Simulation and Visualization of Elementary and Potential Flows

The City College of New York

Dept. of Mechanical Engineering

ME 57200: Aerodynamic Design

Name: Jeremy Maniago

Instructor: Yang Liu

Due Date: 05/22/2024, 11:59pm

Abstract

This project presents a user-friendly MATLAB code able to simulate basic potential flows in 2D. If it can be simulated, it will allow better intuitive connections between the equations and the visualization. After the basic potential flows can be displayed in MATLAB, multiple flows will be able to be visualized, if the code allows the potential flow functions to satisfy Laplace's Equation, which should make the simulation of multiple flows much easier. Implementing a GUI will make interactivity much easier, along with the use of animations to better visualize the flow direction and behavior.

Nomenclature

GUI	Graphical User Interface
Inf	Infinity
NaN	Not a Number
$\boldsymbol{\Gamma}$	Gamma, Circulation
$\boldsymbol{V_\infty}$	Free Stream Velocity
$\boldsymbol{\Lambda}$	Lamba, Strength
$\boldsymbol{\phi}$	Phi, Potential Function
$\boldsymbol{\psi}$	Psi, Stream Function
$\boldsymbol{\theta}$	Theta, Free Stream Velocity angle

Introduction

The potential flows that were covered in ME 57200: Aerodynamic Design set the foundation of most analytical flow analyses. From basic flows such as uniform, source, sink, and vortex flows, we can obtain stream functions and streamlines, which define the flow path of particles affected by such flows. Such derivations can lead to more complex flow patterns.

Firstly, let us examine some the elementary flows. Uniform flow is defined by velocity at a given angle from the horizontal. This can be thought of as V_∞, the free stream velocity. The streamlines are simply straight lines, with the direction defined by the angle. The cartesian velocities are shown in **Equation 1**.

$$u = V_\infty \, cos(\theta) \qquad (1)$$

$$v = V_\infty \, sin(\theta)$$

Source flow is defined by Λ, the strength of the point source. The cartesian velocity components are shown in **Equation 2**. Sink flow utilizes the same velocity equations but is defined with a negative strength.

$$u = \frac{\Lambda}{2\pi}\frac{x}{x^2+y^2}$$
$$v = \frac{\Lambda}{2\pi}\frac{y}{x^2+y^2} \qquad (2)$$

In the case of streamlines, they originate from the source outwards. The opposite occurs for sink flow, thus the negative strength.

Vortex flow is defined by Γ, the circulation. **Equation 3** shows the cartesian velocity components for vortex flow, which look similar to that of source/sink flow.

A negative circulation indicates counterclockwise flow, while a positive circulation indicates a clockwise flow.

$$u = \frac{\Gamma}{2\pi}\frac{y}{x^2+y^2}$$
$$v = \frac{\Gamma}{2\pi}\frac{x}{x^2+y^2} \tag{3}$$

With these components established, we can combine multiple flows. Usually, these flows are defined by ψ and ϕ, the stream function and potential function, respectively. However, through the *Laplace Equation*, shown in **Equation 4**, the combination of velocity components is allowable, as long as the flow is steady-state, incompressible, and irrotational.

$$\nabla^2\phi = 0 \tag{4}$$

The *Laplace Equation* can be proved to work with the stream function as well. Additionally, the *Cauchy-Riemann Equations* allow us to relate the stream function and potential function with the cartesian velocity components. These are shown in **Equation 5**.

$$u = \frac{\partial\phi}{\partial x}, \qquad u = \frac{\partial\psi}{\partial y}$$
$$v = \frac{\partial\phi}{\partial y}, \qquad v = -\frac{\partial\psi}{\partial x} \tag{5}$$

By taking the partial derivates with respect to x or y of the stream or potential functions of the elemental flows, we can obtain the cartesian velocity components shown in **Equations 1, 2, & 3**. Because the velocity components are related through partial derivatives, the *Laplace Equation* should still be applicable when combining the velocities of multiple flows. This will be the main procedure of creating more complicated flows through the velocity components.

Methodology

To visualize elementary and potential flows, MATLAB will be utilized. It is a program capable of doing the basic arithmetic and plotting needed for this project. The structure of the entire program will be laid out as such:

- Main File
  - Uniform Flow
  - Source/Sink Flow
  - Vortex Flow
  - Animation File

The main file will contain all of the parameters and will call the elementary flow functions listed above. **Figure** 1 shows a basic outline/workflow for this program workspace.

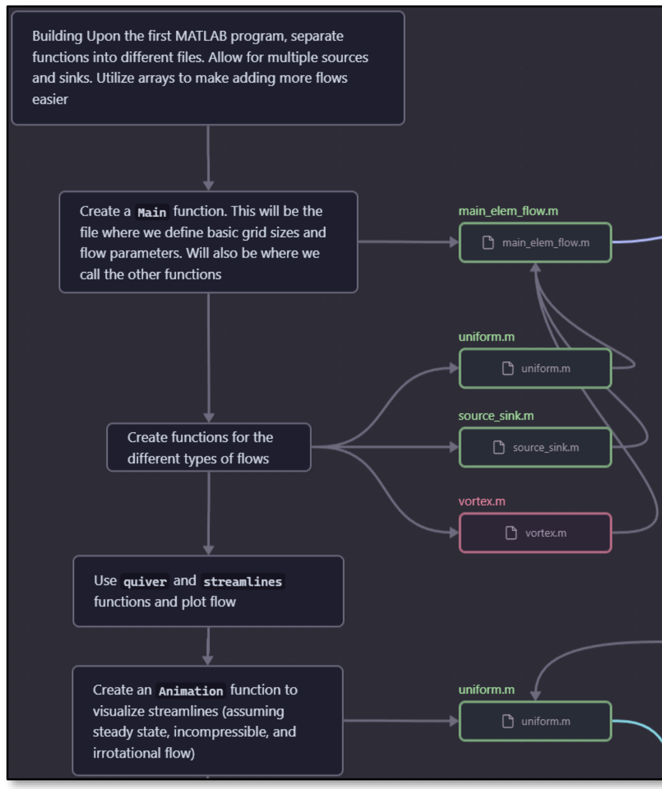


Figure 1: Flowchart of program

For this to work, a 'mesh grid' will be created to initialize the position vectors of each point within our computational domain. E.g., our domain will be from -5 to 5 in the x-axis and -5 to 5 in the y-axis, with 0.25 spacing in between. This results in a 41x41 matrix of the position vectors of our domain. Defining V_∞ will allow us to call the Uniform Flow function and obtain the u and v velocities. As for the Source/Sink flow function, the X and Y position matrix previously mentioned need to be provided, as well as the x and y position of the source/sink flow and strength Λ. The reason for providing the domain position vectors is because we need to calculate the u and v velocities for every point in the domain. This can be achieved by finding the difference between the x position of the source/sink for and the X position of all the position vectors in the domain. Additionally, this is done for the y position. **Equation 6** shows the u and v velocity vectors used within the source/sink MATLAB function.

$$u = \frac{\Lambda}{2\pi} \frac{(x - x_{source})}{(x - x_{source})^2 + (y - y_{source})^2}$$
$$v = \frac{\Lambda}{2\pi} \frac{(y - y_{source})}{(x - x_{source})^2 + (y - y_{source})^2} \tag{6}$$

Similarly, for vortex flow, the difference between the domain position vectors and the position of the vortex flow is need for the MATLAB function. These are shown in **Equation 7**.

$$u = \frac{\Gamma}{2\pi} \frac{(y - y_{source})}{(x - x_{source})^2 + (y - y_{source})^2}$$
$$v = \frac{\Gamma}{2\pi} \frac{(x - x_{source})}{(x - x_{source})^2 + (y - y_{source})^2} \tag{7}$$

Finally, after solving for all of the velocity components, we can sum them up to get the resultant velocity vectors.

$$u = \Sigma\, u = u_{uniform} + u_{source} + u_{sink} + u_{vortex}$$
$$v = \Sigma\, v = v_{uniform} + v_{source} + v_{sink} + v_{vortex} \tag{7}$$

All that is left is to plot the position of the source/sink and vortex flows (whichever is defined and how many). The MATLAB function 'quiver' will be utilized to plot the velocity vectors for each point in our domain. The MATLAB function 'streamline' will plot streamlines given a grid/domain, velocity vectors, and a starting 'seed' where the streamlines will start.

Results

Appendix A shows the results of the code. The quiver velocity vectors are the red arrows. They are scaled based on the sizes of the other vectors in such a way that the arrows will not overlap. The blue lines are the streamlines and are tangent to these velocity vectors. The dots represent either source, sink, or vortex flow – these can be intuitive to tell when looking at the streamlines and velocity vectors. All plots were plotted with a uniform flow stream. Additionally, to properly plot the velocity vectors, all values much be real. Some turned out to be NaN or Inf, which was filtered out and changed into real values. NaN values were replaced with 0 and Inf values were replaced with the maximum real value within the velocity vector domain matrix.

Discussion

This program proved to show success, especially when combining multiple flows. An addition of an animation was also helpful in visualization. However, the code itself is still in its working stage and thus not user-friendly. To fix this, adapting the code with a GUI in another programming language or using MATLAB Apps may provide easier usage. Future work includes adding doublet flow, method of imaging (wall and corner flow), plotting stagnation points and streamlines, and eventually moving into thin airfoil theory and attempting to extract actual values from the calculated velocity vectors.

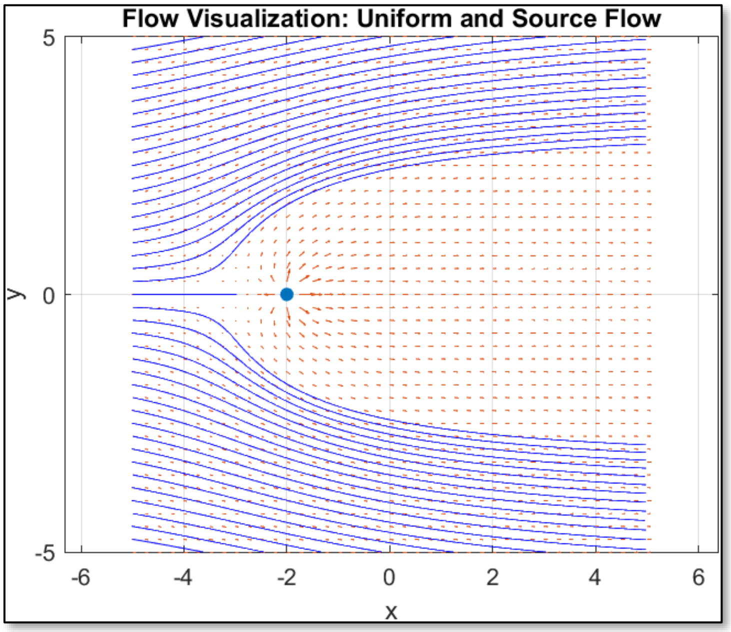
Conclusions

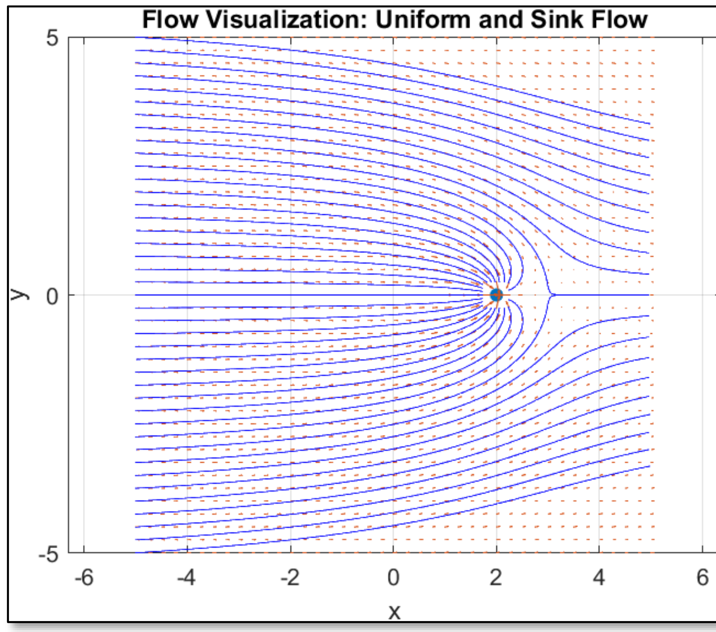
Elementary and Potential Flows do not get enough attention when it comes to simulation and visualization. They are the fundamental building blocks to more complicated Aerodynamics theories, and thus this project is a step bridging this connection between the simple to the complex.

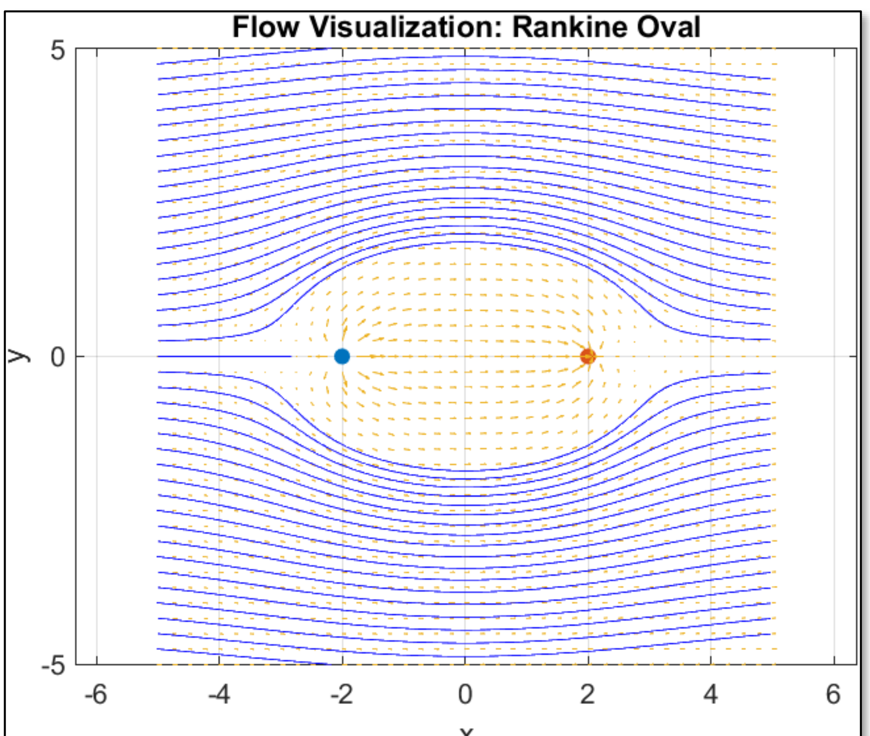
References

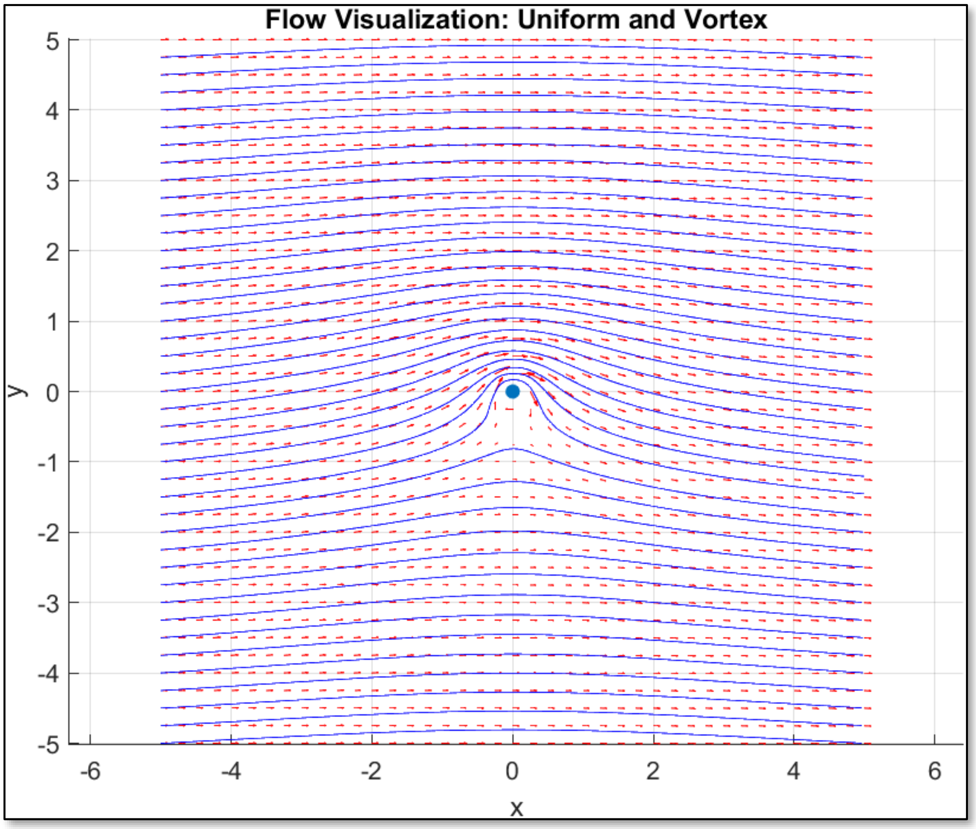
1. Professor Yang Liu's ME 572 Lecture Notes
2. https://potentialflow.com/flow-elements
3. https://web.mit.edu/16.unified/www/FALL/fluids/Lectures/f15.pdf
4. https://web.mit.edu/16.unified/www/FALL/fluids/Lectures/f16.pdf

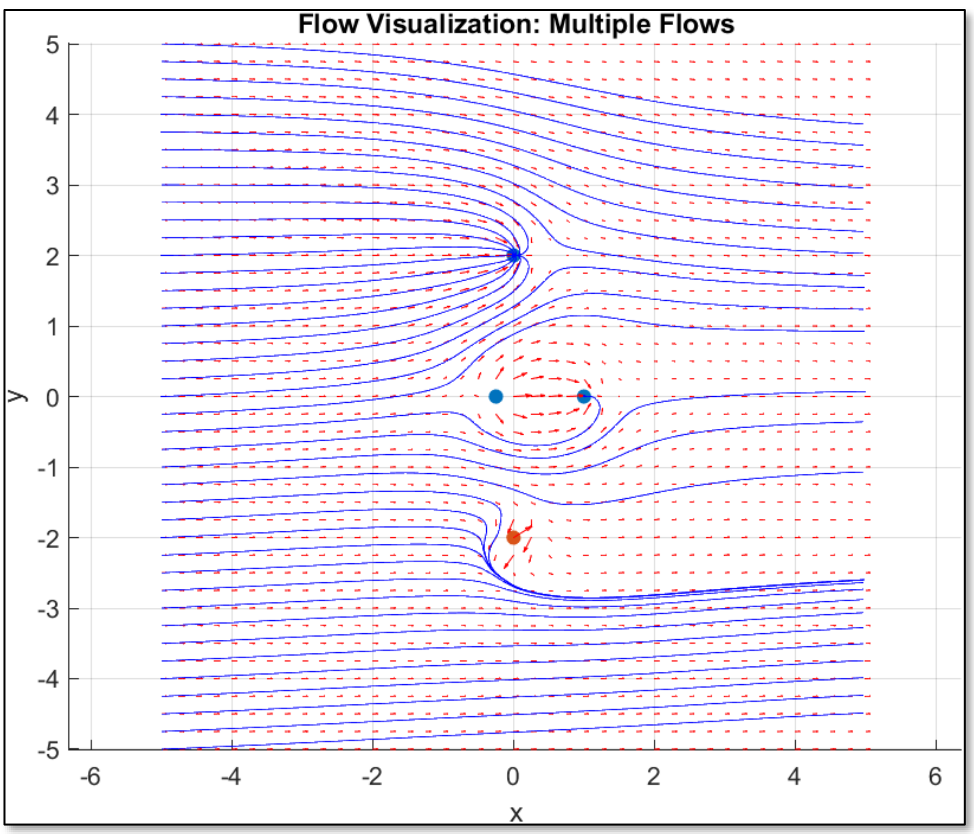
Appendix A








Flow Visualization: Uniform and Vortex
y
x
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6


Flow Visualization: Multiple Flows
y
x
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6