A Thesis

entitled

Analyzing the Efficiency of
an Implicit Dual Time Stepping Solver For Computational Aeroacoustics

by

Md Navid Unjum

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the
Masters of Science Degree in Mechanical Engineering

Dr. Ray Hixon, Committee Chair

Dr. Sorin Cioc, Committee Member

Dr. Reza Rizvi, Committee Member

Dr. Amanda Bryant-Friedrich, Dean
College of Graduate Studies

The University of Toledo
August 2017
Copyright 2017, Md Navid Unjum

This document is copyrighted material. Under copyright law, no parts of this document may be reproduced without the expressed permission of the author.
Stability criteria of explicit scheme is a major issue when dealing with viscous problems. The purpose of the thesis is to evaluate the efficiency of the implicit dual time stepping scheme for aeracoustic calculations. It also investigates whether the relaxation of the stability criteria is aligned with the accuracy of the solution.

The first chapter of the thesis discusses the prospect of computational aeroacoustics and the motivation behind the research of dual time stepping. Second chapter deals with various terminology of computational aeroacoustics as well as various commonly used numerical algorithms. Chapter three consists of various methodology to use along dual time stepping to solve a problem. In chapter four, two benchmark problems from CAA workshop 3 is solved and compared with the available exact solution. The last chapter describes the findings and shortcomings of this thesis and directions for future researchers.
To my grandmothers
Acknowledgments

This work is supported by the NASA Advanced Air Transportation Technologies (AATT) Project. I would like to thank Dr. Edmane Envia of the NASA Glenn Research Center, who was the technical monitor for this work.

I would like to express my gratitude to Dr. Ray Hixon for giving me the opportunity to work in the Aeroacoustics laboratories. This work could not be done without his support.

I would like to thank all my committee members, Dr. Sorin Cioc and Dr. Reza Rizvi. Their ideas and comments have been instrumental in improving this thesis.

I would like to thank Nima Mansouri and Soroush Yazdani for answering all my questions with great patience. It really means a lot to me that you were making time to reply my email with detailed explanation even in your busiest moments. I would also like to thank Riasat and Caleb for their support and friendship over the years. Finally, Iman deserves special thanks for helping me throughout the thesis.

His in-depth knowledge about CAA algorithm and its physical significance, makes my life a lot easier when I stuck with code debugging. I wish a great future for all of them.

I am grateful to my friends Tonmoy, Ismail, Mehedi, Emran, Mubin, Nehal, Masum and Rupak for all the encouragement during the past years. Their company kept me cheerful during the frustrating hours of research life.

To my parents, sister and family members. I could never thank them enough for all the sacrifices they have made for me. Though they are 8500 miles away from me,
my day to day life circles around by thinking the precious time I have spent with them. It’s their faces that keep me moving forward as I know nothing can make them happier than my success.

Finally, thanks to my fiancee, Tonni. I am now a lot better person than I was before because of you. I truly acknowledge the sacrifices you are making to be with me. I will do everything within my capacity to ensure that it is the last time you are the one who is sacrificing most in our relationship. I love you a lot.
Contents

Abstract iii

Acknowledgments v

Contents vii

List of Tables x

List of Figures xi

1 Background 1

1.1 Introduction to CAA . . . . . . . . . . . . . . . . . . . . . . . . . 1
1.2 Discretization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
1.3 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
1.4 Objective . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

2 Basics of CAA Problem Solving Methodology 5

2.1 Analysis of the Numerics . . . . . . . . . . . . . . . . . . . . . . . . 5
2.1.1 Spatial Differencing . . . . . . . . . . . . . . . . . . . . . . . . 6
2.1.1.1 Overview of Finite Differencing . . . . . . . . . . . 7
2.1.1.2 Numerical Wavenumber . . . . . . . . . . . . . . . . 8
2.1.2 Time Integration . . . . . . . . . . . . . . . . . . . . . . . . . 13
2.1.2.1 Stability analysis . . . . . . . . . . . . . . . . . . . . . 16
2.1.2.2 Effect of numerical time integration on explicit scheme 17
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Co-efficients of DRP Schemes-Right Biased Stencils</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>Co-efficients of DRP Schemes-Left Biased Stencils</td>
<td>29</td>
</tr>
<tr>
<td>3.3</td>
<td>Co-efficients of DRP Schemes-Interior Stencils</td>
<td>30</td>
</tr>
<tr>
<td>3.4</td>
<td>Co-efficients of Third Order 2N Storage RK</td>
<td>33</td>
</tr>
<tr>
<td>4.1</td>
<td>Relation among steps per cycle, CFL and $\Delta t$ for Problem 1</td>
<td>53</td>
</tr>
<tr>
<td>4.2</td>
<td>Relation among steps per cycle and Number of Inner Sub-Iterations for Problem 1</td>
<td>53</td>
</tr>
<tr>
<td>4.3</td>
<td>Relation among steps per cycle, CFL and $\Delta t$ for Problem 2</td>
<td>69</td>
</tr>
</tbody>
</table>
List of Figures

2-1 Comparing Numerical Wavenumber with Wavenumber for Central Schemes 11
2-2 Numerical Wavespeed Comparison 12
2-3 Illustration of Time Marching 15

3-1 Schematic of derivative calculation using ghost point 34
3-2 Schematic of a computational domain 39

4-1 Grid Spacing Distribution 49
4-2 Jacobian Distribution 50
4-3 Maximum pressure distribution for 16 steps per cycle 54
4-4 Maximum pressure distribution for 24 steps per cycle 55
4-5 Maximum pressure distribution for 32 steps per cycle 56
4-6 Maximum pressure distribution for 48 steps per cycle 57
4-7 Maximum pressure distribution for 64 steps per cycle 58
4-8 Maximum pressure distribution for 128 steps per cycle 59
4-9 Maximum pressure distribution for 256 steps per cycle 60
4-10 Maximum pressure distribution for 512 steps per cycle 61
4-11 Envelope and instantaneous pressure distribution for 16 steps per cycle 62
4-12 Envelope and instantaneous pressure distribution for 24 steps per cycle 62
4-13 Envelope and instantaneous pressure distribution for 32 steps per cycle 63
4-14 Envelope and instantaneous pressure distribution for 48 steps per cycle 63
4-15 Envelope and instantaneous pressure distribution for 64 steps per cycle 64
<table>
<thead>
<tr>
<th>4-41 Affect of clustering on the mean flow: Downstream Region</th>
<th>82</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-42 Affect of clustering on the mean flow: Upstream Region</td>
<td>83</td>
</tr>
</tbody>
</table>
Chapter 1

Background

1.1 Introduction to CAA

Aeroacoustic is mostly centered around the prediction of noise radiation associated with unsteady flows. Several approaches are developed to predict the radiated noise with the flow data. As the flows behind the noise creation are generally complex, it is simplified based on the physical phenomena of sound generation. The main two categories are linear and non-linear problems which are further sub-divided into various groups. Linear group deals with the sound propagation in uniform medium or in a prescribed non-uniform medium. On the other hand, non-linear solves the sonic boom propagation through atmospheric turbulence and internal flows of thermo-acoustic cooling systems [1].

In order to solve these computational problems, accurate numerical techniques are addressed. As the Computational Fluid Dynamics (CFD) has made tremendous progress in the past decades, it is natural to utilize some of the CFD methods to solve aeroacoustic problems. But as the type, features and intent of aeroacoustic problems are widely distinct from CFD, hence CFD problem solving mechanisms are not adequate to solve CAA problems. For instance, most CFD problems are time independent whereas CAA problems are time dependent. Again, radiation boundary
conditions are of great importance to get a contamination free numerical solution, but for CFD problems, they are usually not imposed. This is why, computational methods unique for aeroacoustic are designed in the recent years. They include but are not limited to high resolution CAA techniques, artificial dissipation, better boundary conditions and development of wave number analysis [2, 3].

1.2 Discretization

Spatial differencing and temporal discretization is an indispensable part to preserve accurate wave propagation and unsteady flow characteristics. The choice of spatial discretization depends on the computational efficiency, less memory storage and implementation capability in varieties of geometries. In CAA studies, finite difference methods are preferred because of the benefit of extending to high order accuracy without much hassle. On the other hand, temporal discretization can be done through two schemes: explicit and implicit [1]. The explicit approach is easy to deal with since every difference equations has only one unknown. On the contrary, in the implicit approach, unknowns are generated by means of a simultaneous solution of the difference equations applied at all the grid points arrayed at a given time level. From the definitions, it is quite understandable that the implicit approach is more complex for setting and programming. Despite this disadvantages, implicit scheme is preferred in the places where stability criteria emerges as a big concern. For the explicit scheme, stability criteria (CFL Number) is restricted by the value of time step. The time step can not be larger than the limit set by the CFL condition, as the computer program becomes unstable in such situation. But for the implicit method, stability can be preserved with larger time steps [4].
1.3 Motivation

For inviscid CAA problems, high order explicit time stepping methods along with high order spatial differencing can predict the solution with minimized errors, if the CFL condition is fulfilled. But for problems with dominant viscous effects, high order explicit algorithm becomes ineffective. For instance, Lockard [7] used a three-dimensional wing to observe acoustic scattering. He implemented the Dispersion-Relation-Preserving (DRP) stencils with a Runge Kutta scheme to calculate the unsteady part. Since he had to cluster the grid at the tip of the wing, the grid spacing becomes extremely small which results in an infinitesimal time step to make the code stable. After 91 hours of single CPU time, the acoustic pulse was still inside the domain because of that tiny time step [5] [6].

As the experiment demonstrated a fundamental flaw in the typical explicit CAA algorithms to solve viscous Navier-Stokes equations, implicit method is usually applied for such calculations because of its larger stability limit. But a low order implicit scheme undercuts the advantage of implicit over explicit since the wave propagation characteristics of first-order implicit are much poorer than fourth-order classical Runge-Kutta (RK) scheme. This is why Shieh and Morris [6] proposed a second-order time accurate and sixth-order space accurate dual time-stepping algorithm as a method for solving viscous problems.

1.4 Objective

For CAA calculations, the priority is to ensure a transient free mean flow within the computational domain. In the dual time stepping approach, initially the steady state solver is carried out for the inner sub-iterations to generate a time accurate mean flow solution. Then this solution is used as initial condition for the next time step. The inner sub-iterations are marched with larger time step which eventually led to the
solving of viscous problems[6]. For this thesis, we are going to concentrate on solving Euler equations which is non-viscous. With the intricacy of the implicit method compared to that of explicit, the absolute first question would be: Why deal with the implicit equations in the first place then? The answer lies in our focus of interest: resolving the mean flow gradients. As for many practical unsteady calculations, the grid spacing is much smaller to resolve the mean flow than the grid spacing required to resolve the unsteady waves. The existing explicit scheme will be inefficient due to the small time step required for stability in the regions of clustered grids. In such cases, implicit dual time stepping can be useful as they relaxed the stability criteria[66].
Chapter 2

Basics of CAA Problem Solving

Methodology

2.1 Analysis of the Numerics

The governing equations of the unsteady fluid flow comprise partial derivatives with respect to both space and time. These equations are solved in order to simulate the flow. The approach that is adopted widely is to approximate the spatial derivatives and thereby obtain a set of ordinary differential equation (ODE). Later, the time derivatives are solved by a time marching method which also produces a system of difference equations \[8\]. Despite having high performance computer, calculating the derivative functions is still very costly. The analytical solutions are also very hard to derive. Due to this problem, derivatives and numerical methods are outlined in a way that computers can handle them easily. But sometimes the numerics can not resolve wave of higher frequency. Then numerical dissipation (filtering) is added to the algorithm without altering the accuracy of physical phenomena. However, the final obstacle of numerical simulation comes in the form of boundary conditions. As the flow problem is completely defined by its physical definition, the solution of the problem is as good as its boundary condition. Thus high accurate boundary conditions
are compulsory to get expected solution [9]. The focus in this chapter is to discuss various aspects of discretization, artificial dissipation and boundary conditions.

2.1.1 Spatial Differencing

In CAA, several discretization methods are used based on problem specifications over the years. For instance, if a problem involves simple geometry and boundary condition (BC), spectral and pseudo-spectral methods are very efficient. They are extensively used to discretize periodic directions such as azimuthal direction of cylindrical polar coordinate system [10]. But in case of periodic BC with temporally evolving flow, the sound generation should be treated very carefully to reach to a final conclusion [11].

Another group is working for the development of finite element and spectral element methods for solving compressible Euler and Navier-Stokes equations. Here the Discontinuous Galerkin (DG) method [12], based on high-order-polynomial basis functions, really stood out. It is more advantageous over other schemes when dealing with complex unstructured meshes. This also comes handy for analyzing full-scale DNS and LES problems [13].

Some other methods are also adopted for various reasons. The Finite Volume schemes are preferred in LES as it allows to impose global conservations principle discretely. It is done through the staggering of fluxes with respect to the conserved variables. Vortex Particle methods are also applied to CAA applications in combination with acoustic analogy [14, 15]. For problems without the interaction of vorticity and entropy waves, boundary element method can be a better substitute [1].

For this thesis, focus will be on the most popular method in CAA studies: The Finite Difference Method. This scheme is easily developed, optimized and implemented in computer programming. It has the advantage of minimizing errors with less computational cost in the research of CAA, LES and DNS. Lele [16] and Tam and
Webb [17] has contributed a lot to this scheme through their work of optimization of compact schemes and optimization of explicit schemes. In the following section, various aspects of finite differences are discussed.

2.1.1.1 Overview of Finite Differencing

As the governing equations contain derivatives of unknown functions in time and space, first it is necessary to find the way to calculate the derivative through numerical methods. To do so, flow solution are approximated numerically at discrete domain in space and time. For instance, if there is 1-D discrete domain in the x-direction with a uniform grid spacing $\Delta x$, we can calculate the spatial differencing $\frac{df}{dx}\bigg|_{x_0}$ in every point. Now Taylor series expansion is applied around the location where the derivative is expected. In this way, the derivatives of the point of interest can be expressed in terms of the values of the function at adjacent points. Denoting $i$ as a grid point counter, the grid point at $x_0$ becomes $x_i = x_0$. The neighbouring point can also be described using the same notation:

\[
x_{i-1} = x_0 - \Delta x \\
x_i = x_0 \\
x_{i+1} = x_0 + \Delta x
\]

Now the Taylor series expansions can be written as:

\[
f_{i-1} = f_i + (-\Delta x) f'_i + \frac{(-\Delta x)^2}{2} f''_i + \frac{(-\Delta x)^3}{6} f'''_i + \ldots \\
f_i = f_i \\
f_{i+1} = f_i + (\Delta x) f'_i + \frac{(\Delta x)^2}{2} f''_i + \frac{(\Delta x)^3}{6} f'''_i + \ldots
\]

(2.1)
Where, \( f' = \frac{df}{dx} \bigg|_i \)

Now to obtain the derivatives, the function values are multiplied with certain coefficients and then manipulated accordingly. With the equations in hand, three forms of first derivative can be represented as follows:

\[
\begin{align*}
\frac{\partial f}{\partial x} \bigg|_i &= \frac{f_i - f_{i-1}}{\Delta x} + O(\Delta x) \quad (2.2) \\
\frac{\partial f}{\partial x} \bigg|_i &= \frac{f_{i+1} - f_i}{\Delta x} + O(\Delta x) \quad (2.3) \\
\frac{\partial f}{\partial x} \bigg|_i &= \frac{f_{i+1} - f_{i-1}}{2\Delta x} + O(\Delta x)^2 \quad (2.4)
\end{align*}
\]

For the first equation, the information used in forming the finite difference quotient comes from the left of grid point \( i \); that is, it uses \( f_{i-1} \) as well as \( f_i \). As no information from the right of \( i \) is used, this equation is called backward difference. In the same way, the next two equations are called forward and central difference respectively.

The symbol \( O(\Delta x) \) is an expression of order of accuracy. The order of the magnitude of the truncation error is represented by this symbol. So the order of accuracy of the scheme is defined by the power that the \( \Delta x \) in the leading error term is raised to.

Hence the first two equations here are 'first order' scheme where the last equation is a 'second order' scheme[4].

\subsection{2.1.1.2 Numerical Wavenumber}

Now to compare the performance of the numerical schemes with exact solution, linear advection equation is used which is defined as:

\[
\frac{\partial u}{\partial t} = -c \frac{\partial u}{\partial x}
\]
This equation has infinite number of solutions in the form of

\[ u(x, t) = F(x - ct) \]

For analysis, a simple harmonic solution is taken as:

\[ u(x, t) = e^{i(kx - \omega t)} \]  \hspace{1cm} (2.5)

where \( k \) is wavenumber and \( \omega \) is frequency

Here the analytical derivative of the simple harmonic solution at \( t=0 \) will be

\[ \frac{\partial}{\partial x}(e^{ikx}) = ike^{ikx} \]

Now implementing the second-order central differencing scheme to calculate the numerical derivative:

\[ \frac{\partial f}{\partial x} \bigg|_{i} = \frac{f_{i+1} - f_{i-1}}{2\Delta x} \]  \hspace{1cm} (2.6)

For \( f = e^{ikx} \)

\[ \frac{\partial f}{\partial x} \bigg|_{i} = \frac{e^{i(k(x_{i}+\Delta x))} - e^{i(k(x_{i}-\Delta x))}}{2\Delta x} \]

\[ = \left( \frac{e^{ik\Delta x} - e^{-ik\Delta x}}{2\Delta x} \right) e^{ikx_{i}} \]

\[ = \left( \frac{2i \sin(k\Delta x)}{2\Delta x} \right) e^{ikx_{i}} \]

\[ = \left( \frac{i}{\Delta x} \right) (\sin(k\Delta x)) e^{ikx_{i}} \]

\[ = \left( \frac{\sin(k\Delta x)}{k\Delta x} \right) ike^{ikx_{i}} \]
It is evident that numerical derivative has some errors when compared to the analytical one. To explain this disparity, a new term 'numerical wavenumber' is introduced in [2] which is defined for the second order central differencing as:

\[(k\Delta x)^* = \sin(k\Delta x)\] (2.8)

Numerical derivative can be obtained for the biased spatial differencing in the same way. For example, numerical derivatives for first order backward differencing using points \(f_i\) and \(f_{i-1}\) and first order forward differencing using points \(f_{i+1}\) and \(f_i\) are:

\[
\left.\frac{\partial f}{\partial x}\right|_B = \left(\frac{\sin(k\Delta x)}{k\Delta x} - i\left(\frac{1 - \cos(k\Delta x)}{k\Delta x}\right)\right) ik e^{ikx} \\
= ik \left(\hat{\gamma} - i\hat{\delta}\right) f(x) \quad (2.9)
\]

\[
\left.\frac{\partial f}{\partial x}\right|_F = \left(\frac{\sin(k\Delta x)}{k\Delta x} + i\left(\frac{1 - \cos(k\Delta x)}{k\Delta x}\right)\right) ik e^{ikx} \\
= ik \left(\hat{\gamma} + i\hat{\delta}\right) f(x) \quad (2.10)
\]

From the derivative it can be seen that the real part of the numerical wavenumber are same as central differencing for all the schemes. For the biased differencing, the imaginary parts are equal and opposite. Two types of errors can be defined from the equation of numerical wavenumber: dispersion error factor \(\hat{\gamma}\) and dissipation error factor \(\hat{\delta}\):

\[
\hat{\gamma} = \frac{\sin(k\Delta x)}{k\Delta x} \quad (2.11)
\]

\[
\hat{\delta} = \frac{1 - \cos(k\Delta x)}{k\Delta x} \quad (2.12)
\]

Dispersion error illustrates waves convecting at a nonphysical speed whereas dissipation error illustrates nonphysical change of wave amplitude. From the equations it can be seen that both the factors depend on the parameter \(k\Delta x\) which can be defined
in terms of points per wavelength (P.P.W):

$$k \Delta x = \frac{2\pi}{P.P.W} \quad (2.13)$$

According to the Nyquist limit, at least two grid points are required to resolve a simple harmonic wave. Thus the limit of $k \Delta x$ becomes:

$$0 \leq k \Delta x \leq \pi \quad (2.14)$$

Figure 2-1 shows the numerical wavenumber against the analytical wavenumber for various spatial differencing. The accuracy of the scheme can be predicted depending on the closeness of $k \Delta x$ to the $(k \Delta x)^*$. As $k \Delta x$ is inversely proportional to the P.P.W, the usage of more points in calculating the spatial differencing ensures more accuracy.
The figure exhibits the superiority of DRP scheme [17] over other schemes in terms of accuracy. For DRP, it is observed that $k\Delta x$ is almost same as $(k\Delta x)^*$ for up to $k\Delta x = 0.9$. So seven mesh points are required per wavelength to resolve the wave. For sixth order central, this value of $k\Delta x$ is around 0.6, so around 10.5 mesh points are required for this scheme. Therefore, an obvious advantage is obtained by using the DRP [2].

In figure (2-2), numerical wavespeed is defined as:

$$Wavespeed = \frac{(k\Delta x)^*}{k\Delta x}$$

This numerical wavespeed graph also demonstrates the promise of high-order schemes for spatial differences. For this thesis, optimized DRP scheme is used for taking the derivatives in space.
2.1.2 Time Integration

When the discretization of spatial derivatives is done in the governing partial differential equations (PDEs), a set of nonlinear ordinary differential equations (ODEs) is obtained in the form:

$$\frac{d\bar{u}}{dt} = F(\bar{u}, t)$$  \hspace{1cm} (2.15)

where

- $u = \text{Length } N \text{ containing the discretized dependent variable on the grid}$
- $N = \text{Total number of degrees of freedom of the system obtained by multiplying the number of mesh points to the number of dependent variables}$
- $F = \text{Nonlinear function of } u$

A time marching method can solve equation (2.15) to yield a time accurate solution for an unsteady problem. But if the problem of interest is a steady flow problem, the equations converted to

$$F(u) = 0$$  \hspace{1cm} (2.16)

As the equations are non-linear, an iterative method is essential to get a satisfactory solution. For instance, Newton’s iterative method can be used for such non-linear equations. Another route is using a time marching method to integrate the unsteady equations. The marching is stopped when the solution is almost similar to that of the steady state. Time marching for steady state can be done very rapidly as the sole purpose is to remove the transient portion of the solution as soon as possible irrespective of time accuracy. This way, time marching method is useful to solve both steady and unsteady problems [8].

13
Considering the scaler part of equation (2.15) and combining some additional notations, it can be rearranged as:

\[
    u_n' = F_n = F(u_n, t_n)
\]  

(2.17)

where

\[ u' = \frac{du}{dt} \]

\[ t_n = nh \]

\[ n = \text{discrete time value} \]

\[ h = \text{time interval } \Delta t \]

Time marching methods for equation (2.15) can be classified as explicit and implicit scheme. Common forms of explicit and implicit schemes can be shown respectively as:

\[
    u_{n+1} = u_n + hu_n' \\
    u_{n+1} = u_n + hu_{n+1}'
\]

(2.18)

(2.19)

Figure 2-3 explains the two types of time marching clearly. Here the marching variable is time \( t \) and \( u \) is known for all grid points at time level \( n \). Now, by time marching, \( u \) at all points at time level \( n + 1 \) can be evaluated from the known values at time level \( n \). By the end of this calculation, \( u \) is known at every points at time level \( n + 1 \) which is used again to calculate values at time level \( n + 2 \). In this way, the final solution is obtained in desired time. Equation of explicit scheme (2.18) resembles the discussed time marching where the known values at time level \( n \) is on the right side while the unknown values at time level \( n + 1 \) is on the left side. As each difference equation contains only one unknown in explicit equation, it can be solved directly.
But for implicit, the right side consists of values from both \( n \) and \( n+1 \) levels which can not yield a solution by itself. So equation 2.19 must be written at all interior grid points that results in a system of algebraic equations and thereby the unknowns are solved simultaneously [4].

In CAA, two types of explicit finite difference schemes are widely used: Runge-Kutta schemes (RK) and Linear Multi-step schemes (LM). The mode of information propagation in time marks the difference between these two schemes. In the RK method, integration can be forwarded from any point without any prior behavior of the solution. For LM method, the solution is dependent on the previous occurrence in time and it facilitates to produce a corrected solution as time step decreases to zero. This method is also beneficial to suppress the spurious solution in time given that a very small time step is chosen. The downside of the LM scheme is that it is not self-starting because of its dependency on the previous time levels. As a result, this method has to settle for lower order for the initial few steps [1].

Implicit methods are useful for situations where suppressing some components of
the solution corresponding to the fastest eigenvalues are required. This is done when there is poorly resolved components in the solution and fast convergence to steady state solution is required. Different groups attempted different implicit methods based on their problem specifications. For steady state convergence, Collis and Lele [18] implemented implicit Euler where for unsteady calculations they adopted the second-order accurate two step implicit scheme. Gaitonde and Visbal [19] use a factored second-order scheme for introducing a high-order accurate Navier-Stokes code. Moin and Mahesh [20] discusses these issues on the basis of Direct Numerical Simulation (DNS).

2.1.2.1 Stability analysis

The effect of numerical time integration can be explored clearly if the spatial differencing is perfect. The linear advection equation is used here to conduct this analysis:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0$$ \hspace{1cm} (2.20)

For a simple harmonic wave with initial function of $u(x,t) = e^{i(kx-\omega t)}$ and a propagating speed of $c$, the perfect spatial differencing would be:

$$\frac{\partial u}{\partial x} = ike^{ikx}$$ \hspace{1cm} (2.21)

Substituting this into equation (2.20):

$$\frac{\partial u}{\partial t} = -ciku$$ \hspace{1cm} (2.22)

$$= -iwu$$ \hspace{1cm} (2.23)
where $w = ck$

Using the algorithm of explicit time marching scheme, it can be re-written as:

\[
\frac{\partial u^n}{\partial t} = -iw u^n \\
\frac{u^{n+1} - u^n}{\Delta t} = -iw u^n \\
u^{n+1} = u^n - iw \Delta t u^n
\]  

For the exact solution, that can be expressed as:

\[
\frac{u^{n+1}}{u^n} = e^{-i\omega \Delta t}
\]  

The stability condition of the code is satisfied as long as the amplitude of the error is not amplified.

\[
\left| \frac{u^{n+1}}{u^n} \right| \leq 1
\]  

\section{2.1.2.2 Effect of numerical time integration on explicit scheme}

Stability of the time marching scheme can be analyzed with the above equations. Defining an 'effective frequency' as:

\[
e^{-i(\tilde{\omega} \Delta t)} = 1 - i\omega \Delta t
\]  

The amplitude of the wave for the time marching becomes:

\[
\left| e^{-i(\tilde{\omega} \Delta t)} \right| = \sqrt{1 + (\omega \Delta t)^2}
\]
This equation exhibits the increase of the wave amplitude at every time step independent of the value of $\Delta t$. So the explicit time marching amplifies the waves exponentially which eventually leads to instability.

2.1.2.3 Effect of numerical time integration on implicit scheme

Following the same procedure as explicit scheme, for a perfect space differencing, implicit time marching scheme can be written as:

$$\frac{\partial u}{\partial t} \bigg|_{n+1} = -i\omega u^{n+1}$$

$$\frac{u^{n+1} - u^n}{\Delta t} = -i\omega u^{n+1}$$

$$u^{n+1} = u^n - i\omega \Delta t u^{n+1}$$

$$\frac{u^{n+1}}{u^n} = 1 - i\omega \Delta t \frac{u^{n+1}}{u^n}$$

$$e^{-i\omega \Delta t} = 1 - i\omega \Delta t e^{-i\omega \Delta t}$$

$$e^{-i\omega \Delta t} (1 + i\omega \Delta t) = 1$$

$$e^{-i(\omega \Delta t)} = \frac{1}{1 + i(\omega \Delta t)}$$

For which the amplitude can be defined as:

$$\left| e^{-i(\omega \Delta t)} \right| = \sqrt{\frac{1}{1 + (\omega \Delta t)^2}}$$

The equation explains the implicit code: with the increase of $\Delta t$, the wave will decrease in amplitude which ensures stability. These characteristics of explicit and implicit time marching is considered when choosing time marching method for this thesis.

Here, a second order accurate implicit linear multi-step scheme is adopted along with third order RK 2N storage scheme. The schemes will be discussed in the following chapter.
2.1.3 Implementation of Artificial Dissipation

Though the previous section shows that the implicit scheme is unconditionally stable, yet in reality the stability limit is restricted. This can be understood clearly when encountered with a strongly nonlinear case: flows with shocks. As we use discretization methods to solve shock problems, numerics can not resolve the appeared scales of motion. It is illustrated by the non-linear interactions of the convective terms in the governing equations. For instance, if two waves interact, there will be a higher and a lower frequency of wave. The finite difference schemes has the capacity of resolving the lower frequency smoothly. But the schemes provide a very low resolution for high wave numbers. When the higher wave numbers go beyond the capacity of the mesh resolution, it produces erroneous low wave numbers as well as increased higher frequencies. If these spurious waves can not be controlled, the solution will be contaminated and eventually becomes unstable [9].

The widely accepted way to deal with the spurious waves is to introduce some form of numerical dissipation. This can be done in many ways including using the upwind scheme where the schemes have inherent dissipation. Steger and Warming [21], Roe [22] and Van Leer [23] used a decomposition of the flux vectors where every element can be differenced in a stable, upwind manner. Several types of upwind schemes are also developed by Yazdani et. al [24], Hixon and Turkel [25], MacCormack [26], Lockard et. al. [27], Zingg [28], Zhuang and Chen [29, 30] and Li [31]. These schemes can produce sharp oscillation free shocks without any external dissipation. But a disadvantage of these schemes is that there is significant amount of damping for the resolved high wave numbers. This problem can be solved with either an increased grid spacing or optimized scheme. However, truncation error of the scheme increases with the increase of grid spacing.

Another popular method is to add non-physical damping terms to the governing
equations to change the behavior of the discontinuity. They have the capability of eliminating high-frequency wave numbers. Here the spatial derivatives are calculated by a central differencing scheme and artificial dissipation will be utilized for the removal of unresolved waves. Pulliam [32] introduced dissipation model for ensuring the stability and accuracy for the Euler equation. Tam and webb [17] proposed dissipation scheme which act on predominately on high wave number data without affecting the already resolved low wave numbers. The advantage of these family of operators is that the amount of dissipation can be selected according to the problem. Mattson et al. [33] used different set of operators to preserve the accuracy of high order finite difference operator. Hixon et al. [34] also proposed an way to utilize stronger dissipation in case of shock capturing while using high-order explicit dissipation for smooth non-linearity.

A dissipative term as a function of even derivatives of $f$ is given here as:

$$D_2(f_i) = \frac{A}{4} (f_{i+1} - 2f_i + f_{i-1})$$

Where, $A$ is a scaling parameter

This term can be added to the non-linear governing equation as:

$$\frac{\partial f}{\partial t} + f \frac{\partial f}{\partial x} - \mu \frac{\partial^2 f}{\partial x^2} - D_2(f) = 0$$

2.1.4 Boundary Conditions for CAA

To accurately predict the flow fluctuations in CAA, a grid is required in the area of interest and the equations are solved based on that grid. A boundary condition(BC) is used to change the governing equations from general to a particular flow problem. But the implementation and evaluation of boundary conditions are very complicated to deal with. There is always a high possibility that these boundary conditions will
generate spurious fluctuations and thereby contaminate the entire solution [35]. This is why, CAA requires deploying robust and highly accurate boundary conditions to predict the unsteady solutions with an accepted level of error.

Pulliam [9] mentioned several things to keep in mind when boundary conditions are handled:

1. BC needs to be specified as such the physical definition of the flow is covered.

2. Physical conditions should be expressed in terms of problem related mathematics. Also the number of BC needs to evaluated.

3. The differencing scheme may need excess BC than provided by the physical definition of the flow.

4. Stability and accuracy should be checked to determine whether the combination of numerical and boundary schemes are compatible with each other.

5. The implementation of the BC should be such so that the flow code could handle a wide-ranging geometry and topology.

There are varieties of boundary conditions for CAA. Tam [36] mentions six categories of boundary conditions in a broad sense:

1. Radiation BC

2. Outflow BC

3. Wall BC

4. Impedance BC

5. Radiation/Outflow BC with incoming acoustics or vorticity waves

6. Radiation BC for ducted environments
Again, a classification in terms of the type of BC is presented by Lele et. al. [1] which is described as follows:

1. Linearized Boundary Conditions:

   Boundary conditions for the Euler equation is developed by linearizing about a comparatively uniform flow. Three types of methods are found here:

   - **Non-reflecting BC**, here the focus is fixed to zero any incoming waves at the computational boundary. Engquist and Majda [37, 38] developed this type of BC for wave equation where Giles [39] introduced that for Euler equations.

   - **Radiation BC**, it assumes that the BC are located far from any source of disturbances. Asymptotic solutions are utilized to specify BC. Initially the idea was presented by Bayliss and Turkel [40, 41] for simple wave equation. Later Tam and Webb [17] extended this analysis. Tam and Dong [42] also suggested a non-uniform boundary conditions. Tam and Webb [17] proposed a 'ghost' point strategy to implement the radiation BC in finite difference calculation.

   - **Perfectly Matched Layer (PML)**, here a buffer region is used to enclose the domain in order to reduce reflection of outgoing waves. Hu [43] formulated this BC for Euler equations. Lately, Hagstrom and Nazarov [44] modified the PML to non-uniform parallel shear flows.

   These above mentioned approaches involve some degree of approximation as they are not entirely non-reflecting.

2. Non-Linear Boundary Conditions:

   - **Thompson’s Approach**: Though the linearized BC yields some advantages, yet it does not allow accurate non-reflectivity for non-uniform flow. For
the non-linear problem, an accurate nonlinear BC can be attained only for a one dimensional flow. Here the coordinate in the direction normal to the boundary is considered as spatial coordinate. For 1-D Euler equations, a set of characteristics equations can be found. Thompson [45, 46], Poinsat and Lele [47] used the characteristic equations to derive boundary conditions. They are also referred as characteristic BC.

3. Buffer Zone Techniques:
When situations such as ‘nonuniform flow crossing the artificial boundary’ or ‘large amplitude disturbances propagating out of the boundary’ arises, both the linear and the non-linear boundary condition performs inadequately. To tackle this issue, a number of absorbing layers are introduced to increase the efficiency of the artificial boundary conditions or eliminate the need of an artificial BC.

- **Damping and artificial convection:** Here the disturbances are damped before they interact with the boundary conditions. This is done through the addition of artificial dissipation or to add a linear damping co-efficient to the governing equations [1].

- **Grid Stretching:** Here the requirement of BC is taken out by mapping the infinite domain to a finite one. However, the efficiency of this techniques is largely dependent on the appropriate mapping. Rai and Moin [48] combined grid stretching and filtering as an absorbing layer for incompressible flows. This is later extended to the compressible aeroacoustic problems by Lele et. al. [49].

4. Wall Boundary Conditions:
The boundary conditions for inviscid flow at solid wall is that the velocity component normal to the wall is zero. This is adequate for evaluating solution for Euler equations as well as for computation where low order finite difference
scheme is used \cite{50}. But this condition is invalid for high order FD scheme as the order of the difference equation is greater than Euler equations.

Wall boundary conditions for high order FD scheme is very difficult for two reasons. First, it requires more BC than the physical BC to get a unique solution. Secondly, in the discretized system, flow variable at a mesh point is governed by an equation. To satisfy the BC at the wall mesh points, there will be too many equations in comparison with the unknowns. Tam and Dong \cite{51} proposed an way to implement wall boundary condition for high order accuracy. They used backward difference stencils as the wall is approached and thereby provide extra BC. To get rid of the problem of extra unknowns, they used ghost points which are defined mesh points outside the computational domain.

To solve acoustic wave scattering problem, Chung and Morris \cite{52} developed an Impedance Mismatched Method (IMM). Here the solid bodies are replaced with a new fluid medium. As the entire computational domain will be regarded as a continuous fluid region, it makes the coding of the problem very simple. However, this condition can only be use for inviscid flows.

Evaluation of the performance of various boundary conditions can be found here \cite{35,53,54}. 


Chapter 3

Implementation and Analysis of Algorithm

3.1 Governing Equation

Here the governing equations are presented in order to analyze the implementation of various algorithm. Quasi 1-D Euler equations are chosen as it is convenient to describe both the acoustic and flow fields assuming the viscosity has very little impact on the flow. The quasi 1-D Euler can be written in conservation form as:

\[ \frac{\partial \vec{Q}}{\partial t} + \frac{\partial \vec{F}}{\partial x} - D(\vec{Q}) = \vec{S} \]  

where the vectors \( \vec{Q}, \vec{F}, D \) and \( S \) are defined as:

\[ Q = \begin{pmatrix} \rho \\ \rho u \\ E_{tot} \end{pmatrix} \]

\[ F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E_{tot} + p) \end{pmatrix} \]
\[
S = -\frac{1}{A} \frac{\partial A}{\partial x} \begin{cases}
\rho u \\
\rho u^2 \\
u (E_{tot} + p)
\end{cases}
\]

where \( \rho \) is the flow density, \( u \) is the velocity, \( p \) is the pressure and \( E \) is the total energy. In these equations,

\[
p = (\gamma - 1) \left( E_{tot} - \frac{\rho u^2}{2} \right)
\]

\[
\gamma = 1.4
\]

Here, \( A \) is the area and \( D \) is an explicit artificial dissipation operator.

To specify the boundary conditions for the problem, stagnation pressure, density and temperature are also defined as:

\[
p_0 = p \left( 1 + \frac{\gamma - 1}{2} M^2 \right)^{\frac{\gamma}{\gamma - 1}}
\]

\[
\rho_0 = \rho \left( 1 + \frac{\gamma - 1}{2} M^2 \right)^{\frac{1}{\gamma - 1}}
\]

\[
T_0 = T \left( 1 + \frac{\gamma - 1}{2} M^2 \right)
\]

\[
M = \frac{u}{c}
\]

\[
c = \sqrt{\frac{\gamma p}{\rho}}
\]

### 3.2 Dual Time Stepping

This algorithm is first proposed by Jameson [5] to tackle issues with unsteady flow problem over airfoils and wings. Later Venkateswaran et. al [54] conducted a study of this algorithm with pre-conditioning. This section follows the work of Shieh and Morris [6] to describe dual time stepping.
To analyze the algorithm, equation 3.1 is written in a compact form as:

\[
\frac{\partial Q}{\partial t} = -R(Q) \quad (3.3)
\]

where \( R \) is represented as the residual of the fluxes. Now with the introduction of a fictitious time, \( \tau \) the equations can be recast as:

\[
\frac{\partial Q}{\partial \tau} = -\left(\frac{\partial Q}{\partial t} + R(Q)\right) = -R^*(Q) \quad (3.4)
\]

Here, \( R^* \) is the new residual which represents physical time derivative along with the flux vectors.

So, The fundamental idea behind this algorithm is:

1. Residual \( R \) of the compact equation 3.3 is replaced by the new residual \( R^* \).

2. The reformulated governing equations 3.4 can be directed to steady state by marching in the fictitious time. An effective steady state flow solver is essential for this purpose.

3. When the artificial steady state is reached, by definition the derivative of \( Q \) with respect to \( \tau \) converts to zero. In this way, the original governing equation 3.3 is recovered.

4. The physical time derivative part is evaluated using an multi-step implicit scheme.

5. Runge-Kutta (RK) scheme is used to iterate the solution in a non-physical time \( \tau \) between each physical time step. The iteration continues unless a pre-defined converged solution is attained.
6. The converged solution becomes the correct solution for each physical time step

So the obvious gain from the method is: the problem is solved in a series of steady state calculation in the 'time like' domain rather than solving in the 'physical domain'. It facilitates us to obtain a time accurate solution in each physical time step.

### 3.3 Implemented Schemes

#### 3.3.1 Spatial Differencing

DRP Scheme [17] is used for the calculation of spatial derivative in equation 3.1. The approximation of the first derivative $\frac{\partial F}{\partial x}$ at the $i^{th}$ node of a uniform grid is defined as:

$$\frac{\partial F}{\partial x} \approx \frac{1}{\Delta x} \sum_{j=-N}^{M} \alpha_j F_{i+j} \quad (3.5)$$

where,

- $i$ = Node counter ranges from 1 to $n$
- $M$ = Values of $F$ to the right of $i^{th}$ node
- $N$ = Values of $F$ to the left of $i^{th}$ node
- $\Delta x$ = Grid spacing for a uniform grid
- $\alpha_j$ = Co-efficient for DRP schemes

The DRP Stencils are provided herewith:
Table 3.1: Co-efficients of DRP Schemes-Right Biased Stencils

<table>
<thead>
<tr>
<th></th>
<th>Point i=1</th>
<th>Point i=2</th>
<th>Point i=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_{-2}$</td>
<td></td>
<td></td>
<td>+0.049041958</td>
</tr>
<tr>
<td>$\alpha_{-1}$</td>
<td>-2.192280339</td>
<td>-0.209337622</td>
<td>-0.468840357</td>
</tr>
<tr>
<td>$\alpha_0$</td>
<td>-4.748611401</td>
<td>+2.147760500</td>
<td>+1.273274737</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>-5.108551915</td>
<td>-1.388928322</td>
<td>-0.518484526</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>+4.461567104</td>
<td>+0.768949766</td>
<td>+0.166138533</td>
</tr>
<tr>
<td>$\alpha_3$</td>
<td>-2.833498741</td>
<td>-0.281814650</td>
<td>-0.026369431</td>
</tr>
<tr>
<td>$\alpha_4$</td>
<td>+1.128328861</td>
<td>+0.048230454</td>
<td></td>
</tr>
<tr>
<td>$\alpha_5$</td>
<td>-0.203876371</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: Co-efficients of DRP Schemes-Left Biased Stencils

<table>
<thead>
<tr>
<th></th>
<th>Point i=n</th>
<th>Point i=n-1</th>
<th>Point i= n-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_{-6}$</td>
<td>+0.203876371</td>
<td>-0.048230454</td>
<td></td>
</tr>
<tr>
<td>$\alpha_{-5}$</td>
<td>-1.128328861</td>
<td>-0.48230454</td>
<td></td>
</tr>
<tr>
<td>$\alpha_{-4}$</td>
<td>+2.833498741</td>
<td>+0.281814650</td>
<td>+0.026369431</td>
</tr>
<tr>
<td>$\alpha_{-3}$</td>
<td>-4.461567104</td>
<td>-0.768949766</td>
<td>-0.166138533</td>
</tr>
<tr>
<td>$\alpha_{-2}$</td>
<td>+5.108551915</td>
<td>+1.388928322</td>
<td>+0.518484526</td>
</tr>
<tr>
<td>$\alpha_{-1}$</td>
<td>-4.748611401</td>
<td>-2.147760500</td>
<td>-1.273274737</td>
</tr>
<tr>
<td>$\alpha_0$</td>
<td>+2.192280339</td>
<td>+1.084875676</td>
<td>+0.474760914</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td></td>
<td>+0.209337622</td>
<td>+0.468840357</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td></td>
<td></td>
<td>-0.049041958</td>
</tr>
</tbody>
</table>
3.3.2 Linear Multi-Step Implicit Scheme

The physical time derivatives of the equation (3.4) can be calculated by either a two-point or a three-point backward differencing scheme which results in two types of implicit linear multi-step methods.

First order accurate:
\[
\frac{\partial Q}{\partial t} = \frac{Q^{n+1} - Q^n}{\Delta t}
\]  

Second order accurate:
\[
\frac{\partial Q}{\partial t} = \frac{3Q^{n+1} - 4Q^n + Q^{n-1}}{2\Delta t}
\]  

where, \( n \) represents the physical time step.

From the equation it is noticeable that the scheme is not a self starting one. The prior time step solution is used as an approximation of initial condition for the current physical time step. A backward three-point differencing scheme is suggested.
to approximate this initial condition:

\[ Q^* = Q^m + \frac{3Q^m - 4Q^{m-1} + Q^{m-2}}{2} \]  \hspace{1cm} (3.8)

However, as this initial condition is just a value for driving the solution to steady state, to develop a more computationally efficient code, this thesis will be using a modified one: \( Q^* = Q^m \)

### 3.3.3 Artificial Dissipation Operator

The \( D \) in equation 3.1 represents explicit artificial dissipation operator. Here, tenth order dissipation is used which is developed by Kennedy and Carpenter [55].

**10th Order Dissipation:**

\[
D_{10} = \frac{1}{1024}
\begin{bmatrix}
-1 & 5 & -10 & 10 & -5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & -26 & 55 & -60 & 35 & -10 & 1 & 0 & 0 & 0 & 0 & 0 \\
-10 & 55 & -126 & 155 & -110 & 45 & -10 & 1 & 0 & 0 & 0 & 0 \\
10 & -60 & 155 & -226 & 205 & -120 & 45 & -10 & 1 & 0 & 0 & 0 \\
-5 & 35 & -110 & 205 & -251 & 210 & -120 & 45 & -10 & 1 & 0 & 0 \\
1 & -10 & 45 & -120 & 210 & -252 & 210 & -120 & 45 & -10 & 1 & 0 \\
0 & 1 & -10 & 45 & -120 & 210 & -252 & 210 & -120 & 45 & -10 & 1 \\
0 & 0 & 1 & -10 & 45 & -120 & 210 & -251 & 205 & -110 & 35 & -5 \\
0 & 0 & 0 & 1 & -10 & 45 & -120 & 205 & -226 & 155 & -60 & 10 \\
0 & 0 & 0 & 0 & 1 & -10 & 45 & -110 & 155 & -126 & 55 & -10 \\
0 & 0 & 0 & 0 & 0 & 1 & -10 & 35 & -60 & 55 & -26 & 5 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -5 & 10 & -10 & 5 & -1
\end{bmatrix}
\]
3.3.4 Explicit Time Marching Scheme

In equation 3.7, Runge-Kutta (RK) scheme is used to iterate the numerical solution in a fictitious time $\tau$. Third-order 2N-storage RK scheme is used here. This scheme is used to analyze the performance of dual time stepping with the implementation of boundary conditions.

- **Third-Order 2N Storage RK:**

  Williamson [60] and Fyfe [61] devised low-storage RK schemes by leaving essential information in the storage register. For consecutive stages, it saves the data onto the same register without zeroing the earlier values. Exploiting this technique, they proposed the N-stage algorithm as:

  \[
  dU_n = A_n dU_{n-1} + hF(U_n) \\
  U_n = U_{n-1} + B_n dU_n
  \]

  where,

  $n = 1$ to $N$ stage

  $h =$ Time Step

  In the equation, as $A_1 = 0$, the algorithm is self starting. And with the storage of $dU$ and $U$, this equation converts to a $2N$ storage algorithm. For a third-order RK scheme, it can be written as:

  \[
  qRK^{(1)} = \alpha_1 qRK^{(0)} + \Delta\tau \left( \frac{\partial}{\partial\tau} \left(Q_i^{(0)}\right) \right) \\
  Q^{(1)} = Q^{(0)} + \beta_1 qRK^{(1)} \\
  qRK^{(2)} = \alpha_2 qRK^{(1)} + \Delta\tau \left( \frac{\partial}{\partial\tau} \left(Q_i^{(1)}\right) \right) \\
  Q^{(2)} = Q^{(1)} + \beta_2 qRK^{(2)} \\
  qRK^{(3)} = \alpha_3 qRK^{(2)} + \Delta\tau \left( \frac{\partial}{\partial\tau} \left(Q_i^{(2)}\right) \right)
  \]
\[ Q^{(3)} = Q^{(2)} + \beta_3 qRK^{(3)} \]  

(3.10)

Table 3.4: Co-efficients of Third Order 2N Storage RK

| $\alpha_1$ | 0 |
| $\alpha_2$ | $-\frac{5}{12}$ |
| $\alpha_3$ | $-\frac{153}{128}$ |
| $\beta_1$ | $\frac{1}{12}$ |
| $\beta_2$ | $\frac{13}{16}$ |
| $\beta_3$ | $\frac{14}{15}$ |

The dual time stepping impose a restriction on the value of $\Delta \tau$. This is done to avoid stability problems which may arise when the fictitious time step surpasses the physical time step. For this thesis, the following equation is maintained for the value of $\Delta \tau$:

\[ \Delta \tau < \frac{2}{3} \left( CFL \right) \Delta t \]

3.3.5 Introduction of Ghost Point

This section discusses the importance of ghost point for implementing the boundary conditions. The ghost points are fictitious points with no physical meaning. It is added adjacent to the boundary to obtain a correct and smooth result for the imposed boundary condition. It facilitates to implement boundary condition that modify the interior equations at the boundary. The ghost point strategy originates from Tam and Dong [51] which later implemented by Hixon [56]. The purpose of the ghost point is to apply correction to some normal derivatives and thereby set expected flow condition at the boundary point based on the interior flow equation. The expected flow conditions are expressed in terms of time derivatives of the flow variables. 1-D
Euler equation in vector form is used to provide an example of how ghost point can be used to impose boundary condition:

\[ Q_t + F_x = 0 \]  

(3.11)

For using the strategy of ghost point, initially the value of ghost point is required. To calculate the value, a fully one-sided difference scheme at the boundary is set equal to the difference scheme when the ghost point is use. A second order explicit difference is used to demonstrate the process following the work of Hixon et.al. [57]

\[
\frac{\partial F}{\partial x} \bigg|_{1,OSD} = \left( \frac{-F_3 + 4F_2 - 3F_1}{2\Delta x} \right) \\
\frac{\partial F}{\partial x} \bigg|_{1,GPT} = \left( \frac{F_2 - F_0}{2\Delta x} \right) \\
\frac{\partial F}{\partial x} \bigg|_{1,OSD} = \frac{\partial F}{\partial x} \bigg|_{1,GPT} \\
\left( \frac{-F_3 + 4F_2 - 3F_1}{2\Delta x} \right) = \left( \frac{F_2 - F_0}{2\Delta x} \right) \]

\[ F_0 = -F_3 + 3F_2 - 3F_1 \]  

(3.12)
To conduct the initial calculation of the interior equations at the boundary, this value is utilized. Now one-sided difference stencils are used to find the time derivative of the flow parameter at the boundary as:

\[ Q_{t, \text{noBC}} = -F_{x, \text{noBC}} \] (3.13)

Then the time derivative of the flow equation is evaluated by using the boundary condition. The difference between these two time derivatives will provide the correction value:

\[ \Delta Q_{t, \text{BC}} = Q_{t, \text{BC}} - Q_{t, \text{noBC}} = -F_{x, \text{BC}} + F_{x, \text{noBC}} = -(F_{x, \text{BC}} - F_{x, \text{noBC}}) = -\Delta F_{x, \text{BC}} \]

\[ \Delta F_{x, \text{BC}} = -\Delta Q_{t, \text{BC}} \] (3.14)

Now this value is used to evaluate the modified ghost point and also set the corrected flow variables. Later the corrected time derivatives are also calculated based on the modified flow variables.

\[ F_{x, \text{corrected}} = F_{x, \text{noBC}} + \Delta F_{x, \text{BC}} \]

\[ Q_{t, \text{corrected}} = -F_{x, \text{corrected}} \] (3.15)

In the current example, as a second-order finite difference equation is used, the change in the flow variable at ghost point, modifies only the boundary derivative. But for the thesis, 7-point DRP scheme is used, which affects three derivatives.
3.3.6 Boundary Condition Specification: Thompson’s Approach

Thompson [45, 46] illustrates the decomposition of hyperbolic equations into wave modes of velocity. He also lays out the way to evaluate the boundary condition for the incoming waves. Through a 1-D characteristic analysis, propagating direction of the wave can be determined with respect to the computational domain. Thompson’s approach is described here following the analysis of Hixon et. al [35]:

Quasi 1-D Euler equations in vector form is chosen to explain the approach:

\[ Q_t + [A] Q_x = S \]  \hspace{1cm} (3.16)

Where, \( S \) is a source term and

\[ Q = \begin{bmatrix} \rho \\ u \\ p \end{bmatrix} \]

and the \([A]\) matrix is:

\[
[A] = \begin{bmatrix} u & \rho & 0 \\ 0 & u & \frac{1}{\rho} \\ 0 & \gamma p & u \end{bmatrix}
\]

An eigen decomposition of the \([A]\) matrix is defined as:

\[ [A] = [T][\Lambda][T]^{-1} \]  \hspace{1cm} (3.17)

Now by applying this similarity transform, equation [3.16] can be transformed into
a 'diagonalized' equation as:

\[
[T]^{-1} Q_t + [\Lambda] [T]^{-1} Q_x = [T]^{-1} S
\]

\[
\dot{Q}_t + [\Lambda] \dot{Q}_x = \dot{S}
\]

(3.18)

Where the time derivative can be evaluated as:

\[
\begin{align*}
\begin{cases}
p_t - c^2 p_t \\
p_t + \rho c u_t \\
p_t - \rho c u_t
\end{cases}
\end{align*}
\]

(3.19)

And the diagonal matrix consisting of the eigenvalues of \([A]\) is expressed as:

\[
[\Lambda] = \begin{bmatrix}
u & 0 & 0 \\
0 & u + c & 0 \\
0 & 0 & u - c
\end{bmatrix}
\]

(3.20)

Comparing equation 3.18 with the linear advection equation, three types of wave can be classified as entropy wave \((A_1)\), a downstream running acoustic wave \((A_2)\) and an upstream running acoustic wave \((A_3)\).

Now the time derivatives of these waves can be evaluated from the time derivative of the flow variable:

\[
\begin{align*}
\frac{\partial A_1}{\partial t} &= \frac{\partial p}{\partial t} - c^2 \frac{\partial \rho}{\partial t} \\
\frac{\partial A_2}{\partial t} &= \frac{\partial p}{\partial t} + \rho c \frac{\partial u}{\partial t} \\
\frac{\partial A_3}{\partial t} &= \frac{\partial p}{\partial t} - \rho c \frac{\partial u}{\partial t}
\end{align*}
\]

(3.21)
Equation 3.21 can also be expressed as:

\[
\frac{\partial p}{\partial t} = \frac{1}{2} \left( \frac{\partial A_2}{\partial t} + \frac{\partial A_3}{\partial t} \right)
\]

\[
\frac{\partial u}{\partial t} = \frac{1}{2\rho c} \left( \frac{\partial A_2}{\partial t} - \frac{\partial A_3}{\partial t} \right)
\]

\[
\frac{\partial \rho}{\partial t} = \frac{1}{2c^2} \left( \frac{\partial A_2}{\partial t} + \frac{\partial A_3}{\partial t} - 2 \frac{\partial A_1}{\partial t} \right)
\]

(3.22)

3.3.6.1 Imposing boundary condition

The methods of implementing Thompson boundary condition for this thesis is discussed in this section:

1. Calculating boundary values with one-sided differencing:

   Initially the time derivatives of the flow variables are calculated without the application of any boundary conditions. As One sided DRP stencils (Table 3.1, 3.2) are used for the calculation of boundary values, it is denoted with subscript 'noBC'. Using equation 3.21, the amplitude of the time derivatives of the waves is found:

   \[
   \left. \frac{\partial A_1}{\partial t} \right|_{noBC} = \left. \frac{\partial p}{\partial t} \right|_{noBC} - c^2 \left. \frac{\partial \rho}{\partial t} \right|_{noBC}
   \]

   \[
   \left. \frac{\partial A_2}{\partial t} \right|_{noBC} = \left. \frac{\partial p}{\partial t} \right|_{noBC} + \rho c \left. \frac{\partial u}{\partial t} \right|_{noBC}
   \]

   \[
   \left. \frac{\partial A_3}{\partial t} \right|_{noBC} = \left. \frac{\partial p}{\partial t} \right|_{noBC} - \rho c \left. \frac{\partial u}{\partial t} \right|_{noBC}
   \]

   (3.23)

2. Specifying the inward and outward propagating waves:

   According to Thompson’s approach, the amplitude of the incoming waves into the computational domain is modified to set the boundary conditions. And for the outgoing waves, as it is dependent on the inside variables of the domain, its amplitudes are left at the primary values.
In figure 2, the upstream x-boundary is taken as the reference inflow boundary while the downstream x-boundary is taken as outflow boundary for the analysis. The sign of the wave in equation 3.20 is a clear indication of whether the wave is an incoming wave or an outgoing one. In the case of inflow boundary, positive eigenvalue represents a entering wave into the domain and its value should be fixed. On the other hand, a negative value indicates a leaving wave which does not require any modification. Four types of boundary conditions can be derived based on this idea:

(a) For Upstream X Boundary

- **Supersonic Inflow**: 
  \[ u > 0, u - c > 0 : A_1, A_2, A_3 \] all are incoming and three conditions must be explicitly defined for them.

- **Subsonic Inflow**: 
  \[ u > 0, u - c < 0 : A_1, A_2 \] are incoming while \( A_3 \) is outgoing. Two conditions is required for this case.

(b) For Downstream X Boundary
• **Supersonic Outflow:**

\[ u > 0, \, u - c > 0 : \, A_1, \, A_2, \, A_3 \text{ all are outgoing, thereby this case does not require any specified condition.} \]

• **Subsonic Outflow:**

\[ u > 0, \, u - c < 0 : \, A_1, \, A_2 \text{ are outgoing where } A_3 \text{ is incoming. So just one condition is suffice for this problem.} \]

3. **Modifying the incoming waves to incorporate with desired boundary conditions:**

Two types of cases are dealt in this thesis:

• **Subsonic Outflow:**

From the previous analysis, it is clear that \( A_3 \) needs specification whereas \( A_1 \) and \( A_2 \) can be set equal to the amplitude value without boundary conditions:

\[
\begin{align*}
\frac{\partial A_1}{\partial t} \bigg|_{BC} &= \frac{\partial A_1}{\partial t} \bigg|_{noBC} \\
\frac{\partial A_2}{\partial t} \bigg|_{BC} &= \frac{\partial A_2}{\partial t} \bigg|_{noBC}
\end{align*}
\]

For this analysis, a pre-defined static pressure is used as the outflow boundary condition. \( A_3 \) should be set such as it satisfies the boundary value.

From equation 3.22, the incoming acoustic wave can be set as:

\[
\frac{\partial A_3}{\partial t} = 2 \frac{\partial p}{\partial t} - \frac{\partial A_2}{\partial t}
\]

\[
\frac{\partial A_3}{\partial t} \bigg|_{BC} = 2 \left( \frac{P_{st} - P_{current}}{\Delta t} \right) - \frac{\partial A_2}{\partial t} \bigg|_{BC}
\]

Where,

\( P_{st} = \) Static Pressure at the outflow boundary

\( P_{current} = \) Pressure at the outflow boundary
\[ \Delta t = \text{Time Step} \]

- **Subsonic Inflow**

  For subsonic inflow, as \( A_1 \) and \( A_2 \) are incoming, the process of determining the amplitude are also detailed here:

  For the outgoing acoustic wave \( A_3 \), the value is kept as the original one:

  \[
  \frac{\partial A_3}{\partial t} \bigg|_{BC} = \frac{\partial A_3}{\partial t} \bigg|_{noBC}
  \]

  For this analysis, a pre-defined stagnation pressure and density is used at the inflow boundary. In the equation 3.2, stagnation pressure is defined as:

  \[
  p_0 = p \left( 1 + \frac{\gamma - 1}{2} M^2 \right)^{\frac{\gamma}{\gamma - 1}}
  \]

  \[
  p_0 = p_0(p, u, \rho)
  \]

  \[
  \Delta p_0 = \frac{\partial p_0}{\partial p} \Delta p + \frac{\partial p_0}{\partial u} \Delta u + \frac{\partial p_0}{\partial \rho} \Delta \rho \quad (3.24)
  \]

  From equation 3.22, the primitive variable can be expressed as:

  \[
  p = \frac{1}{2} (A_2 + A_3)
  \]

  \[
  \rho = \frac{1}{2c^2} (A_2 + A_3 - 2A_1)
  \]

  \[
  u = \frac{1}{2\rho c} (A_2 - A_3)
  \]

  \[
  \Delta p = \frac{\partial p}{\partial A_1} \Delta A_1 + \frac{\partial p}{\partial A_2} \Delta A_2
  \]

  \[
  \Delta u = \frac{\partial u}{\partial A_1} \Delta A_1 + \frac{\partial u}{\partial A_2} \Delta A_2
  \]

  \[
  \Delta \rho = \frac{\partial \rho}{\partial A_1} \Delta A_1 + \frac{\partial \rho}{\partial A_2} \Delta A_2
  \]

  \[
  \frac{\partial p}{\partial A_1} = 0
  \]

  \[
  \frac{\partial p}{\partial A_2} = 0.5
  \]
\begin{align*}
\frac{\partial u}{\partial A_1} &= 0 \\
\frac{\partial u}{\partial A_2} &= \frac{0.5}{\partial \rho} \\
\frac{\partial \rho}{\partial A_1} &= -\frac{1}{c^2} = -\frac{\rho}{\gamma p} \\
\frac{\partial \rho}{\partial A_2} &= \frac{0.5}{c^2} = \frac{0.5\rho}{\gamma p}.
\end{align*}

Now equation 3.24 can be rewritten as:

\begin{align*}
\Delta p_0 &= \left(\frac{\partial p_0}{\partial p} \frac{\partial p}{\partial A_1} + \frac{\partial p_0}{\partial u} \frac{\partial u}{\partial A_1} + \frac{\partial p_0}{\partial \rho} \frac{\partial \rho}{\partial A_1}\right) \Delta A_1 \\
&\quad + \left(\frac{\partial p_0}{\partial p} \frac{\partial p}{\partial A_2} + \frac{\partial p_0}{\partial u} \frac{\partial u}{\partial A_2} + \frac{\partial p_0}{\partial \rho} \frac{\partial \rho}{\partial A_2}\right) \Delta A_2 
\end{align*}

(3.25)

Stagnation pressure can also be defined in the same way:

\begin{align*}
\Delta \rho_0 &= \frac{\partial \rho_0}{\partial \rho} \Delta \rho + \frac{\partial \rho_0}{\partial u} \Delta u + \frac{\partial \rho_0}{\partial p} \Delta p
\end{align*}

Now this equation can be expressed as:

\begin{align*}
\Delta \rho_0 &= \left(\frac{\partial \rho_0}{\partial \rho} \frac{\partial \rho}{\partial A_1} + \frac{\partial \rho_0}{\partial u} \frac{\partial u}{\partial A_1} + \frac{\partial \rho_0}{\partial p} \frac{\partial p}{\partial A_1}\right) \Delta A_1 \\
&\quad + \left(\frac{\partial \rho_0}{\partial \rho} \frac{\partial \rho}{\partial A_2} + \frac{\partial \rho_0}{\partial u} \frac{\partial u}{\partial A_2} + \frac{\partial \rho_0}{\partial p} \frac{\partial p}{\partial A_2}\right) \Delta A_2 
\end{align*}

(3.26)

Here, \( \Delta p_0 \) and \( \Delta \rho_0 \) is defined as:

\begin{align*}
\Delta p_0 &= \frac{p_0\big|_{\text{stagnation}} - p_0\big|_{\text{current}}}{\Delta t} \\
\Delta \rho_0 &= \frac{\rho_0\big|_{\text{stagnation}} - \rho_0\big|_{\text{current}}}{\Delta t}
\end{align*}

(3.27)

Now, equation 3.25 and 3.26 can be solved for \( \Delta A_1 \) and \( \Delta A_2 \).
Finally, the incoming entropy and acoustic wave can be set to:

$$\frac{\partial A_1}{\partial t}_{BC} = \frac{\partial A_3}{\partial t}_{noBC} + \Delta A_1$$

$$\frac{\partial A_2}{\partial t}_{BC} = \frac{\partial A_2}{\partial t}_{noBC} + \Delta A_2$$

4. Evaluating flow variable at the boundary with correct boundary value

Now for both inflow and outflow boundary, by using the equation 3.22, time derivatives of the flow variable at the boundary can be expressed as:

$$\frac{\partial p}{\partial t}_{BC} = \frac{1}{2} \left( \frac{\partial A_2}{\partial t}_{BC} + \frac{\partial A_3}{\partial t}_{BC} \right)$$

$$\frac{\partial u}{\partial t}_{BC} = \frac{1}{2pc} \left( \frac{\partial A_2}{\partial t}_{BC} - \frac{\partial A_3}{\partial t}_{BC} \right)$$

$$\frac{\partial \rho}{\partial t}_{BC} = \frac{1}{2c^2} \left( \frac{\partial A_2}{\partial t}_{BC} + \frac{\partial A_3}{\partial t}_{BC} - 2 \frac{\partial A_1}{\partial t}_{BC} \right)$$

(3.28)

3.3.7 Introduction of Curvilinear Coordinate System

Till now the governing equations are solved on a Cartesian grid of uniform spacing. Though taking numerical derivatives on these types of grids are easier, yet it can not be used for every cases. For instance, when there is a requirement of grid point clustering around a shock, uniformly spaced Cartesian grids will not be suffice. Hence, a definition of non-uniform grids is required which can be done through transferring the governing equations to curvilinear coordinates. Following section is motivated from the work of Vinokur [58, 59].

The quasi Euler equation is used here for the analysis:

$$\frac{\partial \vec{Q}}{\partial t} + \frac{\partial \vec{F}}{\partial x} - D(\vec{Q}) = \vec{S}$$

This equation is transformed from Cartesian coordinates \((x, t)\) to general curvilinear
coordinates \((\xi, \tau)\):

\[
\xi = \xi(x,t) \\
\tau = \tau(t)
\]

Now the Cartesian derivatives can be expressed in terms of curvilinear coordinates as:

\[
\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial \xi} \\
\frac{\partial}{\partial t} = \tau_t \frac{\partial}{\partial \tau} + \xi_t \frac{\partial}{\partial \xi}
\]

Using this definition of Cartesian derivatives, the quasi-1D can be written as:

\[
\tau_t \frac{\partial \vec{Q}}{\partial \tau} + \xi_t \frac{\partial \vec{Q}}{\partial \xi} + \xi_x \frac{\partial \vec{F}}{\partial \xi} - D(\vec{Q}) = \vec{S}
\]

Where the spatial derivative of area in \(S\) can be expressed as:

\[
\frac{\partial A}{\partial x} = \xi_x \frac{\partial A}{\partial \xi}
\]

Now with the introduction of Jacobian of the Transformation\((J)\), the derivative are defined as:

\[
\tau_t = Jx_\xi \\
\tau_x = -Jt_\xi \\
\xi_t = -Jx_\tau \\
\xi_x = Jt_\tau
\]
where the Jacobian is defined as:

\[ J = \frac{1}{t_\tau x_\xi - x_\tau t_\xi} \]  

(3.30)

The Jacobian can be simplified as:

\[ t_x = 0 \]

\[ t_\xi = 0 \]

\[ t_\tau = 1 \]

\[ J \bigg|_i = \xi_x \bigg|_i \]

Dividing equation 3.29 by \( J \) and going through some mathematical manipulation:

\[
\frac{\partial}{\partial t} \left( \frac{\bar{Q}}{J} \right) + \frac{\partial}{\partial \xi} \left( \frac{\xi_t \bar{Q}}{J} + \frac{\xi_x \bar{F}}{J} \right) - \frac{D(\bar{Q})}{J} = \bar{S}
\]

\[
\frac{\partial}{\partial t} \left( \frac{\bar{Q}}{J} \right) + \frac{\partial}{\partial \xi} \left( \frac{\xi_x \bar{F}}{J} \right) - \frac{D(\bar{Q})}{J} = \bar{S}
\]

(3.31)

where, the derivative in \( S \), \( \frac{\partial A}{\partial x} \) is evaluated as \( \frac{\partial A}{\partial \xi} \)
Chapter 4

Validation of the Solver for CAA Problems

4.1 Introduction

In the field of Computational Aeroacoustics, more emphasis is given on the calculation of unsteady flow instead of steady flow calculation. To correctly understand and predict the physics of noise generation and propagation, various numerical schemes are developed. To verify the correct implementation of these schemes and their performance, a range of validation problems with solutions are listed in the CAA Workshops [62, 63, 64].

Two problems of category 1 from CAA Workshop 3 are used here to analyze the efficiency of the dual time stepping based solver. The results are presented in this chapter.
4.2 Category 1 Problems

Quasi 1-D Euler equations are used to solve these two problems. The equations are given in the conservative variables as:

\[
\begin{align*}
\begin{cases}
\rho \\ \rho u \\ E \\
\end{cases} & \begin{cases}
\frac{\partial}{\partial t} + \frac{1}{A} \frac{\partial A}{\partial x} \begin{cases}
\rho u \\ \rho u^2 + p \\
u(E + p) \\
\end{cases} \end{cases} = 0
\end{align*}
\] (4.1)

A one-dimensional nozzle is defined with an area distribution as follows:

\[
A(x > 0) = 0.536572 - 0.198086e^{(-\ln 2)(\frac{x}{D})^2}
\]

\[
A(x < 0) = 1.0 - 0.661514e^{(-\ln 2)(\frac{x}{D})^2}
\] (4.2)

Here, the boundary conditions are specified by using the amplitude of the time derivative of each family of waves at the inflow and outflow:

\[
\begin{align*}
A_1 &= \frac{\partial p}{\partial t} - c^2 \frac{\partial \rho}{\partial t} \\
A_2 &= \frac{\partial p}{\partial t} + \rho c \frac{\partial u}{\partial t} \\
A_3 &= \frac{\partial p}{\partial t} - \rho c \frac{\partial u}{\partial t}
\end{align*}
\] (4.3)

4.2.1 Problem 1: Propagation of Sound Waves through a Transonic Nozzle

This problem models the upstream propagation of sound through a nearly choked nozzle with near sonic conditions. Following are used as characteristic scales for this problem:

length scale = Diameter of nozzle in the uniform region downstream of the throat, D
velocity scale = speed of sound in the same region, \( a_\infty \)

time scale = \( \frac{D}{a_\infty} \)

density scale = mean density of gas in the same region, \( \rho_\infty \)

pressure scale = \( \rho_\infty a_\infty^2 \)

Here, a small amplitude acoustic wave is generated way downstream and propagate upstream through the narrow passage of the nozzle throat. The upstream propagating wave in the downstream boundary is defined as:

\[
\begin{align*}
\begin{bmatrix}
\rho' \\
u' \\
p' \end{bmatrix}_{\text{outflow}} &= \epsilon \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \cos \left[ \omega \left( \frac{x}{1-M_{\text{outflow}}} + t \right) \right] \\
(4.4)
\end{align*}
\]

where, \( \omega = 0.6\pi \) and \( \epsilon = 10^{-5} \). A domain of size 20 is defined, 10 upstream and 10 downstream of the nozzle throat. A grid point of 400 is recommended to find the distribution of maximum acoustic pressure inside the nozzle.

4.2.1.1 Boundary Conditions, Grids and Numerical Details

The mean flow to solve this problem is set as:

\[
\begin{align*}
\begin{bmatrix}
\bar{\rho} \\
\bar{u} \\
\bar{p} \end{bmatrix}_{\text{outflow}} &= \begin{bmatrix} 1 \\ 0.4 \\ \frac{1}{\gamma} \end{bmatrix} \\
(4.5)
\end{align*}
\]

From the definition of upstream running acoustic wave,

\[
\begin{align*}
A_3 &= \frac{\partial p}{\partial t} - \bar{p} \frac{\partial u}{\partial t} \\
\frac{\partial p}{\partial t} &= -\epsilon \omega \sin \left[ \omega \left( \frac{x_{\text{outflow}}}{1-M_{\text{outflow}}} + t \right) \right] \\
\frac{\partial u}{\partial t} &= \epsilon \omega \sin \left[ \omega \left( \frac{x_{\text{outflow}}}{1-M_{\text{outflow}}} + t \right) \right]
\end{align*}
\]
\[ \bar{p} = 1 \]
\[ \bar{c} = \sqrt{\frac{\gamma p}{\bar{p}}} \]
\[ \bar{c} = 1 \]

\[ x_{\text{outflow}} = 10 \]
\[ M_{\text{outflow}} = \frac{\bar{u}}{\bar{c}} \]
\[ = 0.4 \]

\[ A_3 = -2\epsilon \omega \sin \left[ \omega \left( \frac{10}{1-0.4} + t \right) \right] \]

Other two waves: \( A_1 \) and \( A_2 \) is set to zero.

For this problem, a mesh distribution as shown in figure 4-1 is used. A graph showing the Jacobian of the transformation is also presented in figure 4-2.
Figure 4-2: Jacobian Distribution
In this problem, the initial condition is set based on the exact solution acquired from the isentropic flow equations. Then from time 0, disturbance oscillation is imposed on the boundary. 90 cycles of perturbation is used to ensure that the transients pass out of the nozzle. After that, the maximum acoustic pressure is obtained by subtracting the minimum total pressure from the maximum total pressure and then dividing by 2.

4.2.1.2 Result and Discussion

1. This problem is designed to test a scheme’s ability to deal with multiple-length scales problem and to test whether small amplitude wave can be captured from the mean flow. It also aims at evaluating the accuracy of numerical boundary condition implementation.

2. The solution of the problem can be analyzed in three different regions. As in the upstream of the nozzle throat, there exists only the transmitted waves, so the maximum pressure envelope should be nearly a straight line. In the nozzle throat area, due to the smallest area, a sharp peak of pressure is expected here. And finally in the downstream region of the throat, there is an upstream running acoustic wave along with a downstream running reflected wave. The two waves form an interference pattern at that region.

3. In the following pages, two graphs are represented based on the number of steps per cycles. The first graph portrays the transmitted wave amplitude in the upstream region as well as the interference pattern in the downstream. The second graph shows the location and maximum pressure amplitude near the nozzle throat.

4. From table 4.1 it is seen that a larger CFL number can be attained by the dual time marching scheme as discussed in the literature. From the larger value
of \( \Delta t \), it is evident that the computational workload will be reduced by this method, but whether this advantage can be taken depends on the accuracy of the solution with larger CFL number.

5. From table 4.2, it is obvious that with the decrease of \( \Delta t \), less computational effort is required to resolve the wave in the fictitious domain. Here as a 3rd Order RK-2N scheme is used for marching the solution in the 'fictitious domain', \( \Delta \tau \) is taken around 0.006 to maintain the fictitious stability limit. The iterations can be sped up with a scheme that has a larger stability limit. The future work of this project will be to investigate the acceleration techniques of the iterations process.

6. From figure 4-3 to figure 4-10, the effect of larger time steps on the solution accuracy is portrayed effectively.

7. In figure 4-3, 4-4, 4-5 results are shown for high CFL number ranging from 15 to 30. But as the \( \Delta t \) values are larger enough (ranging from 0.104 to 0.208) to resolve the frequency of the acoustic wave, the solution deviates from the exact value. The requirement on the time step is for \( \Delta t \) to be small enough to resolve the frequency.

8. With the reduction of the \( \Delta t \) values, the solution gets better in figure 4-6 to figure 4-10.

9. From the physics of the problem, it is clear that in the upstream of nozzle, the maximum pressure envelope should be a constant line as there is only the transmitted acoustic wave. But in every figures from 4-3 to 4-10, there are wiggles near the left boundary. The test case is run longer enough (900 cycles) to see whether it causes from any existing transient in the domain. But after running this long, the wiggles are still there. The author concludes that there
is problem with the implementation of either boundary condition or boundary stencils and a thorough investigation will be conducted to find out the exact reason behind that in future works.

10. From figures 4-11 to 4-18 envelope of the pressure amplitudes are shown along with the instantaneous pressure distribution at the start of a period. In every figures, the maximum amplitude of instantaneous distribution touches the curve of the maximum envelope as it should be.

Table 4.1: Relation among steps per cycle, CFL and $\Delta t$ for Problem 1

<table>
<thead>
<tr>
<th>Steps per cycle</th>
<th>CFL Number</th>
<th>$\Delta t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>30.28</td>
<td>0.208</td>
</tr>
<tr>
<td>24</td>
<td>20.19</td>
<td>0.139</td>
</tr>
<tr>
<td>32</td>
<td>15.14</td>
<td>0.104</td>
</tr>
<tr>
<td>48</td>
<td>10.09</td>
<td>0.069</td>
</tr>
<tr>
<td>64</td>
<td>7.57</td>
<td>0.052</td>
</tr>
<tr>
<td>128</td>
<td>3.79</td>
<td>0.026</td>
</tr>
<tr>
<td>256</td>
<td>1.89</td>
<td>0.013</td>
</tr>
<tr>
<td>512</td>
<td>0.95</td>
<td>0.006</td>
</tr>
</tbody>
</table>

Table 4.2: Relation among steps per cycle and Number of Inner Sub-Iterations for Problem 1

<table>
<thead>
<tr>
<th>Steps per cycle</th>
<th>Max. Inner Sub-iterations</th>
<th>Min. Inner Sub-iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>161</td>
<td>114</td>
</tr>
<tr>
<td>24</td>
<td>113</td>
<td>79</td>
</tr>
<tr>
<td>32</td>
<td>86</td>
<td>61</td>
</tr>
<tr>
<td>48</td>
<td>59</td>
<td>43</td>
</tr>
<tr>
<td>64</td>
<td>45</td>
<td>33</td>
</tr>
<tr>
<td>128</td>
<td>24</td>
<td>19</td>
</tr>
<tr>
<td>256</td>
<td>13</td>
<td>11</td>
</tr>
</tbody>
</table>
Figure 4-3: Maximum pressure distribution for 16 steps per cycle
Figure 4-4: Maximum pressure distribution for 24 steps per cycle
Figure 4-5: Maximum pressure distribution for 32 steps per cycle
Figure 4-6: Maximum pressure distribution for 48 steps per cycle
Figure 4-7: Maximum pressure distribution for 64 steps per cycle
Figure 4-8: Maximum pressure distribution for 128 steps per cycle
Figure 4-9: Maximum pressure distribution for 256 steps per cycle
Figure 4-10: Maximum pressure distribution for \textbf{512} steps per cycle
Figure 4-11: Envelope and instantaneous pressure distribution for 16 steps per cycle

Figure 4-12: Envelope and instantaneous pressure distribution for 24 steps per cycle
Figure 4-13: Envelope and instantaneous pressure distribution for 32 steps per cycle

Figure 4-14: Envelope and instantaneous pressure distribution for 48 steps per cycle
Figure 4-15: Envelope and instantaneous pressure distribution for 64 steps per cycle

Figure 4-16: Envelope and instantaneous pressure distribution for 128 steps per cycle
Figure 4-17: Envelope and instantaneous pressure distribution for 256 steps per cycle

Figure 4-18: Envelope and instantaneous pressure distribution for 512 steps per cycle
4.2.2 Problem 2: Shock-Sound Interaction

In this problem, a supersonic shock is present at the downstream of the nozzle. A same geometry as problem 1 is also used here with the same governing equations. Here the quantities are non-dimensionalized using the upstream values:

- length scale $= D_{inlet}$
- density scale $= \rho_{inlet}$
- velocity scale $= a_{inlet}$
- pressure scale $= \rho_{inlet}a_{inlet}^2$
- time scale $= \frac{D_{inlet}}{a_{inlet}}$

Here, the problem is focused on the upstream propagation of an acoustic wave through a shock wave. The acoustic wave at the upstream boundary is defined as:

$$
\begin{align*}
\left\{ \begin{array}{c}
\rho' \\
u' \\
p'
\end{array} \right\}_{inflow} &= \epsilon \left\{ \begin{array}{c}
1 \\
1 \\
1
\end{array} \right\} \sin \left[ \omega \left( \frac{x}{1 + M} - t \right) \right]
\end{align*}
$$

(4.6)

where, $\omega = 0.6\pi$ and $\epsilon = 10^{-5}$ and $M_{inlet} = 0.2006533$. At the outflow boundary, the pressure is set such that it creates a shock:

$$p_{exit} = 0.6071752$$

The mean flow to solve this problem is set as:

$$
\begin{align*}
\left\{ \begin{array}{c}
\bar{\rho} \\
\bar{u} \\
\bar{p}
\end{array} \right\} &= \left\{ \begin{array}{c}
1 \\
0.2006533 \\
\frac{1}{\gamma}
\end{array} \right\}
\end{align*}
$$

(4.7)

From the definition of downstream running acoustic wave,

$$A_2 = \frac{\partial p}{\partial t} + \bar{p}c \frac{\partial u}{\partial t}$$
So,

\[ A_2 = -2\epsilon \omega \cos \left( \omega \left( \frac{-10}{1 + 0.2006533} - t \right) \right) \]

4.2.2.1 Result and Discussion

1. The problem consists of three parts: calculating the steady mean distribution, finding perturbation at the start of a period and pressure perturbation at the exit plane through one period. The results for these problems by using dual time stepping is presented here.

2. Figure 4-19 compares the calculated mean pressure with the exact solution. The computed solution matches closely with the exact one except in the shock region. Since a central difference scheme is used in this calculation, oscillations near the shock is expected. However, filtering protects the downstream solution from being contaminated.

3. The performance of the scheme deteriorates with larger values of time steps due to the lack of temporal resolution. Figures 4-20, 4-21, 4-22 and figures 4-28, 4-29, 4-30 clearly shows the inaccuracy of the solution due to larger \( \Delta t \) values.

4. One of the challenges of this problem is to accurately capture the acoustic disturbance in the presence of the shock. From figure 4-24 it is apparent that dual time stepping based code can produce a closely matched solution with a CFL around 8.98. It can also be noted that though there are high spike due to the interaction between acoustic wave and shock wave, the computed transmitted and reflected wave are close to the exact solution.

5. Here, the transmitted wave should exit the nozzle without any reflection whereas the reflected wave contacts with the input wave. From the figures 4-23 to 4-
it is shown that the pressure amplitude after the shock is constant and almost equal to the amplitude of the wave where the amplitude before the shock oscillates in accordance to the input wave.

6. Figure 4-31 to 4-35 exhibits good approximation of the perturbation pressure at the exit plane over a period.

7. Dual time scheme has successfully found the right position of the shock which is downstream of the throat. But much work needs to be done to capture the strength of the shock.

8. Figure 4-36 provides a reasoning behind the choice of convergence criteria in the fictitious time domain. With a convergence limit of $10^{-6}$ and $10^{-7}$, the scheme performs poorly. In figure 4-37, the solution gets closer to the exact one when the convergence criteria reaches to $10^{-9}$. And as the limit $10^{-12}$ doesn’t produce much better result than already attained, the value $10^{-9}$ is fixed as a convergence criteria.

9. From figure 4-38 to 4-42 the importance of clustering is shown for resolving the mean flow. From the zoomed figure in the upstream and downstream region, it is pretty obvious that with lower grid point, the flow solution deviates from the exact one. And at the throat, a clear advantage of using more grid points is exhibited.
Table 4.3: Relation among steps per cycle, CFL and $\Delta t$ for Problem 2

<table>
<thead>
<tr>
<th>Steps per cycle</th>
<th>CFL Number</th>
<th>$\Delta t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>36.46</td>
<td>0.208</td>
</tr>
<tr>
<td>24</td>
<td>24.36</td>
<td>0.139</td>
</tr>
<tr>
<td>32</td>
<td>18.2</td>
<td>0.104</td>
</tr>
<tr>
<td>48</td>
<td>12.03</td>
<td>0.069</td>
</tr>
<tr>
<td>64</td>
<td>8.98</td>
<td>0.052</td>
</tr>
<tr>
<td>128</td>
<td>4.45</td>
<td>0.026</td>
</tr>
<tr>
<td>256</td>
<td>2.21</td>
<td>0.013</td>
</tr>
<tr>
<td>512</td>
<td>1.09</td>
<td>0.006</td>
</tr>
</tbody>
</table>
Figure 4-19: Mean Pressure Distribution with 359 points
Figure 4-20: Perturbation pressure distribution for 16 steps per cycle

Figure 4-21: Perturbation pressure distribution for 24 steps per cycle
Figure 4-22: Perturbation pressure distribution for 32 steps per cycle

Figure 4-23: Perturbation pressure distribution for 48 steps per cycle
Figure 4-24: Perturbation pressure distribution for 64 steps per cycle

Figure 4-25: Perturbation pressure distribution for 128 steps per cycle
Figure 4-26: Perturbation pressure distribution for 256 steps per cycle

Figure 4-27: Perturbation pressure distribution for 512 steps per cycle
Figure 4-28: Perturbation pressure history at exit plane for \textbf{16} steps per cycle

Figure 4-29: Perturbation pressure history at exit plane for \textbf{24} steps per cycle
Figure 4-30: Perturbation pressure history at exit plane for 32 steps per cycle

Figure 4-31: Perturbation pressure history at exit plane for 48 steps per cycle
Figure 4-32: Perturbation pressure history at exit plane for 64 steps per cycle

Figure 4-33: Perturbation pressure history at exit plane for 128 steps per cycle
Figure 4-34: Perturbation pressure history at exit plane for 256 steps per cycle

Figure 4-35: Perturbation pressure history at exit plane for 512 steps per cycle
Figure 4-36: Affect of inner convergence criteria on pressure perturbation for 64 steps per cycle
Figure 4-37: Affect of inner convergence criteria on pressure perturbation for 64 steps at exit plane
Figure 4-38: Affect of clustering on the mean flow

Figure 4-39: Affect of clustering on the mean flow: Upstream Region
Figure 4-40: Affect of clustering on the mean flow: Throat Region

Figure 4-41: Affect of clustering on the mean flow: Downstream Region
Figure 4-42: Affect of clustering on the mean flow: Upstream Region
4.3 Conclusions and Future Work

This work aims to verify the potential of the dual time stepping algorithm as found in various literatures. As the stability criteria of the explicit scheme is a major problem for solving unsteady cases where tiny grid spacing is required, implicit dual time stepping can be helpful for those cases. From the benchmark problems, it is noticed that CFL numbers is taken as 36 for the second problem. Though accuracy drops for this larger CFL, yet we can predict a good solution with a CFL around 9. From the problem investigated, it is clear that $\Delta t$ should be taken such that it is enough for resolving the mean flow gradients.

Though the dual time stepping looks promising, there are certain issues need to be solved before implementing this. The efficiency of the inner sub-iterations may affect the solution. Here 3rd order RK-2N is used to iterate the solution in the fictitious time. As the CFL of RK-2N with DRP is around 1.05, larger $\Delta \tau$ makes the code unstable. A scheme with larger CFL can make the dual time stepping more efficient. Again, implicit residual smoothing and preconditioning can also be used for the enhancements in the convergence.
References


[34] **Hixon, R., Bhate, D., Nallasamy, M., and Sawyer, S.**, “Shock-Capturing Dissipation Schemes for High-Accuracy Computational Aeroacoustics (CAA)


