Simulation of 2-D Compressible Flows on a Moving Curvilinear Mesh with an Implicit-Explicit Runge-Kutta Method

Thesis by
Moataz O. Abu AlSaud, B.S. Mechanical Engineering

Submitted in Partial Fulfillment of the Requirements for the degree of Masters of Science

King Abdullah University of Science and Technology
Physical Sciences and Engineering
Mechanical Engineering

Thuwal, Makkah Province, Kingdom of Saudi Arabia
July, 2012
The undersigned approve the thesis of Your Full Name

Dr. Rooh Khurram
Committee Member

Dr. Fabrizio Bisetti
Committee Member

Dr. Ravi Samtaney
Committee Member (Thesis Supervisor)

King Abdullah University of Science and Technology
Year: 2012
ABSTRACT

Simulation of 2-D compressible flows on a moving curvilinear mesh with an implicit-explicit Runge-Kutta method

Moataz O. Abu AlSaud

The purpose of this thesis is to solve unsteady two-dimensional compressible Navier-Stokes equations for a moving mesh using implicit explicit (IMEX) Runge-Kutta scheme. The moving mesh is implemented in the equations using Arbitrary Lagrangian Eulerian (ALE) formulation. The inviscid part of the equation is explicitly solved using second-order Godunov method, whereas the viscous part is calculated implicitly.

We simulate subsonic compressible flow over static NACA-0012 airfoil at different angle of attacks. Finally, the moving mesh is examined via oscillating the airfoil between angle of attack \( \alpha = 0 \) and \( \alpha = 20 \) harmonically. It is observed that the numerical solution matches the experimental and numerical results in the literature to within 20\%.
ACKNOWLEDGMENTS

The author wishes to express his gratitude to his supervisor Dr. Ravi Samtaney for his constant support and guidance throughout the progression of this thesis. His way of elaborating the concepts related to the work was insightful and kept the author’s enthusiasm and excitement increasing until the end of this work.

The author would also like to thank his parents for their continuous encouragement to finish this thesis. Their great emphasis on learning and education fueled the author’s drive to pursue studies in higher education.

Finally, the author would like to express his appreciation to KAUST for providing all the required resources to complete this research.
# TABLE OF CONTENTS

List of Abbreviations .................................................. 8
List of Symbols .......................................................... 9
List of Illustrations ...................................................... 10
List of Tables ............................................................. 12

I Introduction ................................................................... 13
   I.1 Motivation ............................................................. 13
   I.2 Background ........................................................... 15
   I.3 Thesis approach ...................................................... 17
   I.4 Outline ................................................................. 18

II The governing equations ............................................. 19
   II.1 The basic system of equations .................................. 19
   II.2 Transformation of the equations in curvilinear moving domain ........................................................................ 24

III Numerical method ..................................................... 29
   III.1 Choosing IMEX schemes ........................................ 29
   III.2 Forward-backward Euler (1,2,1) ............................. 30
      III.2.1 First stage ...................................................... 31
      III.2.2 Final stage ..................................................... 35
   III.3 Implicit-explicit midpoint (1,2,2) ............................ 36
   III.4 DIRK (2,3,2) ......................................................... 36
      III.4.1 First stage ...................................................... 37
      III.4.2 Second stage ................................................... 41
      III.4.3 Final stage ..................................................... 44
   III.5 Godunov second order scheme ............................... 45
   III.6 Multigrid method .................................................. 46
   III.7 Numerical method for moving grid .......................... 47

IV Results and discussion ............................................... 50
   IV.1 Grid specifications ................................................ 50
   IV.2 Initial and boundary conditions ............................... 52
   IV.3 Solver parameters ................................................ 52
IV.4 Validation diagnostics ........................................ 53
IV.5 Case studies ....................................................... 55

V Conclusion and future work ........................................ 74
  V.1 Summary ............................................................. 74
  V.2 Further improvement ............................................. 75
  V.3 Future work ........................................................ 75

REFERENCES .............................................................. 76
List of Abbreviations

ALE    Arbitrary Lagrangian Eulerian
DIRK   Diagonally Implicit Runge-Kutta
FSI    Fluid Structure Interaction
IMEX   Implicit Explicit
MAV    Micro Aerial Vehicles
NACA   National Advisory Committee for Aeronautics
List of Symbols

\( x \)  Horizontal physical domain
\( y \)  Vertical physical domain
\( t \)  Time
\( U \)  Solution vector
\( F \)  Inviscid flux in x-direction
\( G \)  Inviscid flux in y-direction
\( F_\nu \)  Viscous flux in x-direction
\( G_\nu \)  Viscous flux in y-direction
\( \rho \)  Density
\( u \)  Horizontal velocity
\( v \)  Vertical velocity
\( E \)  Total energy
\( p \)  Pressure
\( \tau_{ij} \)  Shear stress tensor
\( q \)  Heat flux vector
\( T \)  Temperature
\( Re \)  Reynolds number
\( Pr \)  Prandtl number
\( \gamma_g \)  Adiabatic index
\( R \)  Universal gas constant
\( \mu \)  Viscosity
\( \nu \)  Dynamic viscosity
\( \xi \)  Horizontal computational domain
\( \eta \)  Vertical computational domain
\( J \)  Jacobian
\( u_x \)  Mesh node horizontal velocity
\( u_y \)  Mesh node vertical velocity
\( Ma \)  Mach number
\( c_l \)  Lift coefficient
\( F_L \)  Lift force
\( c \)  Chord length
\( c_d \)  Drag coefficient
\( F_D \)  Drag force
\( \theta \)  Momentum thickness
\( \alpha \)  Angle of attack
$N$  Grid size  
$f$  Reduced frequency  
$\omega$  Frequency of oscillation
## List of Illustrations

<table>
<thead>
<tr>
<th>Illustration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV.1</td>
<td>The physical domain represented by $256 \times 256$ C-grid for the second case, where the airfoil is static at $\alpha = 16$.</td>
</tr>
<tr>
<td>IV.2</td>
<td>Relative magnitude of the ignored term in the implicit stage for the second case, where $t = 21.7$, $\alpha = 16$, $Re = 40000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012.</td>
</tr>
<tr>
<td>IV.3</td>
<td>Steady state horizontal velocity at $\alpha = 5$, $Re = 40000$, $N = 128$, and $U_\infty = 0.4Ma$ for NACA-0012.</td>
</tr>
<tr>
<td>IV.4</td>
<td>Steady state pressure at $\alpha = 5$, $Re = 40000$, $N = 128$, and $U_\infty = 0.4Ma$ for NACA-0012.</td>
</tr>
<tr>
<td>IV.5</td>
<td>Outflow steady state horizontal velocity at $\alpha = 5$, $Re = 40000$, $N = 128$, and $U_\infty = 0.4Ma$ for NACA-0012.</td>
</tr>
<tr>
<td>IV.6</td>
<td>Evolution of vortex shedding at $\alpha = 16$, $Re = 40000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012.</td>
</tr>
<tr>
<td>IV.7</td>
<td>Initial top and bottom vortex shedding at $\alpha = 16$, $Re = 40000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012.</td>
</tr>
<tr>
<td>IV.8</td>
<td>Periodic vortex shedding at $\alpha = 16$, $Re = 40000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012.</td>
</tr>
<tr>
<td>IV.9</td>
<td>Horizontal velocity prior to oscillation at $\alpha_0 = 10$, $Re = 5000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012.</td>
</tr>
<tr>
<td>IV.10</td>
<td>Density prior to oscillation at $\alpha_0 = 10$, $Re = 5000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012.</td>
</tr>
<tr>
<td>IV.11</td>
<td>Lift Coefficient vs. Angle of Attack at $\alpha = \alpha_0 + \alpha_1 \sin(ft)$, $\alpha_0 = 10$, $\alpha_1 = 10$, $Re = 5000$, $f = 0.25$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012.</td>
</tr>
<tr>
<td>IV.12</td>
<td>Lift Coefficient vs. Angle of Attack at $\alpha = \alpha_0 + \alpha_1 \sin(ft)$, $\alpha_0 = 10$, $\alpha_1 = 10$, $Re = 5000$, $f = 0.25$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012 taken from reference [4].</td>
</tr>
<tr>
<td>IV.13</td>
<td>Density at $t = 33.2$, $\alpha = 18.5$ going upward, $f = 0.25$, $Re = 5000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012.</td>
</tr>
<tr>
<td>IV.14</td>
<td>Density at $t = 33.2$, $f = 0.25$, $Re = 5000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012.</td>
</tr>
<tr>
<td>IV.15</td>
<td>Density at $t = 35.1$, $\alpha = 12.5$ going upward, $f = 0.25$, $Re = 5000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012.</td>
</tr>
<tr>
<td>IV.16</td>
<td>Density at $t = 35.1$, $f = 0.25$, $Re = 5000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012 taken from reference [4].</td>
</tr>
</tbody>
</table>
## List of Tables

| IV.1 | First test case: static NACA-0012 at angle of attack $\alpha = 5$ using DIRK (2,3,2) IMEX scheme | 60 |
| IV.2 | Second test case: static NACA-0012 at angle of attack $\alpha = 16$ using DIRK (2,3,2) IMEX scheme | 61 |
| IV.3 | Third test case: oscillating NACA-0012 between $\alpha = 0$ and $\alpha = 20$ using DIRK (2,3,2) IMEX scheme | 65 |
Chapter I

Introduction

I.1 Motivation

The significant growth in world energy demand requires extensive research and development in various branches of energy resources. Renewable energy is an attractive energy resource, since it can be used for generating electricity and transporting fuels to meet the future energy needs and it does not contribute to greenhouse-gas emissions. Specifically, wind energy is a promising source of electricity since it comes directly after fossil fuel based power plants in terms of cost. This made wind energy be considered as a major part of the future electricity generation plan. For instance, the goal of the U.S. Department of Energy is to base 20% of the electricity supply on wind energy by 2030 [7]. Currently, only 2.9% of the electricity is generated from wind. Also, Saudi Arabia is expecting to reach electricity consumption of 67 GW by 2023 whereas currently it consumes around 37 GW [1]. These figures show how wind energy will play a vital role in facing the challenges associated with future energy supply, imposing further development in this energy sector.

The main parameter in harvesting wind energy is to design a wind turbine that has large diameter, since the energy production is proportional to the diameter squared.
However, challenges arise as the wind turbine diameter increases such as noise generation and wind blade large deformation making it unreliable. These issues can be mitigated significantly through better wind turbine design. Optimal wind turbine design can be achieved through high-fidelity simulations. In order to reach this level, deep understanding of the physics behind the wind turbine aeroelasticity is required. A major component in studying aero-elastic interactions is to develop algorithms that are able to handle two main issues:

- Compressibility: This phenomenon becomes crucial in large wind turbines as the velocity of the wind blade tip can reach a high magnitude, becoming a possible source of noise. As a result, the algorithm must handle compressibility effects in order to model wind turbine acoustic noise.

- Wind blade deformation: This is called fluid-structure interaction (FSI) phenomenon and occurs due to high air pressure on the wind blades causing large oscillation or deformation especially at the tip of the blade. The algorithm in high-fidelity simulation should predict solid body deformation to accurately model wind turbines facing FSI.

The aforementioned challenges motivated the author to initiate the groundwork for full simulation of wind turbines, which is developing an efficient numerical method to simulate a wind blade cross section that can handle compressible flows as well as moving domain. The algorithm ability to deal with compressible flows will be used to study the acoustics associated with wind turbines in the future. On the other hand, handling moving domain will be the foundation of FSI, where solid solver will be coupled with the fluid solver.
I.2 Background

Although there has been continuous improvement in developing algorithms to solve compressible Navier Stokes equations, challenges still arise in obtaining the correct solution of the model. These challenges include the numerical method efficiency, and accurate modeling of complex geometries. One of the first approaches to solve the aforementioned equations is MacCormack Method [17], where the viscous and inviscid fluxed are approximated explicitly making it very expensive computationally. Another common method is Beam-Warming Method [5], where the fluxes are computed implicitly. However, the drawback in the above schemes is that they sometimes suffer from numerical instability and unphysical oscillations in the numerical solution.

The unphysical oscillation could be significantly reduced or prevented via another class of schemes called modern shock capturing numerical methods. Such subclasses include Flux Corrected Transport [6], Total Variation Diminishing (TVD) [11], Essentially Non-Oscillatory (ENO) [12], Piecewise Parabolic Method (PPM) [9], Monotonic Upstream-Centered Schemes for Conservation Laws (MUSCL) [31], and Roe schemes [21]. Each of these schemes have advantages and disadvantages in terms of computational expense and its ability to capture shocks accurately.

In addition to the challenges associated with obtaining a numerical method to accurately solve compressible Navier Stokes equations, the geometric complexity of the domain imposes a major challenge in the discretization process. Finite volume method is powerful in solving fluid problems that include complex geometries, and can be formulated for either structured or unstructured grid. Unstructured grids are more flexible, while structured grids are more convenient and accurate in resolving boundary layer around the complex geometry making it more favorable in simulating flow over complex bodies such as airfoils. Structured grids require transformation of the equations. The general transformation of gas dynamic equations to a curvilinear coordinates was first formulated by Viviand [34] and Vinokour [32].
As for the mesh generation, the grid points should be distributed based on the domain geometry creating a body fitted grid. Several methods are used to generate a structured mesh suited for the geometry domain such as algebraic, conformal mapping, and elliptic methods. More details of each class are in reference [8].

Another major issue in modeling gas dynamics is the interaction of fluid with solid body causing a movement or deformation of the body. Thus, the grid points on the boundary will have to move accordingly causing a change in the flux between the cells related to the moving grids. This type of problems falls under FSI class, and has been extensively researched due to its importance in designing structures such as wind turbines. A common approach to simulate FSI is to use Arbitrary Lagrangian Eulerian (ALE) method. This method permits the fluid to either flow in a fixed mesh (Eulerian approach), or a moving mesh (Lagrangian approach) depending on how the domain deforms or moves. [14].

Several techniques have been proposed in the literature on how to move the internal mesh points due to moving boundaries. The performance of the technique is evaluated based on the quality of the mesh at the next times step. It should avoid certain features such cell overstretching, skewness, and slope jumps. These attributes have negative impact causing unexpected errors in the solution [33]. The obvious approach is to implement a mesh generator that re-meshes for the instantaneous boundary point positions at each time step. Although this method generates smooth high quality mesh, it is significantly expensive in terms of computation. An alternative approach is to solve the Laplace equation with specified boundary velocities resulting a smooth distribution of the velocity mesh points within the boundary. Then, the mesh points are updated based on their computed velocities. More rigorous details of this ALE based method could be found in [19] and [20].
I.3 Thesis approach

In this thesis, the numerical method consists of using explicit method for the inviscid flux and implicit method for the viscous flux to have the stability and robustness of implicit methods. As for the inviscid part, the explicit approach is chosen because motion is resolved in the advection time scale. The proposed implicit-explicit (IMEX) scheme is based on Runge-Kutta method. The explicit method used to compute the inviscid term is second order Godunov scheme \[10\]. The viscous flux is obtained implicitly using multigrid method to solve the resulting elliptic equations.

Since the IMEX Runge-Kutta is a linear multistep method, various IMEX schemes can be derived based on the number of stages within each time step. Some of the schemes are forward-backward Euler (1,2,1), implicit-explicit midpoint (1,2,2), and L-stable two-stage second-order DIRK (2,3,2). Details of the mentioned schemes are in chapter \[III\] and more schemes of this type can be seen in reference \[3\].

In generating the grid, the method used in the solution is based on elliptic method. Refer to section \[III.7\] for the equations used to distribute the grid points. As for dealing with moving boundaries, the most convenient method for the cases that are investigated in this paper is ALE. This approach requires a modification of the fluxes in the governing equations. Details of reaching the final strong conservative form of gas dynamic equations in moving curvilinear mesh are in chapter \[II\]. To obtain the velocity at the mesh nodes, Laplace equation is used to find the velocity distribution of the grid points as it is less expensive computationally than generating a grid every time step. Multigrid method is used to obtain the solution of the Laplace equation.

In validating the proposed numerical method, simulation of flow around an airfoil is accomplished. The simulation is performed on both static and oscillating NACA-0012 airfoil cases. The fluid flow conditions are specified to operate at subsonic speed \(U_{\infty} < 0.7Ma\) with modest Reynolds number \(Re < 10^5\). These flow conditions match the operating conditions for important applications such as wind turbines.
and micro aerial vehicles (MAVs) \[18\]. The ability of the numerical method to take
dynamic mesh into consideration makes it useful to such practical applications as
wind turbine blades deform significantly and some MAVs have flapping wings \[26\].
The numerical solution of the chosen cases is compared with the results of \[25\] and
\[4\]. Discussion of the results are in chapter \[IV\].

\section*{I.4 Outline}

The thesis starts with deriving the governing equations in a curvilinear moving domain
in chapter \[II\]. The numerical method used to solve the equations is presented in detail
in chapter \[III\] which includes several flavors of IMEX scheme. Chapter \[IV\] presents
the results of several simulations with comparison to previous results to validate the
numerical method. Finally, conclusions and future work are presented in chapter \[V\].
Chapter II

The governing equations

II.1 The basic system of equations

In modeling fluid dynamics problems, the most common approach is to construct the model on classical continuum mechanics principles obeying the fundamental equations: mass, momentum, and energy conservation equations. The resulting equations, referred to as compressible Navier-Stokes equations, can be grouped and written in the following conservative form in 2D:

\[
\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0,
\]

where \( U \) in equation II.1 is the solution, \( F \) is the flux in the x-direction, and \( G \) is the flux in the y-direction. \( U \), \( F \), and \( G \) are defined as follows:

\[
U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix},
\]

(II.2)
\[ F = \begin{pmatrix} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho u v - \tau_{xy} \\ (E + p)u - u\tau_{xx} - u\tau_{xy} + q_x \end{pmatrix}, \quad (\text{II.3}) \]

\[ G = \begin{pmatrix} \rho v \\ \rho u v - \tau_{xy} \\ \rho v^2 + p - \tau_{yy} \\ (E + p)v - u\tau_{xy} - v\tau_{yy} + q_y \end{pmatrix}, \quad (\text{II.4}) \]

where \( \rho \) is the density, \( u \) is the horizontal velocity, \( v \) is the vertical velocity, \( E \) is the total energy, \( p \) is the pressure, \( \tau_{ij} \) is the shear stress, and \( q \) is the heat flux vector.

The first equation corresponding to the first row represents the mass conservation equation, and the second and third equations are the momentum equations in \( x \) and \( y \) directions respectively. The last equation is the energy equation.

Although the flow type can be turbulent, there is no need for turbulence model at this stage since it is a 2D flow. Moreover, Godunov method is used in the numerical method which has an implicit dissipation. The model can be extended to 3D where turbulence would be significant, but it is beyond the scope of this work.

By observing the system of equations, it can be noticed that the required number of variables to model the flow exceeds the number of equations. Several assumptions are needed to close the system and make it solvable. They are as follows:

- The heat transfer follows Fourier’s law heat of conduction. Thus, the heat fluxes \( q_x \) and \( q_y \) are related to the gradient temperature via:

\[ q_x = -k \frac{\partial T}{\partial x}, \quad q_y = -k \frac{\partial T}{\partial y}, \quad (\text{II.5}) \]

where \( k \) is the thermal conductivity.
• The fluid is newtonian. Therefore, the shear stress is proportional to the rate of strain, and the rate of strain corresponds to the velocity gradient resulting in the following constitutive relation:

$$\tau = [\mu \nabla u + \mu(\nabla u)^T + \lambda I(\nabla \cdot u)],$$

(II.6)

where $\mu$ is the dynamic viscosity, and $\lambda$ is the bulk coefficient of viscosity both related through Stokes’ hypothesis:

$$\lambda = -\frac{2}{3}\mu.$$  

(II.7)

• The gas is assumed to be ideal, which has the following equation of state:

$$p = \rho RT.$$  

(II.8)

• The gas used in the flow is air and its viscosity $\mu$ is constant as the viscosity change due to temperature is almost negligible for a subsonic flow.

• The air is treated as calorically perfect gas making Prandtl number to be constant equal to:

$$Pr = \frac{\mu c_p}{k} = 0.71.$$  

(II.9)

By examining the flux from physics point of view, it can be split into viscous and inviscid parts

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = \frac{\partial F_\nu}{\partial x} + \frac{\partial G_\nu}{\partial y},$$

(II.10)

where the subscript $\nu$ denotes the viscous part. $F, G, F_\nu$, and $G_\nu$ are:
The above governing equations can be written in non-dimensional form. \( F \) and \( G \) remain the same, whereas \( F_\nu \) and \( G_\nu \) become:

\[
F_\nu = \begin{cases}
0 \\
\frac{\tau_{xx}}{Re} \\
\frac{\tau_{xy}}{Re} \\
\frac{u\tau_{xx} + v\tau_{xy} - q_x}{RePr(\gamma - 1)}
\end{cases}, \quad (\text{II.15})
\]

\[
G_\nu = \begin{cases}
0 \\
\frac{\tau_{xy}}{Re} \\
\frac{\tau_{yy}}{Re} \\
\frac{u\tau_{xy} + v\tau_{yy} - q_y}{RePr(\gamma - 1)}
\end{cases}.
\]
\[ G_{\nu} = \begin{pmatrix} 0 \\ \frac{\tau_{xy}}{Re} \\ \frac{\tau_{yy}}{Re} \\ \frac{u\tau_{xy} + v\tau_{yy}}{Re} - \frac{q_y}{RePr(\gamma_g - 1)} \end{pmatrix}, \] (II.16)

where \( \gamma_g \) is the adiabatic index, and \( Re \) is Reynolds number, a dimensionless number characterizing the fluid resistance to motion changes, equal to:

\[ Re = \frac{\rho_{\infty} U_{\infty} L}{\mu}, \] (II.17)

where \( \rho_{\infty}, U_{\infty}, \) and \( L \) are the characteristic density, velocity, and length scales respectively. With the dimensionless variables, Equations II.5, II.6, and II.8 become:

\[ q_x = -\frac{\partial T}{\partial x}, q_y = -\frac{\partial T}{\partial y}, \] (II.18)

\[ \tau = \left[ \nabla u + (\nabla u)^T + \frac{\lambda}{\mu} (\nabla \cdot u) \right], \] (II.19)

\[ p = \rho T \] (II.20)

Since the investigated cases include complex geometry with moving boundary, the system of equations need to be transformed to a rectangular grid to perform the computation to be able to choose mesh points that lay on the body boundary.
II.2 Transformation of the equations in curvilinear moving domain

Since the ultimate goal in this thesis is to simulate an oscillating airfoil, the equations should be written in a moving curvilinear form. This requires the transformation of the physical domain coordinates \(x\) and \(y\) to the variables \(\xi\) and \(\eta\) representing the computational domain. The transformed variables are related to the physical domain with time as well, since the mesh is moving.

\[
\begin{align*}
\xi &= \xi(x, y, t), \quad (\text{II.21}) \\
\eta &= \eta(x, y, t), \quad (\text{II.22}) \\
\tau &= \tau(t). \quad (\text{II.23})
\end{align*}
\]

\(\tau\) and \(t\) are equal since the time variables in physical and computational domains are identical. Using this relationship between the physical and computational domains, the system of equations can be written in the transformed variables via the chain rule:

\[
\frac{\partial U}{\partial \tau} \tau_t + \frac{\partial U}{\partial \xi} \xi_t + \frac{\partial U}{\partial \eta} \eta_t + \frac{\partial F}{\partial \xi} \xi_x + \frac{\partial F}{\partial \eta} \eta_x + \frac{\partial G}{\partial \xi} \xi_y + \frac{\partial G}{\partial \eta} \eta_y = \frac{\partial F}{\partial \xi} \xi_x + \frac{\partial F}{\partial \eta} \eta_x + \frac{\partial G}{\partial \xi} \xi_y + \frac{\partial G}{\partial \eta} \eta_y, \quad (\text{II.24})
\]

where variables \(\xi_t, \eta_t, \text{ etc.}\), are called the metrics. Mathematically, they are the partial derivative of the computational domain variables with respect to the physical domain variables. The system of equations have been transformed to [II.24] to avoid
the nonuniformity of the grid in the physical domain. The above equations can be represented and modeled by a uniform grid in the computational domain. Therefore, the solution can be solved in the computational plane which will correspond to the solution in the physical domain as there is a one to one mapping from the physical variables to the computational variables through equations [II.21], [II.22], and [II.23]. However, it is assumed that there is no analytical expression for the airfoil coordinates. Thus, the transformation is done numerically making it more convenient and accurate to use the inverse transformation:

\[ x = x(\xi, \eta, \tau), \]  
(II.25)

\[ y = y(\xi, \eta, \tau), \]  
(II.26)

\[ t = t(\tau). \]  
(II.27)

A relation between the metric terms and its inverse needs to obtained. This is done by taking differential element of each variable in both the physical and computational domain:

\[ d\xi = \xi_x dx + \xi_y dy + \xi_t dt, \]  
(II.28)

\[ d\eta = \eta_x dx + \eta_y dy + \eta_t dt, \]  
(II.29)

\[ d\tau = dt, \]  
(II.30)
\[ dx = x_\xi d\xi + x_\eta d\eta + x_\tau d\tau, \quad (\text{II.31}) \]
\[ dy = y_\xi d\xi + y_\eta d\eta + y_\tau d\tau, \quad (\text{II.32}) \]
\[ dt = d\tau. \quad (\text{II.33}) \]

Writing the total differential of the variables in a matrix form:
\[
\begin{bmatrix}
  d\xi \\
  d\eta \\
  d\tau
\end{bmatrix}
=\begin{bmatrix}
  \xi_x & \xi_y & \xi_t \\
  \eta_x & \eta_y & \eta_t \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  dx \\
  dy \\
  dt
\end{bmatrix},
\]  
\[
(\text{II.34})
\]
\[
\begin{bmatrix}
  dx \\
  dy \\
  dt
\end{bmatrix}
=\begin{bmatrix}
  x_\xi & x_\eta & x_\tau \\
  y_\xi & y_\eta & y_\tau \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  d\xi \\
  d\eta \\
  d\tau
\end{bmatrix},
\]  
\[
(\text{II.35})
\]

By observing the matrices, it can be noticed that the inverse of the matrix in \[ (\text{II.35}) \] is equal to the matrix in \[ (\text{II.34}) \]. Thus, inverting the matrix in \[ (\text{II.35}) \] gives an expression of the metrics in terms of the the inverse transformation metrics:
\[
\xi_x = \frac{y_\eta}{J}, \quad \xi_y = -\frac{x_\eta}{J}, \quad \xi_t = \frac{x_\eta y_\tau - x_\tau y_\eta}{J},
\]  
\[
(\text{II.36})
\]
\[
\eta_x = -\frac{y_\xi}{J}, \quad \eta_y = \frac{x_\xi}{J}, \quad \eta_t = \frac{x_\xi y_\eta - x_\eta y_\xi}{J},
\]  
\[
(\text{II.37})
\]
\[
J = x_\xi y_\eta - y_\xi x_\eta,
\]  
\[
(\text{II.38})
\]

where \( J \) is the determinant of the transformation matrix, which is referred to as the
Jacobian. Physically, it represents the cell area in the physical domain. Substituting the above expressions in \( \text{II.24} \):

\[
J \frac{\partial U}{\partial \tau} + \frac{\partial U}{\partial \xi} (x_\eta y_\tau - x_\tau y_\eta) + \frac{\partial U}{\partial \eta} (x_\tau y_\xi - x_\xi y_\tau) + \frac{\partial F}{\partial \xi} y_\eta + \frac{\partial F}{\partial \eta} (-y_\xi) + \frac{\partial G}{\partial \xi} (-x_\eta) + \frac{\partial G}{\partial \eta} x_\xi.
\]  

Since finite volume method will be used to solve the compressible Navier-Stokes equations, the equations need to be written in a strong conservative form for better solution accuracy as conservation equations are satisfied even in regions with large discontinuities such as shocks \[2\]. This can be done by applying the product rule derivative for the following term:

\[
\frac{\partial (UJ)}{\partial \tau} = \frac{\partial U}{\partial \tau} J + \frac{\partial J}{\partial \tau} U, \tag{II.40}
\]

Rearranging the above equation to become:

\[
\frac{\partial U}{\partial \tau} J = \frac{\partial (UJ)}{\partial \tau} - \frac{\partial J}{\partial \tau} U. \tag{II.41}
\]

Applying the same technique for the rest of the terms in equation \[\text{II.39}\]:

\[
\frac{\partial F}{\partial \xi} y_\eta = \frac{\partial (F y_\eta)}{\partial \xi} - \frac{\partial y_\eta}{\partial \xi} F, \tag{II.42}
\]

\[
-\frac{\partial F}{\partial \eta} y_\xi = \frac{\partial (-F y_\xi)}{\partial \eta} + \frac{\partial y_\xi}{\partial \eta} F, \tag{II.43}
\]

\[
-\frac{\partial G}{\partial \xi} x_\eta = \frac{\partial (-G x_\eta)}{\partial \xi} + \frac{\partial x_\eta}{\partial \xi} G. \tag{II.44}
\]
\[
\frac{\partial G}{\partial \eta} x_\xi = \frac{\partial (G x_\xi)}{\partial \eta} - \frac{\partial x_\xi}{\partial \eta} G, \tag{II.45}
\]

\[
\frac{\partial F_{\nu}}{\partial \xi} y_\eta = \frac{\partial (F_{\nu} y_\eta)}{\partial \xi} - \frac{\partial y_\eta}{\partial \xi} F_{\nu}, \tag{II.46}
\]

\[-\frac{\partial F_{\nu}}{\partial \eta} y_\xi = \frac{\partial (-F_{\nu} y_\xi)}{\partial \eta} + \frac{\partial y_\xi}{\partial \eta} F_{\nu}, \tag{II.47}\]

\[-\frac{\partial G_{\nu}}{\partial \xi} x_\eta = \frac{\partial (-G_{\nu} x_\eta)}{\partial \xi} + \frac{\partial x_\eta}{\partial \xi} G_{\nu}, \tag{II.48}\]

\[
\frac{\partial G_{\nu}}{\partial \eta} x_\xi = \frac{\partial (G_{\nu} x_\xi)}{\partial \eta} - \frac{\partial x_\xi}{\partial \eta} G_{\nu}. \tag{II.49}\]

By expanding the second term of the right handside of each equation, the second derivatives of the inverse metrics will be cancelled generating the final transformed conservative form of the governing equation:

\[
\vec{\mathbf{u}} \frac{\partial (U J)}{\partial \tau} + \frac{\partial \vec{F}}{\partial \xi} + \frac{\partial \vec{G}}{\partial \eta} = \frac{\partial \vec{F}_{\nu}}{\partial \xi} + \frac{\partial \vec{G}_{\nu}}{\partial \eta}. \tag{II.50}\]

Now, the system of equations is in its final form and suitable for the type of problems that is considered in this paper. It describes the general two dimensional unsteady gas dynamic problem that includes the viscous and the inviscid parts, and it is valid for curvilinear dynamic mesh. Before starting to solve and discretize the equations, the numerical method needs to be chosen first.
Chapter III

Numerical method

III.1 Choosing IMEX schemes

Before choosing the method, the nature of the partial differential governing equations needs to be taken into consideration. By examining the gas dynamic equations in II.10, it is noticed that the equations include hyperbolic term representing the inviscid part and parabolic term corresponding to the diffusion part. The parabolic term arises from the fact that $F_\nu$ and $G_\nu$ include variables, such as shear stress and heat flux, which are related to the first derivative of velocity and temperature respectively. This makes $\frac{\partial F_\nu}{\partial x}$ and $\frac{\partial G_\nu}{\partial y}$ to be second order derivative. If these terms are computed explicitly, the numerical stability condition will be inversely related to the spatial length interval squared. This restricts the time step to be extremely small, which will increase the computational time to reach the steady state solution. Thus, the viscous part is chosen to be solved implicitly. Moreover, implicit methods are more robust and stable. Section III.5 briefly elaborates how multigrid is implemented to calculate the implicit form of the equation.

On the other hand, the hyperbolic term is first order partial differential and non-linear as well. It is more convenient to treat it explicitly as we are interested in
resolving motion in the advective time scale. The numerical scheme used for computing the inviscid flux is second order Godunov scheme, which falls under modern shock capturing methods. It is chosen to avoid unphysical oscillation in the solution. The method is briefly explained in section III.5 which is based on slope limiting [23].

As a result of the previous observation, the numerical method chosen for solving the gas dynamic equations is IMEX scheme and it is based on Runge-Kutta linear multi timesteps. Another reason to choose IMEX Runge-Kutta is to have the option to increase the order of accuracy of the solution and stability region. In the next three sections, three types of IMEX Runge-Kutta scheme are derived for the governing equations, which are forward-backward Euler (1,2,1), implicit-explicit midpoint (1,2,2), and DIRK (2,3,2). The notation (a,b,c) characterizes the scheme where a is the implicit stages, b is the explicit stages, and c is the scheme order of accuracy [3].

III.2 Forward-backward Euler (1,2,1)

According to the notation elaborated in the previous section, this scheme is a first order that consists of two explicit stages as well as one implicit step. The first stage in this scheme is:

\[ U_1 = U_{n-1} + \Delta t[g^D(U_1) + f^H(U_{n-1})], \]  \hspace{1cm} (III.1)

where \( U_{n-1} \) is the solution at the previous time step, \( f^H(U_{n-1}) \) is the inviscid (hyperbolic) term evaluated at the previous time step, and \( g^D(U_1) \) is the diffusive (parabolic) term evaluated at the current time step implicitly. The final stage is:

\[ U_n = U_{n-1} + \Delta t[g^D(U_1) + f^H(U_1)]. \]  \hspace{1cm} (III.2)

The details of density, momentum, and energy equations will be dealt in sequence
for each stage as the solution of the momentum equations depend on the density solution, and the solution of the energy equation depends on both the density and momentum equations.

### III.2.1 First stage

**Density equation**

The density equation does not have diffusion terms. The solution only depends on the inviscid flux which is calculated explicitly according to the first stage formula:

\[
(\rho J)_1 = (\rho J)_{n-1} + \Delta t [f^H(\rho_{n-1})].
\]  

(III.3)

Here, the first stage solution \((\rho J)_1\) is evaluated at the next timestep. As a result, the Jacobian \(J_1\) differs from \(J_{n-1}\) since the mesh is moving. The metrics are updated using the new grid points as well as an equation called geometric conservation law (GCL), which maintains the freestream preservation. Details of this equation is in section [II.7](#).

**Momentum equations**

The momentum equations do have diffusion terms requiring one implicit stage for the Forward-backward (1,2,1). The derivation will be shown for the x-direction momentum as the y-direction momentum follows exactly the same procedure. The first stage has the following form:

\[
(\rho u J)_1 = (\rho u J)_{n-1} + \Delta t [f^H(\rho u)_{n-1} + g^D_{\rho u}(\rho u)_1].
\]  

(III.4)

The viscous flux corresponding to the momentum in [II.10](#) is in terms of shear stress. It can be written as:
For simplification, it is assumed in this stage that $\lambda = -\mu$ in equation (II.19) as this assumption decouples the x and y momentum for the purpose of the implicit step so $u$ and $v$ can be computed separately. This assumption will be relaxed in the future requiring modification of the current implemented multigrid solver. As for now, it is a convenient assumption for low Mach number flows as the divergence of velocity is not significant in affecting the solution. Therefore, the shear stress divergence becomes:

$$\nabla \cdot \tau = \left[ \nabla \cdot (\nabla u) \right].$$  \hspace{1cm} (III.6)

So, the viscous flux becomes the Laplacian of the velocity:

$$g^D_{pu} = \frac{J_1 \nabla^2 u_1}{Re}.$$ \hspace{1cm} (III.7)

Substituting in (III.4)

$$(puJ)_1 = (puJ)_{n-1} + \Delta t \left[ f^H_{pu}(pu)_{n-1} + \frac{J_1 \nabla^2 u_1}{Re} \right].$$ \hspace{1cm} (III.8)

By moving the terms that include $u_1$ to the left side, the right hand side of equation becomes:

$$\tilde{S}_{pu} = (puJ)_{n-1} + \Delta t f^H_{pu}(pu)_{n-1},$$ \hspace{1cm} (III.9)

where $\tilde{S}_{pu}$ includes the solution at the previous time step as well as the inviscid momentum flux calculated explicitly. Dividing by $\rho_1 J_1$ and taking the velocity $u_1$ as a common factor, the equations becomes in the following modified Helmholtz equation:

$$\left( I - \beta_{pu} \nabla^2 \right) u_1 = \frac{\tilde{S}_{pu}}{\rho_1 J_1} = S_{pu},$$ \hspace{1cm} (III.10)
where the quantity $\beta_{\rho u}$ is equal to:

$$\beta_{\rho u} = \frac{\Delta t}{Re\rho_1}. \quad (III.11)$$

Equation $III.10$ is elliptic and can be solved efficiently by multigrid method. Now, the previous solution is ready to be advanced to the first stage:

$$(\rho u J)_1 = (\rho u J)_{n-1} + \Delta t \left[ f^H_{\rho u}(\rho u)_{n-1} + \frac{J_1}{Re} \nabla^2 u_1 \right]. \quad (III.12)$$

**Energy equation**

The energy equation is solved similar to the momentum equation, but the diffusive part has an extra term, which is the shear work term. It will be treated separately as it can be computed explicitly since the velocity and shear stress are known at the next time step from the updated momentum equations. The first stage energy equation is:

$$(EJ)_1 = (EJ)_{n-1} + \Delta t \left[ f^H_E(E_{n-1}) + g^D_E(E_1) \right]. \quad (III.13)$$

Using Fourier’s law in $II.5$, the energy viscous flux in $II.10$ can be written as:

$$g^D_E = \frac{J}{RePr(\gamma_\rho - 1)} \nabla^2 T + \frac{J}{Re} \nabla \cdot (\tau u). \quad (III.14)$$

At the first stage, it becomes:

$$g^D_E(E_1) = \frac{J_1}{RePr(\gamma_\rho - 1)} \nabla^2 T_1 + \frac{J_1}{Re} \nabla \cdot (\tau u)_1. \quad (III.15)$$

Substituting in $III.13$:

$$(EJ)_1 = (EJ)_{n-1} + \Delta t f^H_E(E_{n-1}) + \frac{J_1 \Delta t}{RePr(\gamma_\rho - 1)} \nabla^2 T_1 + \Delta t \frac{J_1}{Re} \nabla \cdot (\tau u)_1. \quad (III.16)$$
The total energy is equal the internal and kinetic energy:

\[ E = \frac{P}{\gamma_g - 1} + \frac{\rho V^2}{2} = \frac{P}{\gamma_g - 1} + \frac{\rho(u^2 + v^2)}{2}. \]  \hspace{1cm} (III.17)

Using equation [II.20] the right hand side of the modified Helmholtz equation includes all the terms that do not have \( T_1 \):

\[ \tilde{S}_T = (EJ)_{n-1} - \frac{J_1 \rho_1 (u_1^2 + v_1^2)}{2} + \Delta t f^H_{E_n-1} + \Delta t \frac{J_1 \nabla \cdot (\tau u)_1}{Re}. \] \hspace{1cm} (III.18)

Substituting the right hand side [III.18] to the first stage equation [III.13]:

\[ \frac{J_1 \rho_1 T_1}{\gamma_g - 1} = \tilde{S}_T + \frac{J_1 \Delta t}{Re Pr (\gamma_g - 1)} \nabla^2 T_1. \] \hspace{1cm} (III.19)

Writing the equation in the modified Helmholtz form using the same technique as done in the momentum:

\[ \left( I - \frac{\Delta t \nabla^2}{Re Pr \rho_1} \right) T_1 = \frac{\tilde{S}_T (\gamma_g - 1)}{J_1 \rho_1}, \] \hspace{1cm} (III.20)

where the the quantities \( \beta_T \) and \( S_T \) are defined to be:

\[ \beta_T = \frac{\Delta t}{Re Pr \rho_1}, \] \hspace{1cm} (III.21)

\[ S_T = \frac{\tilde{S}_T (\gamma_g - 1)}{J_1 \rho_1}. \] \hspace{1cm} (III.22)

Thus, equation [III.20] becomes:

\[ (I - \beta_T \nabla^2) T_1 = S_T. \] \hspace{1cm} (III.23)

Equation [III.23] is in elliptic form as well. \( T_1 \) can be calculated using multigrid
method, and the solution is ready to be advanced to the first stage:

\[
(EJ)_1 = (EJ)_{n-1} + \Delta t f^H_E(E_{n-1}) + \frac{J_1 \Delta t}{Re Pr (\gamma_g - 1)} \nabla^2 T_1 + \Delta t \frac{J_1 \nabla \cdot (\tau u)_1}{Re}. \quad (III.24)
\]

### III.2.2 Final stage

#### Density equation

The final stage of the density equation is:

\[
(\rho J)_n = (\rho J)_{n-1} + \Delta t [f^H_\rho (\rho_1)]. \quad (III.25)
\]

The inviscid flux \( f^H_\rho (\rho_1) \) is evaluated at the solution of the first stage, and the Jacobian \( J_n \) is equal to \( J_1 \) as both are evaluated at the next time step for this scheme.

#### Momentum equations

The final stage is updated similar to the density equation:

\[
(\rho u J)_n = (\rho u J)_{n-1} + \Delta t [f^H_{\rho u} (\rho u)_1 + g^D_{\rho u} (\rho u_1)]. \quad (III.26)
\]

#### Energy equation

The final stage is evaluated to be:

\[
(EJ)_n = (EJ)_{n-1} + \Delta t [f^H_E (E_1) + g^D_E (E_1)]. \quad (III.27)
\]
III.3 Implicit-explicit midpoint (1,2,2)

This scheme is exactly the same as forward-backward Euler (1,2,1), except that the first stage is evaluated at the midpoint corresponding to the half timestep. Thus, the first and final stage become:

\[
U_1 = U_{n-1} + \frac{\Delta t}{2}[g^D(U_1) + f^H(U_{n-1})], \tag{III.28}
\]

\[
U_n = U_{n-1} + \Delta t[g^D(U_1) + f^H(U_1)]. \tag{III.29}
\]

This slight change increases the order of accuracy to become second-order with the same number of explicit and implicit stages as in forward-backward Euler (1,2,1).

III.4 DIRK (2,3,2)

This scheme, diagonally implicit Runge-Kutta (DIRK), is second order accurate. It has the same order of accuracy as implicit-explicit midpoint (1,2,2), but it is L-stable making it less susceptible to numerical instabilities. It consists of two implicit and three explicit stages:

\[
(U.J)_1 = (U.J)_{n-1} + \Delta t\gamma[g^D(U_1) + f^H(U_{n-1})], \tag{III.30}
\]

where \(U_{n-1}\) is the solution at the previous time step, \(f^H(U_{n-1})\) is the inviscid (hyperbolic) term evaluated at the previous time step, and \(g^D(U_1)\) is the diffusive (parabolic) term evaluated at the current time step \(\gamma\Delta t\) implicitly. The second stage is:

\[
(U.J)_2 = (U.J)_{n-1} + \Delta t[(1-\gamma)g^D(U_1)+\gamma g^D(U_2)+\delta f^H(U_{n-1})+(1-\delta)f^H(U_1)], \tag{III.31}
\]
where $U_{n-1}$ is the solution at the previous time step, $f^H(U_1)$ is the inviscid (hyperbolic) term evaluated at the first stage time step, and $g^D(U_2)$ is the viscous (parabolic) term evaluated at the current time step $\Delta t$ implicitly. The constants $\gamma$ and $\delta$ are:

$$
\gamma = \frac{2 - \sqrt{2}}{2}, \quad \delta = -\frac{2\sqrt{2}}{3}.
$$

(III.32)

The final stage to advance to the next timestep is:

$$(UJ)_n = (UJ)_{n-1} + \Delta t \left[(1 - \gamma) \left(g^D(U_1) + f^H(U_1)\right) + \gamma \left(f^H(U_2) + f^H(U_2)\right)\right].$$

(III.33)

The same methodology illustrated in forward-backward (1,2,1) is used to solve the equations, but this scheme requires an extra implicit stage as well as another explicit stage. Section III.2 corresponding to Forward-backward Euler (1,2,1) includes more explanation about the equations.

### III.4.1 First stage

#### Density equations

As mentioned in III.2, the density equation does not have diffusion terms making the density update to be explicit:

$$(\rho J)_1 = (\rho J)_{n-1} + \Delta t \gamma \left[f^H_\rho (\rho_{n-1})\right].$$

(III.34)

Here, the first stage solution $(\rho J)_1$ is evaluated at the timestep which is $\gamma \Delta t$ instead of $\Delta t$ in the case of forward-backward Euler (1,2,1) scheme, or $1/2 \Delta t$ in forward-backward Euler (1,2,2).
Momentum equations

The first stage of this scheme in solving the momentum equations is similar to forward-backward Euler except that the timestep is $\gamma \Delta t$

$$(\rho u J)_{1} = (\rho u J)_{n-1} + \Delta t \gamma \left[ f_{\rho u}^{H}(\rho u)_{n-1} + g_{\rho u}^{D}(\rho u)_{1} \right]. \quad (III.35)$$

The viscous term is equal to:

$$g_{\rho u}^{D} = \frac{J}{Re} \cdot \nabla \cdot \tau. \quad (III.36)$$

Using the same assumption that is mentioned in section [III.2] the diffusive flux becomes:

$$g_{\rho u}^{D} = \frac{J}{Re} \frac{\nabla^{2} u}{Re}. \quad (III.37)$$

Substituting in (III.35)

$$(\rho u J)_{1} = (\rho u J)_{n-1} + \Delta t \gamma \left[ f_{\rho u}^{H}(\rho u)_{n-1} + \frac{J}{Re} \frac{\nabla^{2} u_{1}}{Re} \right]. \quad (III.38)$$

Using the same technique in [III.2] for defining the right hand side of the modified Helmholtz equation:

$$\tilde{S}_{\rho u} = (\rho u J)_{n-1} + \Delta t \gamma f_{\rho u}^{H}(\rho u)_{n-1}. \quad (III.39)$$

The final modified Helmholtz equation is reached via solving for $u_{1}$:

$$\left( I - \beta_{\rho u} \nabla^{2} \right) u_{1} = \frac{\tilde{S}_{\rho u}}{J_{1} \rho_{1}} = S_{\rho u}, \quad (III.40)$$

where the constant $\beta_{\rho u}$ is equal to:
\[ \beta_{pu} = \frac{\gamma \Delta t}{\rho_1 Re}. \]  

(III.41)

\( u_1 \) is obtained from (III.40) via multigrid method as it is an elliptic equation. Then, the Laplacian of \( u_1 \) is calculated to obtain the viscous flux to update the solution at the first stage:

\[ (\rho u J)_1 = (\rho u J)_{n-1} + \Delta t \gamma \left[ f^H_{\rho u}(\rho u)_{n-1} + \frac{J_1 \nabla^2 u_1}{Re} \right]. \]  

(III.42)

**Energy equation**

The first stage of the energy equation is as follows:

\[ (EJ)_1 = (EJ)_{n-1} + \gamma \Delta t \left[ f^H_E(E_{n-1}) + g^D_E(E_1) \right], \]  

(III.43)

Mathematically, the viscous flux is equal to:

\[ g^D_E = \frac{J}{Re Pr (\gamma_g - 1)} \nabla^2 T + \frac{J \nabla \cdot (\tau u)}{Re}. \]  

(III.44)

Substituting in the original equation (III.43)

\[ (EJ)_1 = (EJ)_{n-1} + \gamma \Delta t \left[ f^H_E(E_{n-1}) + \frac{J_1}{Re Pr (\gamma_g - 1)} \nabla^2 T_1 + \frac{J_1 \nabla \cdot (\tau u)_1}{Re} \right]. \]  

(III.45)

Using equation (III.17) and (II.20) to solve in term of temperature, equation (III.45) becomes:
\[
\frac{J_1 \rho_1 RT_1}{\gamma_g - 1} + \frac{J_1 \rho_1 \eta^2}{2} = (EJ)_{n-1} + \gamma \Delta t \left[ f_E^H(E_{n-1}) 
+ \frac{J_1}{Re Pr(\gamma_g - 1)} \nabla^2 T_1 + J_1 \nabla \cdot (\tau u)_1 \right]. 
\]  

(III.46)

Defining the right hand side of the equation to be:

\[
\tilde{S}_T = (EJ)_{n-1} + \frac{J_1 \rho_1 \eta^2}{2} + \gamma \Delta t f_E^H(E_{n-1}) + \gamma \Delta t \frac{J_1 \nabla \cdot (\tau u)_1}{Re}. 
\]  

(III.47)

Thus, equation (III.46) becomes:

\[
\frac{J_1 \rho_1 T}{\gamma_g - 1} = \tilde{S}_T + \frac{J_1 \gamma \Delta t}{Re Pr(\gamma_g - 1)} \nabla^2 T_1. 
\]  

(III.48)

Writing the equation in modified Helmholtz form using the same technique in section (III.2) equation (III.47) becomes:

\[
\left( I - \frac{\gamma \Delta t \nabla^2}{Re Pr \rho_1} \right) T_1 = \frac{S_T(\gamma_g - 1)}{J_1 \rho_1} = S_T. 
\]  

(III.49)

Define \( \beta_T \) to be:

\[
\beta_T = \frac{\gamma \Delta t}{Re Pr \rho_1}. 
\]  

(III.50)

The final form to solve for the temperature is:

\[
\left( I - \beta_T \nabla^2 \right) T_1 = S_T. 
\]  

(III.51)

Once the temperature is computed implicitly, the first stage viscous flux is calculated and substituted in equation (III.43).
\[ (EJ)_1 = (EJ)_{n-1} + \gamma \Delta t \left[ f_E^H(E_{n-1}) + \frac{J_1 k}{RePr(\gamma - 1)} \nabla^2 T_1 + \frac{J_1 \nabla \cdot (\tau u)_1}{Re} \right]. \quad \text{(III.52)} \]

### III.4.2 Second stage

#### Density equation

The second stage of the density equation is evaluated explicitly in a similar manner to the first stage, but the right hand side includes the inviscid flux evaluated at the previous and first stage time steps multiplied by specific constants:

\[ (\rho J)_2 = (\rho J)_{n-1} + \Delta t \left[ \delta f^H_\rho (\rho_{n-1}) + (1 - \delta) f^H_\rho (\rho_1) \right]. \quad \text{(III.53)} \]

The second stage solution \((\rho J)_2\) is computed at \(\Delta t\). Thus, the jacobian \(J_2\) is found via GCL in [III.86] which is calculated at time step \(\Delta t\).

#### Momentum equations

Using the second stage formula in [III.31], the second stage of the momentum equation is equal to:

\[ (\rho u J)_2 = (\rho u J)_{n-1} + \Delta t \left[ (1 - \gamma) g^D_{\rho u} (\rho u)_1 + \gamma g^D_{\rho u} (\rho u)_2 \right. \]
\[ + \left. \delta f^H_{\rho u} (\rho u)_{n-1} + (1 - \delta) f^H_{\rho u} (\rho u)_1 \right]. \quad \text{(III.54)} \]

Using the same technique in defining the right hand side of the momentum equation:

\[ \tilde{S}_{\rho u} = (\rho u J)_{n-1} + \Delta t \left[ (1 - \gamma) g^D_{\rho u} (\rho u)_1 + \delta f^H_{\rho u} (\rho u)_{n-1} + (1 - \delta) f^H_{\rho u} (\rho u)_1 \right]. \quad \text{(III.55)} \]
It is noticed in III.55 that the $\tilde{S}_{\rho u}$ has more terms than the right hand side in the first stage. Substituting in III.54,

\[(\rho u J)_2 = \frac{J_2 \gamma \Delta t \nabla^2 u_2}{Re} + \tilde{S}_{\rho u}.\]  \hspace{1cm} (III.56)

The final modified Helmholtz equation is reached through the same algebraic techniques used previously in the first stage:

\[(I - \beta_{\rho u} \nabla^2)u_2 = \frac{\tilde{S}_{\rho u}}{J_2 \rho_2^2} = S_{\rho u},\]  \hspace{1cm} (III.57)

where $\beta_{\rho u}$ is:

\[\beta_{\rho u} = \frac{\gamma \Delta t}{\rho_2^2 Re}.\]  \hspace{1cm} (III.58)

Solving the elliptic equation for $u_2$ using multigrid, the second stage momentum solution is found via:

\[(\rho u J)_2 = (\rho u J)_{n-1} + \Delta t \left[ (1 - \gamma)g_{\rho u}^{D}(\rho u)_1 + \gamma \frac{J_2 \nabla^2 u_2}{Re} \right.

\left. + \delta f_{\rho u}^{H}(\rho u)_{n-1} + (1 - \delta)f_{\rho u}^{H}(\rho u)_1 \right].\]  \hspace{1cm} (III.59)

**Energy equation**

Using the formulae in III.31 and III.44, the second stage energy equation is:

\[(E J)_2 = (E J)_{n-1} + \Delta t \left[ (1 - \gamma)g_{E}^{D}(E_1) + \gamma \left( \frac{J_2 \Delta t}{RePr(\gamma_g - 1)} \nabla^2 T_2 + \frac{J_2 \nabla \cdot (\tau u)_2}{Re} \right) \right.

\left. + \delta f_{E}^{H}(E_{n-1}) + (1 - \delta)f_{E}^{H}(E_1) \right].\]  \hspace{1cm} (III.60)
The right hand side is defined similar to the previous stage to become:

$$
\bar{S}_T = (EJ)_{n-1} - \frac{J_2 \rho_2 u_2^2}{2} + \Delta t \left[ \delta f^H_E (E_{n-1}) + (1 - \delta) f^H_E (E_1) 
+ (1 - \gamma) g^D_E (E_1) + \gamma J_2 \frac{\nabla \cdot (\tau u)_2}{Re} \right].
$$  \hspace{1cm} (III.61)

Substituting the above right hand side in the second stage equation [III.60]:

$$
\frac{J_2 \rho_2 RT}{\gamma g - 1} = \bar{S}_T + \frac{J_2 \gamma \Delta t}{Re Pr (\gamma g - 1)} \nabla^2 T_2.
$$  \hspace{1cm} (III.62)

The final modified Helmholtz form of the implicit equation becomes:

$$
(I - \beta_T \nabla^2) T_2 = \frac{\bar{S}_T (\gamma g - 1)}{J_2 \rho_2} = S_T,
$$  \hspace{1cm} (III.63)

where $\beta_T$ is:

$$
\beta_T = \frac{\gamma \Delta t}{Re Pr \rho_2}.
$$  \hspace{1cm} (III.64)

The temperature at the second stage is found through multigrid method. Then, it is used to calculate the energy viscous flux to find the total energy at the second stage:

$$
(EJ)_2 = (EJ)_{n-1} + \Delta t \left[ (1 - \gamma) g^D_E (E_1) + \gamma \left( \frac{J_2 \Delta t}{Re Pr (\gamma g - 1)} \nabla^2 T_2 + \frac{J_2 \nabla \cdot (\tau u)_2}{Re} \right)
+ \delta f^H_E (E_{n-1}) + (1 - \delta) f^H_E (E_1) \right].
$$  \hspace{1cm} (III.65)
III.4.3 Final stage

The final stage requires the inviscid flux to be calculated at the second stage. Then, the formula III.33 is used for the density, momentum, and energy equations.

Density equation

The density at the next time step is found using the inviscid flux calculated at the previous time step, first stage, and second stage:

\[ (\rho J)_n = (\rho J)_{n-1} + \Delta t \left[ (1 - \gamma) f^H_\rho (\rho_1) + \gamma f^H_\rho (\rho_2) \right]. \] (III.66)

Momentum equations

The momentum uses the viscous and inviscid fluxes computed at the previous time, first stage, and second stage to become:

\[ (\rho u J)_n = (\rho u J)_{n-1} + \Delta t \left[ (1 - \gamma)(g^D_{\rho u} (\rho u)_1 + g^D_{\rho u} (\rho u)_2) + \gamma (f^H_{\rho u} (\rho u)_2 + f^H_{\rho u} (\rho u)_1) \right]. \] (III.67)

Energy equation

The final step of the numerical method is to find the energy using the calculated fluxes at the aforementioned stages (previous, first, and second stage time steps):

\[ (E J)_n = (E J)_{n-1} + \Delta t \left[ (1 - \gamma)(g^D_E (E)_1 + g^D_E (E)_2) + \gamma (f^H_E (E)_2 + f^H_E (E)_1) \right]. \] (III.68)
III.5 Godunov second order scheme

Godunov scheme is a powerful numerical method that is used to compute the hyperbolic fluxes accurately. As stated before, the non-linearity in the inviscid flux can result in shock waves and regular schemes based on finite difference do not capture shocks correctly as they produce oscillation in the solution. On the other hand, Godunov scheme captures shocks smoothly. It consists of three major:

- Approximate the solution over each cell initially. The second order Godunov scheme approximates the solution as linear piecewise. As a result, there will be discontinuities at the cell interfaces. For instance, the left and right states at the cell interface $\eta_{i+1/2}$ is obtained via:

\[
V_L = V_i + \frac{\Delta \eta}{2} \left( \frac{\partial V}{\partial \eta} \right)_i, \quad (\text{III.69})
\]

\[
V_R = V_{i+1} - \frac{\Delta \eta}{2} \left( \frac{\partial V}{\partial \eta} \right)_{i+1}, \quad (\text{III.70})
\]

where $V_L$ and $V_R$ are the left and right states respectively. In choosing the slope of piecewise linear approximation of the cell values $\left( \frac{\partial V}{\partial \eta} \right)_i$, slope limiting function has been implemented to avoid unphysical oscillations. Mathematically, the slope is defined as:

\[
\left( \frac{\partial V}{\partial \eta} \right)_i = [\Lambda]^{-1} \text{minmod}(\tilde{V}_i, \tilde{V}_{i+1}, \tilde{V}_{i-1}), \quad (\text{III.71})
\]

\[
\tilde{V}_{i+k} = [\Lambda]_i V_{i+k}, \quad (\text{III.72})
\]

where $\text{minmod}$ represents the slope limiting function, and $[\Lambda]$ is a matrix defined.
as the left eigenvectors of the Jacobian $\frac{\partial G}{\partial V}$. This approach results in a second order accuracy.

- The discontinuity at each cell interface represents a Riemann problem, which can be exactly solved or approximated. In Godunov’s scheme, the exact solution is found at each cell interface which represents values of the state variables density, velocity, and pressure.

- The new computed state variables at the interface are used to average the cell value via linear piecewise approximation.

Refer to [30], [23], and [13] for more details about the method.

### III.6 Multigrid method

The purpose of using multigrid method is its computational efficiency in finding the solution of elliptic problems. The implicit approach in the derived IMEX schemes results in an elliptic problem, which is the modified Helmholtz equation. In addition, elliptic problems arise in generating the grid and finding the mesh velocity nodes as Laplace equation, which is an elliptic problem, is solved for both cases (equations are in the next section III.7). The multigrid algorithm is based on two basic principles: error smoothing, and grid coarsening. First, the error or residual in the initial guess is smoothened via Gauss-Seidel relaxation meaning that the high frequency errors are significantly reduced leaving only low frequency errors. When this is accomplished, the error in a coarse grid will be close to the error in the original grid. As a result, solution convergence can be reached via solving the solution on the coarse grid which is less expensive in terms of computation. These two steps are done multiple times to reach very coarse grid. Afterwards, the coarse grid is interpolated to a finer grid and then the new solution approximation is computed after smoothing it again.
The method can be designed more efficiently through selecting when to smooth, coarsen, and interpolate. In the developed solver used for this thesis, V-cycle is selected with a maximum length 10. Also, the number of smoothing iterations is fixed to be 10 for both pre-smoothing and post-smoothing. Details of multigrid algorithm can be found in reference [29].

### III.7 Numerical method for moving grid

Before attempting to solve equation [11.50] the initial mesh coordinates need to be specified. That is, the grid points need to be determined in the physical plane, which will correspond to uniform grid points in the computational plane. The method chosen in generating the mesh is elliptic method suggested by Thompson [28]. Basically, the method requires the coordinates of physical domain boundary which corresponds to the known computational domain rectangular boundary. The internal grid points is determined via solving the following homogeneous laplace equation:

\[
\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} = 0, \quad (III.73)
\]

\[
\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} = 0. \quad (III.74)
\]

Since \( \xi \) and \( \eta \) represent the rectangular computational domain, they are known. \( x \) and \( y \) are unknown in the internal grid points. Thus, an inverse transformation is required which produces the following the equations:

\[
\alpha \frac{\partial^2 x}{\partial \xi^2} - 2\beta \frac{\partial^2 x}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 x}{\partial \xi^2} = 0, \quad (III.75)
\]

\[
\alpha \frac{\partial^2 y}{\partial \xi^2} - 2\beta \frac{\partial^2 y}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 y}{\partial \xi^2} = 0, \quad (III.76)
\]
where $\alpha$, $\beta$, and $\gamma$ are:

$$\alpha = x_\eta^2 + y_\eta^2$$  \hspace{1cm} (III.77)

$$\beta = x_\eta x_\eta + y_\eta y_\eta$$  \hspace{1cm} (III.78)

$$\gamma = x_\xi^2 + y_\xi^2$$  \hspace{1cm} (III.79)

Now, equations [III.75] and [III.76] are discretized and solved via multigrid method. As time advances, the boundaries have velocity values. Since an oscillating airfoil is modeled, the internal boundary have a rotating velocity whereas the outer boundary have zero velocity. As a result, the velocity of the internal grid points can be computed via Laplace equation [15] given the velocity of the boundaries:

$$\alpha \frac{\partial^2 u_x}{\partial \xi^2} - 2\beta \frac{\partial^2 u_x}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 u_x}{\partial \xi^2} = 0,$$  \hspace{1cm} (III.80)

$$\alpha \frac{\partial^2 u_y}{\partial \xi^2} - 2\beta \frac{\partial^2 u_y}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 u_y}{\partial \xi^2} = 0,$$  \hspace{1cm} (III.81)

$$u_x = (u_x)_{airfoil}, u_y = (u_y)_{airfoil}, \quad \text{on the inner boundary},$$  \hspace{1cm} (III.82)

$$u_x = 0, u_y = 0, \quad \text{on the outer boundary}.$$  \hspace{1cm} (III.83)

By solving this elliptic equation using multigrid method, the inner grid points velocities are determined. The x and y coordinates are updated via:

$$x_n = x_{n-1} + u_x dt$$  \hspace{1cm} (III.84)
\[ y_n = y_{n-1} + u_y dt \] (III.85)

This mesh movement causes the grid cells to be deformed. As a result, an extra equation, first mentioned by Thomas and Lombard [27], has to be satisfied to preserve the free stream flow. It represents the change in the Jacobian with respect to time, which can be derived from II.50 via making \( U, F, G, F_\nu, \) and \( G_\nu \) constant. Thus, the equation becomes:

\[
\frac{\partial J}{\partial \tau} + \frac{\partial (y_\tau x_\eta - x_\tau y_\eta)}{\partial \xi} + \frac{\partial (x_\tau y_\xi - y_\tau x_\xi)}{\partial \eta} = 0 \]

(III.86)

The above equation is called the geometric law conservation (GCL), and it is used to update the Jacobian at the next time step explicitly. Several test cases were run with and without the GCL implementation. It was observed that the solution accuracy slightly improved. Therefore, the order of accuracy depends on the IMEX scheme.
Chapter IV

Results and discussion

This chapter presents validation of the numerical method. First, it shows the grid parameters used for the investigated cases, followed by the boundary conditions of the physical domain. Then, the convergence parameters of multigrid and Godunov solvers are stated. Afterwards, equations and quantities showing how some numerical results are compared to results in the literature are elaborated. Finally, the results of each simulated case are illustrated with plots and figures.

IV.1 Grid specifications

The grid type used for the studied cases is C-grid, which has more concentrated points in the wake region than the O-grid type. In terms of domain size, the wake line has been extended to reach 14 airfoil chord lengths and the radius C-shape region is 10 chord length making the domain size 20 chord lengths in the y-direction. The reason behind these specifying these large dimensions is to mitigate and avoid any possible noise generated in the boundary from propagating back to the domain around the airfoil causing inaccurate numerical solution [16]. Figures (a) and (b) in IV.1 show the grid used for the simulation done in the second case, where the airfoil is NACA-0012 that is static at an angle of attack $\alpha = 16$. Refer to table IV.2 for more details
on the parameters used in the model.

Figure IV.1: The physical domain represented by $256 \times 256$ C-grid for the second case, where the airfoil is static at $\alpha = 16$
IV.2 Initial and boundary conditions

Specifying the boundary conditions in the inflow and outflow depends on the flow speed. Since the flow is subsonic, three variables need to be specified at the inflow boundary whereas only one variable is determined at the outflow \[22\]. The following boundary conditions are chosen:

- The inflow boundary: \( p = p_\infty = 1 \), \( u = U_\infty = 0.4Ma \), and \( v = v_\infty = 0 \)
- The outflow boundary: \( p = p_\infty = 1 \), \( \frac{\partial p}{\partial \xi} = 0 \), \( \frac{\partial u}{\partial \xi} = 0 \), and \( \frac{\partial v}{\partial \xi} = 0 \)
- The airfoil boundary: \( u = 0 \), \( v = 0 \) (no slip conditions), and \( \frac{\partial T}{\partial \eta} = 0 \) (adiabatic surface)
- The wake line: the wake line is assumed to be continuous meaning that the cells above the wake line receive flux from the cells below the line and vice versa.

IV.3 Solver parameters

The developed code iterates and calls the function that solves the dynamic gas equations every time step. It stops after reaching the maximum iteration which corresponds to number of time steps specified by the user. Within the fluid solver, there are two main iterative solvers:

- Riemann solution in Godunov’s scheme: In calculating the inviscid flux, Riemann solver uses Newton’s method to calculate the pressure corresponding to the solution of two non-linear curves. The convergence tolerance used for the iterative method is \( tol = 10^{-8} \).

- Implicit scheme: In calculating the viscous flux, iterative multigrid algorithm is used to reduce the residual and find the approximate solution. The convergence tolerance used for the iterative method is \( tol = 10^{-8} \).
In generating the initial C-grid and calculating the velocity mesh points for the mesh motion, the same multigrid solver is used to converge to the solution. As for calculating the error, the infinity norm is used which is more restrictive than the two-norm error making the tolerance more strict.

IV.4 Validation diagnostics

Two specific quantities are used to validate the generated numerical results, which are the lift and drag coefficients. They are computed using the following equations:

\[ c_l = \frac{F_L}{2\rho_\infty U_\infty^2 c}, \]  

\[ (IV.1) \]

\[ c_d = \frac{F_D}{2\rho_\infty U_\infty^2 c}, \]  

\[ (IV.2) \]

where \( c \) is the airfoil chord length. \( F_L \) and \( F_D \) are the lift and drag forces on the airfoil calculated via:

\[ F_L = \int_l dF_L, \]

\[ (IV.3) \]

\[ F_D = \int_l dF_D, \]

\[ (IV.4) \]

where \( l \) represents the airfoil circumference, since it is a two-dimensional model. The force vector can be expressed as:

\[ d\mathbf{F} = dF_D \hat{e}_x + dF_L \hat{e}_y, \]

\[ (IV.5) \]

The contribution of the forces on the airfoil comes from the pressure as well as the shear stress:
The force due to pressure can be divided into normal and tangential to the airfoil surface:

\[ dF = Pdl\hat{\mathbf{n}} + \tau d\hat{\mathbf{n}} = (P\hat{\mathbf{n}} + \tau\hat{\mathbf{n}})dl. \] (IV.6)

The force due to pressure can be divided into normal and tangential to the airfoil surface:

\[ P\hat{\mathbf{n}} = Pn_x\hat{e}_x + Pn_y\hat{e}_y. \] (IV.7)

The stress is a tensor that can be expressed as:

\[ \tau = \tau_{ij}e_i \otimes e_j = \tau_{xx}e_x \otimes e_x + \tau_{xy}e_x \otimes e_y + \tau_{yx}e_y \otimes e_x + \tau_{yy}e_y \otimes e_y. \] (IV.8)

The shear stress force is the shear tensor multiplied by the surface normal:

\[ \tau\hat{\mathbf{n}} = \tau_{xx}n_x\hat{e}_x + \tau_{xy}n_y\hat{e}_x + \tau_{yx}n_x\hat{e}_y + \tau_{yy}n_y\hat{e}_y. \] (IV.9)

The lift force on a small airfoil element takes the normal force component:

\[ dF_L = (Pn_y + \tau_{xy}n_y + \tau_{yy}n_y)dl. \] (IV.10)

The drag force on a small airfoil element takes the tangential component:

\[ dF_D = (Pn_x + \tau_{yx}n_x + \tau_{xx}n_x). \] (IV.11)

The drag coefficient can also be computed in a different method that depends on a parameter called momentum thickness. This parameter can be defined as the distance required for the airfoil or body to move in the same direction of the flow speed to reach the same initial momentum in the inflow. Mathematically for compressible flow [24], it is equal to:
\[ \theta = \int_0^\infty \frac{\rho(\eta)u(\eta)}{\rho_0u_0}(1 - \frac{u(\eta)}{u_0})d\eta. \]  

(IV.12)

The drag coefficient, under incompressible flow limits, becomes:

\[ c_d = 2\theta. \]  

(IV.13)

Besides computing the aforementioned coefficients, visual comparison is done for validating the result.

IV.5 Case studies

Three cases are investigated to validate the proposed numerical method, and all of them are simulated for a laminar flow around NACA-0012 airfoil. These specific cases are chosen, since there are similar available cases done previously in the literature. For the first two cases, the simulation is done on static airfoil at different angle of attacks which is compared to the experimental table in [25]. Tables IV.1 and IV.2 show the flow parameters chosen for the first two cases. For the third case, the airfoil has a harmonic motion, and it is simulated and compared with the numerical result in reference [4]. Table IV.3 provides the essential parameters related to the third case simulation.

In running the code to simulate the test cases, gfortran compiler is used to compile the code with -O3 optimization option. On an average, it takes 1.16 seconds per time step to simulate 256 by 256 grid size on an Intel Xeon 2.67 Ghz CPU. No serious attempts were made to optimize the code performance.

Before studying the simulation cases, the effect of the assumption \( \lambda = -\mu \) made in equation II.19 in the implicit stage is briefly examined. Figure IV.2 is plotted to illustrate the magnitude of the neglected term.
Figure IV.2: Relative magnitude of the ignored term in the implicit stage for the second case, where $t = 21.7$, $\alpha = 16$, $Re = 40000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012

It is observed that the magnitude of the neglected term associated with the assumption is the highest near the stagnation stream line and the region with high acceleration, but it is still relatively low and does not have a significant impact on the solution. In other regions, the magnitude is extremely low and can be assumed negligible.
First case: static NACA-0012 at angle of attack $\alpha = 5$

This case investigates the validation of the numerical method at a low angle of attack. Comparing the numerical and experimental [25] lift and drag coefficients in table IV.1, it is observed that there is small discrepancy, especially in the lift coefficient. The difference is about 10% in the lift coefficient and around 20% in the drag coefficient. Part of this difference is due to the poor quality of the used mesh as the mesh points are not forced to be orthogonal or clustered around the airfoil. In addition, the mesh is coarse as $N = 128$ for this case. These factors affect the results significantly, especially for the computed drag coefficient as the drag coefficient depends mostly on the viscous forces, requiring clustered points and refined mesh around the airfoil to obtain the accurate magnitude.

Figure IV.3 shows the steady state solution of the horizontal velocity, and figure IV.4 illustrates the pressure distribution. There is a high pressure below the airfoil which produces an upward lift. Figure IV.5 shows the horizontal velocity across the outflow boundary. The sudden decrease in the horizontal velocity magnitude occurs in the wake region, which is expected. The small spikes in the wake region observed in figure IV.5 is due to the cut-line that was taken manually in VisIt visualization software as it does not perfectly represent the outflow horizontal velocity magnitude. The above observations show that the IMEX method produces acceptable results for static NACA-0012 at low angle of attack.
Figure IV.3: Steady state horizontal velocity at $\alpha = 5$, $Re = 40000$, $N = 128$, and $U_\infty = 0.4Ma$ for NACA-0012
Figure IV.4: Steady state pressure at $\alpha = 5$, $Re = 40000$, $N = 128$, and $U_\infty = 0.4Ma$ for NACA-0012
First case

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid size $N$</td>
<td>256</td>
</tr>
<tr>
<td>Angle of attack $\alpha$</td>
<td>5</td>
</tr>
<tr>
<td>Reynolds number $Re$</td>
<td>40000</td>
</tr>
<tr>
<td>Flow speed $U_\infty$</td>
<td>0.4Ma</td>
</tr>
<tr>
<td>Computed lift coefficient $c_l$</td>
<td>0.59</td>
</tr>
<tr>
<td>Computed drag coefficient $c_d$</td>
<td>0.022</td>
</tr>
<tr>
<td>Experimental lift coefficient $c_l$</td>
<td>0.53</td>
</tr>
<tr>
<td>Experimental drag coefficient $c_d$</td>
<td>0.017</td>
</tr>
</tbody>
</table>

Table IV.1: First test case: static NACA-0012 at angle of attack $\alpha = 5$ using DIRK (2,3,2) IMEX scheme

Figure IV.5: Outflow steady state horizontal velocity at $\alpha = 5$, $Re = 40000$, $N = 128$, and $U_\infty = 0.4Ma$ for NACA-0012
Second case: static NACA-0012 at angle of attack $\alpha = 16$

The second case is examined to show if the method can simulate more complicated physical phenomenon such as vortex shedding. Table IV.2 shows the flow conditions used for this case.

<table>
<thead>
<tr>
<th>Second case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid size $N$</td>
</tr>
<tr>
<td>Angle of attack $\alpha$</td>
</tr>
<tr>
<td>Reynolds number $Re$</td>
</tr>
<tr>
<td>Flow speed $U_\infty$</td>
</tr>
</tbody>
</table>

Table IV.2: Second test case: static NACA-0012 at angle of attack $\alpha = 16$ using DIRK (2,3,2) IMEX scheme

The evolution of the vorticity near the airfoil is shown in figure IV.6. At the beginning, the vortex starts around the airfoil leading edge. Then, it is spread over the top surface of the airfoil and at some point it gets detached from the airfoil. Physically, this is expected to occur periodically. Figure IV.7 confirms this phenomenon. As the vortex moves past the airfoil, its peak magnitude decrease as it diffuses in space periodically. Figure IV.8 shows the quasi periodic solution at different times as vortices keep shedding from the leading and trailing edges.
Figure IV.6: Evolution of vortex shedding at $\alpha = 16$, $Re = 40000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012
Figure IV.7: Initial top and bottom vortex shedding at \( \alpha = 16 \), \( Re = 40000 \), \( N = 256 \), and \( U_\infty = 0.4Ma \) for NACA-0012
(a) Quasi periodic solution at $t = 18.8$

(b) Quasi periodic solution at $t = 21.7$

Figure IV.8: Periodic vortex shedding at $\alpha = 16$, $Re = 40000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012
Third case: oscillating NACA-0012

This case simulates NACA-0012 airfoil having a harmonic motion. The angle of attack varies according the following periodic motion:

\[ \alpha = \alpha_0 + \alpha_1 \sin(ft), \quad (\text{IV.14}) \]

where \( f = \frac{\omega c}{2U_\infty} \) is the reduced frequency. \( \omega \) is the frequency of oscillation, and \( c \) is the chord length corresponding to the distance from the leading to the trailing edge. Table IV.3 shows the parameters used to simulate the case.

<table>
<thead>
<tr>
<th>Third case</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid size ( N )</td>
<td>256</td>
</tr>
<tr>
<td>Angle of attack ( \alpha )</td>
<td>0 to 20</td>
</tr>
<tr>
<td>Reynolds number ( Re )</td>
<td>5000</td>
</tr>
<tr>
<td>Flow speed ( U_\infty )</td>
<td>0.4Ma</td>
</tr>
<tr>
<td>Frequency ( f )</td>
<td>0.25</td>
</tr>
<tr>
<td>Center of rotation</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table IV.3: Third test case: oscillating NACA-0012 between \( \alpha = 0 \) and \( \alpha = 20 \) using DIRK (2,3,2) IMEX scheme

First, the airfoil is simulated to reach the steady state at \( \alpha = 10 \). Figures IV.10 and IV.9 show the horizontal velocity and density distribution respectively, corresponding to the initial state prior to oscillation.
Figure IV.9: Horizontal velocity prior to oscillation at $\alpha_0 = 10$, $Re = 5000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012

(a) The whole domain

(b) Zoomed
Figure IV.10: Density prior to oscillation at $\alpha_0 = 10$, $Re = 5000$, $N = 256$, and $U_\infty = 0.4 Ma$ for NACA-0012

At the beginning, the airfoil oscillates but does not reach the quasi steady state until it passes the first two cycles. Afterwards, the lift coefficient varies with the angle of attack and follows the trend shown in figure [IV.11]
Figure IV.11: Lift Coefficient vs. Angle of Attack at $\alpha = \alpha_0 + \alpha_1 \sin(ft)$, $\alpha_0 = 10$, $\alpha_1 = 10$, $Re = 5000$, $f = 0.25$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012.

Figure IV.12: Lift Coefficient vs. Angle of Attack at $\alpha = \alpha_0 + \alpha_1 \sin(ft)$, $\alpha_0 = 10$, $\alpha_1 = 10$, $Re = 5000$, $f = 0.25$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012 taken from reference [4].
By examining the above figure IV.11, it is observed that the curve trend fairly matches the result in reference [4] which is shown in figure IV.12. Also, figures IV.13 and IV.15 show density contours at different instantaneous times. They illustrate how the dynamic stall vortex evolves as the airfoil moves upward from $\alpha = 20$ to $\alpha = 0$, which is almost visually identical to figures IV.14 and IV.16 taken from reference [4]. This confirms that the developed IMEX scheme can model curvilinear gas dynamic equations with a moving mesh.
Figure IV.13: Density at $t = 33.2$, $\alpha = 18.5$ going upward, $f = 0.25$, $Re = 5000$, $N = 256$, and $U_{\infty} = 0.4Ma$ for NACA-0012
Figure IV.14: Density at $t = 33.2$, $f = 0.25$, $Re = 5000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012 taken from reference [4].
Figure IV.15: Density at $t = 35.1$, $\alpha = 12.5$ going upward, $f = 0.25$, $Re = 5000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012
Figure IV.16: Density at $t = 35.1$, $f = 0.25$, $Re = 5000$, $N = 256$, and $U_\infty = 0.4Ma$ for NACA-0012 taken from reference [4]
Chapter V

Conclusion and future work

V.1 Summary

In conclusion, the second order IMEX numerical method derived for compressible Navier-Stokes equations on an ALE moving mesh is successfully implemented. The inviscid part of the equation is calculated explicitly using second order Godunov scheme. The diffusive part is implicitly determined by formulating modified Helmholtz equation resulting in an elliptic problem. Then, multigrid method is used to solve the modified Helmholtz equation efficiently.

Different types of IMEX scheme are derived, and each type differs in terms of accuracy order and stability. The mesh motion is resolved by finding the velocity mesh nodes through solving Laplace equation. This is done using multigrid method as well. The GCL equation is implemented and solved explicitly to preserve the freestream. All of this is accomplished on a curvilinear dynamic mesh that is initially generated via elliptic grid generator, which uses multigrid algorithm. It is observed that the IMEX Runge-Kutta method converges to results that can be favorably compared to previous experiments and numerical results. Although the numerical results are satisfactory for both static and oscillating cases at different Reynolds number, they
can still be significantly improved.

V.2 Further improvement

• The mesh generated via elliptic method is not clustered around the airfoil nor it is orthogonal. Forcing terms can be added to equations III.75 and III.76 to cluster and orthogonlize the mesh.

• In deriving the numerical method, the bulk viscosity $\lambda$ is assumed to be zero in the implicit step to avoid coupling of the x and y momentum equations. The result will be more realistic if $\lambda$ is taken into consideration.

• Since the IMEX method is based on Runge-kutta, the order of accuracy can be improved via increasing the number of explicit and implicit stages which will produce more accurate results.

• Other mesh types such as O-grid or H-grid can be tested and might produce better results than the C-grid type.

V.3 Future work

Once the aforementioned issues are studied and implemented, the numerical method can be extended to three dimensions. Moreover, the acoustic noise associated with wind turbines can be investigated as the developed algorithm handles compressible flows. Finally, FSI can be considered via coupling the fluid solver to a solid solver since the numerical method can handle dynamic mesh.
REFERENCES


[8] Cebeci T., Shao J. P., Kafyeke F., and Laurendeau E. *Computational fluid dynamics for engineers*.


