The Pennsylvania State University
The Graduate School
College of Engineering

DETERMINING OPTIMUM CONDITIONS FOR DUAL TIME STEPPING USING CONVERGENCE ACCELERATION TECHNIQUES IN COMPUTATIONAL AEROACOUSTICS

A Thesis in
Aerospace Engineering
by
Vikram Rout

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2015

The thesis of Vikram Rout was reviewed and approved* by the following:

Philip J. Morris
Boeing/A.D. Welliver Professor of Aerospace Engineering
Thesis Adviser

Kenneth S. Brentner
Professor of Aerospace Engineering

Michael M. Micci
Professor of Aerospace Engineering
Director of Graduate Studies

*Signatures are on file in the Graduate School.

Abstract

This thesis is concerned with the evaluation of Dual Time Stepping applied to unsteady compressible flow problems in aeroacoustics. Computational Aeroacoustics(CAA) involves many complex calculations. Thus, like any other computationally intense problems, the computational time and costs are of primary concern. The flow problem in the present thesis consists of the interaction between an acoustic pulse and a plane rigid boundary. The problem includes acoustic, entropy and vorticity waves. This problem, whose analytical solution is known, is used as a benchmark to assess the time marching schemes. The computational domain has a solid wall as the lower boundary, which involves the need for ghost points to apply the proper boundary conditions at the wall. Two types of boundary conditions, namely radiation and outflow boundary conditions, are applied to the waves exiting the domain. Low-dispersion, high-efficiency numerical schemes called Dispersion-Relationship-Preserving(DRP) schemes are used for highly accurate spatial discretization. Three different convergence acceleration techniques have been used in this thesis. These are Dual Time Stepping(DTS), Implicit Residual Smoothing(IRS) and Multigrid. Each of these convergence acceleration methods have been implemented individually, as well as in conjunction with each other to determine the most numerically accurate and cost efficient combination of these techniques.

Contents

List of Figures

List of Tables

Acknowledgment

I would like to express my sincere and deepest gratitude towards my advisor, Prof. Philip Morris, whose expertise, understanding, and patience added considerably to my graduate experience. He has been supportive since the very first day I started working on this topic. With his extensive knowledge and experience in the field of aeroacoustics, Dr. Morris has helped me throughout my research and during the process of writing this thesis. He has been a great inspiration for me, and his guidance has been paramount to the successful completion of this thesis.

I would like to thank all my friends here at Penn State, my roommates in particular: Racheet Matai, Abhishek Jain, Ankit Tiwari and Raghu Abhilash, who made my transition into this new cultural and working environment, a very smooth and memorable one. I would also like to thank my friend, Vinod Anantharaman, for his help and assistance with LaTeX and numerous other things.

Finally, I would like to thank the most important people in my life, my parents and elder brother, who have supported me unconditionally in all my ventures in life, and without whose love and encouragement none of this would have been possible.

Chapter 1

Introduction

Computational Aeroacoustics (CAA) is a relatively new field of research compared to Computational Fluid Dynamics. CAA problems are time dependent and involve high-frequency components while CFD problems are time independent or involve only low-frequency unsteadiness [1]. Some of the computational challenges in the field of Computational Aeroacoustics are listed here.

1. A majority of aeroacoustics problems involve a broad range of frequencies. Resolving these waves to distinctly capture high-frequency waves as well as lower frequency waves is a major stumbling block for accurate numerical results.

2. The amplitudes of acoustic waves are very small, by up to five or six orders of magnitude, compared to that of the mean flow. Thus any numerical scheme being used should have very high resolution to resolve both the mean flow and the acoustic waves. It should also contain very low numerical error so that the error doesn't get superimposed on the acoustic waves.

3. The size of a computational domain is limited. In CFD problems, the flow disturbances usually decay at a rapid rate as they move away from a body or their source and hence are very small and almost negligible at the boundaries of the computational domain. However, in CAA problems, acoustic waves decay much more slowly and they do reach the boundaries of the computational domain. To avoid the reflection of these waves back into the computational domain and thus cause numerical inaccuracy, different types of boundary conditions, namely radiation and outflow boundary conditions, have been developed. CFD problems do not generally require such boundary conditions.

4. CAA problems often involve largely varying length scales. The length scale

of an acoustic source is often smaller by order(s) of magnitude than the acoustic wavelength. The numerical methods developed must be able to handle these multiscale problems in different parts of the computational domain.

The above reasons highlight the important factors that warrant the need for separate numerical schemes for CAA problems from those used in traditional CFD.

The linearized Euler equations are the primary governing equations of all CAA problems. From wave propagation theory, the number of wave modes and their wave propagation characteristics are included in the dispersion relation of the governing equations[2].

In any problem solved computationally, the accuracy of the results, and the time and cost of the computation are some of the primary factors that are considered to determine if the solution is feasible. Obtaining highly accurate results close to the exact solution can easily increase the computational cost and time. The computational time, and consequently cost, can be reduced by using faster convergence methods and higher order schemes with greater stability. Some of these methods for faster convergence, namely Dual Time Stepping(DTS), Implicit Residual Smoothing(IRS) and Multigrid, have been implemented in this thesis.

Implementation of the standard CFD second-order schemes usually requires 18 to 25 grid points per wavelength for acceptable accuracy. Clearly, this cannot be used for practical problems due to the very high computational needs. This roadblock led to the development of large-stencil high-resolution CAA schemes that have the ability to provide sufficiently accurate results with the use of only six to seven grid points per wavelength. The difference in the amplitudes of the mean flow and the disturbances might lead to the suppression of the actual radiated sound by the inherent error of the numerical scheme in use. The development of high-resolution CAA schemes has helped to avoid this problem. Wave-number analysis, through the use of Fourier-Laplace transforms, has shown that the order of the scheme is not the primary factor in achieving highly accurate results. The primary component to obtain accurate results is the resolved bandwidth in the wave number space. In wave propagation problems, space and time are very closely related. A numerical scheme with the same dispersion relations as the original partial differential equations is called a dispersion-relationship-preserving(DRP) scheme. Use of such schemes ensures the accuracy of the results, while also preserving the acoustic, vorticity and entropy wave modes along with the original

characteristics of the partial differential equations. Wave number analysis enables the understanding of the characteristics of spurious waves in much greater detail.

Another important aspect of CAA is the development of numerical boundary conditions. The development and implementation of the boundary conditions is a major factor in any CAA problem. Time stepping is another important factor in CAA problems. This can be achieved in two ways: the single time step method and multistage time discretization. The multistage time discretization schemes include second and higher order Runge-Kutta schemes.

The dispersion-relationship-preserving schemes were first developed by Tam and Webb[2]. Using a consistent, convergent and stable high order finite difference scheme does not necessarily guarantee an accurate solution for time dependent problems, which include a majority of acoustics problems. The linearized Euler equations support three types of waves, which are acoustic, entropy and vorticity waves. The acoustic waves are isotropic, non-dispersive and non-dissipative in nature and they travel at the speed of sound. The entropy and vorticity waves are non-dispersive and non-dissipative like the acoustic waves, but they are also highly directional. They propagate in the direction of the mean flow at the speed of the mean flow.

To preserve the dispersion relation of the linearized Euler equations, a finite difference approximation is needed whose Fourier-Laplace transform in space or time is the same as the original partial derivative from the Euler equations. Spatial finite difference schemes vary slightly from the finite difference scheme associated with time. The use of high order finite difference schemes inevitably introduces extraneous or irrelevant numerical solutions. It is of primary concern that these extraneous solutions be damped. This is achieved with the appropriate selection of the computational parameters. Some of the methods used to damp these solutions are artificial dissipation and filtering. Various filtering schemes have been used in the present thesis.

Various methods have been used to develop the radiation and outflow boundary conditions[3;4]. The boundary conditions used in this thesis were developed by Tam and Webb[2] by finding the far field asymptotic solutions to the linearized Euler equations.

The primary contribution of the present thesis is a systematic assessment of the accuracy and efficiency of time-marching schemes in CAA. Emphasis is placed on

the Dual Time Stepping method as this alleviates some of the time step constraints imposed on explicit time marching schemes.
The next chapter describes the numerical methods used in the thesis. These include the convergence acceleration methods. The problem description and the associated initial and boundary conditions are also described. This is followed by a description of the results. Finally, the conclusions drawn from the calculations are given.

Chapter 2

Numerical Methods

This chapter describes the numerical methods used in the current thesis. The advantages and implementation of the Dual Time Stepping(DTS) scheme are described first. This is followed by a description and the formulation of the multigrid technique. The details of another convergence accelerating algorithm called Implicit Residual Smoothing(IRS) follows the DTS section. The actual physical problem is described in detail in the subsequent section along with the necessary initial and boundary conditions required to solve the problem on a customized computational grid. The final section of this chapter delves into the need and application of different filtering techniques used in the solution of the described problem.

2.1 Dual Time Stepping

The concept of Dual Time Stepping was first introduced by Jameson [5]. It makes use of a fictitious time derivative in addition to the physical time derivative: hence the name Dual Time Stepping(DTS). For an unsteady flow, the use of an explicit time marching scheme requires the use of a very small time step to obtain a stable solution. This time step is less than the time step required for acceptable accuracy. Using a very small time step, in turn, results in the use of a large number of time steps. Implicit time marching schemes enable the use of much larger time steps. Each time step involves more computation compared to the explicit method but the overall reduction in computation time can be significant.

As an introduction to DTS, consider the one-dimensional wave equation in the following form:

$$\frac{\partial \mathbf{U}}{\partial t} = Res(\mathbf{U}). \tag{2.1}$$

The purpose of the Dual Time Stepping scheme is to convert an unsteady problem in the physical time domain into a steady state problem in the fictitious time domain. A fictitious time derivative term can be introduced such that,

$$\frac{\partial \mathbf{U}}{\partial \tau} = -\frac{\partial \mathbf{U}}{\partial t} + Res(\mathbf{U}) = Res^*(\mathbf{U}). \tag{2.2}$$

When $Res^*(U)$ reduces to zero, the modified governing equations effectively return to the original equations. A three point backward difference scheme is used to represent the time derivative in the physical domain as such a method is unconditionally stable. This gives,

$$\left[\frac{\partial \mathbf{U}}{\partial \tau}\right]^{n+1} = -\frac{\mathbf{U}^{n,m+1} - 4\mathbf{U}^{m} + 3\mathbf{U}^{m-1}}{2\Delta t} + Res(\mathbf{U}^{n,m+1}) = Res^*(\mathbf{U}^{n,m+1}) \tag{2.3}$$

Here, m represents the physical time step stage and n represents the fictitious time step stage. t and τ denote the physical and fictitious times respectively. $\mathbf{U}^{n,m+1}$ is the predicted value of the variable at the next time step and is obtained by a using a Taylor's series approximation,

$$\mathbf{U}^{n,m+1} = \mathbf{U}^{n,m} + \frac{\mathbf{U}^{n,m-2} - 4\mathbf{U}^{n,m-1} + 3\mathbf{U}^{n,m}}{2} \tag{2.4}$$

During every physical time step, the solution of the governing equations are iterated in the fictitious time domain using a second order Runge-Kutta scheme until $Res^*(U) = 0$. When this condition is reached, the flow solution is in a converged state in the fictitious time domain and the converged value in the fictitious domain $U^{(n+1,m+1)}$ is the value of the variable at the beginning of the next physical time step. The implementation of the Runge-Kutta scheme for one fictitious time cycle is shown below. It is implemented until $Res^*(U) \approx 0$ is achieved.

$$\mathbf{U}^{(1)} = \mathbf{U}^{(n)} \tag{2.5}$$

$$\mathbf{U}^{(2)} = \mathbf{U}^{(n)} - \Delta\tau Res^*(\mathbf{U}^{(1)}) \tag{2.6}$$

$$\mathbf{U}^{(3)} = \mathbf{U}^{(n)} - \Delta\tau \frac{(Res^*(\mathbf{U}^{(1)}) + Res^*(\mathbf{U}^{(2)}))}{2} \tag{2.7}$$

$$\mathbf{U}^{(n+1)} = \mathbf{U}^{(3)} \tag{2.8}$$

The solution in the fictitious time domain is not time-accurate, i.e, the final result

does not depend on the amount of physical time passed. Due to this reason, other methods to accelerate the convergence of the solution can be implemented. Two of the methods that have been used here are Implicit Residual Smoothing and the Multigrid method.

The selection of the physical time step is another important aspect of DTS. Choosing a time step that is too large can lead to convergence and stability problems. In the sample flow problem used in this thesis, a ratio of 4 has been used for the time step with DTS and time step without DTS. This implies that the ratio of the CFL numbers with and without DTS is 4. The ratio has only been chosen for this particular problem as an example. In general practice, it is quite common to use a CFL number ratio of up to 100 at the grid locations which have a very small grid size for accurate solutions. In DTS, the speed of convergence depends to a large extent on the size of the fictitious time step. The fictitious time step is chosen such that the CFL number is constant throughout the domain in the fictitious time scale. This, in turn, requires the use of a different time step at each point on the grid. A constant CFL number and varying time step in DTS is in direct contrast to solving the problem without DTS which requires a constant time step over the entire domain and varying CFL number at each grid point.

Consider a one-dimensional hyperbolic system in the ξ direction. The equation describing this system can be represented in a diagonalized form as [6]

$$\frac{\partial q}{\partial \tau} + \Lambda \frac{\partial q}{\partial \xi} = S \tag{2.9}$$

where, Λ is a diagonal matrix with five eigenvalues.

$$\lambda_\xi^1 = \lambda_\xi^2 = \lambda_\xi^3 = \xi_x u + \xi_y v + \xi_z w = \overline{U} \tag{2.10}$$

$$\lambda_\xi^4 = \overline{U} + c(\xi_x^2 + \xi_y^2 + \xi_z^2)^{1/2} \tag{2.11}$$

$$\lambda_\xi^5 = \overline{U} - c(\xi_x^2 + \xi_y^2 + \xi_z^2)^{1/2} \tag{2.12}$$

The non-dimensional formulas are written as

$$\lambda_\xi^1 = \lambda_\xi^2 = \lambda_\xi^3 = U \tag{2.13}$$

$$\lambda_\xi^4 = U + \frac{1}{M_r}\xi_x^2 + \xi_y^2 + \xi_z^2)^{1/2} \tag{2.14}$$

$$\lambda_\xi^5 = U + \frac{1}{M_r}(\xi_x^2 + \xi_y^2 + \xi_z^2)^{1/2} \tag{2.15}$$

where $\overline{U}$ is the contra-variant velocity component. $c = \sqrt{\gamma R T_r}$, $U = \overline{U}/U_r$, $M_r = U_r/c$. and $\xi_x = \frac{\partial \xi}{\partial x}$, $\xi_y = \frac{\partial \xi}{\partial y}$, $\xi_z = \frac{\partial \xi}{\partial z}$.
In the grid transformation used for this flow problem, both ξ_y and ξ_z are equal to zero. According to the CFL criteria,

$$\left| \lambda^{\bar{l}} \frac{\Delta t}{\Delta \xi} \right| \leq 1 \tag{2.16}$$

where $\Delta\xi$ is the grid size of the uniform grid.

2.2 Multigrid

Multigrid methods are used to accelerate the convergence of flow problems by using one or a higher number of coarser grids in addition to the original grid. The primary advantage of this method is to remove the low-frequency error that cannot be easily resolved on the finer grids. Upon transferring the flow problem to a coarser grid, these low-frequency waves become a part of a higher frequency range and hence, can be resolved with greater ease since the relaxation stage is more effective in this case. This method was first introduced by Brandt [7]. The simplest method of applying the multigrid scheme is to remove every alternate grid point in each direction of the grid. To obtain an efficient coarse grid from the fine grid, it is advisable to use $2^n + 1$ grid points in each direction, where n is a positive integer. This enables a much smoother transition from the fine to coarse grid compared to a grid with a different number of grid points in each direction. An example depicting the advantage of using a grid with $2^n + 1$ points is shown in Figures 2.1 and 2.2.

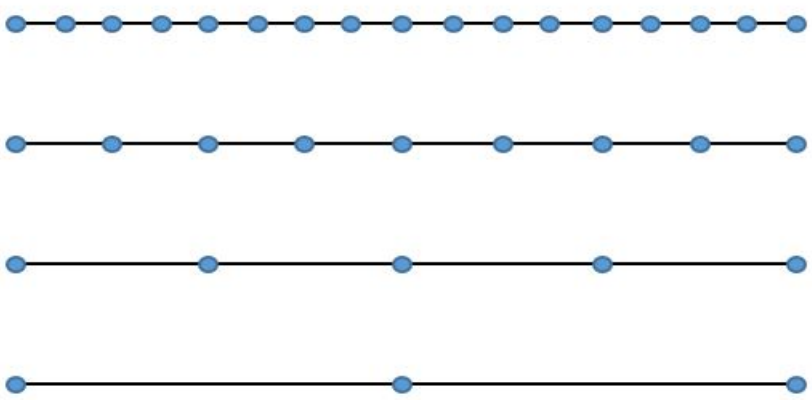


Figure 2.1: Multigrid scheme applied to a domain with $2^n + 1(17)$ points

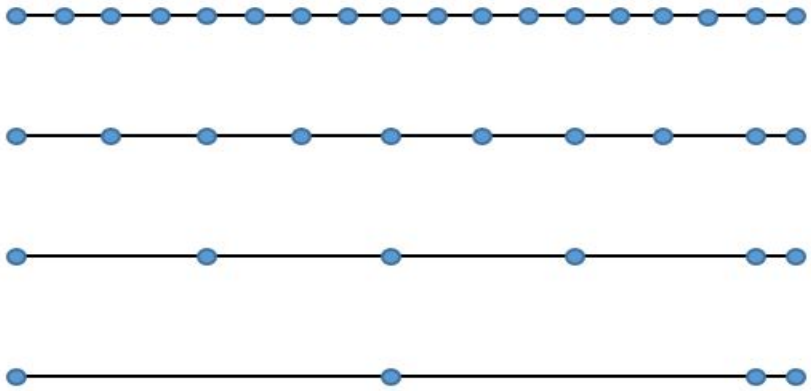


Figure 2.2: Multigrid scheme applied to a domain with arbitrary number(18) of points

The step-by-step process for a multigrid cycle is given below. The first index in the parentheses represents the time step and the second index represents the grid level, with 0 being the finest grid and the coarseness increases with increase in the index number.

1. Save the value of the variable at the end of the physical time step in $\mathbf{U}(0,0)$.
2. Store the residual of the variable values $\mathbf{U}(0,0)$ in $Res(0,0)$.
3. Solve the equations with $\mathbf{U}(0,0)$ and $Res(0,0)$ at the next time step by the smoothing process(2^{nd} order Runge-Kutta) to obtain the variable values $\mathbf{U}(1,0)$.
4. Find the residual values from $\mathbf{U}(1,0)$ and store them in $Res(1,0)$.
5. Restrict the variables $\mathbf{U}(0,0)$, $\mathbf{U}(1,0)$ onto $\mathbf{U}(0,1)$ and $\mathbf{U}(1,1)$ respectively.
6. Restrict the residual values from $Res(1,0)$ onto $Res(1,1)$.
7. Find the residual value on the coarse grid using $\mathbf{U}(1,1)$ and store it in $Res(2,1)$.
8. Calculate a forcing function (fine-grid correction).

$$F(\mathbf{U}) = Res(1,1) - Res(2,1) \tag{2.17}$$

9. Update the coarse grid residual value by adding the forcing function.

$$Res^*(2,1) = Res(2,1) + F(\mathbf{U}) \tag{2.18}$$

10. Solve the flow problem on the coarse grid using the updated residual $Res^*(2,1)$ and taking a time step forwards in the fictitious time domain. Store the final variable values in $\mathbf{U}(2,1)$.

11. Prolong the difference between the solution on the coarse grid and the initial restricted value from the fine grid onto the coarse grid.

$$PROLONG(\mathbf{U}(2,1) - \mathbf{U}(1,1)) \tag{2.19}$$

12. Make the correction to the value of the variable on the fine grid. This completes one multigrid cycle.

$$U(2,0) = U(1,0) + PROLONG(\mathbf{U}(2,1) - \mathbf{U}(1,1)) \tag{2.20}$$

These steps are described in greater detail below.
The purpose of the forcing function is to ensure that the fine-grid accuracy is also maintained on the coarse grid. If the number of smoothing passes or time steps taken on the coarse grid is just one, then the final residual is equal to the restricted residual value from the fine grid as seen from equations 2.17 and 2.18.

There are a number of different multigrid cycles that have been developed. These include sawtooth cycles, V-cycles and W-cycles. Choice of the multigrid cycle depends on the computational resources and time available.

In a sawtooth cycle, the flow progresses in time, i.e. a time step is taken only during the restriction stage on each grid. During the prolongation stage from a coarse grid to a subsequent finer grid, no time steps are taken and the solution is just transferred from one grid to another. After the correction has been applied from a coarse grid to a fine grid, no further time steps are taken. A three-level sawtooth cycle is shown in Figure 2.3.

In a V-cycle, time steps are taken during the restriction stage similar to the sawtooth cycle. But, and in addition to this, the flow also marches in time during the prolongation stage from a coarse grid to a finer grid. Time steps are taken after every correction from a coarser to a finer grid. A three-level V-cycle is shown in Figure 2.4 with each component of the schematic having the same function as shown in Figure 2.3.

In a W-cycle multigrid scheme, the coarse grid corrections are smoothed before transferring the results to the finer grid. Theoretically, this is the most time-

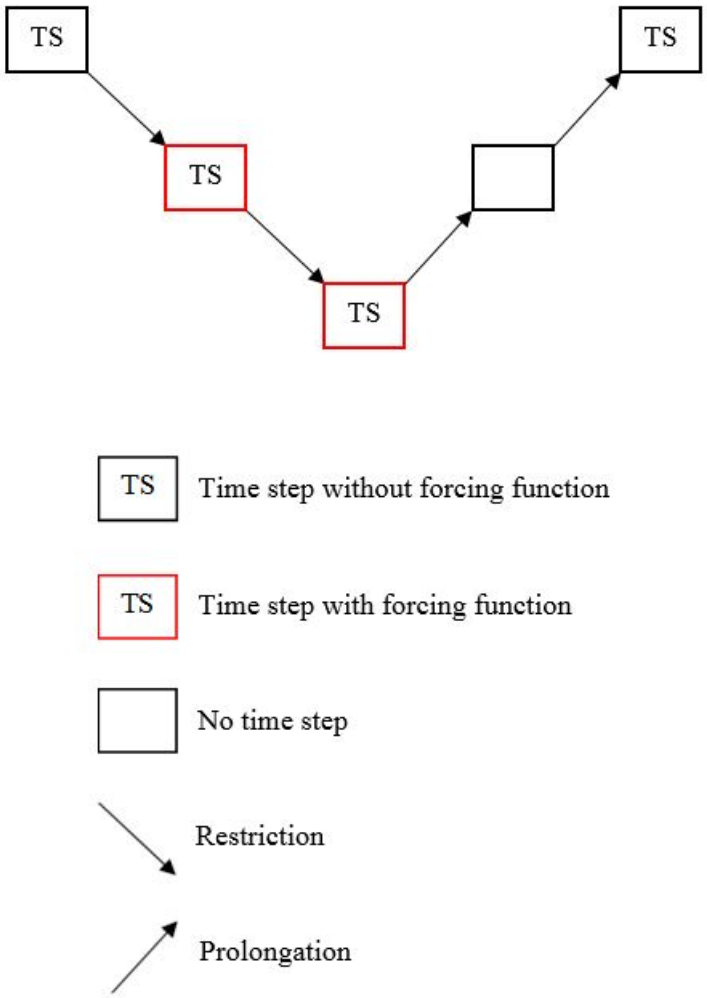


Figure 2.3: Three-level Sawtooth Multigrid Algorithm

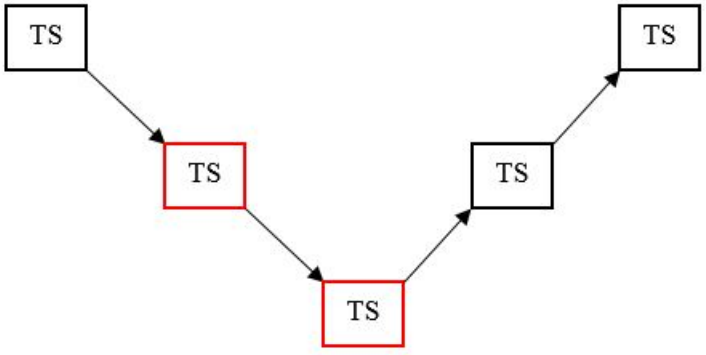


Figure 2.4: Three-level V-cycle Multigrid Algorithm

efficient cycle since a greater number of time steps are taken on the coarser grid compared to the finer grid, thus reducing the computational time. Figure 2.5 represents the schematic of a three-level W-cycle and each part of the schematic

serves the same purpose as shown in Figure 2.3.

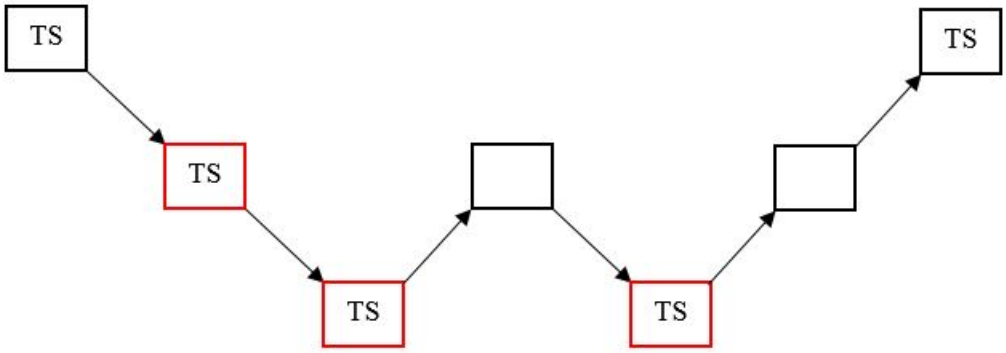


Figure 2.5: Three-level W-cycle Multigrid Algorithm

2.2.1 Restriction

Restriction is the process of transferring the information from a fine grid to a subsequent coarser grid. This can be done in a number of ways like “direct injection", “half-weighting" and “full-weighting". In “direct injection", the data is directly transferred from a grid point on the fine grid to its corresponding coincident grid point on the coarse grid.

In the last two methods, when the data from a grid point is transferred from a fine grid to a coarser grid, the neighboring grid points also contribute to the values of the variables on the coarse grid. The restriction method is classified as “half-weighting" or “full-weighting" depending on the contributions of each neighboring grid point to the primary grid point. The full-weighting operator is given by:

$$\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \tag{2.21}$$

In the above array, the $(2, 2)$ location is the grid point on the finer grid which is transferred to the coarser grid. The adjacent array locations indicate the weights of the values of the variable at their corresponding grid locations on the fine grid. Only the grid point at $(2, 2)$ is transferred from the fine grid to the coarse grid and all other grid points are eliminated. An identical approach is used to understand

the half-weighting operator which is given by

$$\begin{bmatrix} 0 & \frac{1}{8} & 0 \\ \frac{1}{8} & \frac{1}{2} & \frac{1}{8} \\ 0 & \frac{1}{8} & 0 \end{bmatrix} \tag{2.22}$$

2.2.2 Prolongation

Prolongation is the process of transferring the data from the coarser grid back to the finer grid. The most commonly used two-dimensional prolongation operator is given below:

$$\begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} \tag{2.23}$$

Different weights are used for different grid points depending on their position on the fine grid compared to the coarse grid.

1. Type I grid points are present on both the fine and coarse grids. The information can be transferred directly from the coarse to fine grid without using any weighting factors. This is similar to “direct-injection" during the restriction stage.

2. Type II grid points lie between and are collinear to two Type I grid points. The variable value at these points is found by taking a simple average of the variable values at the two Type I grid points.

3. Type II grid points lie at the centre of four Type I grid points. This classification of grid points is shown in Figure 2.6.

Figure 2.6: Classification of grid points for Multigrid scheme

The above array represents the weights assigned to the nodes at and adjacent to the central array location of $(2, 2)$. The array location $(2, 2)$ indicates a Type I

grid point and the different array locations indicate the weights of the $(2, 2)$ grid point at the adjacent grid points on the fine grid. Figure 2.7 depicts the different weights assigned by the Type I grid points to different grid points. The bold brown lines indicate a weighting factor of 0.5 whereas, the thin black lines have a weighting factor of 0.25.

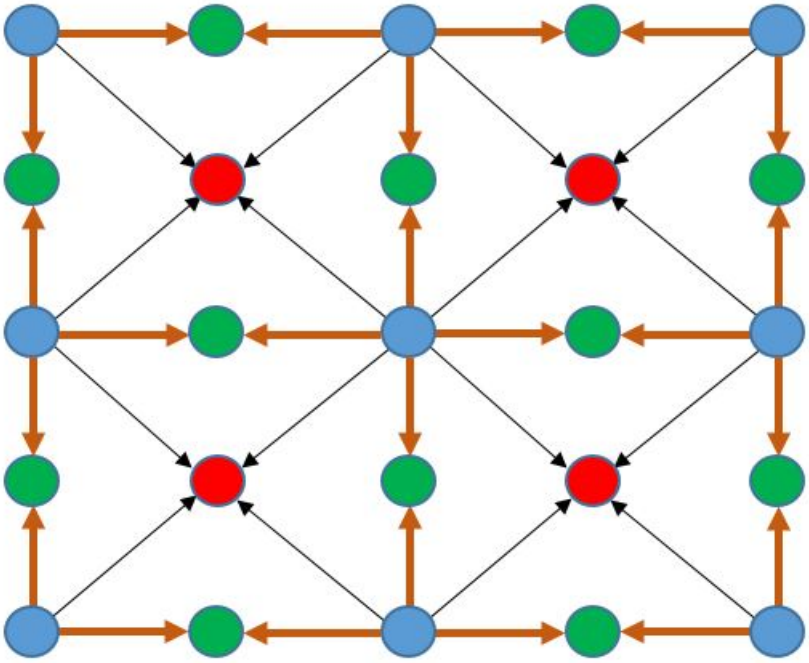


Figure 2.7: Prolongation

2.3 Implicit Residual Smoothing

Another method used to accelerate the convergence of steady-state flow solutions is Implicit Residual Smoothing(IRS). This was also introduced by Jameson [8]. It is implemented at each stage of the time-stepping scheme. IRS artificially extends the stability region of the basic explicit time-stepping scheme and, thus enables the use of a higher Courant-Friedrichs-Lewy(CFL) number [9]. IRS can also improve the regular damping properties of the basic scheme. This is of great advantage in the Multigrid scheme where the low frequencies are transferred from finer to coarser grids. The residual smoothing is applied in both x and y directions(in a two-dimensional problem), with the smoothed residuals after the first pass replacing the unsmoothed residuals for the second pass. Central Implicit Residual Smoothing(CIRS) is applied implicitly to the flow problem.

$$-\epsilon Res_{j-1}^{n+1} + (1 + 2\epsilon) Res_j^{n+1} - \epsilon Res_{j+1}^{n+1} = Res_j^n \tag{2.24}$$

where ϵ is the smoothing coefficient of the IRS scheme. $j-1$, j and $j+1$ are the grid locations. The superscript n indicates the residual value before smoothing and $n+1$ superscript denotes the smoothed residual value. For a second-order central spatial differencing scheme, it can be shown that the maximum permissible ratio of the CFL numbers of the smoothed(σ) to the unsmoothed(σ^*) scheme is given by[6]

$$\frac{\sigma}{\sigma^*} = \sqrt{1+4\epsilon} \tag{2.25}$$

Consider an example using the first order wave equation [6;10]

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{2.26}$$

$$\frac{\partial u}{\partial t} = Res(u) \tag{2.27}$$

If the second order time derivative of the flux term is taken, then a pseudo-residual can be represented by:

$$\frac{\partial^2}{\partial t^2}(-a\frac{\partial u}{\partial x}) = \widetilde{Res}_j(u) = -\frac{a}{\Delta x}\delta^2 u_j \tag{2.28}$$

$$\widetilde{Res}_j(u) = -\frac{a}{\Delta x}(u_{j+1} - 2u_j + u_{j-1}) \tag{2.29}$$

The indices have been removed to show that the following procedure can be applied at any grid point in the domain. Using a predicted change in the pseudo-residual as a correction, the residual at the n^{th} step can be corrected by:

$$Res^{n*} = Res^n + (\widehat{Res} - \widetilde{Res^n}) \tag{2.30}$$

where the pseudo-residual $\widehat{Res}$ is predicted implicitly:

$$\widehat{Res} = (1-\alpha)\widetilde{Res^n} + \alpha\widetilde{Res}^{n+1} \qquad where, 0 \leq \alpha \leq 1 \tag{2.31}$$

α is a weighting factor and n and $n+1$ are the current and next time steps respectively.
Then,

$$Res^{n*} = Res^n + \alpha(\widetilde{Res}^{n+1} - \widetilde{Res}^n) \tag{2.32}$$

Using Taylor's series, the pseudo-residual at the next time-step is determined by:

$$\widetilde{Res}^{n+1} = \widetilde{Res}^n + \left(\frac{\partial \widetilde{Res}}{\partial u}\Delta u\right)^n \tag{2.33}$$

Substituting the Taylor series expansion into Res^{n*},

$$Res^{n*} = Res^n + \alpha \left(\frac{\partial \widetilde{Res}}{\partial u} \Delta u \right)^n \tag{2.34}$$

Now substituting this in the discretized form of the wave equation gives,

$$\left[\left(\frac{1}{\Delta t} - \alpha \frac{\partial \widetilde{Res}}{\partial u} \right) \Delta u \right]^n = Res^n \tag{2.35}$$

From equations 2.28 and 2.29, it can be seen that the differential term in 2.35 can be replaced by

$$\frac{\partial \widetilde{Res}}{\partial u} = -\frac{a}{\Delta x} \delta^2 \tag{2.36}$$

After the smoothing process, the residual becomes $\overline{Res} = \Delta u / \Delta t$. Then equation 2.35 reduces to

$$(1 - \epsilon \delta^2)(\overline{Res}) = Res \tag{2.37}$$

Upon expansion at a grid point i, equation 2.37 yields equation 2.24.

$$-\epsilon Res_{j-1}^{n+1} + (1 + 2\epsilon) Res_j^{n+1} - \epsilon Res_{j+1}^{n+1} = Res_j^n \tag{2.38}$$

and $\epsilon = \alpha a \Delta t / \Delta x$. Also, $a \Delta t / \Delta x$ is the CFL number. All the residuals become zero when the solution converges and thus the residual smoothing does not affect the accuracy of the solution. A restriction on the value of the smoothing coefficient ϵ is given by[10]

$$\epsilon > \frac{1}{3} \left(\frac{1}{1.158} \left(\frac{\sigma^* / \sigma}{1.6227} + 0.1 \right) \right)^2 \tag{2.39}$$

2.4 Computational Grid

The geometric grid used is created in such a way that there is much greater resolution near the boundaries and the center of the grid compared to the outer boundaries because of higher frequency disturbances being present in these parts of the grid. Thus, these high frequency waves have to be resolved as much as possible. The transformation variables are as follows

$$A(y) = \frac{\beta + (1 - \frac{y}{h})}{\beta - (1 - \frac{y}{h})} \tag{2.40}$$

$$B = \frac{\beta + 1}{\beta - 1} \tag{2.41}$$

$$C(x) = \frac{\beta + (1 - \frac{x}{l})}{\beta - (1 - \frac{x}{l})} \tag{2.42}$$

$$\eta = 1 - \frac{ln[A(y)]}{ln[B]} \tag{2.43}$$

$$\xi = 1 - \frac{ln[C(x)]}{ln[B]} \tag{2.44}$$

where β is a stretching parameter $1 < \beta < \infty$. ξ and η represent the coordinate system in the computational domain and are functions of x and y respectively. β is chosen such that the largest grid spacing on the grid is approximately one.
The computational grid used for the present flow problem is shown in Figure 2.8 with magnified versions of regions I, II and III shown in Figures 2.9, 2.10 and 2.11 respectively.
The non-uniform grid is then transformed into a uniform grid in the computation domain and the following substitutions are used to update the governing equations:

$$\frac{\partial \mathbf{U}}{\partial x} = \frac{\partial \mathbf{U}}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial \mathbf{U}}{\partial \eta}\frac{\partial \eta}{\partial x} = \frac{\partial \mathbf{U}}{\partial \xi}\xi_x \tag{2.45}$$

$$\frac{\partial \mathbf{U}}{\partial y} = \frac{\partial \mathbf{U}}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial \mathbf{U}}{\partial \eta}\frac{\partial \eta}{\partial y} = \frac{\partial \mathbf{U}}{\partial \eta}\eta_y \tag{2.46}$$

where,

$$\xi_x = \frac{\partial \xi}{\partial x} = \frac{1}{l\left(\beta^2 - \left(1 - \frac{x}{l}\right)^2\right)}\frac{2\beta}{ln[B]} \tag{2.47}$$

$$\eta_y = \frac{\partial \eta}{\partial y} = \frac{1}{h\left(\beta^2 - \left(1 - \frac{y}{h}\right)^2\right)}\frac{2\beta}{ln[B]} \tag{2.48}$$

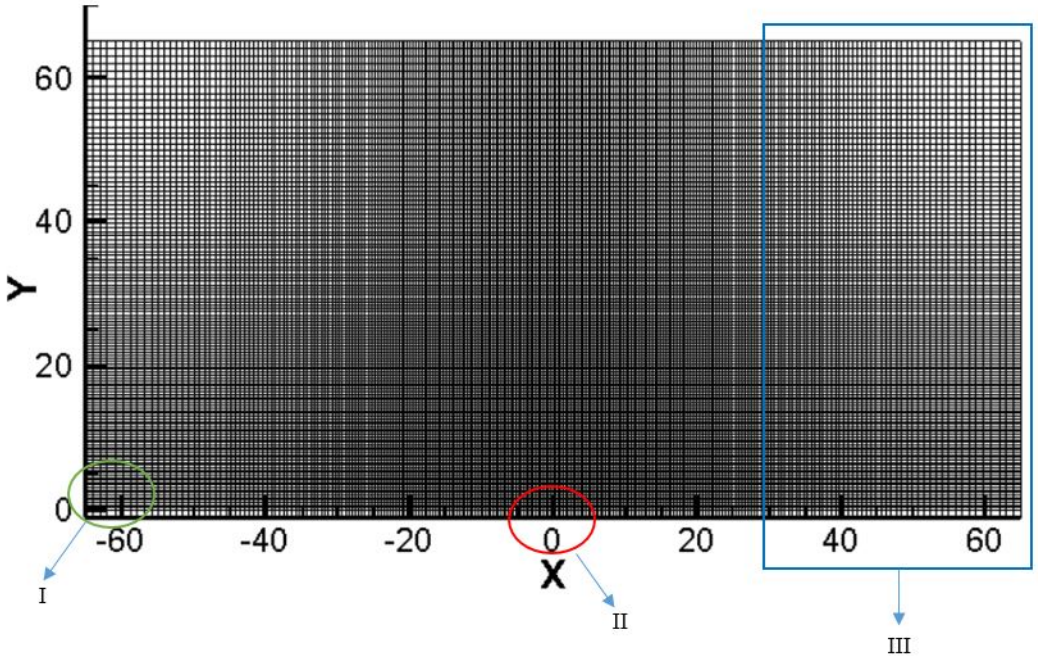


Figure 2.8: Computational grid

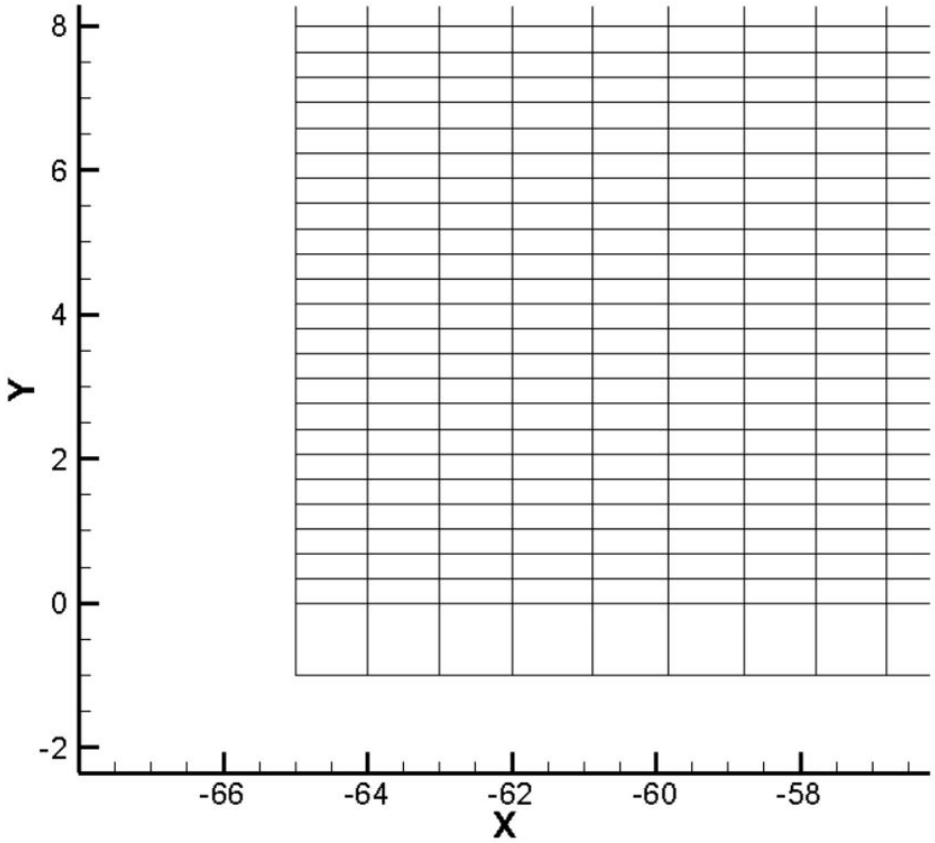


Figure 2.9: Magnified region I from Figure 2.8 showing the left boundary of the domain

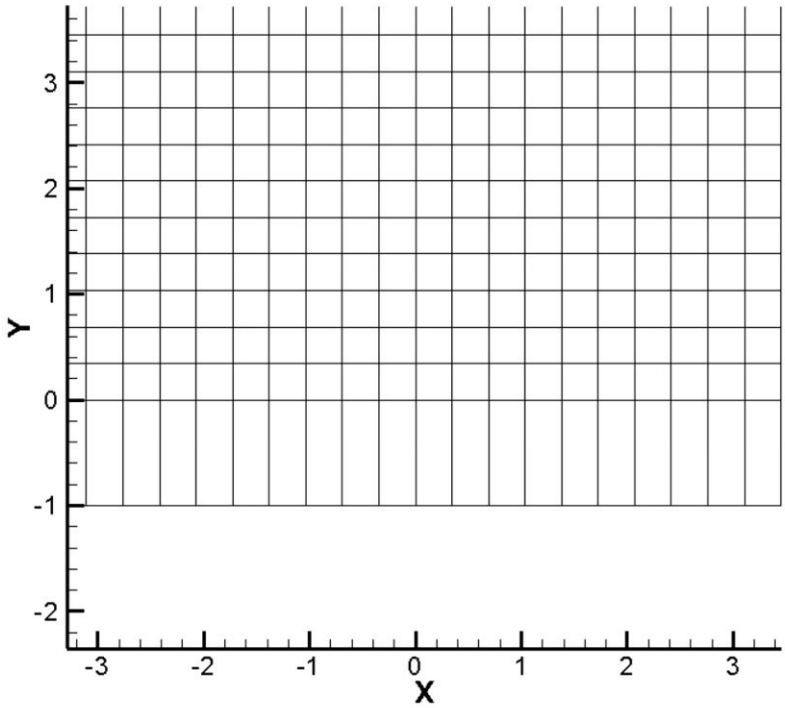


Figure 2.10: Magnified region II from Figure 2.8 showing the clustering of grid points near the centre of the domain

2.5 Problem Description

Consider a rectangular domain with a length to breadth ratio of 2:1. The lower side of the domain is a solid non-permeable wall. The other three sides of the domain are open and allow the waves to exit. A magnified version of the solid wall region from Figure 2.8 is shown in Figure 2.12.

Consider a uniform flow in the x-direction with a velocity u_0, density ρ_0 and pressure p_0 superimposed with small amplitude disturbances in the given domain. The disturbances include acoustic, vorticity and entropy waves from sources in the domain. It is assumed that the boundaries of the computational domain are far enough away from these sources so that they don't interfere with the initial propagation of the waves. Since the boundaries of the domain are considered to be very far from the sources, we can find the solutions at the boundaries by finding the asymptotic solution to the discretized form of the governing equations [2]. The governing linearized Euler equations are given by

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} = \mathbf{H} \tag{2.49}$$

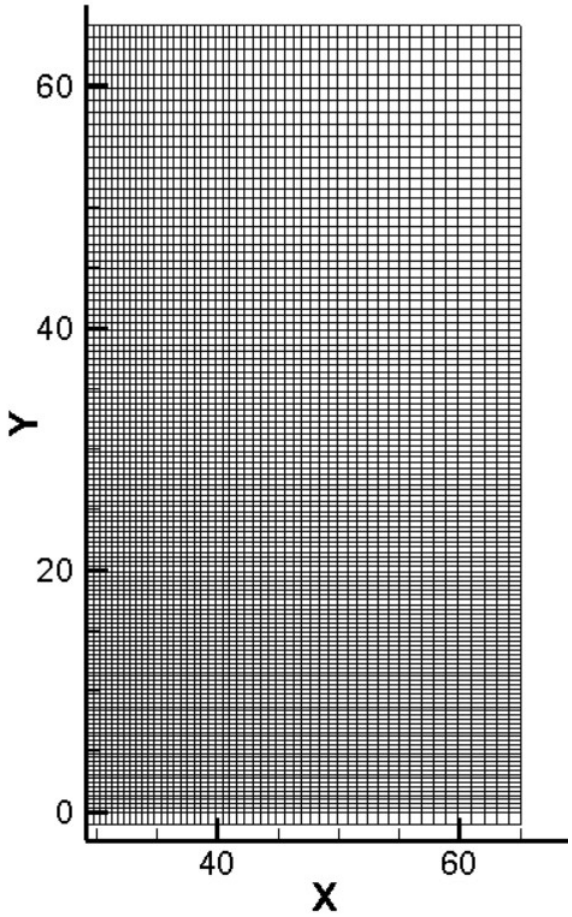


Figure 2.11: Magnified region III from Figure 2.8 showing the increasing grid spacing from the wall towards the boundary

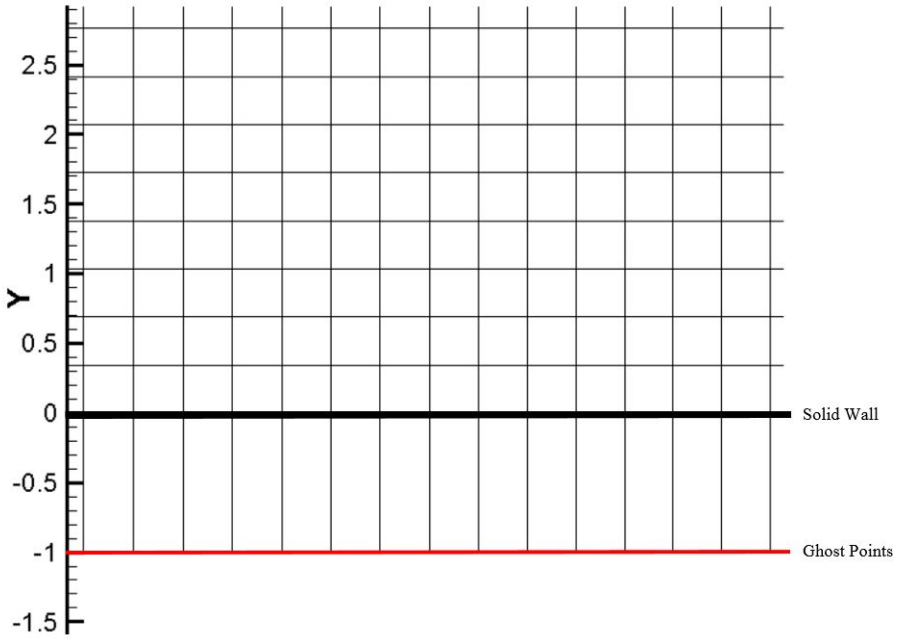


Figure 2.12: Solid Wall and Ghost Points in Computational Grid

$$\mathbf{U} = \begin{bmatrix} \rho \\ u \\ v \\ p \end{bmatrix} \tag{2.50}$$

$$\mathbf{E} = \begin{bmatrix} \rho_0 u + \rho u_0 \\ u_0 u + p/\rho_0 \\ u_0 v \\ u_0 p + \gamma p_0 u \end{bmatrix} \tag{2.51}$$

$$\mathbf{F} = \begin{bmatrix} \rho_0 v \\ 0 \\ p/\rho_0 \\ \gamma p_0 v \end{bmatrix} \tag{2.52}$$

The term on the right hand side of the equation is a nonhomogeneous term $\mathbf{H}$ and it represents the distributed unsteady sources. Since the problem being handled in the present thesis is an initial value problem, the right hand side is set to 0.

2.6 Initial Conditions

The small disturbances that are superimposed on the uniform flow are introduced as entropy, vorticity and acoustic waves. Applying the Fourier-Laplace transform to the linearized Euler equations in 2.49, the equation for $\tilde{\mathbf{U}}$ is given by,

$$\mathbf{A}\tilde{\mathbf{U}} = \tilde{\mathbf{G}} \tag{2.53}$$

where,

$$\mathbf{A} = \begin{bmatrix} \omega - \alpha u_0 & -\rho_0\alpha & -\rho_0\beta & 0 \\ 0 & \omega - \alpha u_0 & 0 & -\alpha/\rho_0 \\ 0 & 0 & \omega - \alpha u_0 & -\beta/\rho_0 \\ 0 & -\gamma p_0\alpha & -\gamma p_0\beta & \omega - \alpha u_0 \end{bmatrix} \tag{2.54}$$

and $\tilde{\mathbf{G}} = i(\tilde{\mathbf{H}}+\tilde{\mathbf{U}}_{initial})/2\pi$ is the sum of the transforms of the initial conditions and any source terms. α, β and ω are the variables in the Fourier-Laplace transform corresponding to x, y and t coordinates respectively. The dispersion relations of the waves[11] are found by determining the eigenvalues of $\mathbf{A}$. The eigenvalues are:

$$\lambda_1 = \lambda_2 = (\omega - \alpha u_0) \tag{2.55}$$

$$\lambda_3 = (\omega - \alpha u_0) + a_0(\alpha^2 + \beta^2)^{0.5} \tag{2.56}$$

$$\lambda_4 = (\omega - \alpha u_0) - a_0(\alpha^2 + \beta^2)^{0.5} \tag{2.57}$$

2.6.1 The Entropy Wave

An entropy wave consists of only density fluctuations and does not have any disturbance in pressure or velocity. The entropy wave solution is obtained by taking an inverse Fourier-Laplace transform. This gives,

$$\rho(x, y, t) = \int_\Gamma \int \int_{-\infty}^{\infty} \frac{C_1}{(\omega - \alpha u_0)} e^{i(\alpha x + \beta y - \omega t)} d\alpha d\beta d\omega \tag{2.58}$$

where, C_1 is an element of the coefficient vector, which is described in greater detail in Appendix A. Γ is a line parallel to the real axis in the complex ω-plane above all poles and singularities of the integrand.

The dispersion relation is obtained by finding a pole of the integrand, which involves the zeroes of the denominator. This is equivalent to equating the eigenvalue to zero.

$$\lambda_1 = (\omega - \alpha u_0) = 0 \tag{2.59}$$

The asymptotic solution can be simplified as:

$$\rho(x, y, t) = \begin{cases} \chi(x - u_0 t, y), & x \to \infty \\ 0, & x \to -\infty \end{cases} \tag{2.60}$$

where,

$$\chi(x - u_0 t, y) = \int_\Gamma \int_{-\infty}^{\infty} \frac{2\pi i C_1}{-u_0} e^{i(x/u_0 - t)\omega + i\beta y} d\beta d\omega \tag{2.61}$$

With an initial Gaussian distribution, the solution in a stationary coordinate system is given by:

$$p = u = v = 0 \tag{2.62}$$

$$\rho = \epsilon_2 e^{-\alpha_2[(x - Mt)^2 + y^2]} \tag{2.63}$$

where, M is the Mach number of the mean flow, ϵ_2 is the amplitude of the entropy pulse which is taken to be 0.001 and $\alpha_2 = ln2/b^2$, where b is the half-width of the entropy pulse and is set equal to 5.0.

2.6.2 The Acoustic Wave

The acoustic wave consists of pressure, density and velocity fluctuations. The dispersion relation is given by:

$$\lambda_3\lambda_4 = (\omega - \alpha u_0)^2 - a_0^2(\alpha^2 + \beta^2) = 0 \tag{2.64}$$

The asymptotic solution($r \to \infty$) in polar coordinates(r and θ) is given by[2]

$$\begin{bmatrix} \rho \\ u \\ v \\ p \end{bmatrix} = \frac{F(r/V(\theta) - t, \theta)}{r^{1/2}} \begin{bmatrix} \frac{1}{a_0^2} \\ \frac{\widehat{u}\theta}{\rho_0 a_0} \\ \frac{\widehat{v}\theta}{\rho_0 a_0} \\ 1 \end{bmatrix} + O(r^{-3/2}) \tag{2.65}$$

where,

$$V(\theta) = u_0 cos\theta + a_0(1 - M^2 sin^2\theta)^{1/2} \tag{2.66}$$

$$M = u_0/a_0 \tag{2.67}$$

$$\widehat{u}(\theta) = \frac{cos\theta - M(1 - M^2 sin^2\theta)^{1/2}}{(1 - M^2 sin^2\theta)^{1/2} - M cos\theta} \tag{2.68}$$

$$\widehat{v}(\theta) = sin\theta[(1 - M^2 sin^2\theta)^{1/2} + M cos\theta]. \tag{2.69}$$

$V(\theta)$ is the effective convection velocity in the θ direction.

The initial Gaussian distribution is given by:

$$u = v = 0 \tag{2.70}$$

$$p = \rho = \epsilon_1 e^{-\alpha_1 r^2} \tag{2.71}$$

where, ϵ_1 is the amplitude of the acoustic wave and equals 0.01 and $\alpha_2 = ln2/b^2$, where b is the half-width of the acoustic pulse and is equal to 3.0.

2.6.3 The Vorticity Wave

The vorticity wave includes only velocity fluctuations without any effect of pressure or density fluctuations. The dispersion relation is the same as that for the entropy wave and is given by:

$$\lambda_2 = (\omega - \alpha u_0) = 0 \tag{2.72}$$

Thus the entropy and vorticity waves have the same wave convection characteristics. The full solution in a stationary frame of reference is given by:

$$p = \rho = 0 \tag{2.73}$$

$$u = \epsilon_3 y e^{-\alpha_3[(x-Mt)^2+y^2]} \tag{2.74}$$

$$v = -\epsilon_3 (x - Mt) e^{-\alpha_3[(x-Mt)^2+y^2]} \tag{2.75}$$

where, ϵ_3 is the amplitude of the vorticity wave and is equal to 0.0004 and $\alpha_2 = ln2/b^2$, where b is the half-width of the vorticity pulse and is equal to 5.0. The solution for the vorticity wave is given in greater detail in Appendix A.

The initial conditions for the problem discussed in this thesis are obtained by adding the full solution in a stationary frame for all the four physical variables from the acoustic, vorticity and entropy waves. Thus the initial conditions are:

$$p = \epsilon_1 e^{-\alpha_1 r^2} \tag{2.76}$$

$$\rho = \epsilon_1 e^{-\alpha_1 r^2} + \epsilon_2 e^{-\alpha_2[(x-Mt)^2+y^2]} \tag{2.77}$$

$$u = \epsilon_3 y e^{-\alpha_3[(x-Mt)^2+y^2]} \tag{2.78}$$

$$v = \epsilon_3 x e^{-\alpha_3[(x-Mt)^2+y^2]} \tag{2.79}$$

2.7 Boundary Conditions

A majority of the problems involved in computational aeroacoustics are open boundary problems. Since it is not possible to have an infinite domain for a computational problem, it is a common practice to make the computational domain as

large as computationally viable so that all the possible sources that affect the flow are included in the domain. This is done because we cannot determine the effect of any sources outside the computational domain and a large computational domain enables the inclusion of a majority of the influencing sources. Proper boundary conditions need to be applied to simulate the effects of the open environment on the flow within the domain. Another potential problem with the boundaries is the possible reflection of the waves when they exit the domain and thus contaminate the original solution by interfering with the flow, as well as the proper acoustic field in the domain. Thus the boundary conditions are designed such that minimum reflection of the waves exiting the domain takes place.

As shown above, the linearized Euler equations support three types of waves, namely, acoustic, vorticity and entropy waves. The mean flow is responsible for convecting the vorticity and entropy waves downstream in the domain. But, in case of subsonic flow, the acoustic waves propagate and radiate in all directions as well as convecting with the mean flow. Thus, the only waves exiting at the inflow region of the domain are acoustic waves. However, at the outflow region, the outgoing waves consist of the vorticity and entropy waves convected by the mean flow, in addition to the acoustic waves. This gives rise to the need for different boundary conditions for the inlet and outlet regions.

The outflow boundary conditions are applied to the right side boundary. The left and top boundaries will have the boundary conditions accounting for the exit of the acoustic waves. Thus, only the bottom boundary which consists of a solid wall, has to be assigned its specific boundary condition. In the case of a viscous flow, a no-slip boundary condition is applied at the solid wall. But since the present flow problem deals with inviscid flow, a no-penetration flow boundary condition is used at the wall. This means that the component of velocity normal to the wall(v) is zero.

2.7.1 Radiation Boundary Conditions

Radiation boundary conditions are used to represent the asymptotic solutions on the boundaries where the only waves exiting the computational domain are the acoustic waves. So the radiation boundary conditions are effectively the asymptotic solution for the acoustic waves given by equation 2.65.

$$\begin{bmatrix} \rho \\ u \\ v \\ p \end{bmatrix} = \begin{bmatrix} \rho_a \\ u_a \\ v_a \\ p_a \end{bmatrix} = \frac{F(r/V(\theta) - t, \theta)}{r^{0.5}} \begin{bmatrix} \frac{1}{a_0^2} \\ \frac{\widehat{u}\theta}{\rho_0 a_0} \\ \frac{\widehat{v}\theta}{\rho_0 a_0} \\ 1 \end{bmatrix} + O(r^{-1.5}) \tag{2.80}$$

Taking the derivatives of the above equation with respect to time(t) and radial distance(r), the general function F can be eliminated and the disturbances satisfy a reduced set of equations. They can be written as:

$$\left(\frac{1}{V(\theta)} \frac{\partial}{\partial t} + \frac{\partial}{\partial r} + \frac{1}{2r} \right) \begin{bmatrix} \rho \\ u \\ v \\ p \end{bmatrix} = 0 + O(r^{-2.5}) \tag{2.81}$$

In Cartesian coordinates, this can be written,

$$\left(\frac{1}{V(\theta)} \frac{\partial}{\partial t} + cos\theta \frac{\partial}{\partial x} + sin\theta \frac{\partial}{\partial y} + \frac{1}{2r} \right) \begin{bmatrix} \rho \\ u \\ v \\ p \end{bmatrix} = 0 \tag{2.82}$$

2.7.2 Outflow Boundary Conditions

The outflow region of the computational domain consists of acoustic, vorticity and entropy waves exiting the computational domain. So we have to combine the asymptotic solutions of these three types of waves to obtain the outflow boundary conditions. The combined solutions are given by:

$$\begin{bmatrix} \rho \\ u \\ v \\ p \end{bmatrix} = \begin{bmatrix} \chi(x - u_0 t, y) + \rho_a \\ \frac{\partial \psi}{\partial y}(x - u_0 t, y) + u_a \\ -\frac{\partial \psi}{\partial x}(x - u_0 t, y) + v_a \\ p_a \end{bmatrix} + ... \tag{2.83}$$

where ρ_a, u_a, v_a and p_a can be found from the radiation boundary conditions.
As observed from the nature of the three types of waves, the pressure fluctuations are affected only by the acoustic waves. Hence the outflow boundary condition for

pressure is the same as the radiation boundary condition for pressure.
Taking the derivatives of the equation for ρ with respect to time(t) and x in the above equation, the density is found to satisfy,

$$\frac{\partial \rho}{\partial t} + u_0 \frac{\partial \rho}{\partial x} = \frac{\partial \rho_a}{\partial t} + u_0 \frac{\partial \rho_a}{\partial x} \tag{2.84}$$

Substituting $\rho_a = p_a/a_0^2 = p/a_0^2$, with p is known from the radiation boundary conditions, we obtain:

$$\frac{\partial \rho}{\partial t} + u_0 \frac{\partial p}{\partial x} = \frac{1}{a_0^2}\left(\frac{\partial p}{\partial t} + u_0 \frac{\partial p}{\partial x}\right) \tag{2.85}$$

Following a similar procedure for the two components of velocity, differentiation of the equations with respect to time(t) and the x-coordinate location, shows that they satisfy the equations,

$$\frac{\partial u}{\partial t} + u_0 \frac{\partial u}{\partial x} = \frac{\partial u_a}{\partial t} + u_0 \frac{\partial u_a}{\partial x} \tag{2.86}$$

$$\frac{\partial v}{\partial t} + u_0 \frac{\partial v}{\partial x} = \frac{\partial v_a}{\partial t} + u_0 \frac{\partial v_a}{\partial x} \tag{2.87}$$

The acoustic components of the equations satisfy the linearized Euler equations and upon their substitution into the boundary equations, we obtain:

$$\frac{\partial u}{\partial t} + u_0 \frac{\partial u}{\partial x} = -\frac{1}{\rho_0}\frac{\partial p}{\partial x} \tag{2.88}$$

$$\frac{\partial v}{\partial t} + u_0 \frac{\partial v}{\partial x} = -\frac{1}{\rho_0}\frac{\partial p}{\partial y} \tag{2.89}$$

2.7.3 Wall Boundary Conditions

We need to introduce the concept of ghost points to apply the boundary condition at a solid wall. The number of ghost points needed per boundary point is equal to the number of boundary conditions being implemented. Since the present problem deals with an inviscid fluid, the boundary condition does not allow any penetration of flow through the wall. Hence we need only one ghost point per boundary point on the solid wall. This is implemented by setting the v-component of velocity, which is the velocity in the y-direction, as zero on each grid point on the wall[12]. The wall exerts a pressure on the fluid inside the domain such that $v = 0$ on the wall. Thus the pressure exerted by the wall is simulated by the ghost point just below the boundary. Since we only need the pressure at the ghost points, we do not concern ourselves with the other physical variables at the ghost points. These ghost points are only used in the calculation of the pressure derivative in the y-

direction. The derivatives of the other three physical variables in the y-direction do not involve any ghost points. The pressure at the ghost point is established such that $v = 0$ at all the boundary points on the wall. Using the time-marching DRP scheme for v, the velocity at the next time step $n+1$ is given by:

$$v_{l,0}^{(n+1)} = v_{l,0}^{(n+1)} + \Delta t \sum_{j=0}^{3} b_j \mathbf{R}_{l,0}^{(n-j)} \tag{2.90}$$

Here, $\mathbf{R}$ is the residual form the discretized version of the linearized Euler equation for v and is given by:

$$\mathbf{R}_{l,0}^{(n-j)} = -\frac{u_0}{\Delta x} \sum_{i=-3}^{3} a_i v_{l+i,0}^{n-j} - \frac{1}{\rho_0 \Delta y} \sum_{i=-1}^{5} a_1^{51} p_{l,i}^{(n-j)} \tag{2.91}$$

a_i^{51} are the optimized coefficients of the backward difference DRP scheme(Appendix A). Equation 2.91 can be used to find the pressure at the ghost points $p_{l,-1}^n$ by setting the velocity at the next time step on the wall equal to zero ($v_{l,0}^{n+1} = 0$). The pressure at the ghost point is thus given by:

$$p_{l,-1}^n = -\frac{1}{a_{-1}^{51}} \sum_{i=0}^{5} a_i^{51} p_{l,i}^n \tag{2.92}$$

The above boundary condition is necessary to ensure that $\partial p/\partial y = 0$ on the wall boundary points. If no ghost points are introduced and v is set equal to zero along the wall, then $\partial p/\partial y = 0$ will not necessarily be equal to zero.

The above equation can be only used on a uniform grid. On a non-uniform grid, such as the first coarse grid of a multigrid scheme, where the ghost points are closer to the wall than the interior grid points, a modified equation must be derived using the same process as above. The derivation of the formulas for two multigrid levels is shown in Appendix B.

2.7.4 Ghost Points

The central difference Dispersion Relation Preserving(DRP) scheme uses a seven point stencil. Thus it includes values at three grid points on each side of the reference grid point with the direction depending on which spatial derivative is being calculated. Inevitably, at the boundary points(the final three grid points of the computational domain in each direction) there are not enough grid points on one side to implement the central DRP scheme. This gives rise to the need for ghost points outside the initial computational domain. We introduce only a finite number of ghost points in such a way that the central DRP scheme can be used for

the grid points lying on the boundary of the initial domain and thus, only 3 ghost points are added outside the domain for each grid point on the top, left and right boundaries. The physical variables at the ghost points need to be calculated to be used in the DRP scheme. This is done using the radiation boundary conditions for the left and top boundary ghost points and outflow boundary conditions for the ghost points outside the right boundary.

2.8 Filtering

The need for highly accurate solutions requires numerical schemes with low dispersion, low dissipation and large spectral bandwidth. This is achieved by optimizing the dispersion and dissipation characteristics for low wave numbers [13]. Most centered discretization schemes are non-dissipative in nature and thus are prone to numerical instabilities due to the growth of high-frequency modes. The elimination of these high-frequency modes is therefore necessary to obtain numerically stable and accurate results. This is achieved by implementing a high-order implicit filtering technique [14]. If a physical variable after one time step is represented by f, then the filtered variable $\widehat{f}$ is given by:

$$\alpha_l \widehat{f}_{i-1} + \widehat{f}_i + \alpha_l \widehat{f}_{i+1} = \sum_{n=0}^{N} \frac{a_n}{2}(f_{i+n} + f_{i-n}) \tag{2.93}$$

where α_l is a free parameter such that $0 < |\alpha_l| \leq 0.5$. This gives a tridiagonal system of equations that can be easily solved by implementing the Thomas algorithm. The higher the value chosen for α_l, the less the dissipation in the filter. It has been established that a value of α_l chosen between 0.3 and 0.5 is ideal. The filter is chosen in such a manner that its order of accuracy is greater than that of the differential scheme by a minimum of two. Depending on the number of points chosen to filter the value at each grid point, we can obtain a filter scheme of order $2N$ for a $2N+1$ point stencil. The coefficients of the variables $a_0 ... a_n$ are derived in terms of α_l and are given in Table 2.1.

Since this is a central filtering scheme, it cannot be used at the boundaries and the need arises for a different scheme at and near the boundaries. Implicit non-centered filters have been developed by Visbal and Gaitonde [14;15]. These can be used on the boundaries but are not optimized in the Fourier space to implement wall boundary conditions. Non-centered selective filters optimized in wave number space have been developed by Berland [13]. These use a seven or eleven-point explicit finite difference scheme and are implemented on the boundary points.

Scheme	a_0	a_1	a_2	a_3	a_4	a_5	Order of accuracy
F2	$\frac{1}{2}+\alpha_l$	$\frac{1}{2}+\alpha_l$	0	0	0	0	2
F4	$\frac{5}{8}+3\frac{\alpha_l}{4}$	$\frac{1}{2}+\alpha_l$	$-\frac{1}{8}+\frac{\alpha_l}{4}$	0	0	0	4
F6	$\frac{11}{16}+5\frac{\alpha_l}{8}$	$\frac{15}{32}+17\frac{\alpha_l}{16}$	$-\frac{3}{16}+3\frac{\alpha_l}{8}$	$\frac{1}{32}-\frac{\alpha_l}{16}$	0	0	6
F8	$\frac{93+70\alpha_l}{128}$	$\frac{7+18\alpha_l}{16}$	$\frac{-7+14\alpha_l}{32}$	$\frac{1}{16}-\frac{\alpha_l}{8}$	$-\frac{1}{128}+\frac{\alpha_l}{16}$	0	8
F10	$\frac{193+126\alpha_l}{256}$	$\frac{105+302\alpha_l}{256}$	$\frac{15(-1+2\alpha_l)}{64}$	$\frac{45(1-2\alpha_l)}{512}$	$\frac{5(-1+2\alpha_l)}{256}$	$\frac{-1+2\alpha_l}{512}$	10

Table 2.1: Coefficients of Filtering Scheme for the Interior Grid

On a uniform grid, the filtered variable $\widehat{f}(x_i)$ is given by:

$$\widehat{f}(x_i) = f(x_i) - \sigma \sum_{j=-P}^{Q} d_j f(x_i + j\Delta x) \tag{2.94}$$

where Δx is the grid size and d_j are the coefficients of the filter. σ is the filtering strength and is chosen between 0 and 1. Optimization of this scheme produces an 11-point stencil with 1 and 9 points on each side of the concerned grid point. However, this filtering scheme introduces a high level of dissipation over a wide range of wavelengths. Thus, in the present study, a seven point stencil is used on the last but one point of the computational domain. This is much more stable than the 11-point stencil. In spite of being less dissipative than the 11-point stencil, the 7-point stencil is still quite dissipative and would produce excessive damping which will give inaccurate results. Thus a second-order accurate four-point filter is implemented. The coefficients of this filter are given in Table 2.2. SF stands for selective filter since it filters out only short waves. The two integers in the subscript denote the number of grid points on either side of the concerned grid point at which the filter is being applied. This filter is less accurate in phase compared to the seven and eleven point stencils but is comparable in amplitude accuracy.

a_j	SF_{03}
a_0	0.320882352941
a_{-1}	-0.465
a_{-2}	0.179117647059
a_{-3}	0.035

Table 2.2: Coefficients of SF_{03} Filtering Scheme for Boundary Points

For wavelengths satisfying the condition $k\Delta x < \pi/2.5$, the dissipation observed is one order of magnitude greater than the dissipation observed in the 11-point centered filter. To compensate for the amount of dissipation on the first and last boundary points in each direction, the filtering strength used for the filter SF_{03}, $\sigma_{boundary}$ is taken to be one-tenth of the filtering strength, σ for the interior grid points. The coefficients of the weights used in the filtering scheme for the first two grid points adjacent to the boundary are given in Table 2.3. The weighting coefficients for the next two adjacent grid points from the wall are given in Table 2.4.

a_j	SF_{15}	SF_{28}
a_{-2}	-	0.0307159855992469
a_{-1}	-0.085777408969	-0.148395705486028
a_0	0.277628171524	0.312055385963757
a_1	-0.356848072173	-0.363202245195514
a_2	0.223119093072	0.230145457063431
a_3	-0.057347064865	-0.0412316564605079
a_4	-0.000747264596	-0.0531024700805787
a_5	-0.000027453993	0.0494343261171287
a_6	-	-0.0198143585458560
a_7	-	0.00339528102492129
a_8	-	0

Table 2.3: SF_{15} and SF_{28} Filtering Scheme Coefficients for Near-Boundary Points

a_j	SF_{37}	SF_{46}
a_{-4}	-	0.008391235145
a_{-3}	-0.000054596010	-0.047402506444
a_{-2}	0.042124772446	0.121438547725
a_{-1}	-0.173103107841	-0.200063042812
a_0	0.299615871352	0.240069047836
a_1	-0.276543612935	-0.207269200141
a_2	0.131223506571	0.122263107843
a_3	-0.023424966418	-0.047121062819
a_4	0.013937561779	0.009014891495
a_5	-0.024565095706	0.001855812216
a_6	0.013098287852	-0.001176830044
a_7	-0.002308621090	-

Table 2.4: SF_{37} and SF_{46} Filtering Scheme Coefficients for Near-Boundary Points

In the next chapter, the numerical methods introduced in this chapter are applied to the benchmark problem and the results are described.

Chapter 3

Results and Discussions

Four different methods have been used to solve the benchmark CAA problem considered in this thesis. The first method uses Dual Time Stepping(DTS), while the second method uses Implicit Residual Smoothing(IRS) as well as DTS. The final two methods include a Multigrid scheme in addition to the first and second techniques.

The first two schemes, i.e. the DTS method and IRS coupled with DTS method show identical results. This can be seen in Figures 3.1 and 3.2. The $y = 0$ location represents the solid wall. Figure 3.1 shows a comparison of the pressure waveforms, prior to reflection from the solid wall, along $x = 0$ after $t = 5.5s$ using DTS and DTS combined with IRS. The waveform along $x = 0$ represents the pulse propagation in a direction normal to the wall. The waveforms in both these cases perfectly superimpose on each other showing that the these methods do not compromise the accuracy of the numerical solution. Figure 3.2 shows the pressure waveforms after reflection from the solid wall along $x = 0$ after $t = 31.5s$. The left peak in the waveform represents the reflected wave, which now travels in the same direction as the non-reflected wave. A comparison of both the methods using the multigrid scheme also exhibits similar results and the waveforms superimpose in both cases as seen in Figures 3.3 and 3.4.

A comparison of the schemes involving only DTS and the one using Multigrid along with DTS shows that both waveforms propagate with the same speed in the domain. It is seen that the amplitude of the waveform using the Multigrid scheme is smaller than the amplitude observed with only DTS as seen in Figures 3.5 and 3.6. These figures represent the pressure waveforms along $x = 0$, before and after reflection from the solid wall, at $t = 5.5s$ and $t = 31.5s$ respectively. This discrepancy can be potentially attributed to the loss of information or data during the restriction and prolongation stages of the Multigrid scheme.

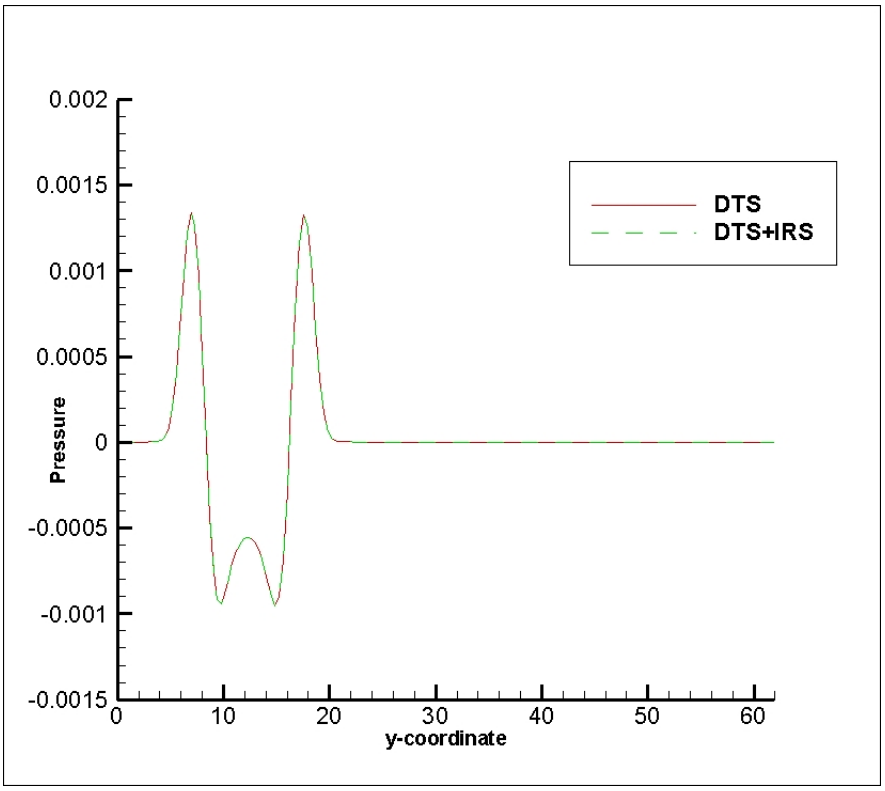


Figure 3.1: Comparison of pressure waveforms along $x = 0$ at $t = 5.5s$ using DTS and DTS+IRS techniques

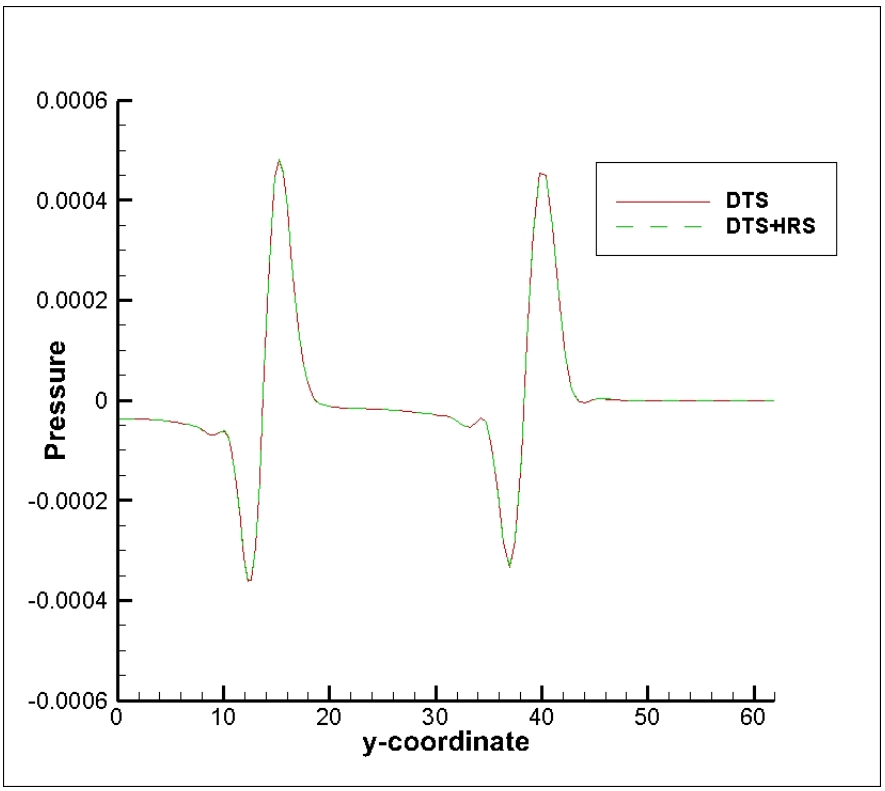


Figure 3.2: Comparison of pressure waveforms along $x = 0$ at $t = 31.5s$ using DTS and DTS+IRS techniques

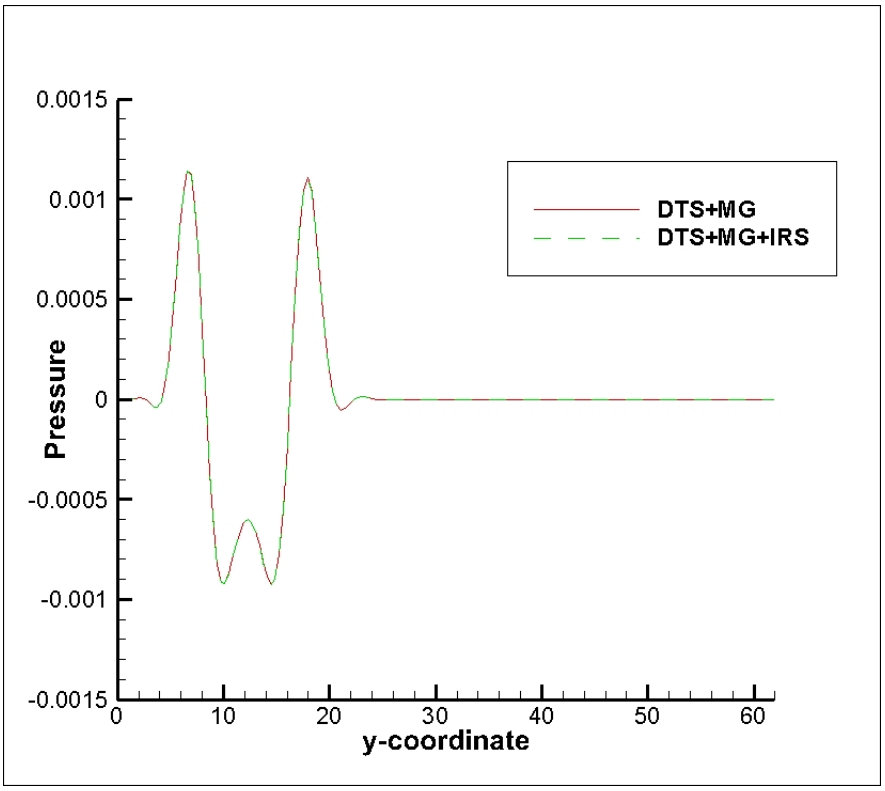


Figure 3.3: Comparison of pressure waveforms along $x = 0$ at $t = 5.5s$ using DTS+MG and DTS+MG+IRS techniques

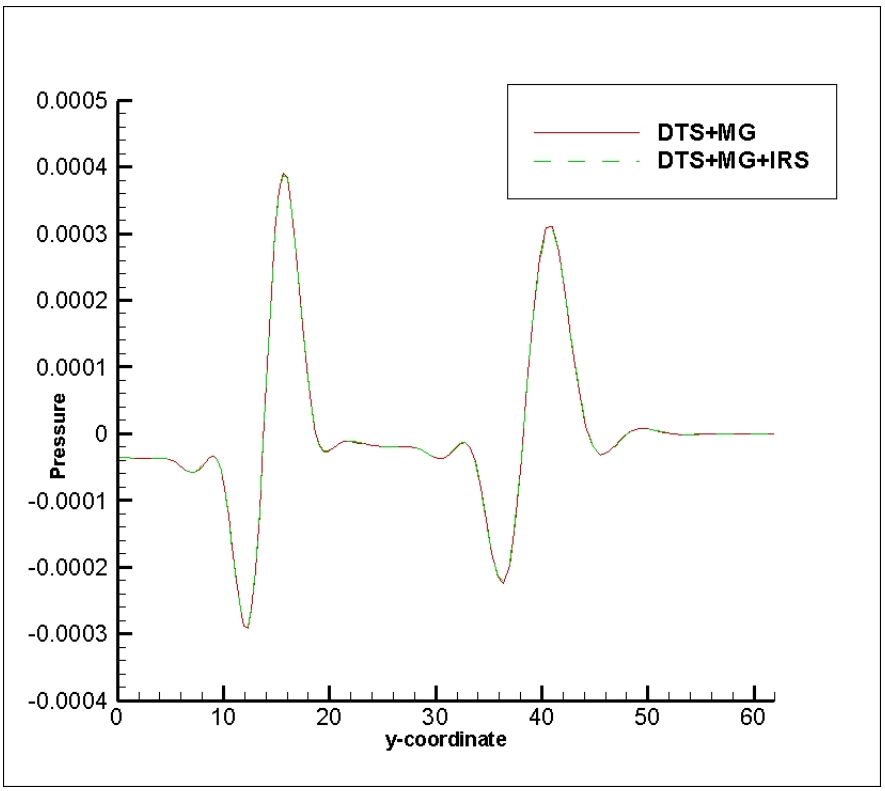


Figure 3.4: Comparison of pressure waveforms along $x = 0$ at $t = 31.5s$ using DTS+MG and DTS+MG+IRS techniques

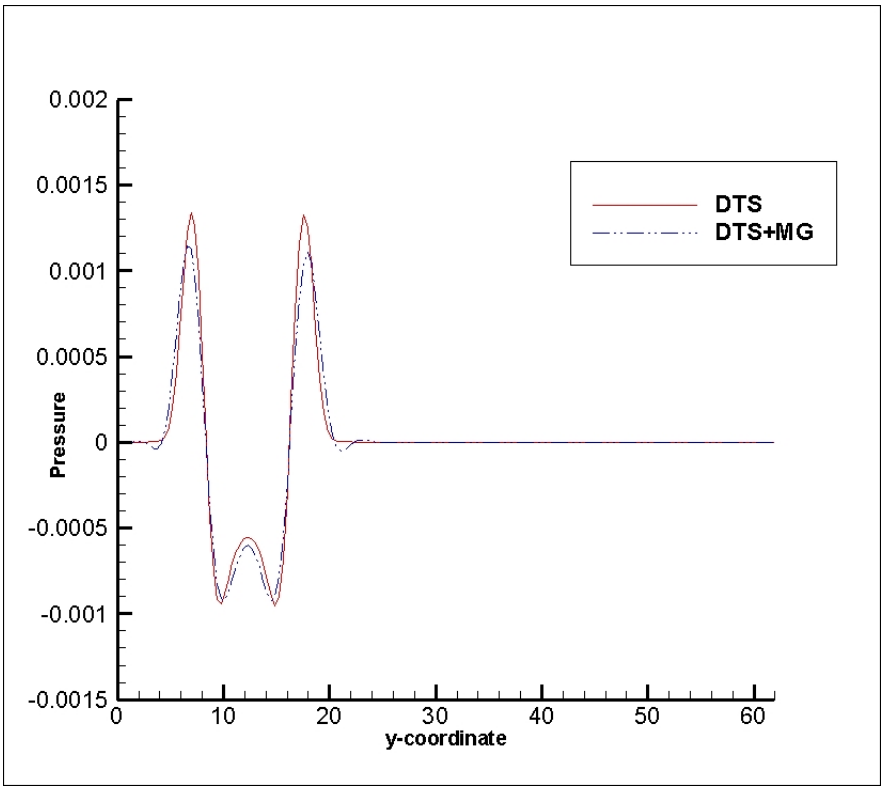


Figure 3.5: Comparison of pressure waveforms along $x = 0$ at $t = 5.5s$ using DTS and DTS+MG techniques

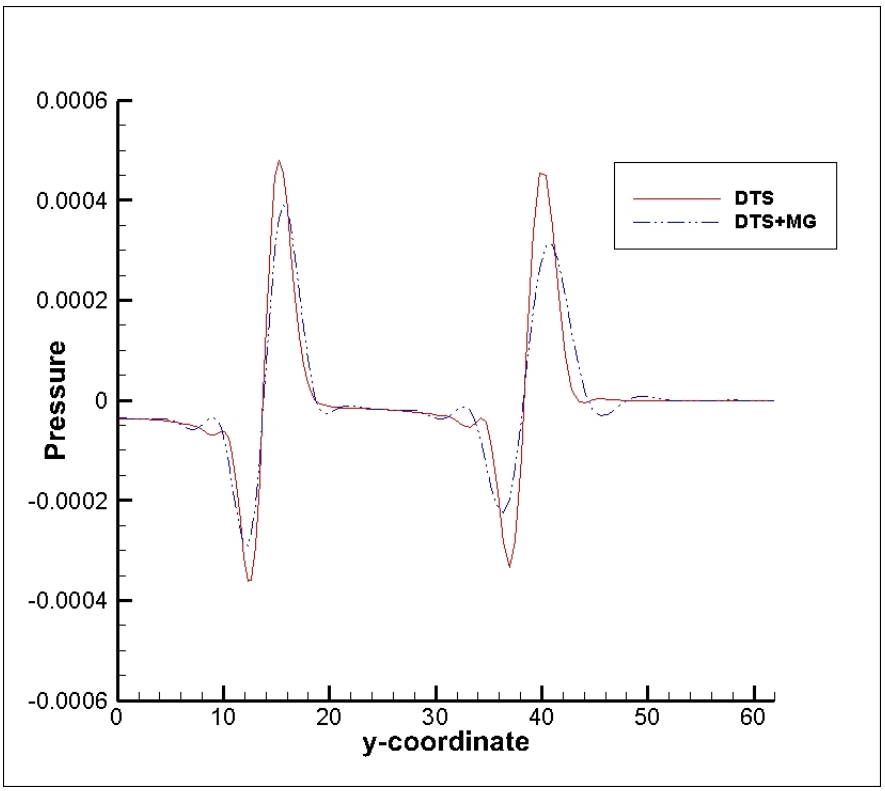
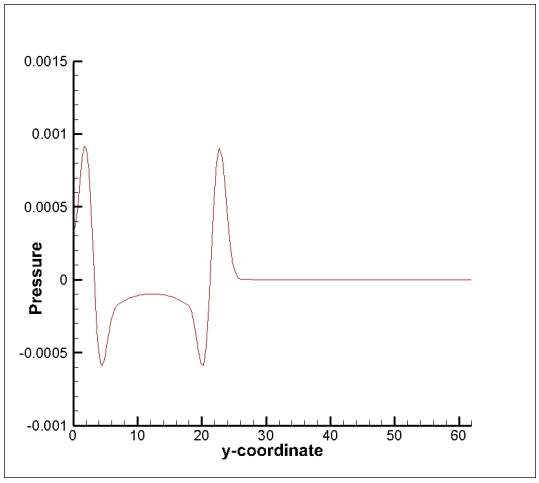
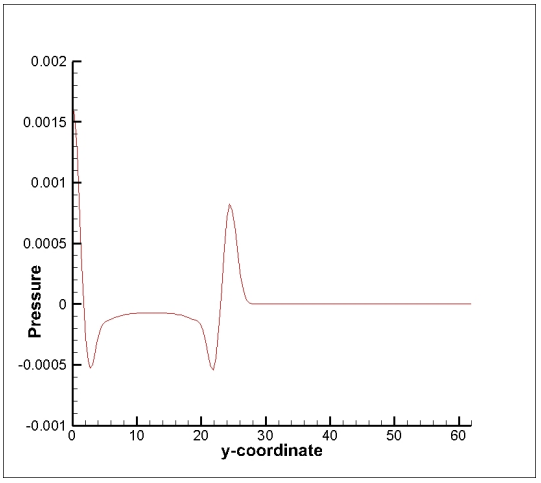


Figure 3.6: Comparison of pressure waveforms along $x = 0$ at $t = 31.5s$ using DTS and DTS+MG techniques

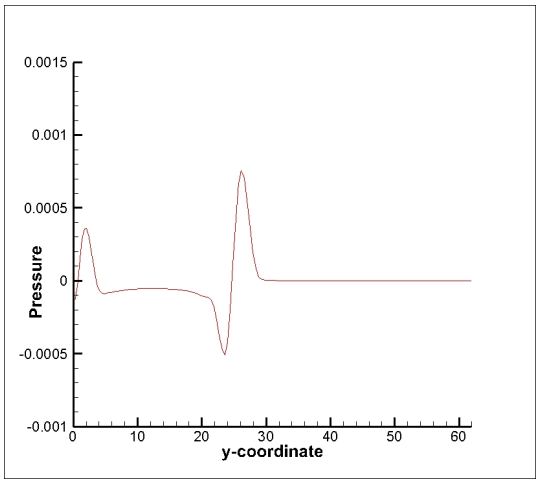
The plots along the $x = 0$ line show the propagation of the pulse in the direction normal to the wall. Figures 3.7(a)-(d) show the specular reflection of the wave by the solid wall. This is possible only with the correct implementation of the wall boundary conditions and the introduction of one ghost point for each grid point on the wall. The boundary condition used is no penetration through the wall, i.e. the component of velocity normal to the wall(v) is zero. This in turn causes the derivative of pressure perpendicular to the wall to be zero. Figures 3.8(a) and (b) depict the density contours, at $t = 11.5s$ and $t = 17.5s$ respectively, which consist of the acoustic and entropy pulses. The acoustic pulse, which is seen on the left in these figures, is convected downstream, while expanding radially. The specular reflection of the acoustic wave by the solid wall can be seen in these figures. The entropy pulse, which is closer to the right boundary of the domain, does not expand and is only convected downstream.



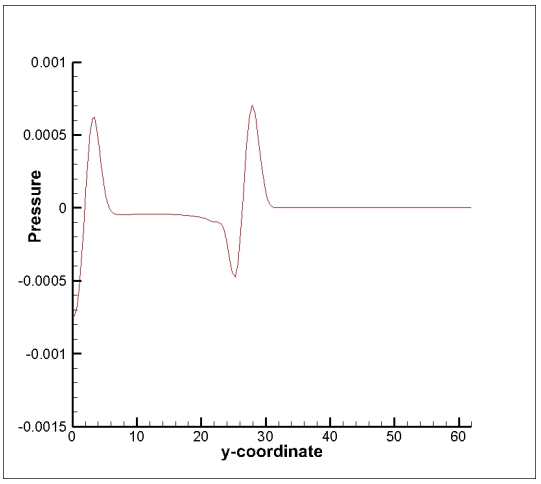
(a) $t = 11.5s$



(b) $t = 13.5s$

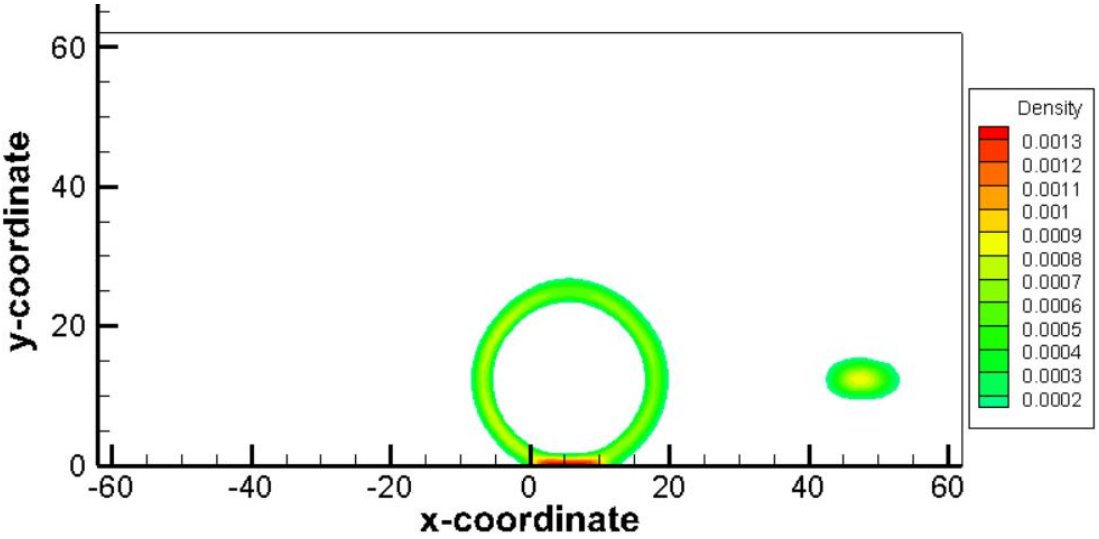


(c) $t = 15.5s$

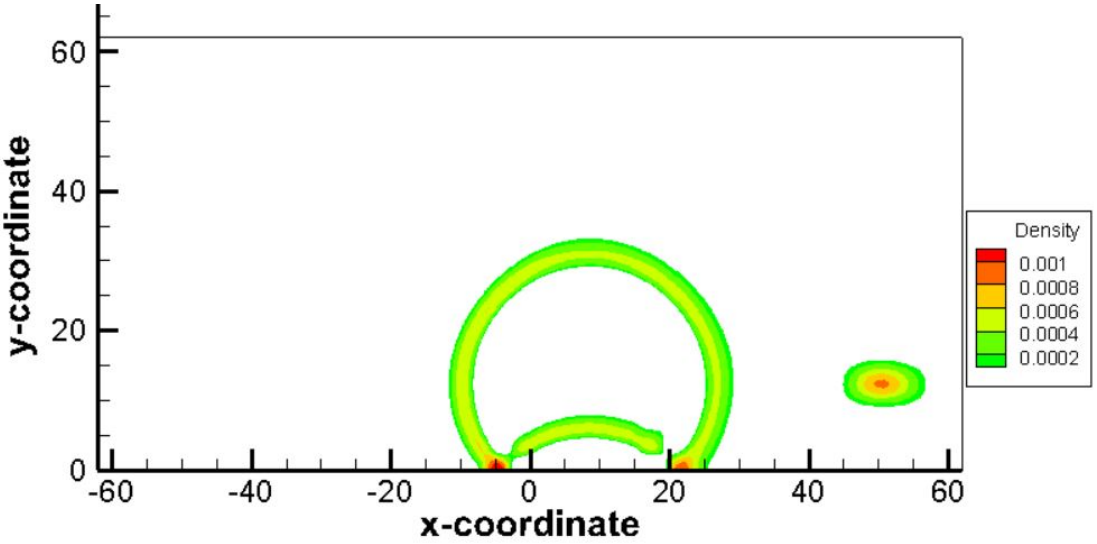


(d) $t = 17.5s$

Figure 3.7: Pressure waveform depicting the specular reflection of the pulse by the solid wall



(a)



(b)

Figure 3.8: Density contours showing specular reflection against the solid wall at $a = 11.5s$ and $b = 17.5s$

The contour plot in Figure 3.9 shows the initial location of the acoustic wave, which is at the center of the domain, and that of the entropy wave which is further downstream of the acoustic wave at a distance equal to one third of the computational domain from the flow boundary. The vorticity wave is also placed at the same location as the entropy wave. The acoustic wave travels at three times the speed of the vorticity and entropy waves. The deliberate positioning of all the three waves ensures that all three pulses exit the domain simultaneously as seen in Figure 3.10 at $t = 41.5s$. The green component of the contour represents the acoustic wave, while the red and yellow parts of the contour represent the entropy wave. This helps in correctly testing the efficiency of the outflow boundary conditions. It is also interesting to note that since the outflow boundary conditions are developed from the far-field asymptotic solutions, the three pulses have to be initially sufficiently far from the boundaries. This is necessary because the boundary conditions depend on the angle θ at which the waves are incident on the boundaries, and it does not play a major role if the boundaries are very far from the source. However, if the sources are close to the boundary then the angle θ can disrupt the intended effect of the boundary conditions. Since the sources are at a distance far enough from the boundary in the given problem, no such inconveniences are observed. This can also be seen from the density contour at $t = 43.5s$ in Figure 3.11 where no reflection of the waves is observed from the boundaries where outflow boundary conditions have been applied. Similarly, the effectiveness of the radiation boundary conditions can be seen in Figure 3.12, which shows the density contour at $t = 130.5s$ with no reflected or spurious waves due to the interaction of the pulse with the left boundary.

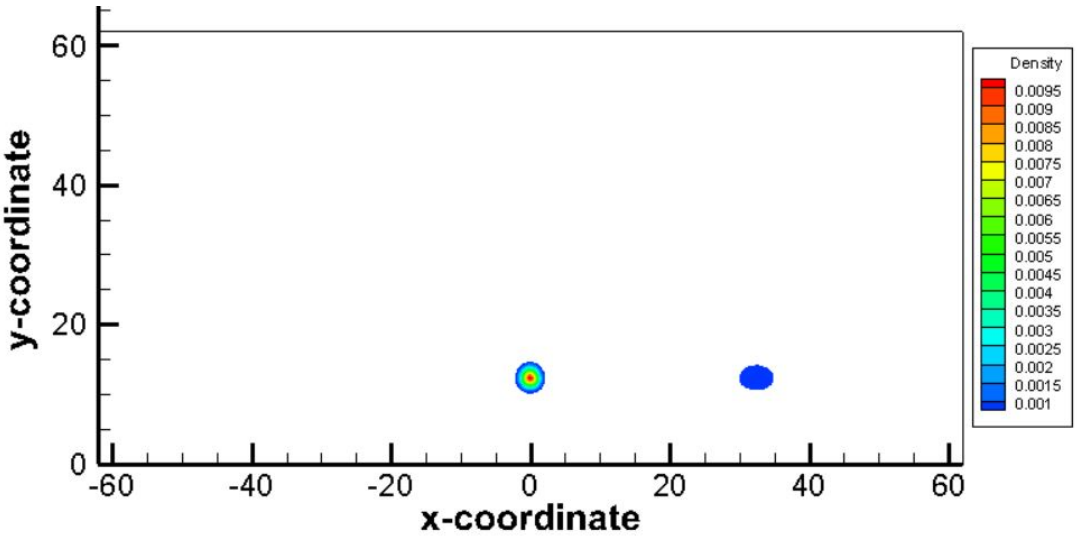


Figure 3.9: Density contour at $t = 0$

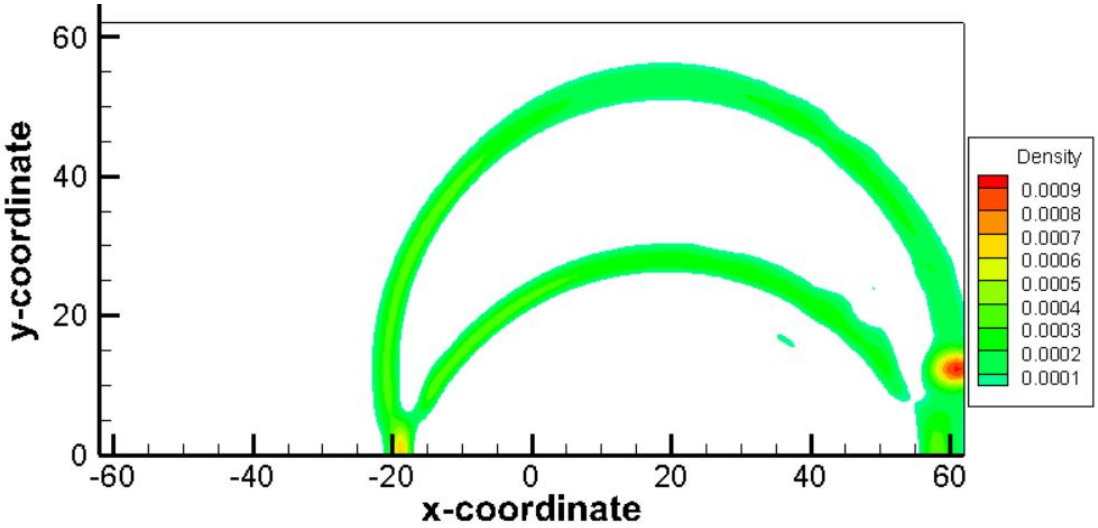


Figure 3.10: Density contour at $t = 41.5s$

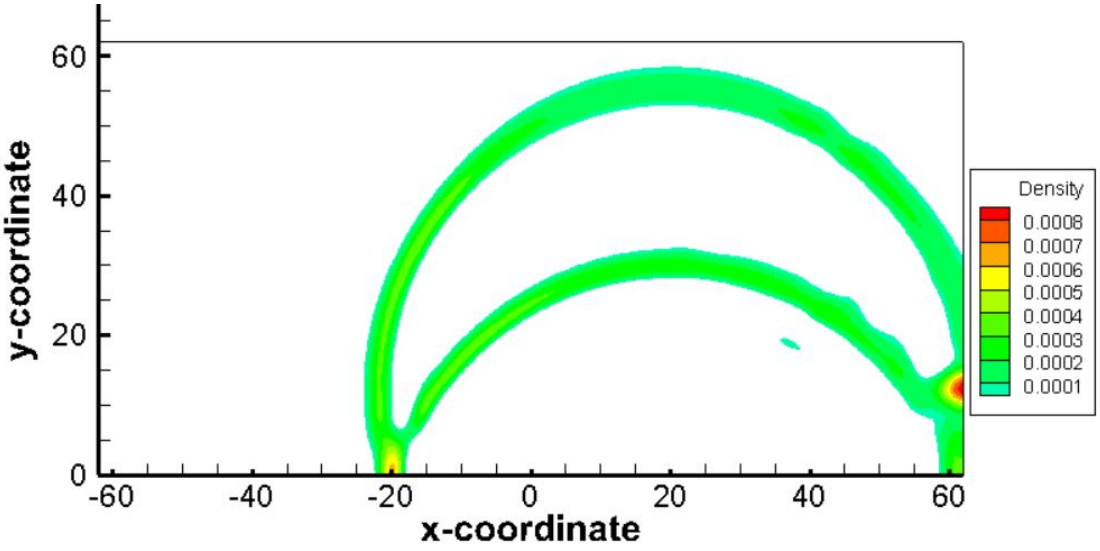


Figure 3.11: Density contour at $t = 43.5s$

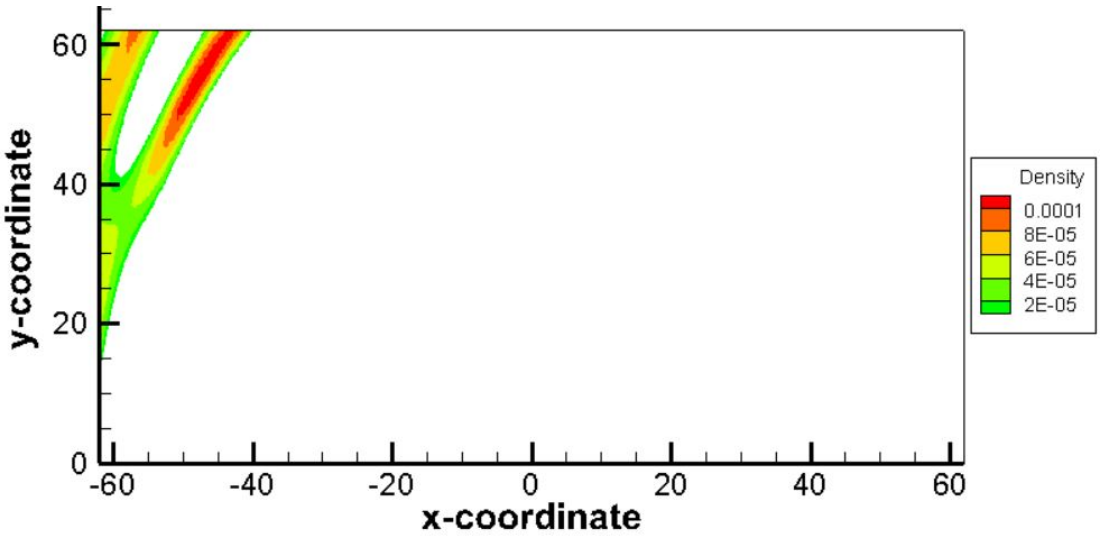
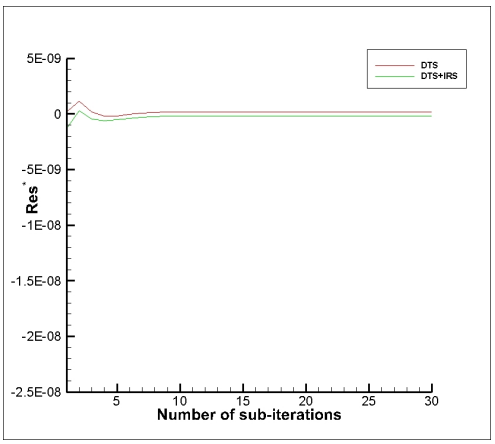


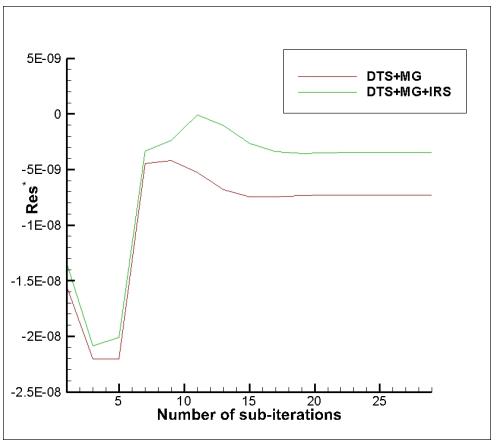
Figure 3.12: Density contour at $t = 130.5s$

A comparison of the residuals in methods 1 and 2(DTS and DTS+IRS) shows that, at the earlier time steps, both these methods converge after approximately an equal number of iterations in the fictitious time scale. Figures 3.13(a) and (b), which represent the density residual at $x = 0$ on the wall after $t = 3.5s$, show this behavior. While the ideal case in Dual Time Stepping requires the residual(Res^*) to become zero, it is extremely difficult to get absolute zero for the residual values because of the restrictions on the computational abilities of the solver. Thus, the best results are obtained when the residual takes a value as close to zero as possible. Comparison of cases 1 and 2 shows that, near the boundaries, the density residual in DTS+IRS has a final convergence value closer to zero than DTS. It is also seen that the inclusion of multigrid increases the values of the final converged residual value. This can be attributed to loss of information during the restriction and prolongation stages of the multigrid scheme. At later time steps though, it is observed that the inclusion of IRS does not affect the residual values in a significant manner, as the residual values are very similar since the first sub-iteration in the fictitious time scale and the plots superimpose. A similar pattern is observed in cases 3 and 4 for the density residuals at $x = 0$ on the solid wall after $t = 23.5s$ as seen in Figures 3.14(a) and (b), which includes the Multigrid scheme. The residuals are closer to zero, for the initial time steps, for the method including the IRS scheme and are similar to the case with Multigrid at the later time steps.

Away from the boundaries, the residuals are identical for cases 1 and 2 even at earlier time steps and the residual plots superimpose for both these cases. A similar behavior is observed for cases 3 and 4, which incorporate the Multigrid method. This is shown in Figures 3.15(a) and (b).

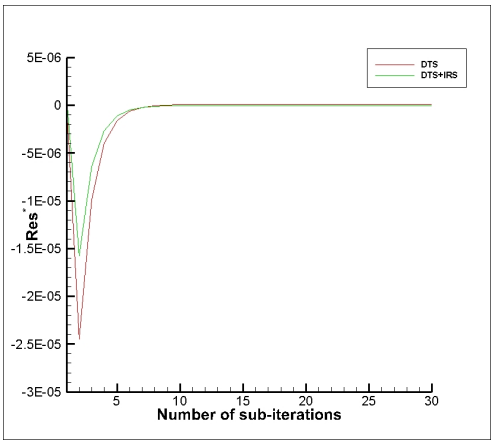


(a) DTS vs. DTS+IRS

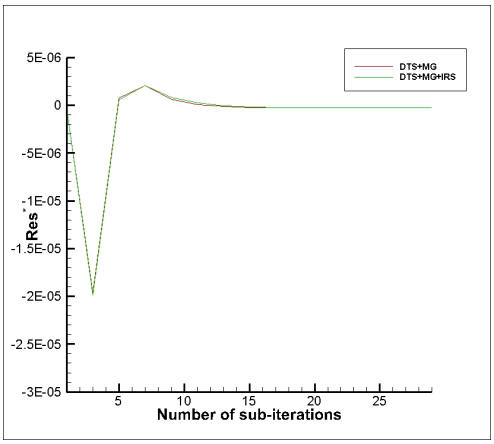


(b) DTS+MG vs. DTS+MG+IRS

Figure 3.13: Comparison of density residuals at $t = 3.5s$ at location $x = 0$ on the wall

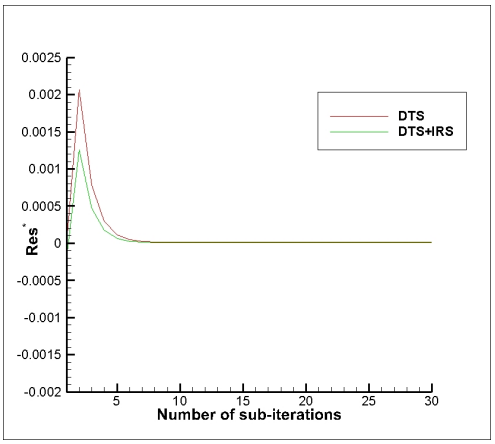


(a) DTS vs. DTS+IRS

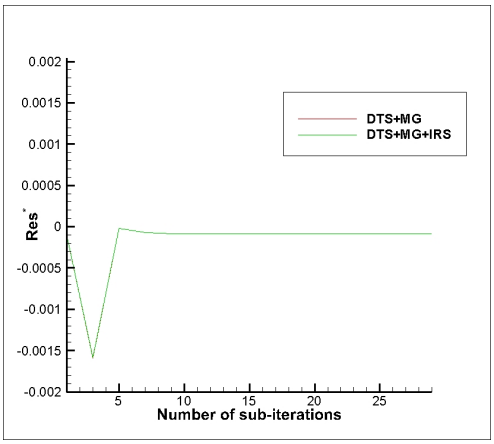


(b) DTS+MG vs. DTS+MG+IRS

Figure 3.14: Comparison of density residuals at $t = 23.5s$ at location $x = 0$ on the wall



(a) DTS vs. DTS+IRS



(b) DTS+MG vs. DTS+MG+IRS

Figure 3.15: Comparison of density residuals at $t = 3.5s$ at grid location $(0, 30)$

A comparison of the computational time taken for the pulse to travel the same distance is shown in Figure 3.16. As expected, when both cases 1 and 2 are allowed to run up to the same number of iterations with the same convergence criteria in the fictitious time domain, case 2 takes more time because of implementation of the IRS scheme. A similar reasoning is applied to cases 3 and 4, with case 4 taking more time than the former one because of the use of IRS. It is seen that the fastest method among all the four techniques used is case 3, which uses DTS with Multigrid. This can be attributed to the fact that half of the calculations in the fictitious time domain are performed on a coarser grid with many fewer grid points than the fine grid. Thus, the computational time is reduced compared to using only DTS to solve the problem.

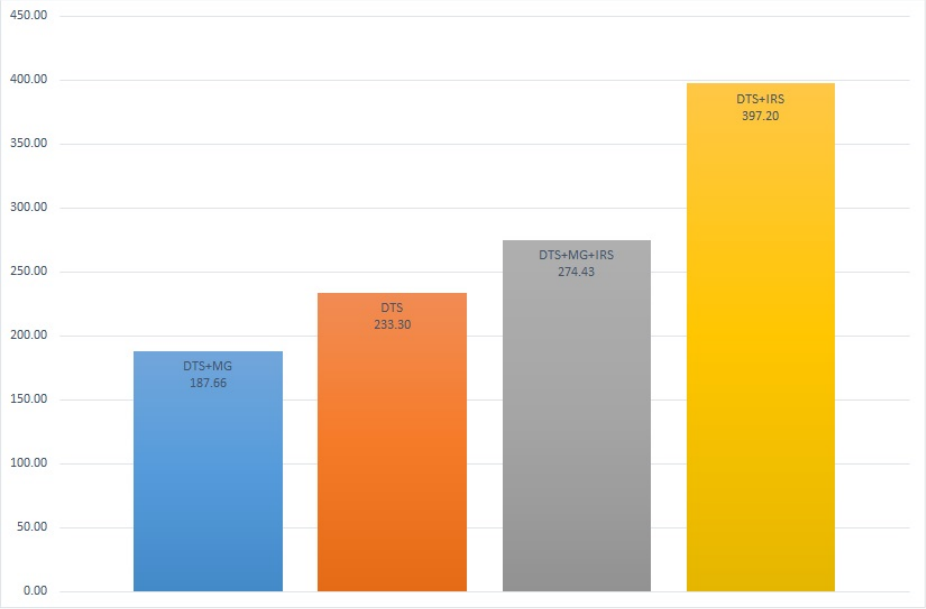


Figure 3.16: Comparison of time taken(in *seconds*) for 235 dual time steps with $dt = 0.2s$

The number of sub-iterations allowed during each physical time step was limited to 30 for the different methods used in this thesis. The convergence criterion was kept such that the change in the residual at the current time step to the residual at the previous time step is less than 1%. This criterion has to be satisfied all over the domain for all four physical variables for the solution to be considered as converged. The average number of sub-iterations taken by each method over 235 physical time steps is shown in Figure 3.17. It is seen that the least number of sub-iterations are taken when only DTS is implemented and the number of sub-iterations increases for DTS+MG, DTS+IRS and DTS+MG+IRS respectively.

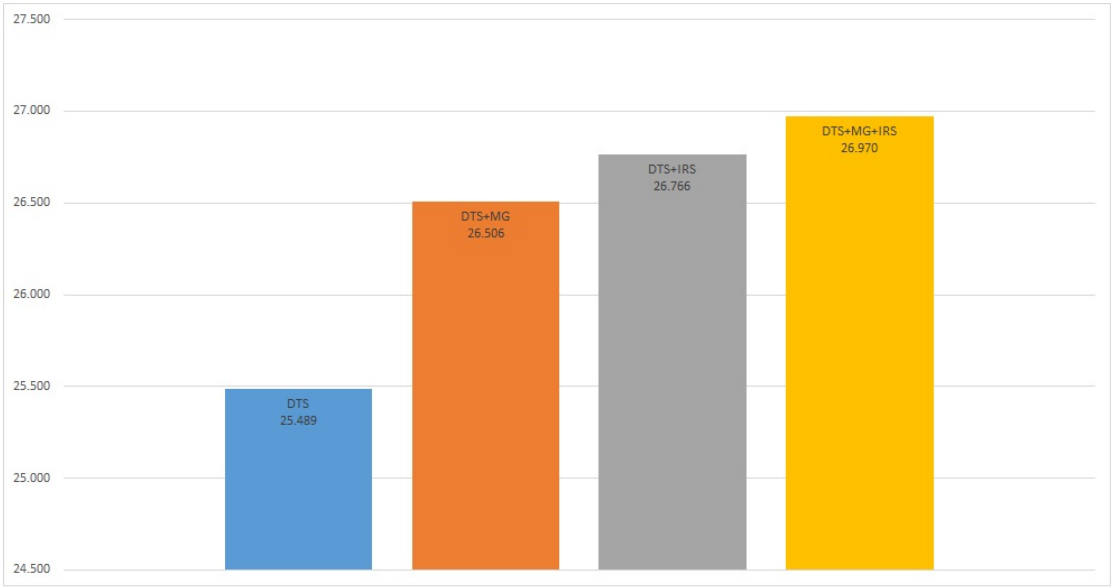


Figure 3.17: Comparison of average number of sub-iterations in the fictitious time domain over a range of 235 dual time steps in the physical time domain

The biggest advantage of using DTS is the implementation of a large time step compared to the time step used without DTS. However, even though the time step that can be used safely without causing any convergence or stability problems is large, there exists a ceiling for the maximum time step. As stated earlier, the ratio of the CFL numbers with and without DTS is 4. It was observed that the results are in accordance with the expected solutions even for a ratio of the CFL numbers in fictitious to physical time scales of 12. This is the maximum threshold of the time step used in DTS for this particular flow problem without causing any accuracy problems. For any time step above this value, the solution blows up after a few time steps. The larger the time step used, the fewer the number of time steps before which the solution blows up. In general practice, the computational grid would contain a much smaller minimum grid size and a much higher ratio of CFL numbers can be implemented. It is common to use a CFL number ratio of up to 100 for accurate resolution of extremely fine grids.

In the DTS scheme, every time step in the physical time domain is referred to as one iteration. Every time step taken in the fictitious time domain is referred to as one sub-iteration. Time progression in the fictitious time domain results in the solution achieving convergence with the residual going towards zero. Three different cases have been used to show the impact of the number of sub-iterations

on the accuracy of the solution. The first case allows the complete convergence of the solution, whereas, the second and third cases limit the number of sub-iterations to 2 and 5 respectively. It can be seen from Figure 3.18 that, at an early stage in the physical time domain, the case with just 2 sub-iterations produces very different results from the other two cases. Also, results with 5 sub-iterations are comparable to the fully converged results. Figure 3.19 shows that at later time steps, even the case with 5 sub-iterations produces very different results from the fully converged results. This is because the error from the earlier time steps, due to the non-convergence of the solution with 5 sub-iterations, accumulates as time progresses and is high enough to cause a large difference in the results.

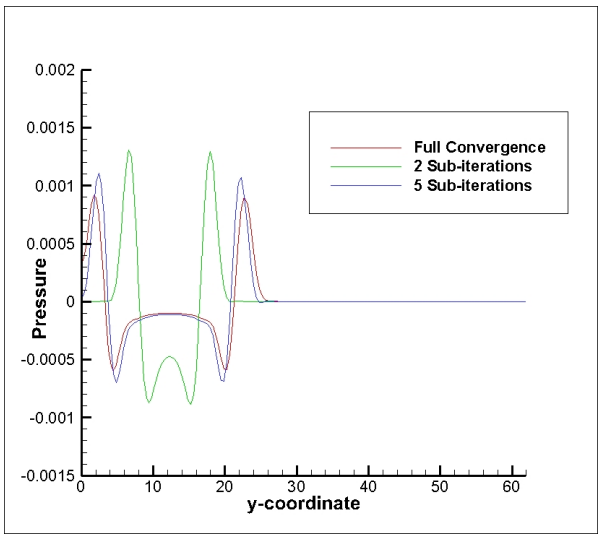


Figure 3.18: Comparison of pressure waveform with different number of sub-iterations at $t = 13.5s$

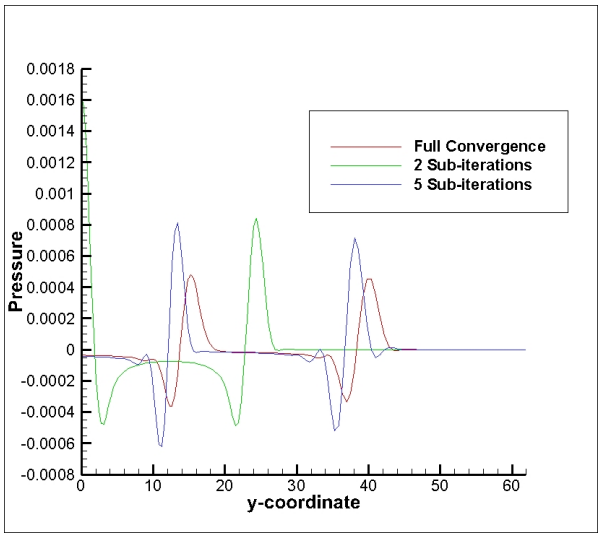


Figure 3.19: Comparison of pressure waveform with different number of sub-iterations at $t = 31.5s$

Chapter 4

Conclusions

A benchmark flow problem has been solved using various convergence acceleration techniques, individually and with each of the other techniques. It is observed that the most computationally cost-effective method is the Multigrid technique used with DTS, as it uses the least amount of computational time. A 24% improvement in the computational time is seen in this case, when compared to the case in which only DTS is implemented. This is expected, since the coarse level in Multigrid is less computationally demanding than the fine grid. On the average, the fastest convergence in the fictitious time domain was achieved by using the DTS technique by itself. Another intriguing observation that has been made is the increase in the computational time with the inclusion of Implicit Residual Smoothing, without any substantial increase in accuracy. For this flow problem, the accuracy of the results decreases with the use of only a single coarse grid in the Multigrid scheme.

4.1 Future Work

The flow problem in this thesis is a basic benchmark problem and was solved to determine the most efficient conditions to implement the Dual Time Stepping scheme. This work can form the foundation of a more detailed and thorough analysis into this field. The flow problem dealt with an inviscid flow. This can be extended to solve more complicated problems with viscous flow, which will require the addition of the stress gradient terms to the governing equations along with an additional wall boundary condition of no-slip. The Multigrid scheme used in this thesis is a simple single-level sawtooth algorithm. Other schemes like V-cycle and W-cycle can be implemented with multiple coarser grids to determine if the inclusion of a higher number of coarser grids curbs the loss of accuracy observed during the restriction and prolongation stages. Finally, the non-uniform grid can be changed to make it much finer at the places where high accuracy is desired.

This will also enable the use of a much higher time step in DTS compared to the time step used without DTS, and thus use the true power of the Dual Time Stepping scheme.

Appendix A

Full Solutions of Acoustic, Entropy and Vorticity Waves

As shown in equation 2.54, the coefficient matrix $\mathbf{A}$ in the Fourier-Laplace transform of the linearized Euler equations is given by

$$\mathbf{A} = \begin{bmatrix} \omega - \alpha u_0 & -\rho_0 \alpha & -\rho_0 \beta & 0 \\ 0 & \omega - \alpha u_0 & 0 & -\alpha/\rho_0 \\ 0 & 0 & \omega - \alpha u_0 & -\beta/\rho_0 \\ 0 & -\gamma p_0 \alpha & -\gamma p_0 \beta & \omega - \alpha u_0 \end{bmatrix} \tag{A.1}$$

The eigenvectors of this matrix are

$$\mathbf{X}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.2}$$

$$\mathbf{X}_2 = \begin{bmatrix} 0 \\ \beta \\ -\alpha \\ 0 \end{bmatrix} \tag{A.3}$$

$$\mathbf{X}_3 = \begin{bmatrix} \frac{1}{a_0^2} \\ -\frac{\alpha}{\rho_0 a_0 (\alpha^2+\beta^2)^{0.5}} \\ -\frac{\beta}{\rho_0 a_0 (\alpha^2+\beta^2)^{0.5}} \\ 1 \end{bmatrix} \tag{A.4}$$

$$\mathbf{X}_4 = \begin{bmatrix} \frac{1}{a_0^2} \\ \frac{\alpha}{\rho_0 a_0 (\alpha^2+\beta^2)^{0.5}} \\ \frac{\beta}{\rho_0 a_0 (\alpha^2+\beta^2)^{0.5}} \\ 1 \end{bmatrix} \tag{A.5}$$

where, $a_0 = (\gamma p_0/\rho_0)^{0.5}$ is the speed of sound and α, β and ω are the variables in the Fourier-Laplace transform corresponding to x, y and t coordinates respectively. Thus the solution can be expressed in terms of the eigenvalues and eigenvectors as

$$\tilde{\mathbf{U}} = \frac{C_1}{\lambda_1}\mathbf{X}_1 + \frac{C_2}{\lambda_2}\mathbf{X}_2 + \frac{C_3}{\lambda_3}\mathbf{X}_3 + \frac{C_4}{\lambda_4}\mathbf{X}_4 \tag{A.6}$$

The above equation shows the decomposed solution of the linearized Euler equations with $\mathbf{X}_1$ and $\mathbf{X}_2$ representing the entropy and vorticity waves respectively. $\mathbf{X}_3$ and $\mathbf{X}_4$ are the two acoustic waves.
C_1,C_2,C_3 and C_4 are the elements of the coefficient vector $\mathbf{C}$ which is given by

$$\mathbf{C} = \mathbf{X}^{-1}\tilde{\mathbf{G}} \tag{A.7}$$

where $\mathbf{X}^{-1}$ is the inverse of the eigenvector matrix and is

$$\mathbf{X}^{-1} = \begin{bmatrix} 1 & 0 & 0 & -\frac{1}{a_0^2} \\ 0 & \frac{\beta}{(\alpha^2+\beta^2)} & -\frac{\alpha}{(\alpha^2+\beta^2)} & 0 \\ 0 & -\frac{1}{2}\frac{\rho_0 a_0 \alpha}{(\alpha^2+\beta^2)}^{0.5} & -\frac{1}{2}\frac{\rho_0 a_0 \beta}{(\alpha^2+\beta^2)}^{0.5} & \frac{1}{2} \\ 0 & \frac{1}{2}\frac{\rho_0 a_0 \alpha}{(\alpha^2+\beta^2)}^{0.5} & \frac{1}{2}\frac{\rho_0 a_0 \beta}{(\alpha^2+\beta^2)}^{0.5} & \frac{1}{2} \end{bmatrix} \tag{A.8}$$

and $\tilde{\mathbf{G}}$ is the sum of the Fourier-Laplace transforms of the source term and initial conditions and equals

$$\tilde{\mathbf{G}} = i(\tilde{\mathbf{H}} + \tilde{\mathbf{U}}_{initial}/2\pi). \tag{A.9}$$

Appendix B

Vorticity Wave Solution

The solution for the vorticity wave is determined by taking the inverse Fourier-Laplace transform of the second vector in equation A.6. This yields

$$U(x,y,t) = \int_{\Gamma}\int\int_{-\infty}^{\infty} \frac{C_2}{\omega - \alpha u_0}\mathbf{X}_2 e^{i(\alpha x+\beta y-\omega t)} d\alpha d\beta d\omega \tag{B.1}$$

There are no pressure or density fluctuations in the vorticity waves as the first and last elements of $\mathbf{X}_2$ are zero. Upon further analysis, the form of the velocity components can be written as

$$u(x,y,t) = \begin{cases} \frac{\partial \psi}{\partial y}, & x \to \infty \\ 0, & x \to -\infty \end{cases}$$

$$v(x,y,t) = \begin{cases} -\frac{\partial \psi}{\partial x}, & x \to \infty \\ 0, & x \to -\infty \end{cases}$$

where $\psi = \psi(x - u_0 t, y)$.

Appendix C

Derivation of one-sided difference formula for non-uniform grid

The wall boundary condition is implemented by using the criterion that the component of velocity normal to the wall(v) is zero, which in turn makes $\partial p/\partial y = 0$ at the wall. The difference scheme used at the wall to calculate the pressure derivative in the y-direction is a one-sided difference scheme on a uniform grid as defined in Appendix A. When the multigrid scheme is applied to the present flow problem, the ghost points are still at the same distance from the wall. But the grid points closest to the wall in the interior of the domain are now further apart compared to the fine grid. This gives rise to a non-uniform grid near the wall and thus warrants a customized difference scheme to apply the wall boundary condition effectively.

C.1 First Multigrid Level

In the first coarse grid, the grid spacing near the wall is shown in Figure C.1

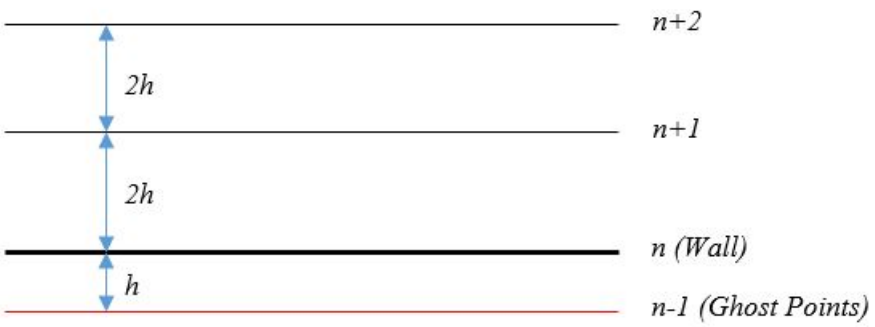


Figure C.1: First Multigrid Level

Let $f(x_0)$ be the value of an arbitrary function f at grid point n. Using Taylor's

series expansion,

$$f_{n+1} = f(x_0 + 2h) = f(x_0) + 2hf'(x_0) + 2h^2 f''(x_0) + \frac{4h^3}{3} f'''(x_0) + ... \quad \text{(C.1)}$$

$$f_{n+2} = f(x_0 + 4h) = f(x_0) + 4hf'(x_0) + 8h^2 f''(x_0) + \frac{32h^3}{3} f'''(x_0) + ... \quad \text{(C.2)}$$

$$f_n = f(x_0) = f(x_0) \quad \text{(C.3)}$$

$$f_{n-1} = f(x_0 - h) = f(x_0) - hf'(x_0) + \frac{h^2}{2} h^2 f''(x_0) - \frac{h^3}{6} f'''(x_0) + ... \quad \text{(C.4)}$$

Introducing four coefficients a,b,c,d and multiplying them to equations C.1-C.4 respectively yields,

$$af(x_0 + 2h) + bf(x_0 + 4h) + cf(x_0) + df(x_0 - h) = (a + b + c + d)f(x_0)$$
$$+h(2a+4b-d)f'(x_0)+h^2\left(2a + 8b + \frac{d}{2}\right) f''(x_0)+h^3\left(\frac{4a}{3} + \frac{32b}{3} - \frac{d}{6}\right) f'''(x_0)+... \quad \text{(C.5)}$$

The values of the coefficients have to be assigned in such a manner that only the coefficient of $f'(x_0)$ is unity. All other coefficients of f and its derivatives are zero. Thus a system of linear equations is formed and can be represented in matrix form as shown in equation C.6

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 4 & 0 & -1 \\ 2 & 8 & 0 & \frac{1}{2} \\ \frac{4}{3} & \frac{32}{3} & 0 & -\frac{1}{6} \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} = \begin{Bmatrix} 0 \\ 1/h \\ 0 \\ 0 \end{Bmatrix} \quad \text{(C.6)}$$

Solving the above system gives,

$$a = \frac{1}{3h} \qquad b = -\frac{1}{20h} \qquad c = \frac{1}{4h} \qquad d = -\frac{8}{15h}$$

Thus,

$$f'(x_0) = f'_n = \frac{20f_{n+2} - 3f_{n+1} + 15f_n - 32f_{n-1}}{60h} + \mathcal{O}(h^3) \tag{C.7}$$

This formula is third-order accurate.
Note that a higher order scheme could be derived with the use of additional interior grid points.

C.2 Second Multigrid Level

The grid spacing near the wall at the second multigrid level is shown in Figure C.2

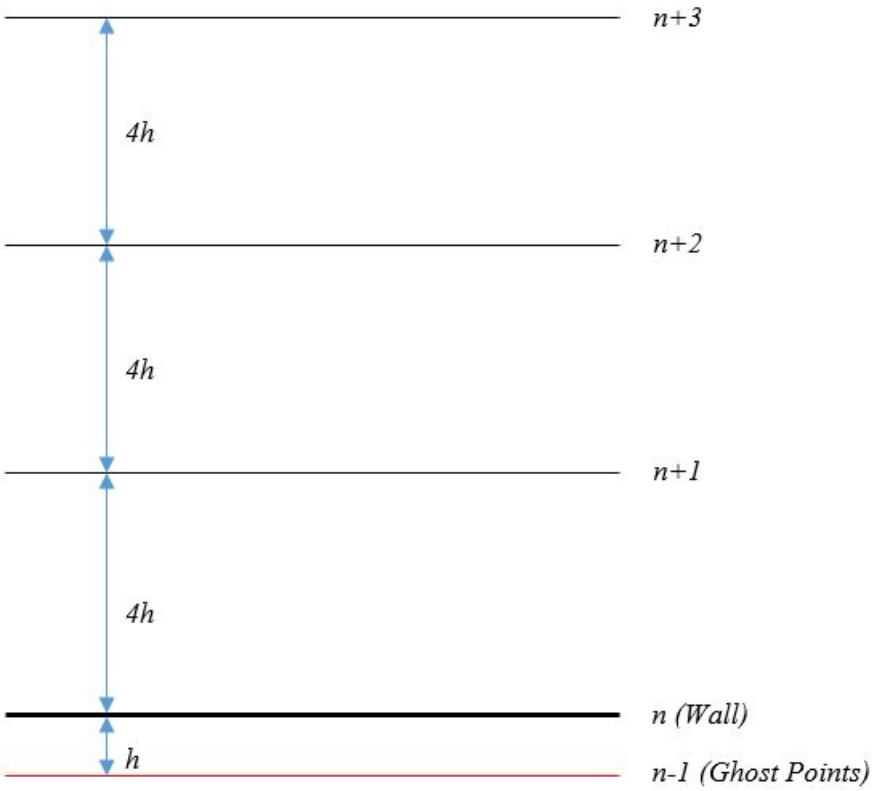


Figure C.2: Second Multigrid Level

A fourth-order accurate difference scheme is developed here. Following a similar terminology as in the case of first multigrid level,

$$f_{n+3} = f(x_0+12h) = f(x_0)+12hf'(x_0)+72h^2f''(x_0)+288h^3f'''(x_0)+864h^4f^{(iv)}(x_0)+... \tag{C.8}$$

$$f_{n+2} = f(x_0+8h) = f(x_0)+8hf'(x_0)+32h^2f''(x_0)+\frac{256h^3}{3}f'''(x_0)+\frac{512h^4}{4}f^{(iv)}(x_0)+... \tag{C.9}$$

$$f_{n+1} = f(x_0+4h) = f(x_0)+4hf'(x_0)+8h^2f''(x_0)+\frac{32h^3}{3}f'''(x_0)+\frac{32h^4}{3}f^{(iv)}(x_0)+... \tag{C.10}$$

$$f_n = f(x_0) = f(x_0) \tag{C.11}$$

$$f_{n-1} = f(x_0-h) = f(x_0) - hf'(x_0) + \frac{h^2}{2}h^2f''(x_0) - \frac{h^3}{6}f'''(x_0) + \frac{h^4}{24}f^{(iv)}(x_0)... \tag{C.12}$$

Taking four coefficients a,b,c,d,e and multiplying them to equations C.8-C.12 respectively and representing the system of equations in matrix form,

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 12 & 8 & 4 & 0 & 1 \\ 72 & 32 & 8 & 0 & 1 \\ 288 & \frac{256}{3} & \frac{32}{3} & 0 & -\frac{1}{6} \\ 864 & \frac{512}{3} & \frac{32}{3} & 0 & \frac{1}{24} \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \\ d \\ e \end{Bmatrix} = \begin{Bmatrix} 0 \\ 1/h \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{C.13}$$

Solving the above system of equations gives,

$$a = \frac{77}{9132h} \qquad b = -\frac{339}{6088h} \qquad c = \frac{639}{3044h} \qquad d = \frac{6245}{18264h} \qquad e = -\frac{384}{761h}$$

Thus, we obtain a fourth-order accurate difference scheme given by,

$$f'(x_0) = f'_n = \frac{154f_{n+3} - 1017f_{n+2} + 3834f_{n+1} + 6245f_n - 9216f_{n-1}}{18264h} + \mathcal{O}(h^4) \tag{C.14}$$

C.3 Dispersion Relation Preserving(DRP) Coefficients

The coefficients of the four-level DRP time-marching scheme are:

$b_0 = 2.3025580884 \quad b_1 = -2.4910075998 \quad b_2 = 1.5743409332 \quad b_3 = -0.385891422$

The coefficients of the seven-point central difference DRP scheme are:

$$a_1 = -a_{-1} = 0.799266427 \qquad a_2 = -a_{-2} = -0.189413142$$
$$a_3 = -a_{-3} = 0.026519952 \qquad a_0 = 0.0$$

The coefficients of the three different 7-point backward difference stencils for the DRP scheme are:

Scheme Name	a_j^{42}	a_j^{51}	a_j^{60}
a_{-2}	0.049041958	-	-
a_{-1}	-0.468840357	-0.209337622	-
a_0	-0.474760914	-1.084875676	-2.192280339
a_1	1.273274737	2.147776050	4.748611401
a_2	-0.518484526	-1.388928322	-5.108851915
a_3	0.166138533	0.768949766	4.461567104
a_4	-0.026369431	-0.28181465	-2.833498741
a_5	-	0.048230454	1.128328861
a_6	-	-	-0.203876371

Table C.1: Coefficients of backward difference DRP schemes

Bibliography

[1] C.K.W. Tam. *Computational Aeroacoustics: A Wave Number Approach*. Number 33. Cambridge University Press, 2012.

[2] C.K.W. Tam and J.C. Webb. Dispersion-Relation-Preserving Finite Difference Schemes for Computational Acoustics. *Journal of Computational Physics*, 107(2):262 – 281, 1993.

[3] R. Kosloff and D. Kosloff. Absorbing boundaries for wave propagation problems. *Journal of Computational Physics*, 63(2):363 – 376, 1986.

[4] K.W. Thompson. Time-dependent boundary conditions for hyperbolic systems, II. *Journal of Computational Physics*, 89(2):439 – 461, 1990.

[5] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA Paper*, 91-1596, 1991.

[6] C. Liu. High performance computation for DNS/LES. *Applied Mathematical Modelling*, 30(10):1143–1165, 2006.

[7] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.

[8] A. Jameson and T.J. Baker. Solution of the Euler equations for complex configurations. *AIAA Paper*, 83-1929-CP, 1983.

[9] J. Blazek, N. Kroll, R. Radespiel, and C.C. Rossow. Upwind implicit residual smoothing method for multi-stage schemes. *AIAA Paper*, 91–1532, 1991.

[10] Y. Du. *Supersonic jet noise prediction and noise source investigation for realistic baseline and chevron nozzles based on hybrid RANS/LES simulations*. PhD thesis, The Pennsylvania State University, 2011.

[11] G.B. Whitham. *Linear and Nonlinear Waves*, volume 42. John Wiley & Sons, 2011.

[12] C.K.W. Tam and Z. Dong. Radiation and outflow boundary conditions for direct computation of acoustic and flow disturbances in a nonuniform mean flow. *Journal of Computational Acoustics*, 4(02):175–201, 1996.

[13] J. Berland, C. Bogey, O. Marsden, and C. Bailly. High-order, low dispersive and low dissipative explicit schemes for multiple-scale and boundary problems. *Journal of Computational Physics*, 224(2):637–662, 2007.

[14] M.R. Visbal and D.V. Gaitonde. On the use of higher-order finite-difference schemes on curvilinear and deforming meshes. *Journal of Computational Physics*, 181(1):155–185, 2002.

[15] D.V. Gaitonde and M.R. Visbal. Padé-Type Higher-Order Boundary Filters for the Navier-Stokes Equations. *AIAA Journal*, 38(11):2103–2112, 2000.