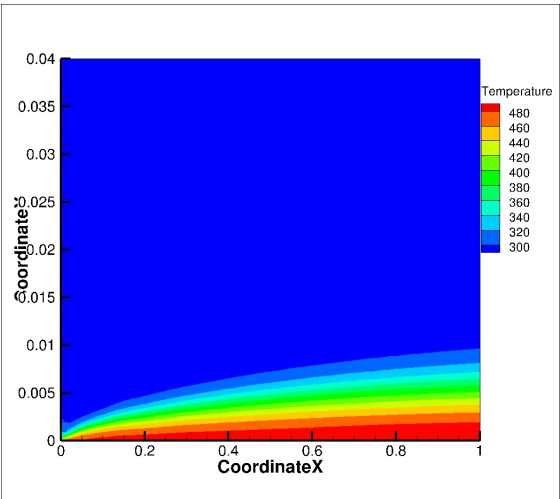


(c) VelocityX

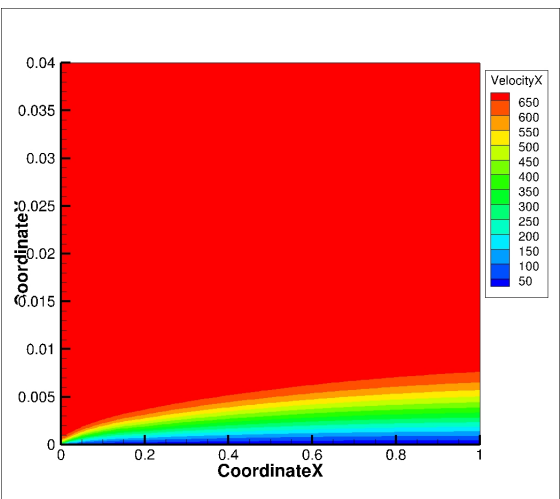
Figure 5.10: MacCormack contours



(a) Machnumber



(b) Temperature



(c) VelocityX

Figure 5.11: AUSM contours

5.3 Subsonic Flow Over a Flat Plate Ma=0.5, RE=10,000

We now consider the subsonic flow over a 2-D Flatplate. The inflow conditions are summarized in Table 5.3. and the present results have been compared with analytical Results [17] . A Courant number of 0.3 was used for both AUSM+ and MacCormack.

Computational Domain:

$$L_H = 1m$$

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

$$L_V = 5\delta$$

$$L_V = 0.25m$$

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

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

$$\mu = 0.0208445651 \text{ kg/(m-sec)}$$

$$Pr = 0.71$$

$$k = \frac{\mu * Cp}{Pr}$$

Initial Condition:

IC	Part
Pressure	101325 Pa
Temperature	288.16K
U velocity	170.134 m/sec
V velocity	0
W velocity	0

Boundary Conditions:

Wall:

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

Non-dimensional y-distance:

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

Quantity	Inflow	Outflow
Pressure	101325 Pa	101325 Pa
Temperature	288.16 K	-
U velocity	170.134 m/sec	-
V velocity	0	-
W velocity	0	-

Table 5.3: Boundary conditions $M = 0.5$, $\alpha = 0^\circ$

For outlet or trailing edge i.e. $x=L$, \bar{y} becomes

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

The steady-state contours using MacCormack are shown in Fig. 5.13 and the contours from AUSM+ are shown in Fig. 5.14. Both the MacCormack and AUSM+ gives similar results and both have high accuracy, corresponding to the analytical results [17] shown in Fig. 5.12. AUSM+ scheme however gives a solution with less oscillations as we can see in contours.

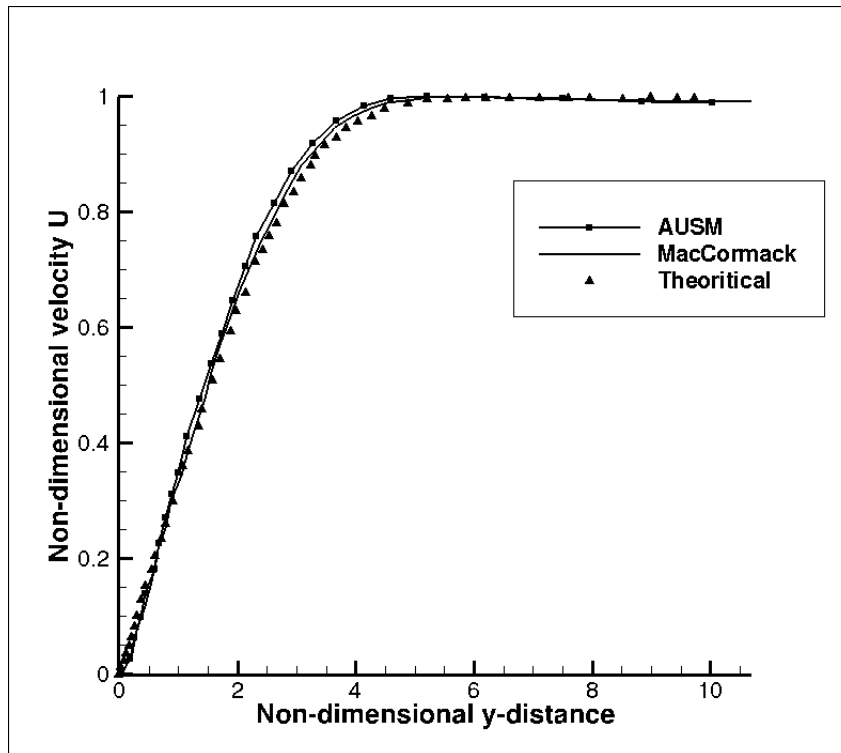
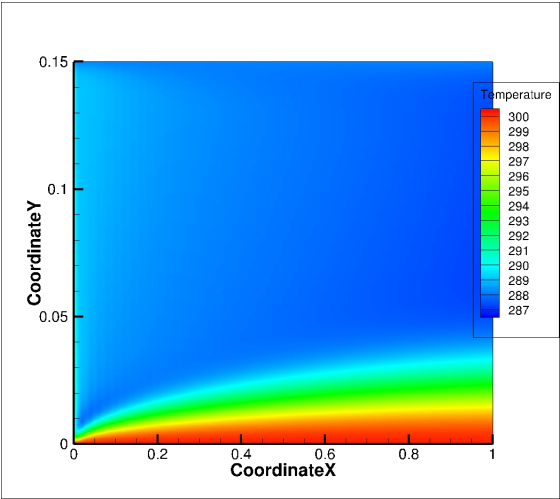
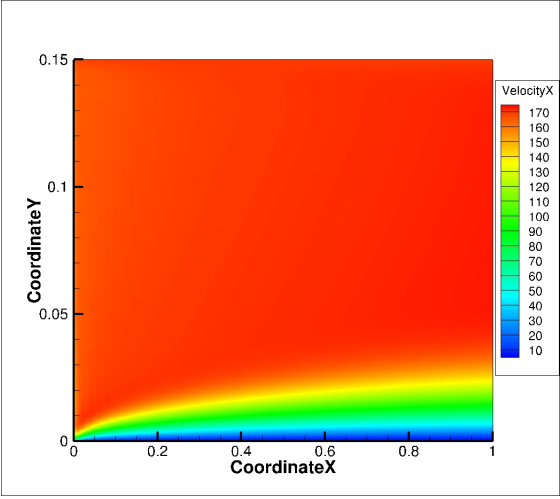


Figure 5.12: Plot for Velocity-x w.r.t non-dimensional Y

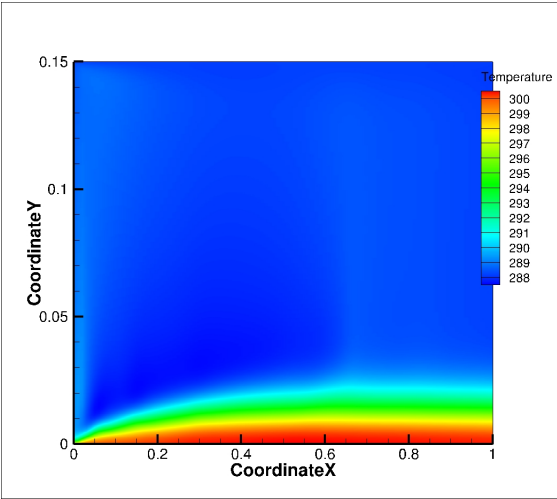


(a) Temperature

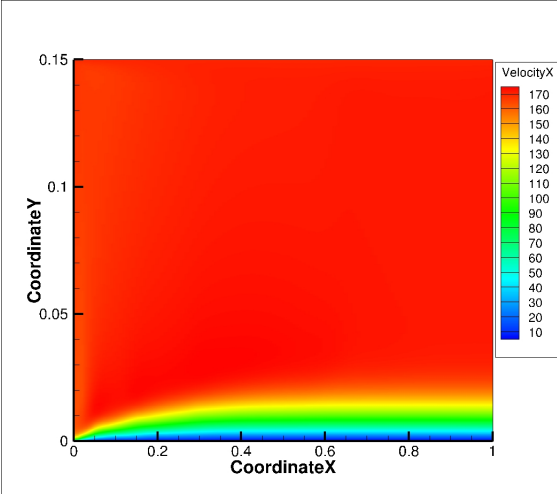


(b) VelocityX

Figure 5.13: MacCormack contours



(a) Temperature



(b) VelocityX

Figure 5.14: AUSM contours

5.4 Supersonic Flow Over a Wedge

We now consider the supersonic flow over a 2-D wedge with wedge angle 15° , as shown in Fig. 5.15 with Reynolds number of 1000. The inflow conditions are summarized in Table 5.4 and the present results have been compared with those from the commercial software Fluent. A Courant number of 0.3 was used for both AUSM+ and MacCormack. The steady-state contours of Mach number shown in Fig. 5.16, for static pressure in Fig. 5.17, and temperature in Fig. 5.18. Mach number at the outlet has been compared with Fluent results in Fig. 5.19.

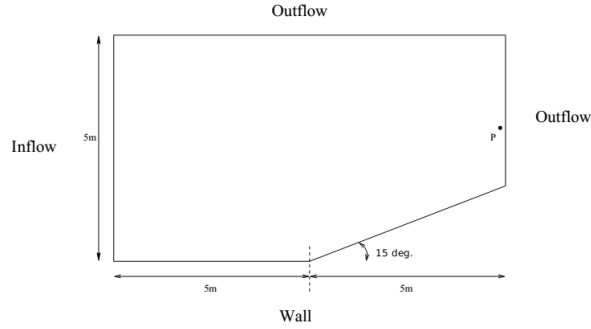


Figure 5.15: Computational domain.

Properties of the fluid

dynamic Viscosity $\mu = 0.83378260405 \text{ kg}/(\text{m}\cdot\text{sec})$

Prandtl number $\text{Pr}=0.71$

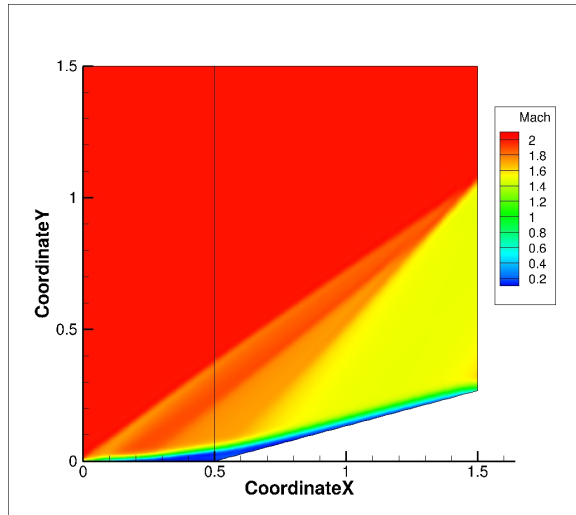
Thermal conductivity $K=1181.8927 \text{ W}/(\text{m}\cdot\text{K})$

Boundary Conditions

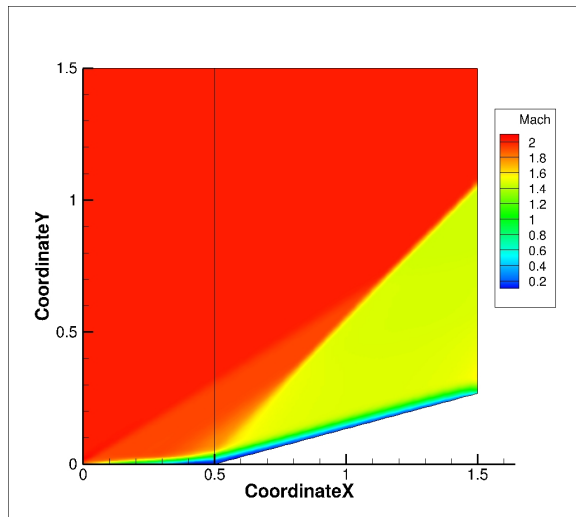
Quantity	Inflow	Outflow
Pressure	101325 Pa	-
Temperature	288.16 K	-
U velocity	2 Mach	-
V velocity	0	-
W velocity	0	-

Table 5.4: Boundary conditions for supersonic wedge

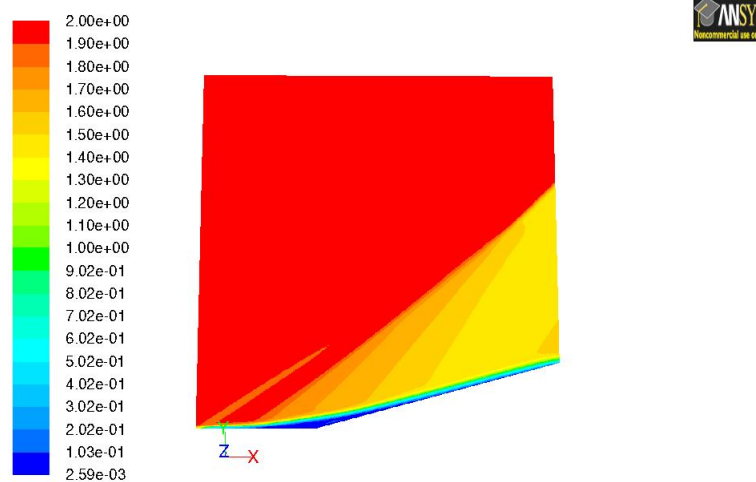
Both the MacCormack and AUSM+ gives similar results and both have high accuracy, corresponding with the Fluent results. AUSM+ scheme however gives a solution with less oscillations.



(a) MacCormack Scheme



(b) AUSM+ Scheme.

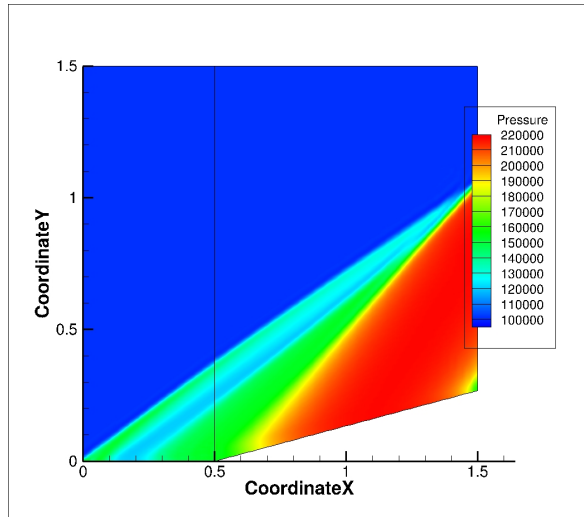


Contours of Mach Number

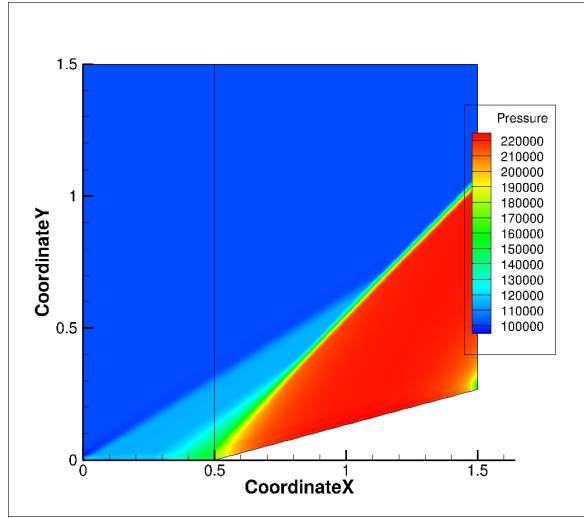
ANSYS FLUENT 12.1 (3d, dbns imp, lam) Jun 01, 2015

(c) Fluent

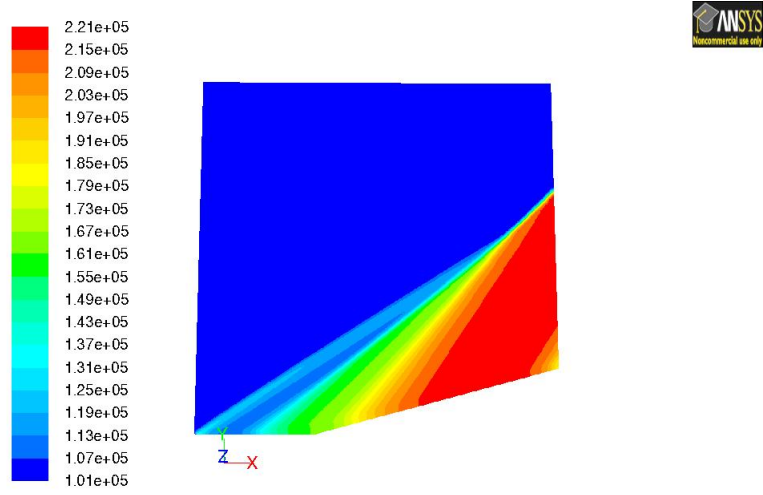
68
Figure 5.16: Contours for Mach Number



(a) MacCormack Scheme



(b) AUSM+ Scheme.

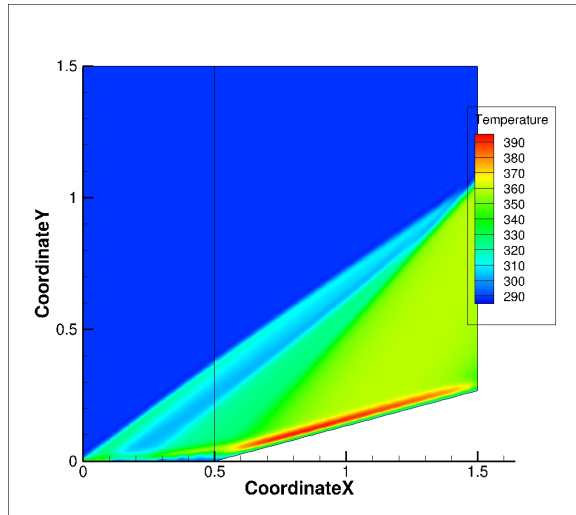


Contours of Static Pressure (pascal)

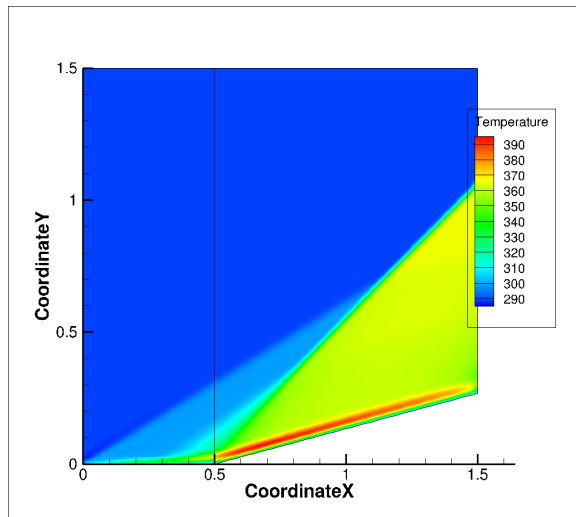
ANSYS FLUENT 12.1 (3d, dbns imp, lam) Jun 01, 2015

(c) Fluent

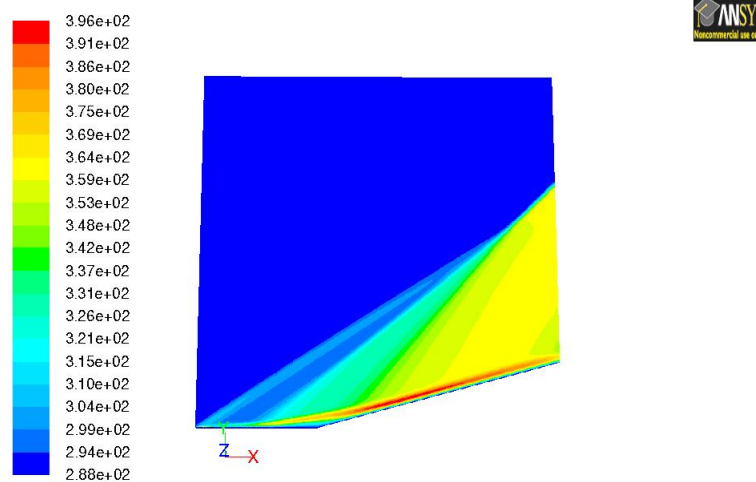
69
Figure 5.17: Contours for Pressure



(a) MacCormack Scheme



(b) AUSM+ Scheme.



Contours of Static Temperature (k)

ANSYS FLUENT 12.1 (3d, dbns imp, lam) Jun 01, 2015

(c) Fluent

70
Figure 5.18: Contours for Temperature

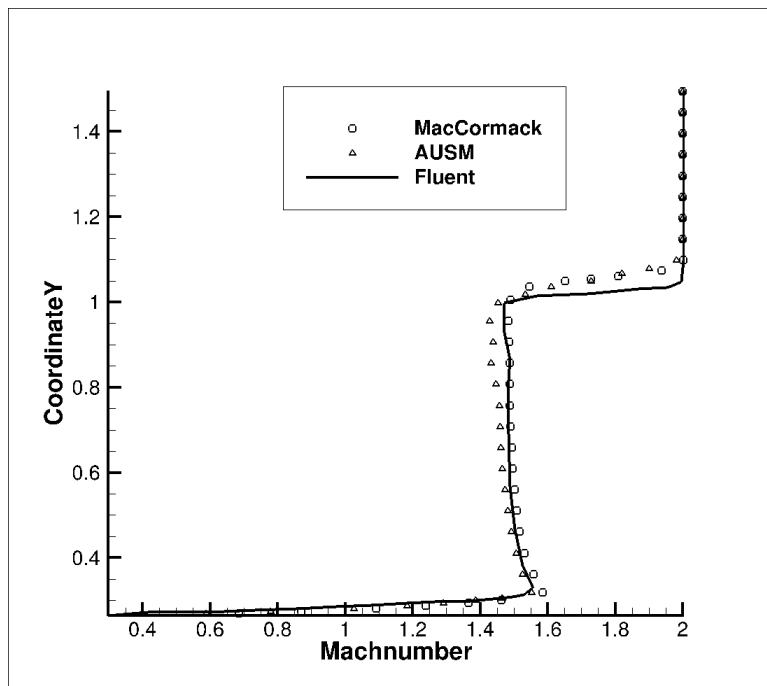


Figure 5.19: Comparison of mach number along outlet

Chapter 6

Conclusions and Future work

Building on the earlier work of Kalkote [11], Ashwani [7] we have created a stand-alone version of the AnuPravaha Solver for compressible flows. Two discretization schemes are implemented for viscous fluxes in the solver, and validated for subsonic and supersonic cases. Boundary conditions for complex and curved boundaries have also been implemented.

As a preliminary part of the thesis we developed finite-difference 2-D rectangular solver based on the MacCormack scheme [5] (Sec. 4.7.1). We, however, used central difference for the viscous term in the predictor and corrector steps of the scheme (Sec. 4.7.1). This formulation gave spurious oscillations, which was removed by adding artificial viscosity term. Then we developed the 2-D finite volume rectangular code using the same scheme (Sec. 4.7.3). In this for discretization of the viscous term we used the formulation of Chandrashekar [6]. A point to be noted was using this formulation which needed no artificial viscosity term, we did not get any spurious oscillations.

In the later part of the thesis, the discretization of viscous terms for momentum and energy equations was done, which were then implemented in the 3D Anupravaha solver (Sec. 4.2). The Least square cell method was used to find the gradients at the cell-center (Sec. 4.6.2).

Also, the viscous terms have been implemented using the AUSM+ scheme for the convective part of the governing equation. The code was validated for both MacCormack and AUSM+ formulation for the subsonic and supersonic cases.

At the end, we are presenting a standalone density-based solver catering for aerospace applications exclusively.

We list down following points which can be used for future work.

1. An Implicit version of the code can be developed. Also, a 2D version of the code for faster execution of two-dimensional cases is needed.
2. Acceleration Techniques(multigrid, residue smoothing) can be implimented to get the solution even more faster.

3. Turbulence models should be implemented to handle real aerospace applications.
4. A review of convergence criteria has to be performed. At present, our measure for steady-state has been based on post-processing results.
5. Preconditioning can be implemented in order to capture the flows which are Mach number($Ma < 0.3$).

Our future goal is to develop an unstructured version of *AnuPravaha* density based solver for all-speed flows.

Appendices

Appendix A

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.

$$\frac{\partial U}{\partial t} = -\frac{\partial U}{\partial x} - \frac{G}{y} + J \quad (\text{A.1})$$

where U is the solution vector, $U = \left[\rho \quad \rho u \quad \rho v \quad \rho(e + V^2/2) \right]$.

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

$$\begin{aligned} S_{i,j}^t = & C_x \frac{|p_{i+1,j}^t - 2p_{i,j}^t + p_{i-1,j}^t|}{p_{i+1,j}^t - 2p_{i,j}^t + p_{i-1,j}^t} (U_{i+1,j}^t - 2U_{i,j}^t + U_{i-1,j}^t) \\ & + C_y \frac{|p_{i,j+1}^t - 2p_{i,j}^t + p_{i,j-1}^t|}{p_{i,j+1}^t - 2p_{i,j}^t + p_{i,j-1}^t} (U_{i,j+1}^t - 2U_{i,j}^t + U_{i,j-1}^t) \end{aligned} \quad (\text{A.2})$$

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

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

$$\begin{aligned} \bar{S}_{i,j}^t = & C_x \frac{|\bar{p}_{i+1,j}^t - 2\bar{p}_{i,j}^t + \bar{p}_{i-1,j}^t|}{\bar{p}_{i+1,j}^t - 2\bar{p}_{i,j}^t + \bar{p}_{i-1,j}^t} (\bar{U}_{i+1,j}^t - 2\bar{U}_{i,j}^t + \bar{U}_{i-1,j}^t) \\ & + C_y \frac{|\bar{p}_{i,j+1}^t - 2\bar{p}_{i,j}^t + \bar{p}_{i,j-1}^t|}{\bar{p}_{i,j+1}^t - 2\bar{p}_{i,j}^t + \bar{p}_{i,j-1}^t} (\bar{U}_{i,j+1}^t - 2\bar{U}_{i,j}^t + \bar{U}_{i,j-1}^t) \end{aligned} \quad (\text{A.3})$$

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

The value of $S_{i,j}^t$ and $\bar{S}_{i,j}^t$ 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,

$$\bar{\rho}_{i,j}^{t+\Delta t} = \rho_{i,j}^t + \left(\frac{\partial \rho}{\partial t} \right)_{i,j}^t \Delta t + S_{i,j}^t \quad (\text{A.4})$$

On the corrector step,

$$\rho_{i,j}^{t+\Delta t} = \rho_{i,j}^t + \left(\frac{\partial \rho}{\partial t} \right)_a v \Delta t + \bar{S}_{i,j}^{t+\Delta t} \quad (\text{A.5})$$

References

- [1] C. Hirsch. Numerical computation of internal and external flows fundamentals of computational fluid dynamics. Elsevier, Amsterdam, 2007.
- [2] C. Hirsch. Numerical computation of internal and external flows vol 2. Wiley, Chichester [England]; New York, 1988.
- [3] J. Blazek. Computational fluid dynamics principles and applications. Elsevier, Amsterdam; San Diego, 2005.
- [4] D. L. Whitfield and J. M. Janus. Three-dimensional unsteady Euler equations solution using flux vector splitting 1984. 00121.
- [5] J. D. Anderson. Computational fluid dynamics: the basics with applications. McGraw-Hill, New York, 1995.
- [6] K. Chandrasekhar. Numerical Simulation of Turbulent Two-Phase Flows. Master Thesis, Indian Institute of Technology Hyderabad, India, 2009.
- [7] A. Assam. Development of a General-Purpose Compressible Flow AnuPravaha Based Solver. Master Thesis, Indian Institute of Technology Hyderabad, India, 2014.
- [8] M.-S. Liou. A sequel to AUSM, Part II: AUSM-up for all speeds. *Journal of Computational Physics* 214, (2006) 137–170.
- [9] M.-S. Liou and C. J. Steffen Jr. A new flux splitting scheme. *Journal of computational physics* 107, (1993) 23–39.
- [10] M.-S. Liou. Ten Years in the Making: AUSM-family. National Aeronautics and Space Administration, Glenn Research Center, 2001.
- [11] N. K. Narayan. Development of Compressible Flow Module In AnuPravaha. Master Thesis, Indian Institute of Technology Hyderabad, India, 2013.
- [12] B. Engquist and A. Majda. Radiation boundary conditions for acoustic and elastic wave calculations. *Communications on Pure and Applied Mathematics* 32, (1979) 313–357.

- [13] G. Hedstrom. Nonreflecting boundary conditions for nonlinear hyperbolic systems. *Journal of Computational Physics* 30, (1979) 222–237.
- [14] D. H. Rudy and J. C. Strikwerda. A nonreflecting outflow boundary condition for subsonic navier-stokes calculations. *Journal of Computational Physics* 36, (1980) 55–70. Cited by 0234.
- [15] C. B. Laney. Computational gasdynamics. Cambridge Univ. Press, Cambridge u.a., 1998.
- [16] A. H. Shapiro. The dynamics and thermodynamics of compressible fluid flow, Vol. 2. *Ronald, New York* 1157.
- [17] D. Drikakis and S. Tsangaris. On the solution of the compressible Navier-Stokes equations using improved flux vector splitting methods. *Applied mathematical modelling* 17, (1993) 282–297.