DEVELOPMENT OF NUMERICAL PROGRAM ON 2 DIMENSIONAL INVISCID BURGERS EQUATION WITH FDM BASED ON MCCORMACK METHOD WITHIN A STRUCTURED GRID

MAJOR PROJECT 1
Submitted as a partial fulfillment for completing
WF5012 - Computational Fluid Dynamics 1

By:

M. Lucky Witjaksono	23623303
Pramudya Vito A.	23624020

MASTER PROGRAM IN AEROSPACE ENGINEERING
FACULTY OF MECHANICAL AND AEROSPACE ENGINEERING
INSTITUT TEKNOLOGI BANDUNG
2025

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLE

Chapter 1

Introduction

1.1 Problem Definitions

The problem given are as follows:

1. Derive the discrete equation of the 2-Dimensional Burger's equation using Mac Cormack scheme and its boundary condition,
2. Derive the grid equation based on Transfinite Interpolation (TFI) technique,
3. Create algorithms for the discrete equation of the Burger's equation and grid equation,
4. Create a program for generating uniform grids on the domain with grid size (i) N= 21 and (ii) N= 41 and the numerical solution of the discrete Burger's equation,
5. Compute the convergence solution,
6. Study effect of artificial viscosity on numerical solution,
7. Determine solution of characteristic equation $\left(\frac{\partial u}{\partial y}\right)$ and draw in flow field based on steady state analytic solution for above 2D Burger's equation $\left(\frac{\partial u}{\partial t} = 0\right)$ and boundary condition,
8. Compare the numerical solution to analytic solution
9. Create the report and presentation

1.2 Governing Equation, Domain Specification, and Boundary Conditions

In this task, the derivation of the numerical procedure is following the inviscid Burgers equation in a 2 dimensional plane. For the domain of $0 \leq x \leq 1$ and $0 \leq y < 1$ on uniform grid of $N \times N$ points, subjected to a boundary conditions of:

$$u(0,y) = 1.5u(1,y) = -0.5u(x,0) = 1.5 - 2x \tag{1.1}$$

1.3 Analytical Solution for 2D steady state Burger's Equation

The analytical solution of which are separated into two section whereas $y \leq 0.5$ and $y \geq 0.5$:

- for $y \leq 0.5$

$$u(x,y) = \begin{cases} 1.5, & \text{if } x \leq 1.5y \\ \frac{1.5-2x}{1-2y}, & \text{if } 1.5y \leq x \leq (1-0.5y) \\ -0.5, & \text{if } x \geq (1-0.5y) \end{cases} \tag{1.2}$$

- for $y \geq 0.5$

$$u(x,y) = \begin{cases} 1.5, & \text{if } x \leq (0.5+0.5y) \\ -0.5, & \text{if } x > (0.5+0.5y) \end{cases} \tag{1.3}$$

Chapter 2

Basic Theory

2.1 Grid Generation

Transfinite interpolation is a mathematical technique used to create grids by blending information from the boundaries of a computational domain. It generates interior points in a smooth and systematic way based on data specified along curves (1D), surfaces (2D), or volumes (3D). The term "transfinite" refers to the fact that the method blends contributions from finite-dimensional boundary data to define the entire domain, without requiring intermediate points or cells to be explicitly defined beforehand. This would lead to a simpler use for a structured grid generation for a semi-complex geometries and/or domains and it would produce a smooth transition as it is controlled by ξ and η (in a 2D FTI), this could generate a uniform growth throughout the domain regardless of the actual dimension used. The equation of a 2D TFI are:

$$u(\xi, \eta) = (1 - \xi)u(0, \eta) + \xi u(1, \eta) + (1 - \eta)u(\xi, 0) + \eta u(\xi, 1) - \text{corner corrections} \tag{2.1}$$

2.2 Inviscid Burger Equation

Inviscid Burgers' equation is a fundamental partial differential equation (PDE) that appears in various fields of physics and applied mathematics, especially if the studies involving a discontinuity in the primitive variables. Its simplicity makes it a powerful tool for understanding more complex nonlinear phenomena such as a shockwave formation. The equation of which are shown as:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0 \tag{2.2}$$

2.3 McCormack Method in finite difference

The idea of using a McCormack method is to achieving a higher order accuracy with the approach of a first order. In this context, the method is by using a separated central difference scheme with predictor and corrector step [1]. In order to implement the McCormack method into a inviscid burger equation, a modification is required, those modification would be:

$$\frac{\partial u}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0 \tag{2.3}$$

Where E and F are:

$$E = f(u) = \frac{u^2}{2}, F = g(u) = u \tag{2.4}$$

Such that the numerical implementations in a x-direction are represented in equation 2.5 and 2.6 respectively and later the combination of predictor and corrector are combined with equation 2.7.

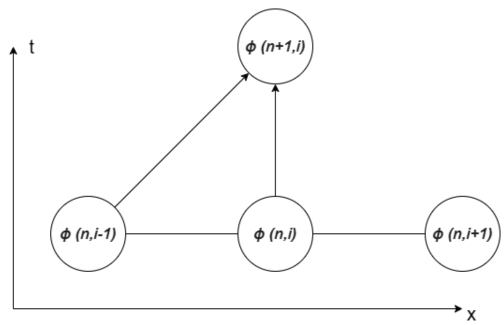
$$u_i^* = u_i^n - \frac{\Delta t}{\Delta x}\left(E_{i+1}^n - E_i^n\right) \tag{2.5}$$

$$u_i^{**} = u_i^n - \alpha\frac{\Delta t}{\Delta x}\left(E_i^* - E_{i-1}^*\right) \tag{2.6}$$

$$u_i^{n+1} = \frac{1}{2}\left[u_i^{**} + u_i^*\right] \tag{2.7}$$

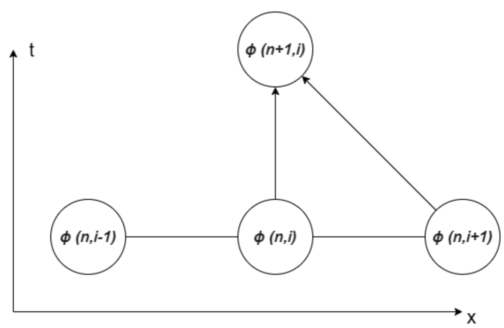
2.4 Space and Time Discretization

Discretization is the process of converting a continuous data, in this case space and time domain, into a collection of unique discrete-able data [2]. A forward time discretization is selected as it is more simple and intuitive to program. While space discretization will utilizing two method as in this study, since the McCormack method is used and the discretization would follow forward and backward space. Combining both time and space discretization would result in forward time backwards space (FTBS) or known as upwind and forward time forward space (FTFS), the illustrations and numerical implementation of which are displayed in figure 2.2 and equation 2.8 for FTBS and figure 2.2 and equation 2.9 for FTFS. Furthermore, in this study, a addition of central difference scheme will be used in the implementation of artificial velocity. In which will be discussed in chapter 2.6.



$$U_i^{n+1} = U_i^n - \alpha \frac{\Delta t}{\Delta x}\left(U_i^n - U_{i-1}^n\right) \quad (2.8)$$

Figure 2.1 FTBS



$$U_i^{n+1} = U_i^n - \alpha \frac{\Delta t}{\Delta x}\left(U_{i+1}^n - U_i^n\right) \quad (2.9)$$

Figure 2.2 FTFS

2.5 Modified McCormack Method

Considering that the given boundary condition and the requirement of using a McCormack method, a modification would be necessary. The given boundary condition of the upper domain (y=1) is unknown and would be treated as a Neumann boundary condition, this would result that the forward difference is unable to be done near the upper boundary, thus is why the modification would need to be conducted. The modification would be placed in both corrector and predictor for the y direction. In an original McCormack, the predictor would use the forward difference while the predictor would use backward difference. in this modified version, both predictor and corrector would follow a backward difference scheme. The detail of the modification would be as follows:

$$u^{*}_{(i,j)} = u^{n}_{(i,j)} - \frac{\Delta t}{\Delta x}\Big(E^{n}_{(i+1,j)} - E^{n}_{i,j}\Big) - \frac{\Delta t}{\Delta y}\Big(F^{n}_{(i,j)} - F^{n}_{i,j-1}\Big) \tag{2.10}$$

$$u^{**}_{(i,j)} = u^{n}_{(i,j)} - \frac{\Delta t}{\Delta x}\Big(E^{*}_{(i,j)} - E^{*}_{i-1,j}\Big) - \frac{\Delta t}{\Delta y}\Big(F^{*}_{(i,j)} - F^{*}_{i,j-1}\Big) \tag{2.11}$$

2.6 Artificial Viscosity

Artificial viscosity or in a more general term, numerical dissipation is a method used to denote the diffusivity behavior of a numerical solution. This is used to mitigate the different phase of a wave that could shown "wave" like wiggles before and/or after the wave and it is a type of fluid behavior called dispersion. This dispersion comes from the use of odd-order derivatives, in which are present in this studies. It should be denoted that an artificial viscosity does not represent a real viscosity and while the use of it does compromise the final solution, it will stabilize the solution. Thus, if the case is having a strong gradients such as shockwave, the solution would be attainable rather without using it [1].

$$\begin{aligned} S^{t}_{(i,j)} =& C_x \frac{|p^{t}_{(i+1,j)} - 2p^{t}_{(i,j)} + p^{t}_{(i-1,j)}|}{p^{t}_{(i+1,j)} + 2p^{t}_{(i,j)} + p^{t}_{(i-1,j)}} \left(U^{t}_{(i+1,j)} - 2U^{t}_{(i,j)} + U^{t}_{(i-1,j)}\right) \\ &+ C_y \frac{|p^{t}_{(i,j+1)} - 2p^{t}_{(i,j)} + p^{t}_{(i,j-1)}|}{p^{t}_{(i,j+1)} + 2p^{t}_{(i,j)} + p^{t}_{(i,j-1)}} \left(U^{t}_{(i,j+1)} - 2U^{t}_{(i,j)} + U^{t}_{(i,j-1)}\right) \end{aligned} \tag{2.12}$$

The numerical implementation in this case is by adding an extra fourth equivalent order in the truncation error explicitly. The left part of the equation 2.12 is what so called "shock sensor" to determine if there's a discontinuity in the region calculated. In this studies, implementation of the equation would follow with $U = E$ and $p \approx u$.

Chapter 3

Methodology

Since the task is about developing a numerical program, thus the methodology will follow the program workflow. In short, it would be explained in the flowchart in figure 3.1 and the details of which are explained in the later chapters.

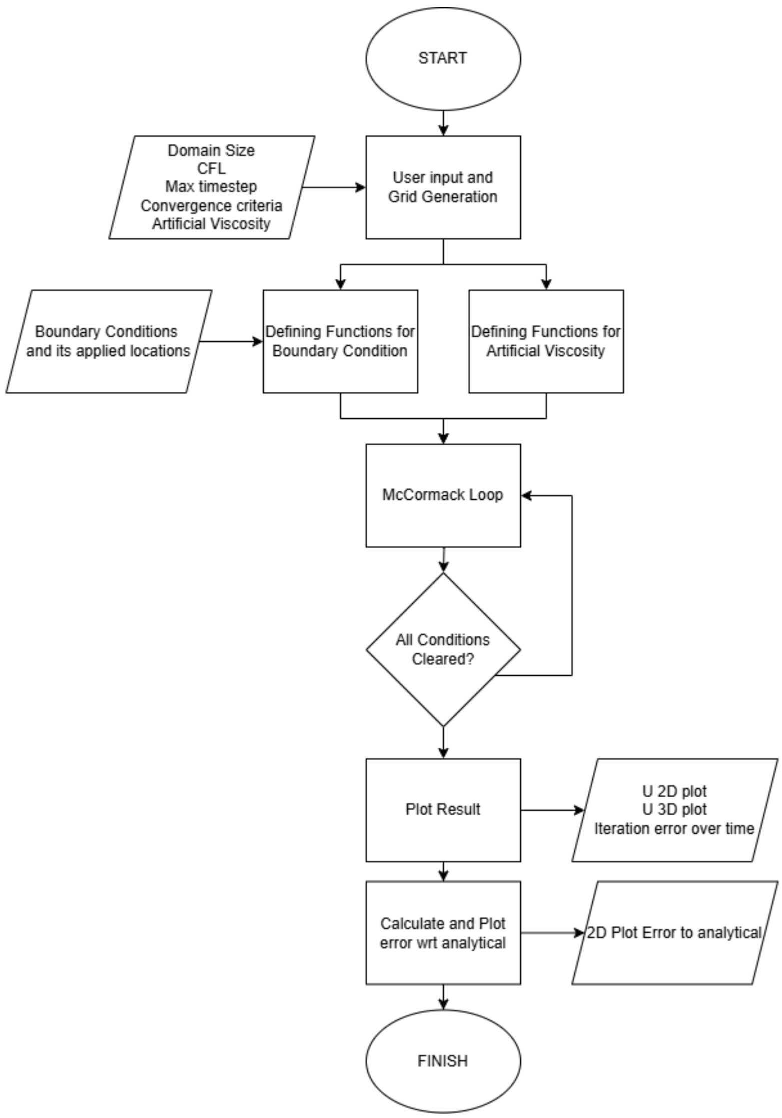


Figure 3.1 Program Flow Chart

3.1 Grid Generation

The grid generation would follow the transfinite interpolation method (TFI) as stated in chapter 1.1. The workflow of which are displayed below:

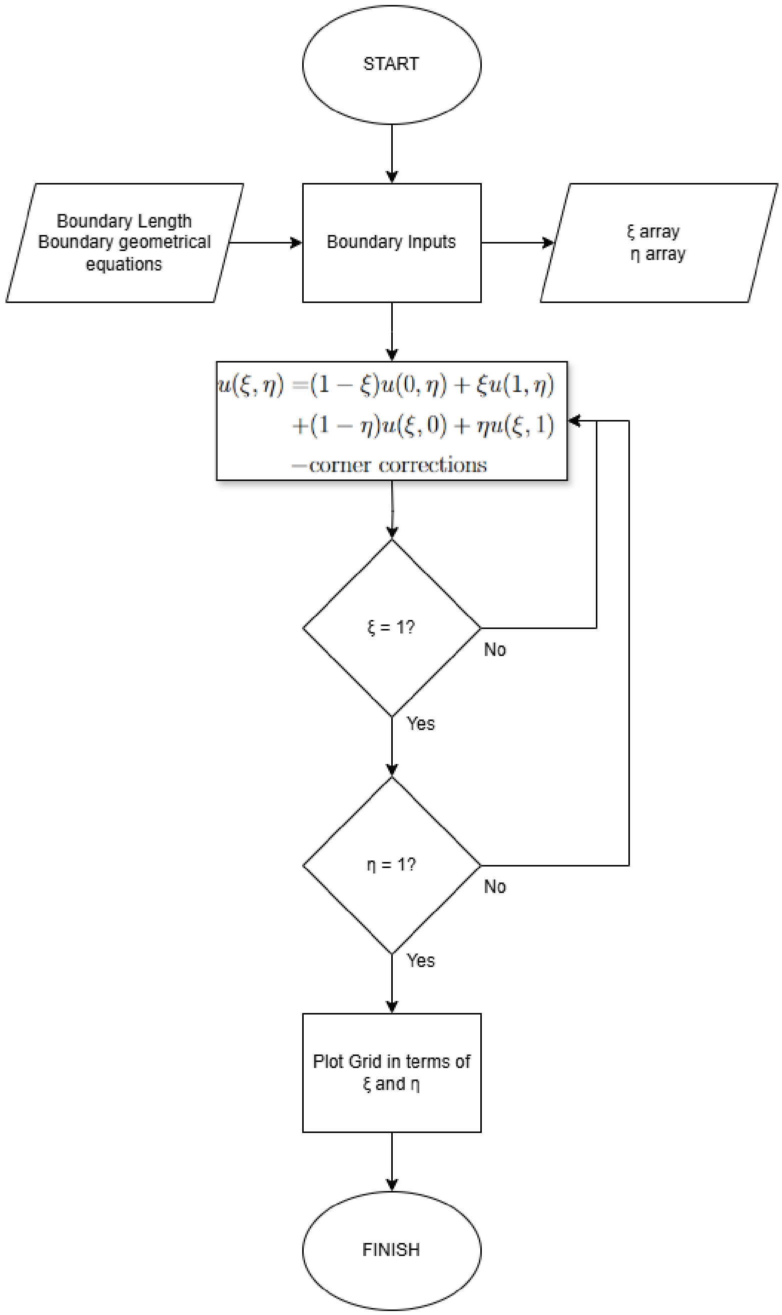


Figure 3.2 Transfinite Interpolation Flowchart

3.2 Boundary Condition Function

The purpose of this function is to replace the u data in the boundary condition and its location that was specified by the user, it would be beneficial to make a function to do this rather than implementing it in the main program due to its speed and simplicity as it is called for every predictor, corrector, and time iterations. The flowchart of which are illustrated below:

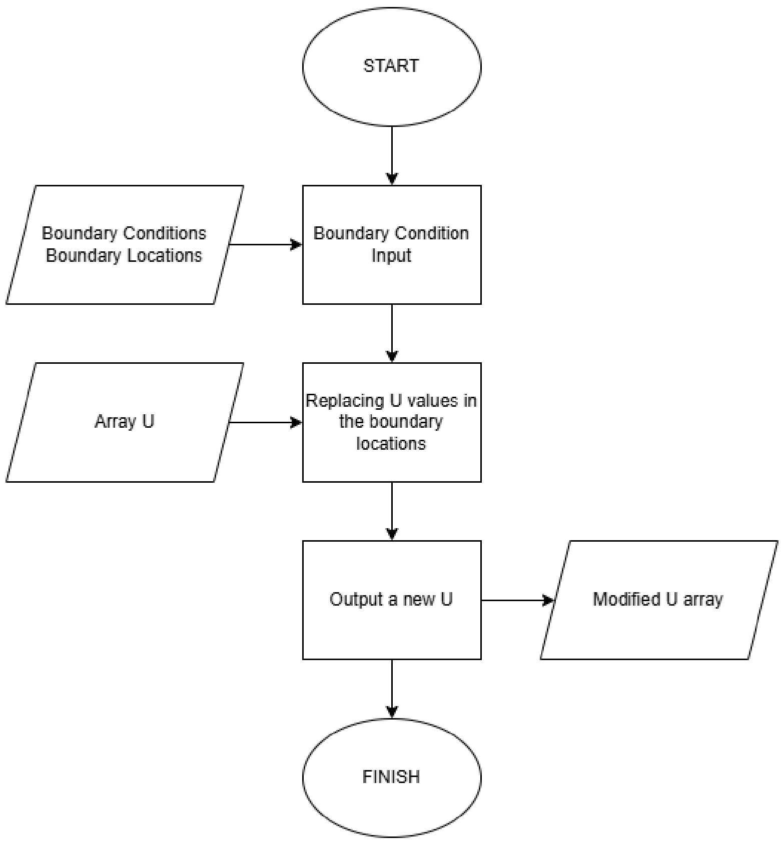


Figure 3.3 Boundary Condition Function Workflow

3.3 Artificial Viscosity Function

The purpose of this function is to calculate the term S to be added later explicitly after the corrector and predictor. As it is always called at every predictor, corrector, and time iterations, making it a function is very beneficial for time reduction. Such that, the flowchart is illustrated below:

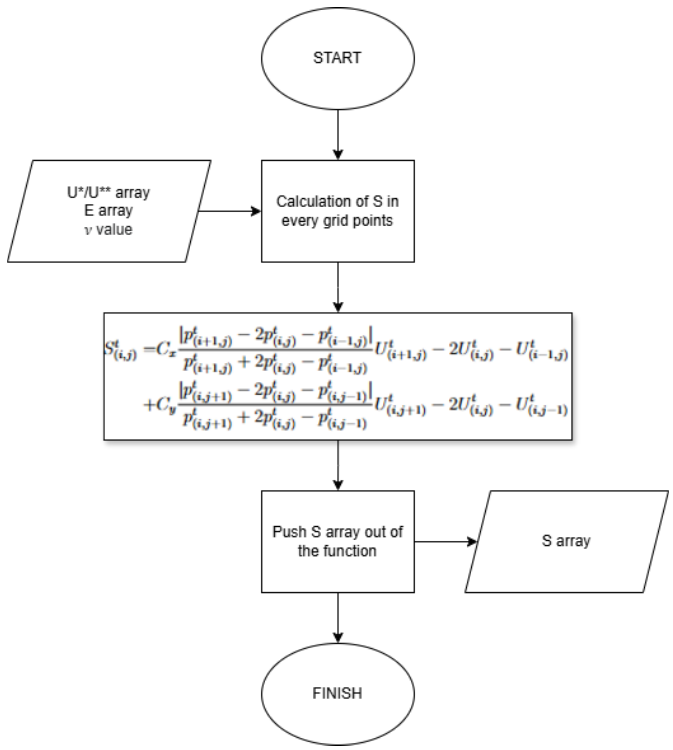


Figure 3.4 Artificial Viscosity Function Workflow

3.4 McCormack Method Loops

As explained in chapter 2.3 and further modified and clarified in chapter 2.5. The flowchart of the McCormack method loops would be as follows:

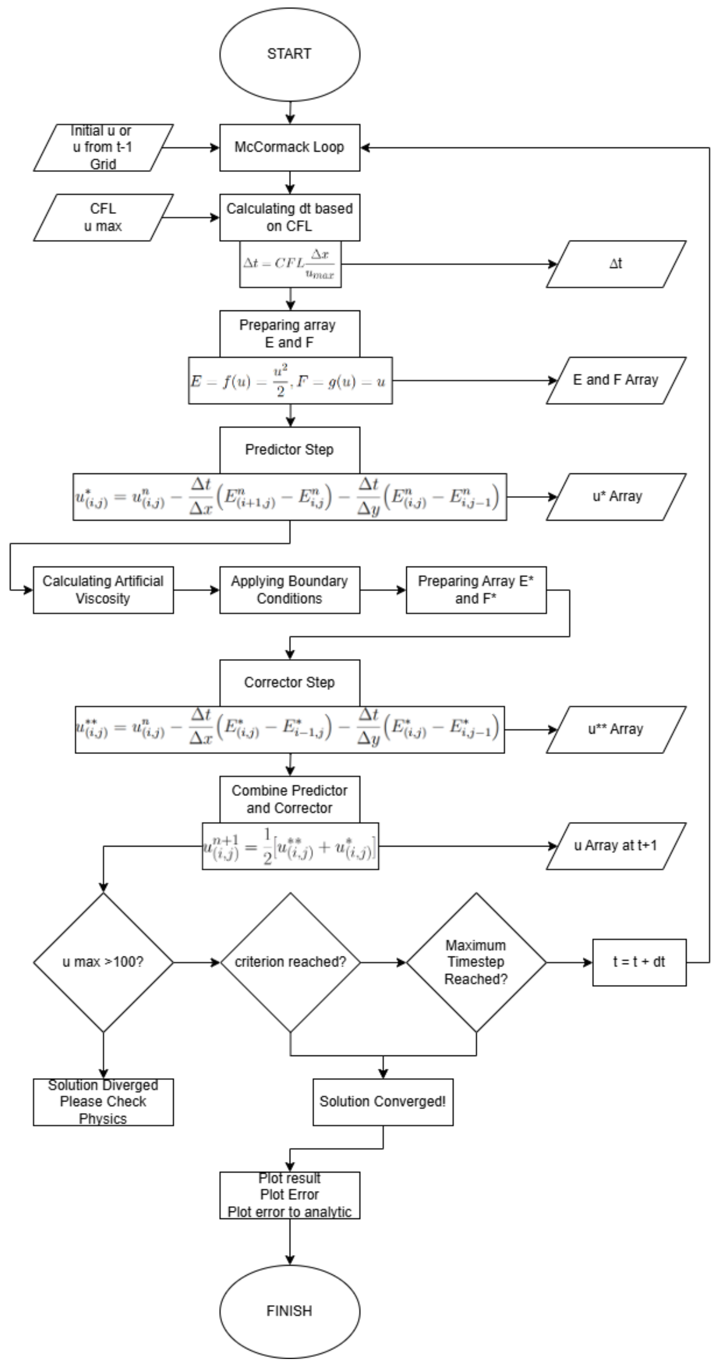


Figure 3.5 McCormack Method Loops with conditional checking

Chapter 4

Results and Discussion

4.1 Grid Generation

Since the domain specified is inherently a square shape, the TFI would not yield a significant differences compared to a uniformly distributed grid. Therefore, the result would be displayed in figure 4.1 for 21 by 21 grid and figure 4.2 for 41 by 41 grid.

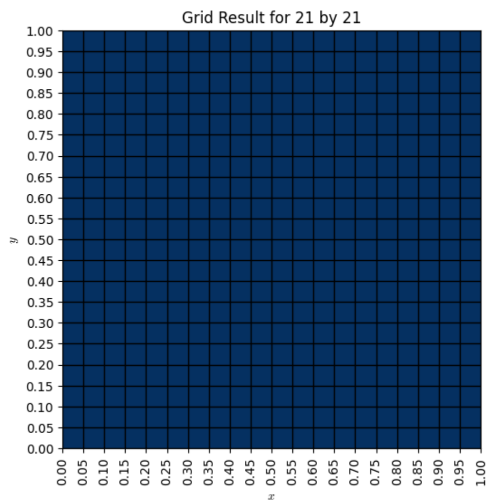


Figure 4.1 Domain with 21 by 21 grid

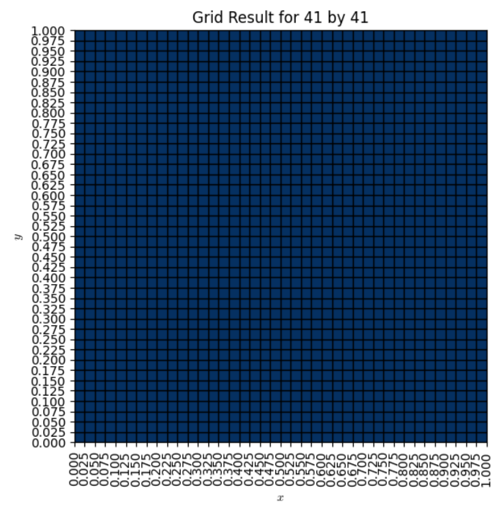


Figure 4.2 Domain with 41 by 41 grid

4.2 Analytical Result

The analytical result are performed from the equations provided and the result can be seen in figure 4.3 for 21 by 21 grid and figure 4.4 for 41 by 41 grid in 2D form, with addition of 3D plot in figure 4.5 and 4.6 respectively.

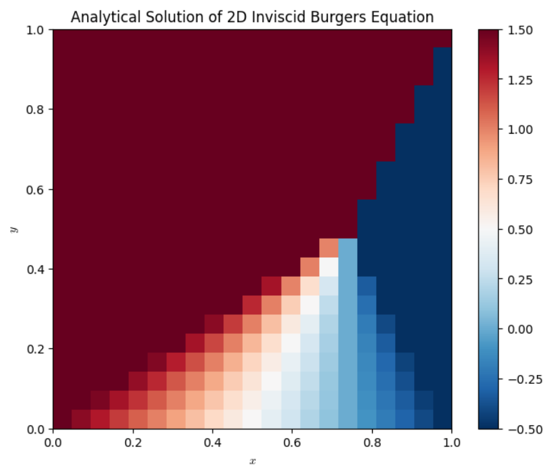


Figure 4.3 Analytical result for 21 by 21 grid

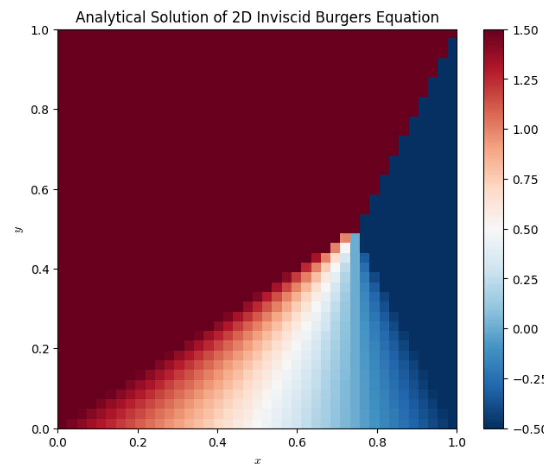


Figure 4.4 Analytical result for 41 by 41 grid

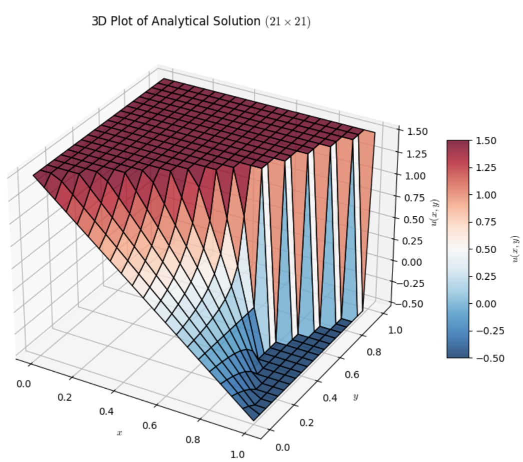


Figure 4.5 3D plot of analytical result for 21 by 21 grid

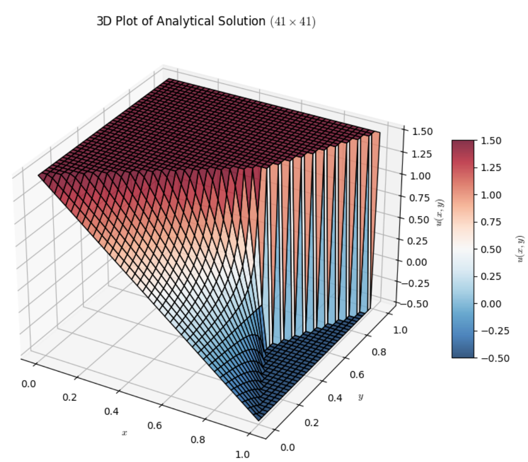


Figure 4.6 3D plot of analytical result for 41 by 41 grid

The provided figures above represent the **analytical solution of the 2D inviscid Burgers' equation**. The plots illustrate the variation of the solution across the spatial domain (x, y), with both x and y ranging from 0 to 1. The colormap encodes the values of the solution $u(x, y)$, as indicated by the color bar on the right. The values range from -0.5 (dark blue) to 1.5 (dark red), with intermediate gradients represented by white and light blue shades. The sudden transition from red to blue visualized a non-linear change in the solution, which is the characteristic of the Burgers' equation. This region can represent a shock or discontinuity zone. The plot will give more accurate solution when the grid size is increased.

4.3 Solution of Characteristic Equation $\partial x/\partial y$

For the steady state 2D Burger's equation, while $\frac{\partial u}{\partial t} = 0$, the equation becomes:

$$\begin{aligned} u\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} &= 0 \\ \frac{\partial u}{\partial y} &= -u\frac{\partial u}{\partial x} \\ \frac{\partial x}{\partial u}\frac{\partial u}{\partial y} &= -u\frac{\partial u}{\partial x}\frac{\partial x}{\partial u} \\ \frac{\partial x}{\partial y} &= -u \end{aligned} \tag{4.1}$$

With the information of the analytical solution of $u(x, y)$ for steady state 2D Burger's equation, the solution of characteristic equation $\frac{\partial x}{\partial y}$ can be plotted as shwon in Figure.

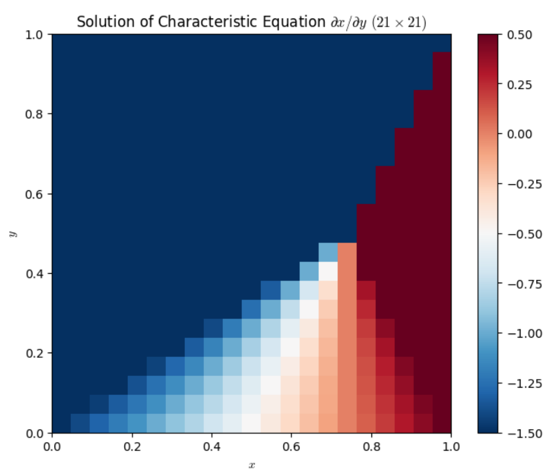


Figure 4.7 Solution of characteristic equation for 21 by 21 grid

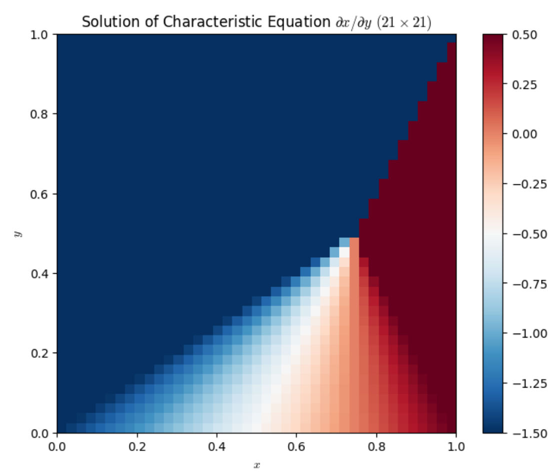


Figure 4.8 Solution of characteristic equation for 41 by 41 grid

4.4 Numerical Result

4.4.1 21 by 21 Grid

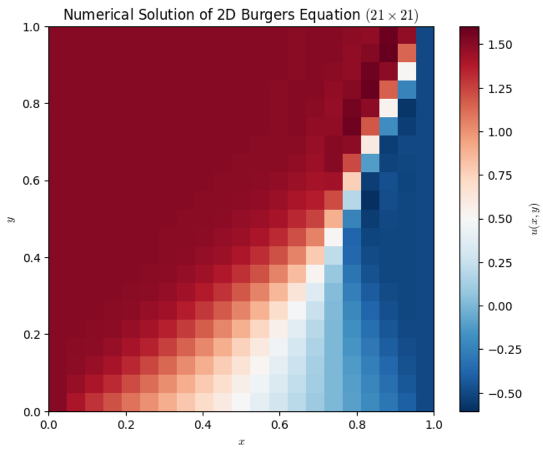


Figure 4.9 2D contour plot of numerical solution for 21 by 21 grid

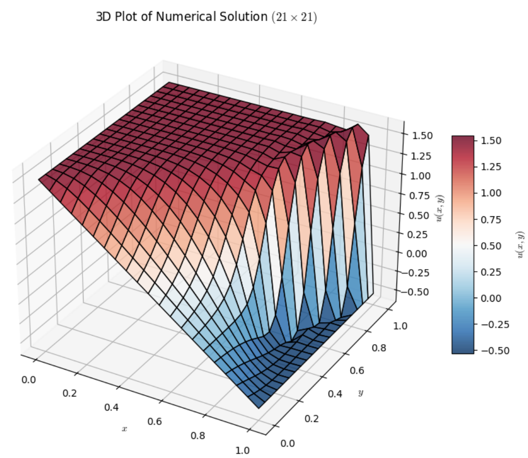


Figure 4.10 3D plot of numerical solution for 21 by 21 grid

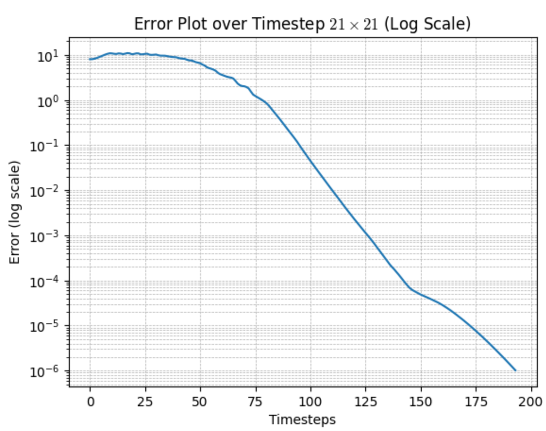


Figure 4.11 Error plot of numerical solution for 21 by 21 grid

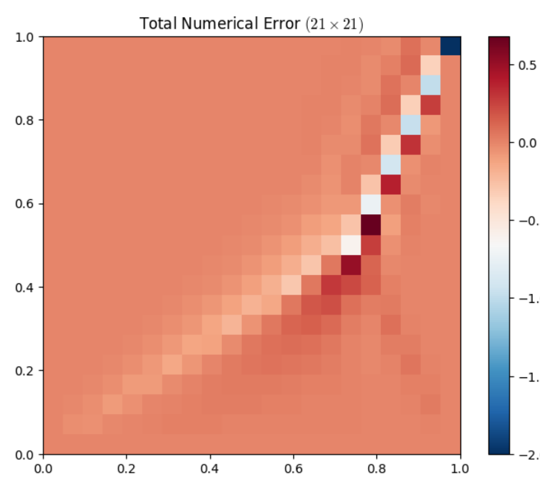


Figure 4.12 Total numerical error for 21 by 21 grid

For the 21 by 21 grid without artificial viscosity, the simulation was converged in 194 iterations. The plot shown in Figure 4.9 to 4.11 can capture the overall behaviour of the numerical solution. A notable observation is the presence of a wiggle along the discontinuity line, particularly in the central transition region. Additionally, since the grid is relatively coarse (21×21), the plot shows a blocky or pixelated appearance. This is because a small number of grid points results in lower resolution. As a result, sharp details or smooth transitions are not well-resolved, and individual grid cells become visible.

4.4.2 41 by 41 Grid

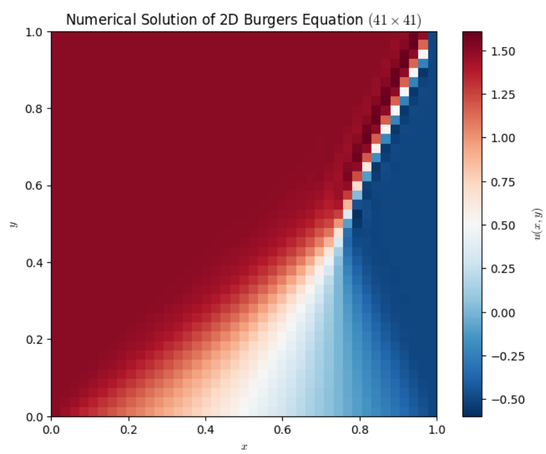


Figure 4.13 2D contour plot of numerical solution for 41 by 41 grid

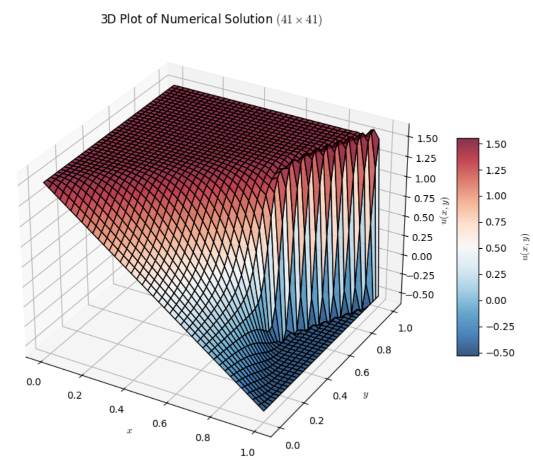


Figure 4.14 3D plot of numerical solution for 41 by 41 grid

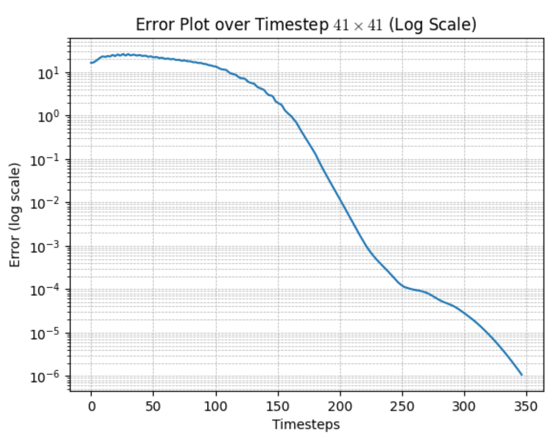


Figure 4.15 Error plot of numerical solution for 41 by 41 grid

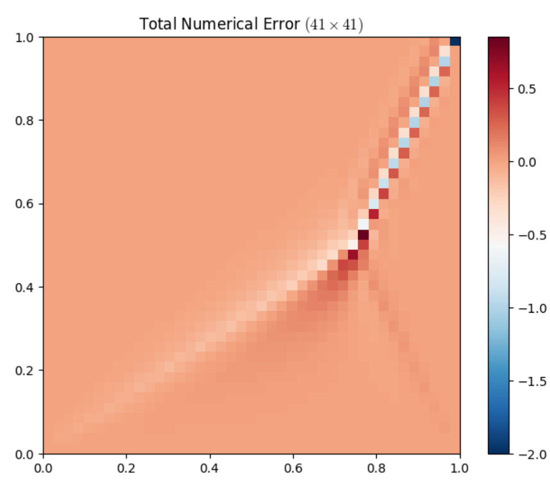


Figure 4.16 Total numerical error for 41 by 41 grid

For the 41 by 41 grid without artificial viscosity, the simulation was converged in 347 iterations, which is almost 2 times of 21 by 21 grid. As shown in Figures 4.13 to 4.16, the plots provide higher resolution compared to the previous case. A wiggle is also observed around the discontinuity, marked by dark red and dark blue colors. This can be seen clearly from the 3D plot in Figure 4.14.

4.5 Artificial Viscosity

The use of artificial viscosity is theoretically will reduce the oscillation during iteration, thus reduce the wiggle effect. It will be applied in both cases, for $N = 21$ and $N = 41$. The used artificial viscosity ν is varied: 0.01, 0.05, and 0.125. Those values were chosen because the lower value will give small effect, while the higher makes the simulation divergent. For a more drastic comparison, the grid $N = 21$ are used and the result in 2D plot can be seen in figure 4.17 until 4.19.

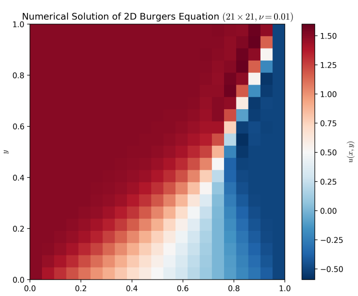


Figure 4.17 2D contour plot of numerical solution with $\nu = 0.01$

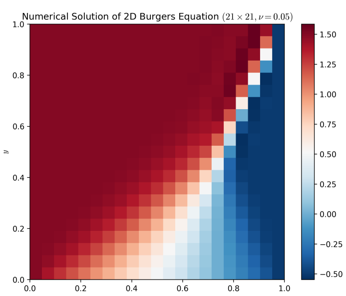


Figure 4.18 2D contour plot of numerical solution with $\nu = 0.05$

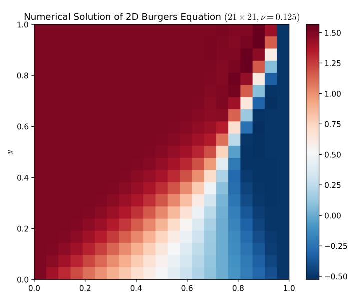


Figure 4.19 2D contour plot of numerical solution with $\nu = 0.125$

From the 2D contour plot, the result can seems to be the same, thus the plotting in 3D is performed in figure 4.20 until 4.22.

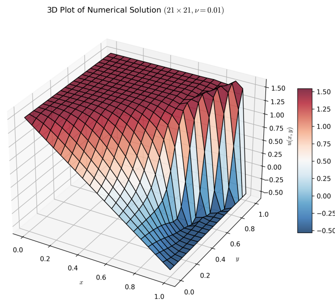


Figure 4.20 3D contour plot of numerical solution with $\nu = 0.01$

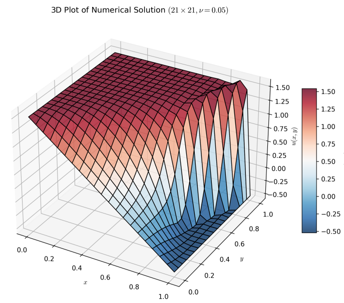


Figure 4.21 3D contour plot of numerical solution with $\nu = 0.05$

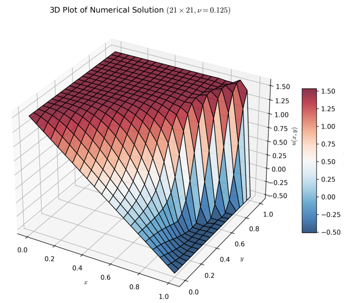


Figure 4.22 3D contour plot of numerical solution with $\nu = 0.125$

Again from the 3D plot alone, the results seems to remain unchanged by the artificial viscosity.

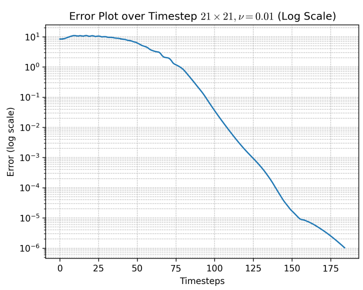


Figure 4.23 Numerical errors w.r.t iterations with $\nu = 0.01$

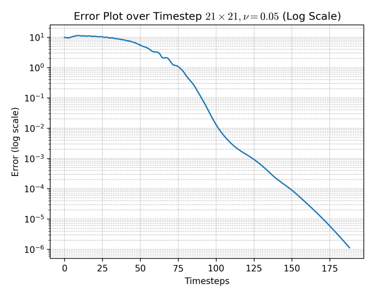


Figure 4.24 Numerical errors w.r.t iterations with $\nu = 0.05$

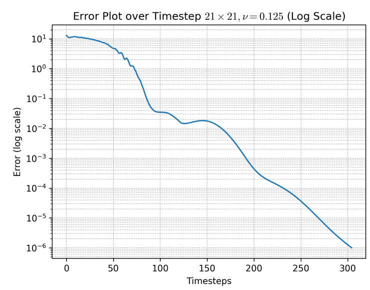


Figure 4.25 Numerical errors w.r.t iterations with $\nu = 0.125$

If we compare it with error to iteration wise in figure 4.23 until 4.25, the numerical result with viscosity $\nu = 0.125$ has a higher amount of iterations performed to reach the same convergence criteria, this indicates that the solution is "taking more time" to converge and to a certain extent more stable. With this, the error with respect to analytical solution is performed in figure 4.26 until 4.28.

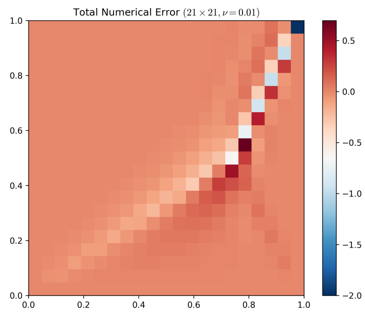


Figure 4.26 2D contour plot of error wrt analytical with $\nu = 0.01$

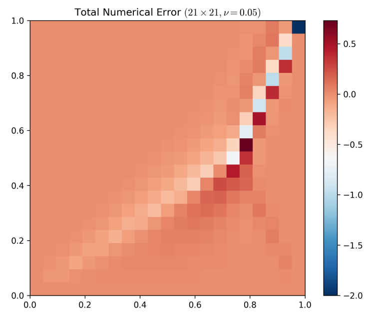


Figure 4.27 2D contour plot of error wrt analytical with $\nu = 0.05$

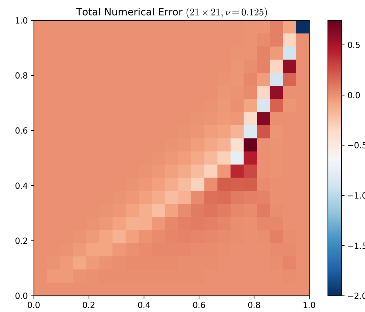


Figure 4.28 2D contour plot of error wrt analytical with $\nu = 0.125$

In this example, again, not much can be seen by the plot, thus the sum the absolute of those values are displayed in table 4.1. Here we can see that it is correct that artificial viscosity indeed add some more errors as explained in chapter 2.6, the higher the ν value is, the higher the error is.

Table 4.1 Values of total absolute error for different grids and artificial viscosity ν

Grid	$\nu = 0$	$\nu = 0.01$	$\nu = 0.05$	$\nu = 0.125$
21 × 21	19.84826	19.89419	20.31916	21.33440
41 × 41	42.22017	42.30640	42.86701	43.54519

Chapter 5

Conclusion

The conclusion of this studies are as follows:

1. The derivation of 2-dimensional Burger's equation using McCormack has been performed with its appropriate scheme,
2. The grid is generated by using a transfinite interpolation method as stated in chapter 1.1,
3. The numerical and analytic solutions are calculated based on the grid of $N = 21$ and $N = 41$, with satisfying result in numerical convergence wise.
4. The solution of characteristic equation are also performed and approached by mathematical derivation,
5. The effects on artificial viscosity, while the solution is affected by it. It is only affecting it in adding additional errors and iterations, while having a little effects on solution stability. Further studies would need to be required to have a great dissipation terms in a burger's equation.

REFERENCE

[1] J. D. Anderson, *Computational Fluid Dynamics: The Basics with Applications.* McGraw-Hill Book Company Europe, 1995.

[2] C. Hirsch, "Chapter 4 - the finite difference method for structured grids," in *Numerical Computation of Internal and External Flows (Second Edition)* (C. Hirsch, ed.), pp. 145–201, Oxford: Butterworth-Heinemann, second edition ed., 2007.

[3] M. A. Moelyadi, "Lecture note," 2020.

APPENDIX A

Below is the link for the Python programming language script used in this project:
https://colab.research.google.com/MajorTask1CFD1