A Mathematical Investigation and Modelling of the Joukowski Transformation

Charles Turrell

May 4, 2022

Abstract

In this project I will be investigating the mathematics used in the early days of Aerodynamics. Modern methods involve the use of software and supercomputers to model air flow around a wing, but over 100 years ago mathematicians were attempting to model it with mathematical theory. This is what I will be investigating throughout this project. I will introduce some aspects of fluid dynamics and complex variable theory, which are the foundation for this theory. Using some basic, but important, flows I will build towards modelling flow around a cylinder. Once we have this solution, then a special type of conformal mapping called the Joukowski Transformation will be used in order to create an aerofoil shape. We will see different flows around an aerofoil and see what it is that causes lift. Along side the theoretical mathematics, I have used Python to create illustrations of the several types of flow and modelled the Joukowski Transformation. The project will then finish with a brief section on some lift and pressure calculations; outlining some of the theory involved.

Please note that, for the electronic reader, all referenced equations, figures, and sections are hyperlinked and clickable numbers are the off-colour. So the reader can easily navigate the document by clicking on the number. As well as being able to click on any section in the contents to navigate to particular points.

Supervised By
Dr Lloyd Bridge

Word Count: 10493

Page Count [including appendix]: 47 [68]

Contents

1 Introduction **3**
- 1.1 Aims and Objectives . . . 3
- 1.2 The Forces of Flight . . . 3
- 1.3 History of Aerodynamics . . . 4

2 Fluid Dynamics **4**
- 2.1 Vector Calculus . . . 5
- 2.2 Basic Assumptions . . . 6
- 2.3 Potential Flow Theory . . . 7

3 Complex Variable Theory **8**
- 3.1 Cauchy-Riemann Equations . . . 9
- 3.2 Harmonic Conjugate Functions . . . 9
- 3.3 Complex Potential . . . 10

4 Simple Flows **12**
- 4.1 Uniform Flows . . . 13
- 4.2 Source and Sink . . . 16
- 4.3 Line Vortex . . . 18
- 4.4 Superposition of Flows . . . 19
  - 4.4.1 Dipole . . . 20
  - 4.4.2 Superimposed Uniform Flow with Dipole . . . 22

5 Flow Around Cylinders **23**
- 5.1 Uniform Flow Past a Stationary Cylinder . . . 24
- 5.2 Uniform Flow Past a Cylinder with Circulation . . . 26
- 5.3 Stagnation Points . . . 28

6 Flow Around Joukowski Aerofoil **32**
- 6.1 Conformal Mapping . . . 33
- 6.2 Joukowski Transformation . . . 33
- 6.3 Flow Past Aerofoils . . . 37
  - 6.3.1 Flow Past an Aerofoil . . . 37
  - 6.3.2 Flow Around Aerofoil with Angle of Attack . . . 39

7 A Brief Look at Lift **41**
- 7.1 Kutta-Joukowski Lift Theorem . . . 41
- 7.2 Pressure Coefficient . . . 43

8 Discussion **45**

A Appendix **47**
- A.1 Flow Code . . . 47
- A.2 Joukowski Transformation Code . . . 57
- A.3 Pressure Code . . . 65

1 Introduction

Before we commence on the mathematics behind this theory, we will understand the aims and objectives for this project, as well as covering the scope of this project. Followed by some of the absolute basics of theory of flight and finishing with some key events in the history of aerodynamics.

1.1 Aims and Objectives

The Joukowski Transformation was used in the early days of aeroplanes, over 100 years ago. It was used as a mathematical method to model air flow around an aerofoil. The method involves having the known solution to flow past a cylinder then transforming the cylinder into an aerofoil using a conformal mapping. Unfortunately, it does not have much use in the modern world but it is an incredibly fascinating topic.

By the end of the project we aim to have sufficient information of the fundamental mathematics from fluid dynamics and complex variable theory. We will cover major topics including potential flow theory, harmonic functions, and the complex potential, amongst others. Followed by an outline of some of the underlying mathematics that will be needed in understanding these concepts. Along with some basic assumptions that will be used throughout this project.

The main part of the project will include hand calculations on a variety of simple flows, such as uniform flow, a source and sink, and a line vortex. Which will lead onto different types of flow around cylinders. Finishing with an introduction of conformal mappings and utilising the Joukowski Transformation, which are fascinating topics and provide a way to solve a complex mathematical problems.

Along side the mathematics there will be illustrations to supplement the project that will give a visual experience to some of the complex mathematics. The python package `matplotlib` will mostly be used to help create the illustrations. The images will consist of all the calculated simple flows, all the different types of flows around cylinders, and will finish with images of the Joukowski Transformation in action.

The main objective for this project is to model different flows around a cylinder, this will include uniform flow, flow with circulation, and flow at an angle of attack. Once these are completed we will see these flows mapped onto another plane using the Joukowski Transformation. And, as just mentioned, all these aspects will be supplemented with python illustrations. Finishing the project with some discussion about the work completed, different options for further research, and my thoughts throughout this long research journey.

It is not within the scope of this project but there is a brief section on lift and pressure at the end. Since this is the real reason for why aeroplanes fly, and it would be the next logical step of research. This section is not extensive but will cover some of the aspects that have appeared during the research for this project.

1.2 The Forces of Flight

There are four basic forces acting on aeroplane during take-off, landing and in-flight. These are lift, weight, thrust, and drag.

Lift enables the object to move upwards, relative to the ground. Which we will be exploring towards the end of this project. Weight is the mass of an object multiplied by gravity, which is 9.81 ms^{-1}, on Earth. Weight opposes lift and wants the object on the ground. If an object creates a lift force greater than the weight force, then the object will overcome gravity and rise.

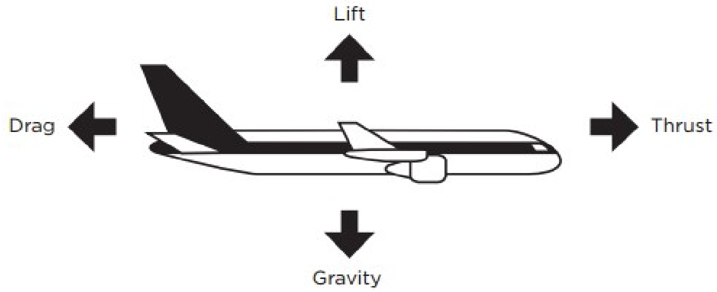


Figure 1.1: The four basic forces of flight.
Image obtained from: https://www.scienceworld.ca/resource/four-forces-flight/

Thrust is usually performed by the planes engines or propellers, propelling the object forwards. Vehicles such as gliders use current speed to create lift and thrust. All movement then creates a drag force, acting in the opposite direction. Primarily in the form of wind resistance. As long as the thrust is greater than the drag, then the object will move forwards.

1.3 History of Aerodynamics

The Majority of the following information was obtained from Acheson (2003), which is an excellent overview of the history of the mathematics behind aerodynamics and compliments the project nicely.

Before the Wright brothers managed to complete the first ever powered flight on 17^{th} December 1903, the theory of flight had been mused for centuries. More than 500 years ago, Leonardo Da Vinci, famously, designed a flying machine that mimicked the flapping wings of a bird. But, it was not until the late 19^{th} Century when some serious breakthroughs occurred on the mathematics of lift.

In 1894, an English engineer named Frederick W. Lanchester presented a paper, *The soaring of birds and the possibilities of mechanical flight*, which is the first known paper to have included circulation as a theory of lift. Unfortunately, in 1897, the paper was rejected by the Physical Society. A few years later, in 1902, a German mathematician named Martin Kutta released a paper connecting circulation and lift.

In Kutta's paper, *Lifting forces in flowing fluids*, he found the solution to two-dimensional flow past a circular arc, with circulation round the surface and a finite velocity at the trailing edge. But it was not in the form of the general theorem that we know today. Coincidentally, a few years later, in 1906, a Russian Mathematician named Nikolai Joukowski published a paper containing the lift theorem in the form that we use today.

Stranger still, the following year, in 1907, Lanchester published his book, *Aerodynamics*, which contained important results dating back to 1892. It appears that the three mathematicians, Lanchester, Kutta, and Joukowski, all arrived at their own conclusions completely independent from each other. Even separate from the Wright brothers, whom succeeded using an experimental approach rather than a theoretical one.

2 Fluid Dynamics

The first aim of this project is to establish the fundamental calculus behind fluid dynamics, showing the vitally important formulae that will be referred back to constantly. We will then make some basic assumptions about the fluid, which we will be using. Finishing on the introduction of potential flow, which is important throughout the project.

2.1 Vector Calculus

Fluid dynamics is largely the application of vector calculus. In this section we will cover a few key concepts and definitions. Throughout this project we will be working in cartesian coordinates and cylindrical polar coordinates, which are defined as ($\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$) and ($\mathbf{e}_r$, $\mathbf{e}_\theta$, $\mathbf{e}_z$) as their three-dimensional coordinate systems, respectively. In mathematics we use polar coordinates when dealing with circular entities, which enables us to solve the problems easier.

All the following formulae are obtained from the Fluid Dynamics module (Henderson, 2021). The essential part to vector calculus is the 'del' operator, also known as the nabla symbol. The del opertator is defined as

$$\nabla \equiv \mathbf{i}\frac{\partial}{\partial x} + \mathbf{j}\frac{\partial}{\partial y} + \mathbf{k}\frac{\partial}{\partial z} \tag{2.1}$$

The del operator acts on a scalar function $\Omega(x, y, z)$ or a vector function $\mathbf{F}(x, y, z) = F_1\,\mathbf{i} + F_2\,\mathbf{j} + F_3\,\mathbf{k}$, in cartesian coordinates, and $\mathbf{F}(r, \theta, z) = F_r\,\mathbf{e}_r + F_\theta\,\mathbf{e}_\theta + F_z\,\mathbf{e}_z$, in cylindrical polar coordinates. Thus giving rise to the field quantities: gradient, divergence, curl, and the laplacian. We will now outline each of these fields in cartesian coordinates and cylindrical polar coordinates.

The Gradient

The gradient, or grad, of a scalar field results in a vector field. In cartesian coordinates the grad is defined as

$$\nabla\Omega = \frac{\partial\Omega}{\partial x}\mathbf{i} + \frac{\partial\Omega}{\partial y}\mathbf{j} + \frac{\partial\Omega}{\partial z}\mathbf{k} \tag{2.2}$$

and in cylindrical polar coordinates

$$\nabla\Omega = \frac{\partial\Omega}{\partial r}\mathbf{e}_r + \frac{1}{r}\frac{\partial\Omega}{\partial\theta}\mathbf{e}_\theta + \frac{\partial\Omega}{\partial z}\mathbf{e}_z \tag{2.3}$$

The Divergence

The divergence, or div, of a vector field results in a scalar field. In cartesian coordinates the div is defined as

$$\nabla \cdot \mathbf{F} = \frac{\partial F_1}{\partial x} + \frac{\partial F_2}{\partial y} + \frac{\partial F_3}{\partial z} \tag{2.4}$$

and in cylindrical polar coordinates

$$\nabla \cdot \mathbf{F} = \frac{1}{r}\frac{\partial\,(rF_r)}{\partial r} + \frac{1}{r}\frac{\partial\,(F_\theta)}{\partial\theta} + \frac{\partial F_z}{\partial z} \tag{2.5}$$

The Curl

The curl of a vector field results in a vector field. In cartesian coordinates the curl is defined as

$$\nabla \times \mathbf{F} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ F_1 & F_2 & F_3 \end{vmatrix} \tag{2.6}$$

and in cylindrical polar coordinates

$$\nabla \times \mathbf{F} = \frac{1}{r} \begin{vmatrix} \mathbf{e}_r & r\mathbf{e}_\theta & \mathbf{e}_z \\ \frac{\partial}{\partial r} & \frac{\partial}{\partial \theta} & \frac{\partial}{\partial z} \\ F_r & rF_\theta & F_z \end{vmatrix} \tag{2.7}$$

The Laplacian

The Laplacian operator, often called 'del-squared', for a scalar field in cartesian coordinates is defined as

$$\nabla^2 \Omega = \frac{\partial^2 \Omega}{\partial x^2} + \frac{\partial^2 \Omega}{\partial y^2} + \frac{\partial^2 \Omega}{\partial z^2} \tag{2.8}$$

and in cylindrical polar coordinates

$$\nabla^2 \Omega = \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial \Omega}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 \Omega}{\partial \theta^2} + \frac{\partial^2 \Omega}{\partial z^2} \tag{2.9}$$

2.2 Basic Assumptions

In order to simplify the problem, we need to make some basic assumptions about the fluid flow. The following assumptions are valid for subsonic flow where Mach, the speed of sound, is $M < 0.3$.

Direction of Flow

To avoid confusion, we will set the direction of flow to be consistent throughout this project. Mainly, the flow will be travelling from left to right. And we will set this direction to be our direction of positive flow. This means that the flow will be positive in the increasing x direction.

If, we have a flow that is only flowing vertically, with zero horizontal flow. Then, we will set the flow to be positive from bottom to top. That is, positive flow the increasing y direction. But, usually, it is sufficient to treat the flow with respect to its horizontal direction.

Two-dimensional flow

We will be looking at flows in only two dimensions. This enables us to look at the problem in solely the xy-plane. Meaning that when we look at flow around a cylinder and an aerofoil, we are actually looking at the cross-section that is infinitely long. This is, of course, just a mathematical model of the real thing, but it allows us to neglect the effects that occur at the end of the object. Enabling us to solely look at the mathematics of the fluid flowing around the object.

A 2D flow with velocity $\mathbf{v} = (\mathbf{r}, t)$, where $\mathbf{r}$ can represent any point in space and t represents time, is of the form

$$\mathbf{v} = u(x, y, t)\mathbf{i} + v(x, y, t)\mathbf{j}\,;$$

or in cylindrical coordinates

$$\mathbf{v} = v_r(r, \theta, t)\mathbf{e}_r + v_\theta(r, \theta, t)\mathbf{e}_\theta\,,$$

Throughout this project we will be confining ourselves to two-dimensional flow and thus will neglect the z coordinate from now on. Instead we will be referring to the xy-plane as the z-plane, which is common nomenclature for the cartesian plane.

Steady flow

The flow is stationary or steady. In other words, the velocity only depends on its position and does not depend on time. That is to say, if we have a fluid flow of velocity $\mathbf{v} = (\mathbf{r}, t)$ then for a steady flow we have

$$\frac{\partial \mathbf{v}}{\partial t} = 0$$

so, $\mathbf{v}$ does not depend on t and thus depends on $\mathbf{r}$ alone. Hence, a *two-dimensional steady flow* in cartesian coordinates is of the form

$$\mathbf{v} = u(x, y)\mathbf{i} + v(x, y)\mathbf{j}\,,$$

or in cylindrical coordinates

$$\mathbf{v} = v_r(r, \theta)\mathbf{e}_r + v_\theta(r, \theta)\mathbf{e}_\theta$$

Now, that is all the general assumptions completed. We will now move onto some particular assumptions about the fluid which form the basis of *potential flow theory.*

2.3 Potential Flow Theory

Potential flow is very useful within fluid dynamics, and is of vital importance to conformal mapping and the Joukowski Transformation. Which we will see in Section 6. It is a flow which we assume to be *incompressible*, *irrotational*, and *inviscid.* Potential flow means that we are able to derive a velocity $\mathbf{v}$ from a scalar velocity potential such that

$$\mathbf{v} = \nabla\Phi\,, \tag{2.10}$$

which you will notice is the gradient of a scalar (2.3). For equation (2.10) to hold, it is a necessary condition that the flow is *irrotational* (Hughes and Brighton, 1999).

Irrotational flow

The rotation of a fluid is given by the *vorticity* vector ω. Vorticity describes the infinitesimal rotation of a vector field and is given by the curl of the velocity. It is the localised rotation of fluid elements and has nothing to do with any global rotation of the fluid.

An irrotational flow is defined as

$$\omega = \nabla \times \mathbf{v} = 0\,, \tag{2.11}$$

the direction of the rotation is determined by the vorticity.

Incompressible flow

In Fluid Dynamics, fluids are either compressible or incompressible. For obvious reasons, liquids are treated as incompressible and gases are usually treated as compressible. Under certain conditions, where pressure is negligible, gases may be considered incompressible. This is the case for this project, in which we will be dealing with subsonic flow.

An incompressible fluid means that the density of the flow is constant. Incompressibility can be discovered using

$$\nabla \cdot \mathbf{v} = 0\,, \tag{2.12}$$

this equation is also known as the *continuity equation* and will be referenced numerous times throughout this project. Therefore, for a compressible fluid, $\nabla \cdot \mathbf{V} > 0$ means that the fluid is expanding, and conversely, if $\nabla \cdot \mathbf{V} < 0$ then the fluid is contracting.

Inviscid flow

In other words, the fluid has no viscosity and thus has no internal friction. An example of a fluid with high viscosity would be honey. It sticks to the spoon really well and moves slowly off it when poured, especially in colder temperatures. Conversely, water and air are good examples of fluids that have low viscosity.

We assume the fluid to be non-viscous so that we can neglect the boundary layer over the aerofoil. Obviously, inviscid fluid is just a mathematical concept and there is no such thing in the real world. But, the boundary layer in modelling air over a wing is not important for aerofoil theory and we can safely assume to neglect the effects.

3 Complex Variable Theory

Complex numbers are a fascinating and incredible realm of mathematics. They are so abstract that they are even called *imaginary* numbers, although there is not much imaginary about them. Since they have some truly remarkable real world uses.

In this project we will use the nomenclature $z = x + iy$, where $i = \sqrt{-1}$ and $x, y \in \mathbb{R}$, for cartesians coordinates. We will also use the commonly used $z = re^{i\theta}$, for problems requiring polar coordinates. Furthermore, a complex function $\zeta = f(z)$ is defined as: $f(z) = f(x+iy) = u(x, y) + iv(x, y)$, where $u(x, y)$ is called the real part and $v(x, y)$ is called the imaginary part.

This section concentrates on some of the applications of complex numbers. We will begin by explaining some key features of complex variable theory, which includes more vital aspects of this project. Namely, the *Cauchy-Riemann equations* and *Harmonic functions*. The section will finish with the introduction of the *complex potential*, which is a type of complex function that will be utilised in most of the project.

3.1 Cauchy-Riemann Equations

The Cauchy-Riemann equations are integral for this project, they allow us to test if a complex function is analytic. If a function $f(z)$ satisfies the Cauchy-Riemann equations, then it is said to be an analytic function. If a complex function $f(z)$ is differentiable at every point of the given region, then it is said to be analytic (Spiegel *et al.*, 2009). The Cauchy-Riemann equations are defined as

$$\begin{aligned}\frac{\partial u}{\partial x} &= \frac{\partial v}{\partial y}\,,\\ \frac{\partial v}{\partial x} &= -\frac{\partial u}{\partial y}\,.\end{aligned} \tag{3.1}$$

In addition to what we have stated above: If we have two real and continuous functions $u(x, y)$ and $v(x, y)$, that have continuous first partial derivatives and satisfy equations (3.1) in a given region, then the complex function $f(z)$ is analytic in that region (Kreyszig, 2011). This is an important property of a complex function, that we will see in the following sections on harmonic functions and the complex potential.

3.2 Harmonic Conjugate Functions

Harmonic functions are known to be one of the most important partial differential equations (PDEs), providing immensely important results within engineering and physics (Kreyszig, 2011). In this project, we are going to be utilising them with reference to fluid flows.

Recall the Laplacian operator (2.8) from earlier. If a function satisfies *Laplace's equation*

$$\nabla^2 f = 0\,, \tag{3.2}$$

then the function is known as a *harmonic function*. As mentioned in Section 3.1, if $f(z) = u(x, y) + iv(x, y)$ is analytic, then the Cauchy-Riemann equations

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}\,, \tag{3.3a}$$

$$\frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}\,. \tag{3.3b}$$

are satisfied.

Theorem 3.1. *If a function $f(z)$ is analytic, then both the real and imaginary parts of the analytic function are harmonic.*

Proof. If we differentiate (3.3a) with respect to x, we find

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2} &= \frac{\partial^2 v}{\partial y \partial x}\\ \Leftrightarrow \frac{\partial^2 u}{\partial x^2} &= \frac{\partial}{\partial y}\left(\frac{\partial v}{\partial x}\right)\end{aligned}$$

substituting (3.3b) in, we obtain

$$
\begin{aligned}
&\frac{\partial^2 u}{\partial x^2} = \frac{\partial}{\partial y}\left(-\frac{\partial u}{\partial y}\right) \\
\Leftrightarrow &\frac{\partial^2 u}{\partial x^2} = -\frac{\partial^2 u}{\partial y^2} \\
\Leftrightarrow &\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0
\end{aligned}
$$

thus the function $u(x, y)$ satisfies Laplace's equation. Similarly, differentiating (3.3b) with respect to x gives us

$$
\begin{aligned}
&\frac{\partial^2 v}{\partial x^2} = -\frac{\partial^2 u}{\partial y \partial x} \\
\Leftrightarrow &\frac{\partial^2 v}{\partial x^2} = -\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x}\right)
\end{aligned}
$$

then substituting in (3.3a), we see that

$$
\begin{aligned}
&\frac{\partial^2 v}{\partial x^2} = -\frac{\partial}{\partial y}\left(\frac{\partial v}{\partial y}\right) \\
\Leftrightarrow &\frac{\partial^2 v}{\partial x^2} = -\frac{\partial^2 u}{\partial y^2} \\
\Leftrightarrow &\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0
\end{aligned}
$$

thus the function $v(x, y)$ also satisfies Laplace's equation. Hence, since both the real and imaginary parts satisfy Laplace's equation they are therefore harmonic. □

Harmonic functions are vital to this project because if we are given one of $u(x, y)$ or $v(x, y)$, then we can find the other such that $f(z) = u(x, y) + iv(x, y)$ is analytic. It all ties together so nicely!

3.3 Complex Potential

In this section we shall derive the *complex potential* function $F(z)$ which we will be using to calculate the flow around objects. The complex potential comprises of two harmonic functions: the *velocity potential* Φ, which is the real part, and the *stream function* Ψ, which is the imaginary part. The complex potential is one of the most powerful tools in potential flow theory and forms the basis of subsonic aerodynamics (Hughes and Brighton, 1999).

Velocity Potential

The first part of the complex potential is the velocity potential, which is the real part of the the function. If we let $\Phi =$ contant, we can find the *equipotiential lines* that are parallel to the vertical axis. Recall from Section 2.3, that it is a necessary condition of irrotationality that the velocity is derivable from the scalar velocity potential

$$
\mathbf{v} = \nabla\Phi\,, \tag{3.4}
$$

thus, we have the velocity components

$$u = \frac{\partial \Phi}{\partial x}\,; \quad v = \frac{\partial \Phi}{\partial y}\,. \tag{3.5}$$

The velocity potential can be used in three dimensions but since we have assumed two-dimensional for this project, we shall continue as such. Recall also that we have assumed incompressibility. If we substitute (3.4) into the continuity equation (2.12) we can see that

$$\begin{aligned}
\nabla \cdot \mathbf{v} &= \nabla \cdot (\nabla \Phi) \\
&= \left(\frac{\partial}{\partial x}\,\mathbf{i} + \frac{\partial}{\partial y}\,\mathbf{j}\right) \cdot \left(\frac{\partial \Phi}{\partial x}\,\mathbf{i} + \frac{\partial \Phi}{\partial y}\,\mathbf{j}\right) \\
&= \frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = 0\,.
\end{aligned}$$

Hence, the velocity potential Φ is indeed a harmonic function.

Stream Function

Another important function is the *stream function* Ψ, which can be defined for any two-dimensional flow irrespective of whether it is irrotational and incompressible. But since, as stated before, we are using potential flow, then we shall carry on as such. If we let $\Psi = \textit{constant}$ then we can obtain *streamlines*, which are lines parallel to the horizontal axis. We will be using streamlines to model flow in the next section.

From the continuity equation (2.12) we have

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0\,,$$

which is satisfied when

$$u = \frac{\partial \Psi}{\partial y}\,; \quad v = -\frac{\partial \Psi}{\partial x}\,, \tag{3.6}$$

and in polar coordinates, we have

$$v_r = \frac{1}{r}\frac{\partial \Psi}{\partial \theta}\,; \quad v_\theta = -\frac{\partial \Psi}{\partial r}\,. \tag{3.7}$$

Equations (3.6) and (3.7) can both be used to easily find the stream function from a velocity profile, in cartesian coordinates and polar coordinates, respectively. Depending on whichever is best suited to the problem.

It can then be shown that the stream function is analytic and harmonic. From irrotationality (2.11), we find that

$$\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} = 0\,,$$

and when we substitute (3.6) in, we obtain

$$\frac{\partial}{\partial y}\left(\frac{\partial \Psi}{\partial y}\right) - \frac{\partial}{\partial x}\left(-\frac{\partial \Psi}{\partial x}\right) = \frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial^2 \Psi}{\partial y^2} = 0\,.$$

Hence, the stream function Ψ is a harmonic function.

Moreover, if we equate (3.5) and (3.6) we notice that the functions are the Cauchy-Riemann equations! As we can see here:

$$u = \frac{\partial \Phi}{\partial x} = \frac{\partial \Psi}{\partial y}\,; \quad v = \frac{\partial \Phi}{\partial y} = -\frac{\partial \Psi}{\partial x}\,. \tag{3.8}$$

Hence the velocity potential and stream function satisfy the Cauchy-Riemann equations and are harmonic functions. The two equations in (3.8) also represent the velocity components of an irrotational, incompressible flow in the z-plane (Acheson, 2003).

The Complex Function

In the previous section we showed that the velocity potential Φ and stream function Ψ are both analytic and harmonic functions and therefore *harmonic conjugate functions*. Thus we have the function:

$$F(z) = \Phi(x, y) + i\Psi(x, y)\,, \tag{3.9}$$

which is known as the *complex potential*, such that the function $F(z)$ is an analytic function of $z = x + iy$. Since Φ and Ψ are uniquely determined conjugate functions except for an additive real constant, we may called it *the* complex potential (Kreyszig, 2011).

Complex Velocity

If we differentiate the complex potential (3.9), we obtain

$$\frac{dF}{dz} = \frac{\partial \Phi}{\partial x} + i\frac{\partial \Psi}{\partial x}\,,$$

which, of course, is interchangeable with the Cauchy-Riemann equations. Conveniently, we can substitute in equation (3.8) to find the velocity components u and v, thus giving us

$$\frac{dF}{dz} = u - iv \tag{3.10}$$

this is known as the *complex velocity.*. It follows that

$$\left|\frac{dF}{dz}\right|^2 = u^2 + v^2 = v_r^2 + v_\theta^2 = V^2 \tag{3.11}$$

and V^2 can be used in Bernoulli's equation enabling us to perform calculations of the pressure.

4 Simple Flows

Now that we have the fundamental mathematics, we will begin to model flow graphically. Each section will have a velocity potential Φ, a stream function Ψ, or a velocity profile $\mathbf{v}$, acquired during the research for this project. Followed by calculations to find the complex potential in its simplest form, which will lead onto calculations for the complex velocity. Every section will then be completed with calculations to find V^2. Which does not have much use within the scope of this project, but it is a good point at which to conclude the calculations, seeing as that would be the next logical step. See appendix A.1 to view all the code used in this section.

4.1 Uniform Flows

Uniform Flows are the most basic type of flow. It is a constant flow in a straight line, with no deviation, unless another flow or an object interfere with it. We will be looking at uniform flow parallel to both axes, followed by uniform flow that can be at any angle on the plane.

Parallel to horizontal axis

Firstly, we will look at uniform flow parallel to the x-axis. Suppose we have a velocity profile $\mathbf{v} = U_0\,\mathbf{i}$, obtained from Hughes and Brighton (1999). As stated in Section 3.3, we can find the velocity potential Φ using $\mathbf{v} = \nabla\Phi$. Thus integrating the velocity profile we find

$$\Phi = U_0 x + c_1\,. \tag{4.1}$$

Then, using equations (3.6), we can find the stream function. As follows

$$\begin{aligned} u &= \frac{\partial \Psi}{\partial y}\,; \quad v = -\frac{\partial \Psi}{\partial x} \\ \Rightarrow \frac{\partial \Psi}{\partial y} &= U_0\,; \quad \frac{\partial \Psi}{\partial x} = 0 \end{aligned}$$

integrating we obtain

$$\Psi = U_0 y + c_2\,. \tag{4.2}$$

Hence, combining (4.2) and (4.1), we now have the complex potential for uniform flow parallel to the x-axis, which is

$$\begin{aligned} F(z) &= U_0 x + iU_0 y \\ &= U_0(x+iy) \\ \Rightarrow F(z) &= U_0 z \end{aligned} \tag{4.3}$$

And we can easily find the complex velocity to be

$$\frac{dF}{dz} = U_0\,,$$

thus

$$V^2 = U_0^2\,. \tag{4.4}$$

As mentioned in Section 3.3, streamlines can be obtained if we let $\Psi = A$, where ($A =$ *constant*). So we have $A = U_0\,y + c_2 \Rightarrow y = B/U_0$, where $B = A - c_2$. Thus the streamlines will occur parallel to the x-axis, as seen in Figure 4.1.

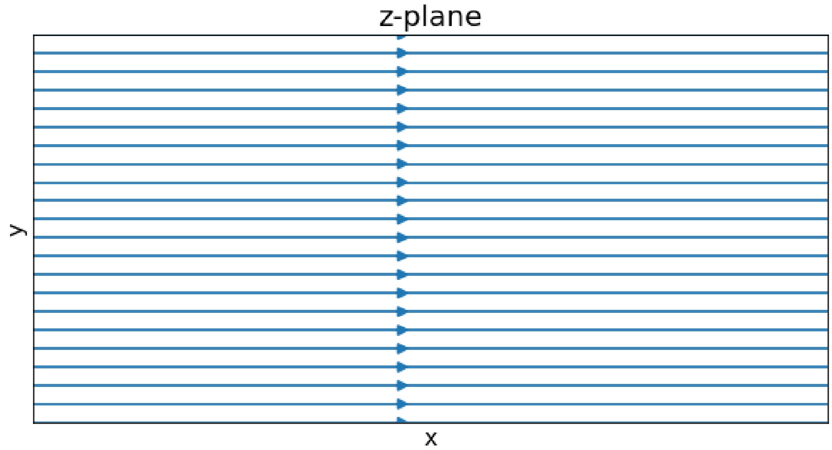


Figure 4.1: Python output of uniform flow parallel to x-axis.

Parallel to vertical axis

Now, we will look at uniform flow parallel to the y-axis. Suppose we have a velocity profile $\mathbf{v} = U_0\,\mathbf{j}$, that was obtained using knowledge from the previous section. Using the same process as above, we can find the velocity potential and stream function:

$$\Phi = U_0 y + c_1\,; \tag{4.5}$$
$$\Psi = -U_0 x + c_2\,. \tag{4.6}$$

Combining equations 4.5 and 4.6, we have the complex potential

$$\begin{aligned} F(z) &= U_0 y - iU_0 x \\ &= U_0(y - ix) \\ &= iU_0(x + iy) \\ \Rightarrow F(z) &= iU_0 z\,. \end{aligned} \tag{4.7}$$

Now we can obtain the complex velocity

$$\frac{dF}{dz} = iU_0\,,$$

therefore

$$V^2 = u^2 + v^2 = U_0^2\,. \tag{4.8}$$

We can find the streamlines if we let $\Psi = A$, where ($A = \textit{constant}$), so the streamlines occur at $x = B/U_0$, where $B = c_2 - A$. Thus streamlines will occur parallel to the y-axis, as seen in Figure 4.2.

Uniform flow at an angle

Suppose we have the velocity profile $\mathbf{v} = U_0 \cos\alpha\,\mathbf{i} + U_0 \sin\alpha\,\mathbf{j}$, where α is the angle of the flow to the horizontal axis. Later on we will see that it is also the angle of attack. The velocity profile was obtained with the idea from Acheson (2003). We can then find the velocity potential

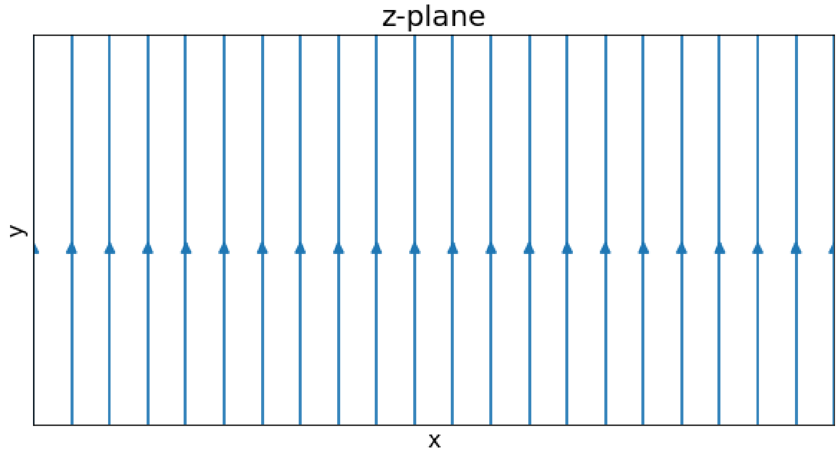


Figure 4.2: Python output of uniform flow parallel to y-axis.

$$\Phi = U_0 x \cos\alpha + U_0 y \sin\alpha + c_1 \tag{4.9}$$

then, similar to what we have done previously, we can find the stream function as follows

$$\begin{aligned} \frac{\partial \Psi}{\partial y} &= U_0 \cos\alpha\,; & \frac{\partial \Psi}{\partial x} &= -U_0 \sin\alpha \\ \Rightarrow \Psi_1 &= U_0 y \cos\alpha + c_2\,; & \Psi_2 &= -U_0 x \sin\alpha + c_3 \end{aligned}$$

thus

$$\Psi = U_0 y \cos\alpha - U_0 x \sin\alpha + c_4\,. \tag{4.10}$$

We can now obtain, with some manipulation, the complex potential

$$\begin{aligned} F(z) &= (U_0 x \cos\alpha + U_0 y \sin\alpha) + i(U_0 y \cos\alpha - U_0 x \sin\alpha) \\ &= U_0 x(\cos\alpha - i\sin\alpha) + U_0 y(\sin\alpha + i\cos\alpha) \\ &= U_0 x e^{-i\alpha} + U_0 y(\sin\alpha + i\cos\alpha) \\ &= U_0 x e^{-i\alpha} + iU_0 y(\cos\alpha - i\sin\alpha) \\ &= U_0 x e^{-i\alpha} + iU_0 y e^{-i\alpha} \\ \Rightarrow F(z) &= U_0 z e^{-i\alpha} \end{aligned} \tag{4.11}$$

and the complex velocity can easily found to be

$$\begin{aligned} \frac{dF}{dz} &= U_0 e^{-i\alpha} \\ &= U_0(\cos\alpha - i\sin\alpha) \\ \Rightarrow \frac{dF}{dz} &= U_o \cos\alpha - i\sin\alpha\,, \end{aligned}$$

therefore

$$
\begin{aligned}
V^2 &= U_0^2 \cos^2\alpha + U_0^2 \sin^2\alpha \\
&= U_0^2(\cos^2\alpha + \sin^2\alpha) \\
\Rightarrow V^2 &= U_0^2\,. \qquad (4.12)
\end{aligned}
$$

Notice that V^2 in equations (4.4), (4.8) and (4.12) are all identical, which we would expect to see. These being equal tell us the obvious fact that the velocity is equal to the strength of the flow, since there are no obstacles and the flow is constant. Uniform flow at angle α can be seen in Figure 4.3.

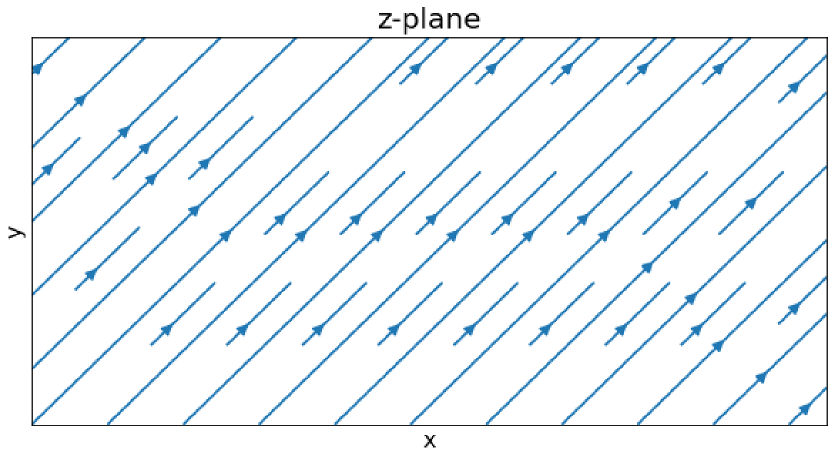


Figure 4.3: Python output of uniform flow at angle $\alpha = \frac{\pi}{4}$.

4.2 Source and Sink

In this section we will look at the behaviour of fluid from points where fluid flows out, known as a source, and points where fluid flows into, known as a sink.

Suppose we have a velocity profile $\mathbf{v} = m/2\pi r\, \mathbf{e}_r$, where m is the strength of the source or sink. This velocity profile was obtained from Henderson (2021). If $m > 0$, we have positive flow and therefore it is a source. And visa versa, if $m < 0$, we have negative flow and thus a sink. Then, with the use of cylindrical coordinates, we find the velocity potential

$$
\begin{aligned}
\frac{\partial \Phi}{\partial r} &= \frac{m}{2\pi r}\mathbf{e}_r \\
\Rightarrow \Phi &= \frac{m}{2\pi}\ln r + c_1\,. \qquad (4.13)
\end{aligned}
$$

We can then obtain the stream function as follows

$$
\begin{aligned}
v_r &= \frac{1}{r}\frac{\partial \Psi}{\partial \theta}\,; \quad v_\theta = -\frac{\partial \Psi}{\partial r} \\
\Rightarrow \frac{1}{r}\frac{\partial \Psi}{\partial \theta} &= \frac{m}{2\pi r}\,; \quad \frac{\partial \Psi}{\partial r} = 0
\end{aligned}
$$

therefore

$$\frac{\partial \Psi}{\partial \theta} = \frac{m}{2\pi}$$
$$\Rightarrow \Psi = \frac{m}{2\pi}\theta + c_2 \tag{4.14}$$

now we have the complex potential

$$F(z) = \frac{m}{2\pi}\ln r + i\frac{m}{2\pi}\theta$$
$$= \frac{m}{2\pi}(\ln r + i\theta)$$
$$\Rightarrow F(z) = \frac{m}{2\pi}\ln z\,. \tag{4.15}$$

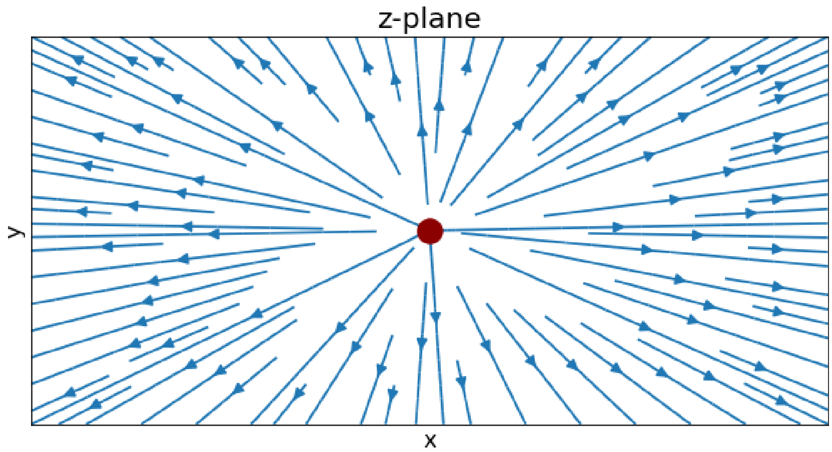


Figure 4.4: Python output of a sink centred at the origin with strength $m = 5$.

It follows that we are able to find the complex velocity

$$\frac{dF}{dz} = \frac{m}{2\pi z}$$
$$= \frac{m}{2\pi r}e^{-i\theta}$$
$$\Rightarrow \frac{dF}{dz} = \frac{m}{2\pi r}\cos\theta - i\frac{m}{2\pi r}\sin\theta$$

therefore

$$V^2 = \left(\frac{m}{2\pi r}\cos\theta\right)^2 + \left(\frac{m}{2\pi r}\sin\theta\right)^2$$
$$= \left(\frac{m}{2\pi r}\right)^2(\cos^2\theta + \sin^2\theta)$$
$$\Rightarrow V^2 = \frac{m^2}{4\pi^2 r^2}\,.$$

Streamlines and equipotential lines are found by letting $\Phi = A$ and $\Psi = B$, then rearranging to find $\theta = A2\pi/m$ and $r = e^{B2\pi/m}$, respectively. In Figure 4.4, we can see a source centred at the origin with the flow coming out in all directions but that point. Figure 4.5 depicts a sink, also centred at the origin, we can clearly see the flow going towards that point from all directions.

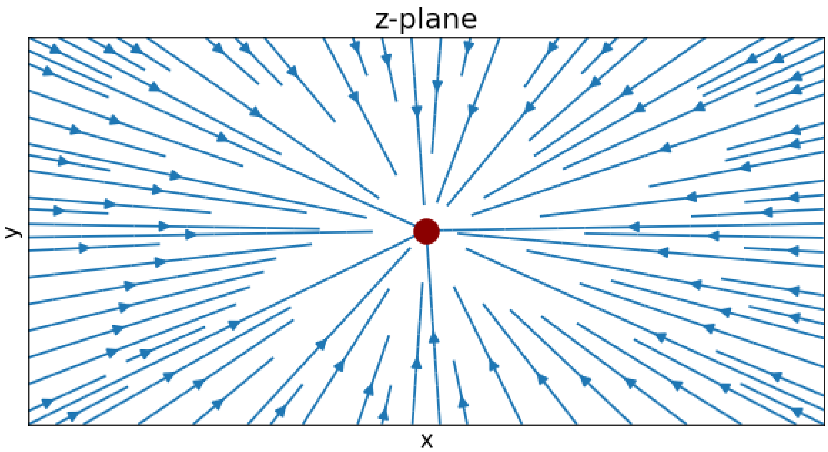


Figure 4.5: Python output of a sink centred at the origin with strength $m = -5$.

4.3 Line Vortex

Suppose we have the velocity profile $\mathbf{v} = \Gamma/2\pi r\, \mathbf{e}_\theta$, where Γ is known as the circulation. This is the velocity profile of a line vortex, obtained from Acheson (2003). We can then calculate the velocity potential

$$\begin{aligned} \frac{1}{r}\frac{\partial \Phi}{\partial \theta} &= \frac{\Gamma}{2\pi r} \\ \Rightarrow \Phi &= \frac{\Gamma}{2\pi}\theta + c_1 \end{aligned} \tag{4.16}$$

followed by the stream function

$$\begin{aligned} \frac{1}{r}\frac{\partial \Psi}{\partial \theta} = 0\,; \quad \frac{\partial \Psi}{\partial r} &= -\frac{\Gamma}{2\pi r} \\ \Rightarrow \Psi &= -\frac{\Gamma}{2\pi}\ln r + c_2 \end{aligned} \tag{4.17}$$

As done previously, we combine equations (4.16) and (4.17) to obtain the complex function

$$F(z) = \frac{\Gamma}{2\pi}\theta - i\frac{\Gamma}{2\pi}\ln r$$

and, with some manipulation, we find

$$\begin{aligned} F(z) &= \frac{\Gamma}{2\pi}(\theta - i\ln r) \\ &= -\frac{\Gamma}{2\pi}(i\ln r - \theta) \\ &= -i\frac{\Gamma}{2\pi}(\ln r + i\theta) \\ \Rightarrow F(z) &= -i\frac{\Gamma}{2\pi}\ln z\,. \end{aligned} \tag{4.18}$$

We can then find the complex velocity as follows

$$
\begin{aligned}
\frac{dF}{dz} &= -i\frac{\Gamma}{2\pi z} \\
&= -i\frac{\Gamma}{2\pi r}e^{-i\theta} \\
&= -i\frac{\Gamma}{2\pi r}(\cos\theta - i\sin\theta) \\
\Rightarrow \frac{dF}{dz} &= -\frac{\Gamma}{2\pi r}\sin\theta - i\frac{\Gamma}{2\pi r}\cos\theta\,,
\end{aligned}
$$

therefore

$$
\begin{aligned}
V^2 &= \left(\frac{\Gamma}{2\pi r}\sin\theta\right)^2 + \left(\frac{\Gamma}{2\pi r}\cos\theta\right) \\
&= \left(\frac{\Gamma}{2\pi r}\right)^2(\cos^2\theta + \sin^2\theta) \\
\Rightarrow V^2 &= \frac{\Gamma^2}{4\pi^2 r^2}
\end{aligned}
$$

By letting $\Phi = A$ and $\Phi = B$, Where A and B are constants, and by rearranging we can find the equations for the streamlines and equipotential lines to be: $r = e^{-2A\pi/\Gamma}$ and $\theta = 2B\pi/\Gamma$, respectively. Notice that these are opposite to the equations used for a source and sink. Which is exactly what we would expect to see. Figure 4.6 depicts at vortex centred at the origin with positive flow in the anti-clockwise direction, which has relevance to aerofoil theory that we shall see later on.

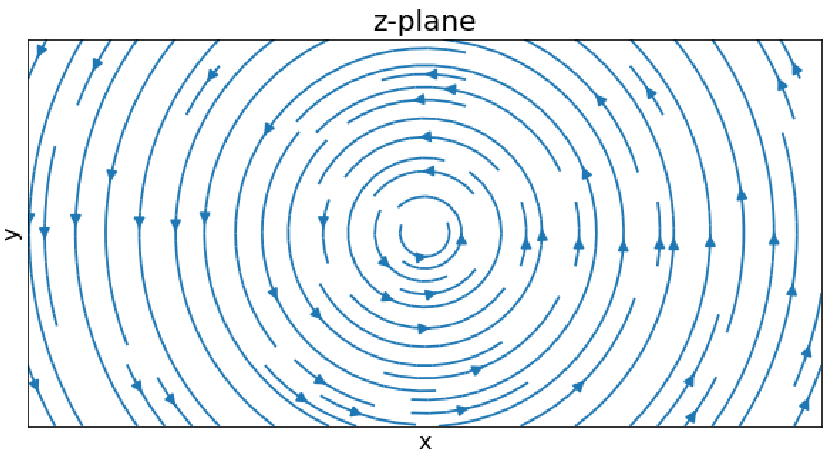


Figure 4.6: Python output of a line vortex, centred at the origin, with strength $\Gamma = 10$.

4.4 Superposition of Flows

In this section we will look at the superposition of potential flows. Which is where are able to linearly add potential flows to create more complex flows. Which we shall also see when we look at flow around cylinders in Section 5. Since potential flows are governed by harmonic functions, we can then combine different functions to create new functions.

4.4.1 Dipole

Here we will be looking at the special case called a dipole, where we will superimpose a source with a sink. It has a very important function within this project, which we shall see very shortly. This section will be following closely to the work done in the Fluid Dynamics module (Henderson, 2021).

Suppose we have a source at the origin and a source at distance, a, from the origin along the x-axis, giving the coordinates: $(a, 0)$. As we did previously, the sink and source have strength m. Using our result from Section 4.2, we have

$$\Phi = \frac{m}{2\pi}\ln r - \frac{m}{2\pi}\ln\bar{r}\,, \tag{4.19}$$

where $\bar{r} = \sqrt{(x-a)^2 + y^2}$. Which is the distance of the sink from the origin, i.e., the source. Then, by expanding and converting to cylindrical coordinates, we find

$$\begin{aligned}
\bar{r} &= (x^2 - 2ax + a^2 + y^2)^{\frac{1}{2}} \\
&= (r^2\cos^2\theta - 2ar\cos\theta + a^2 + r^2\sin^2\theta)^{\frac{1}{2}} \\
&= (r^2 - 2ar\cos\theta + a^2)^{\frac{1}{2}} \\
&= r\left(1 - \frac{2ar}{r^2}\cos\theta + \frac{a^2}{r^2}\right)^{\frac{1}{2}} \\
&= r\left(1 - \frac{2a}{r}\cos\theta + \frac{a^2}{r^2}\right)^{\frac{1}{2}}\,,
\end{aligned} \tag{4.20}$$

substituting (4.20) into the original natural logarithm, we obtain

$$\begin{aligned}
\ln\bar{r} &= \ln\left(r\left(1 - \frac{2a}{r}\cos\theta + \frac{a^2}{r^2}\right)^{\frac{1}{2}}\right) \\
&= \ln r + \ln\left(\left(1 - \frac{2a}{r}\cos\theta + \frac{a^2}{r^2}\right)^{\frac{1}{2}}\right) \\
&= \ln r + \frac{1}{2}\ln\left(1 - \frac{2a}{r}\cos\theta + \frac{a^2}{r^2}\right).
\end{aligned}$$

We assume that a is very small and also much smaller than r. Thus, using Maclaurin series, where $\ln(1+x) \approx x - \frac{1}{2}x^2$ for small x, we have

$$\ln\bar{r} \approx \ln r - \frac{a}{r}\cos\theta\,,$$

and substituting back into (4.19)

$$\begin{aligned}
\Phi &\approx \frac{m}{2\pi}\ln r - \frac{m}{2\pi}\left(\ln r - \frac{a}{r}\cos\theta\right) \\
&\approx \frac{ma}{2\pi r}\cos\theta\,.
\end{aligned}$$

If we let $a \to 0$ but keep it bounded, such that $ma \to \mu$, where μ is a constant. Then we can obtain the velocity potential

$$\Phi = \frac{\mu}{2\pi r}\cos\theta\,. \tag{4.21}$$

We can now convert to a velocity profile, giving us

$$\begin{aligned}
\mathbf{v} = \nabla\Phi &= \frac{\partial\Phi}{\partial r}\mathbf{e}_r + \frac{1}{r}\frac{\partial\Phi}{\partial\theta}\mathbf{e}_\theta \\
&= -\frac{\mu}{2\pi r^2}\cos\theta\,\mathbf{e}_r + \frac{1}{r}\left(-\frac{\mu}{2\pi r}\sin\theta\right)\mathbf{e}_\theta \\
\Rightarrow \mathbf{v} &= -\frac{\mu}{2\pi r^2}\cos\theta\,\mathbf{e}_r - \frac{\mu}{2\pi r^2}\sin\theta\,\mathbf{e}_\theta
\end{aligned}$$

and obtain the stream function, as follows

$$\begin{aligned}
\frac{1}{r}\frac{\partial\Psi}{\partial\theta} &= -\frac{\mu}{2\pi r^2}\cos\theta \quad ; \quad -\frac{\partial\Psi}{\partial r} = -\frac{\mu}{2\pi r^2}\sin\theta \\
\frac{\partial\Psi}{\partial\theta} &= -\frac{\mu}{2\pi r}\cos\theta \quad ; \quad \frac{\partial\Psi}{\partial r} = \frac{\mu}{2\pi r^2}\sin\theta \\
\Rightarrow \Psi &= -\frac{\mu}{2\pi r}\sin\theta + f(r) \quad ; \quad \Psi = -\frac{\mu}{2\pi r}\sin\theta + g(\theta)
\end{aligned}$$

$$\Rightarrow \Psi = -\frac{\mu}{2\pi r}\sin\theta + c\,. \tag{4.22}$$

We now, finally, can combine equations (4.21) and (4.22) to obtain the complex potential, followed by some manipulation. Thus, we have

$$\begin{aligned}
F(z) &= \frac{\mu}{2\pi r}\cos\theta - i\frac{\mu}{2\pi r}\sin\theta \\
&= \frac{\mu}{2\pi r}(\cos\theta - i\sin\theta) \\
&= \frac{\mu}{2\pi r}e^{-i\theta} \\
&= \frac{\mu}{2\pi r e^{i\theta}} \\
\Rightarrow F(z) &= \frac{\mu}{2\pi z}\,,
\end{aligned} \tag{4.23}$$

and we can easily find the complex velocity to be

$$\begin{aligned}
\frac{dF}{dz} &= -\frac{\mu}{2\pi z^2} \\
&= -\frac{\mu}{2\pi r^2}e^{-i2\theta} \\
&= -\frac{\mu}{2\pi r^2}(\cos 2\theta - i\sin 2\theta) \\
\Rightarrow \frac{dF}{dz} &= -\frac{\mu}{2\pi r^2}\cos 2\theta + i\frac{\mu}{2\pi r^2}\sin 2\theta
\end{aligned}$$

therefore

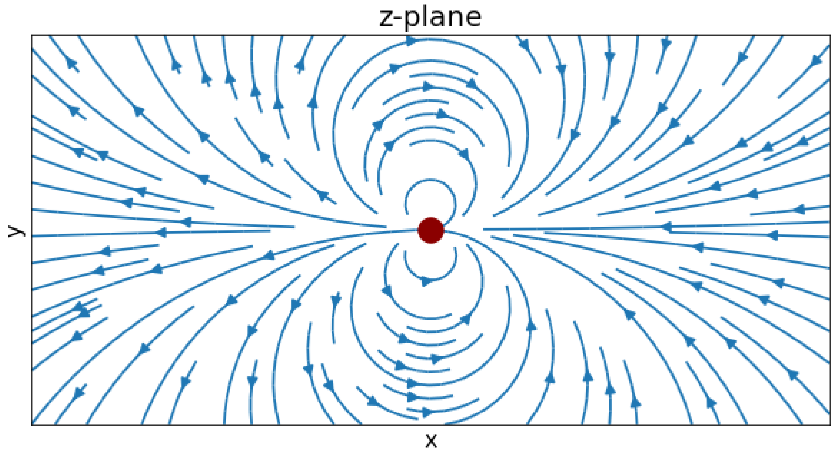


Figure 4.7: Python output of a dipole, centred at the origin, with strength $\mu = 10$.

$$
\begin{aligned}
V^2 &= \left(\frac{\mu}{2\pi r^2}\cos 2\theta\right)^2 + \left(\frac{\mu}{2\pi r^2}\sin 2\theta\right)^2 \\
&= \left(\frac{\mu}{2\pi r^2}\right)^2 \left((\cos 2\theta)^2 + (\sin 2\theta)^2\right) \\
\Rightarrow V^2 &= \frac{\mu^2}{4\pi^2 r^4}\,. \qquad (4.24)
\end{aligned}
$$

Streamlines can be found when $\Psi = A$, where A is a constant. Thus, from (4.22), we have $A = -\frac{\mu}{2\pi r}\sin\theta + c \Rightarrow r = B\sin\theta$, where $B = -\frac{\mu}{2\pi A}$. In Figure 4.7 we can see a dipole with both the source and sink located at the origin, which works in the python to create the plots but recall that the centres are approaching each other such that $a \to 0$.

4.4.2 Superimposed Uniform Flow with Dipole

When uniform flow is superimposed with a dipole, we notice a very interesting result. In Figure 4.8 there appears to be a cylinder shape at the centre that the flow is going around. In fact, the solution to flow around a cylinder is actually the superposition of uniform flow and a dipole (Brennen, 2004).

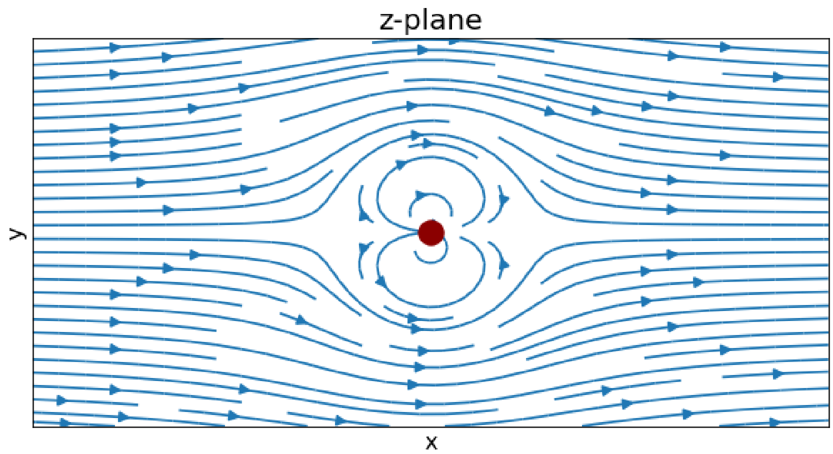


Figure 4.8: Python output of uniform flow superimposed with a dipole to show flow around a cylinder.

If we superimpose the complex potential of uniform flow parallel to the horizontal axiss (4.3) and the complex potential of a dipole (4.23), we have

$$F(z) = U_0 z + \frac{\mu}{2\pi z}\,. \tag{4.25}$$

Let $\mu = r^2 2\pi U_0$ (we will see where this comes from later on - it is the rearranged formula for the radius of a dipole that is in uniform flow). Thus, we find that

$$F(z) = U_0\left(z + \frac{r^2}{z}\right), \tag{4.26}$$

which is actually the complex potential for uniform flow past a cylinder (5.8), except that we have $r = a$.

Hence, simply superimposing uniform flow with a dipole, we can create the exact streamlines that would be expected in uniform flow past a stationary cylinder. We can neglect the streamlines that are inside the radius of the dipole, since they do not effect the flow except as acting as a solid cylinder. In the following section we will put a circle, with radius equal to the radius of the dipole, to cover these streamlines.

5 Flow Around Cylinders

Now our interest has finally progressed to flow past a cylinder, starting with uniform flow, then superimposing circulation into to the flow. We have just seen that the solution to uniform flow around a cylinder is actually the superposition of uniform flow and a dipole. Following a different route, we will see that we end up with the same solution for both.

We will then superimpose a line vortex into our calculations this will simply add circulation, which is an important aspect into why aerofoils can generate lift. The section will conclude with finding the stagnation points in the two types of flow, which have a significant use when looking at pressure and lift calculations.

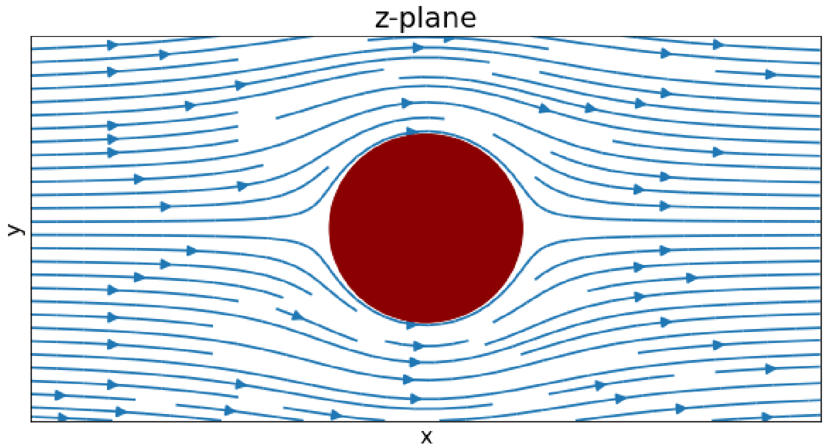


Figure 5.1: Uniform flow past a stationary cylinder that is centred at origin, with $U_0 = 1$ and $\mu = 6$. The streamlines in this plot show the direction of flow.

Figures 4.8 and 5.1 are identical, except that there is a circle with equal radius to the dipole on top of the dipole. We will now have a look at the mathematics of flow around a cylinder and see that the mathematics behind these two illustrations are also identical. Please see appendix A.1 for all the code written for this section.

5.1 Uniform Flow Past a Stationary Cylinder

We have the velocity potential, obtained from Henderson (2021):

$$\Phi = U_0 \left(r + \frac{a^2}{r} \right) \cos\theta \, . \tag{5.1}$$

It follows that we can obtain the velocity profile, using polar coordinates

$$\begin{aligned} \mathbf{v} &= U_0 \left(1 - \frac{a^2}{r^2} \right) \cos\theta \, \mathbf{e}_r + \frac{1}{r} \left(-U_0 \left(r + \frac{a^2}{r} \right) \sin\theta \right) \mathbf{e}_\theta \\ \Rightarrow \mathbf{v} &= U_0 \left(1 - \frac{a^2}{r^2} \right) \cos\theta \, \mathbf{e}_r - U_0 \left(1 + \frac{a^2}{r^2} \right) \sin\theta \, \mathbf{e}_\theta \, , \end{aligned} \tag{5.2}$$

now we can begin the find the stream function, as follows

$$\frac{1}{r} \frac{\partial \Psi}{\partial \theta} = U_0 \left(1 - \frac{a^2}{r^2} \right) \cos\theta \tag{5.3}$$

$$-\frac{\partial \Psi}{\partial r} = -U_0 \left(1 + \frac{a^2}{r^2} \right) \sin\theta \tag{5.4}$$

if we we arrange (5.3) and integrate with respect to θ we find

$$\begin{aligned} \frac{\partial \Psi}{\partial \theta} &= U_0 \left(r - \frac{a^2}{r} \right) \cos\theta \\ \Rightarrow \Psi_1 &= U_0 \left(r - \frac{a^2}{r} \right) \sin\theta + g(r) \end{aligned} \tag{5.5}$$

then, integrate (5.4) with respect to r to find

$$\begin{aligned} \frac{\partial \Psi}{\partial r} &= U_0 \left(1 + \frac{a^2}{r^2} \right) \sin\theta \\ \Rightarrow \Psi_2 &= U_0 \left(r - \frac{a^2}{r} \right) \sin\theta + f(\theta) \end{aligned} \tag{5.6}$$

finally, we combine (5.5) and (5.6) to obtain our stream function, which is

$$\Psi = U_0 \left(r - \frac{a^2}{r} \right) \sin\theta + c \, . \tag{5.7}$$

Now using equations (5.1) and (5.7) we can create the complex potential and simplify using some manipulation, as follows

$$
\begin{aligned}
F(z) &= U_0\left(r+\frac{a^2}{r}\right)\cos\theta + iU_0\left(r-\frac{a^2}{r}\right)\sin\theta \\
&= U_0\left[r\cos\theta + ir\sin\theta + \frac{a^2}{r}\cos\theta - i\frac{a^2}{r}\sin\theta\right] \\
&= U_0\left[r(\cos\theta + i\sin\theta) + \frac{a^2}{r}(\cos\theta - i\sin\theta)\right] \\
&= U_0\left[r\,e^{i\theta} + \frac{a^2}{r}e^{-i\theta}\right] \\
&= U_0\left[r\,e^{i\theta} + \frac{a^2}{r\,e^{i\theta}}\right] \\
\Rightarrow F(z) &= U_0\left(z+\frac{a^2}{z}\right).
\end{aligned} \tag{5.8}
$$

Notice that equation (5.8) is equivalent to the superposition of the complex potential of uniform flow and a dipole (4.26). Now to find the complex velocity and expand to find the real and imaginary parts. We have

$$
\begin{aligned}
\frac{dF}{dz} &= U_0\left(1-\frac{a^2}{z^2}\right) \qquad (5.9)\\
&= U_0\left(1-\frac{a^2}{r^2}e^{-i2\theta}\right) \\
&= U_0\left(1-\frac{a^2}{r^2}(\cos 2\theta - i\sin 2\theta\right) \\
&= U_0\left(1-\frac{a^2}{r^2}\cos 2\theta - i\frac{a^2}{r^2}\sin 2\theta\right) \\
\Rightarrow \frac{dF}{dz} &= U_0\left(1-\frac{a^2}{r^2}\cos 2\theta\right) - i\frac{U_0a^2}{r^2}\sin 2\theta\,.
\end{aligned}
$$

This time we will use a slightly different method to obtain V^2. We will take the cylindrical polar velocity components, v_r and v_θ, from equation (5.2) and calculate V^2 using these. The velocity components are:

$$
\begin{aligned}
v_r &= U_0\left(1-\frac{a^2}{r^2}\right)\cos\theta \\
v_\theta &= -U_0\left(1+\frac{a^2}{r^2}\right)\sin\theta
\end{aligned}
$$

therefore

$$
\begin{aligned}
V^2 &= u^2 + v^2 = v_r^2 + v_\theta^2 \\
&= \left[U_0^2\left(1-\frac{a^2}{r^2}\right)^2\cos^2\theta\right] + \left[U_0^2\left(1+\frac{a^2}{r^2}\right)^2\sin^2\theta\right] \\
&= U_0^2\left[\cos^2\theta\left(1-\frac{2a^2}{r^2}+\frac{a^4}{r^4}\right)+\sin^2\theta\left(1+\frac{2a^2}{r^2}+\frac{a^4}{r^4}\right)\right] \\
&= U_0^2\left[\cos^2\theta-\frac{2a^2}{r^2}\cos^2\theta+\frac{a^4}{r^4}\cos^2\theta+\sin^2\theta+\frac{2a^2}{r^2}\sin^2\theta+\frac{a^4}{r^4}\sin^2\theta\right] \\
&= U_0^2\left[(\cos^2\theta+\sin^2\theta)+\frac{a^4}{r^4}(\cos^2\theta+\sin^2\theta)+\frac{2a^2}{r}(\sin^2\theta-\cos^2\theta)\right] \\
\Rightarrow V^2 &= U_0^2\left[1+\frac{a^4}{r^4}+\frac{2a^2}{r^2}(\sin^2\theta-\cos^2\theta)\right]\,. \qquad (5.10)
\end{aligned}
$$

Figure 5.2 is very similar to the previous image we have seen of flow around a cylinder, except there is a different style of streamline in this image. Since we have already assumed that flow is going from left to right, we no longer need to keep showing the arrows in our images. The way these streamlines bunch up around the cylinder, we can get a more complete picture of what is happening in the steady flow.

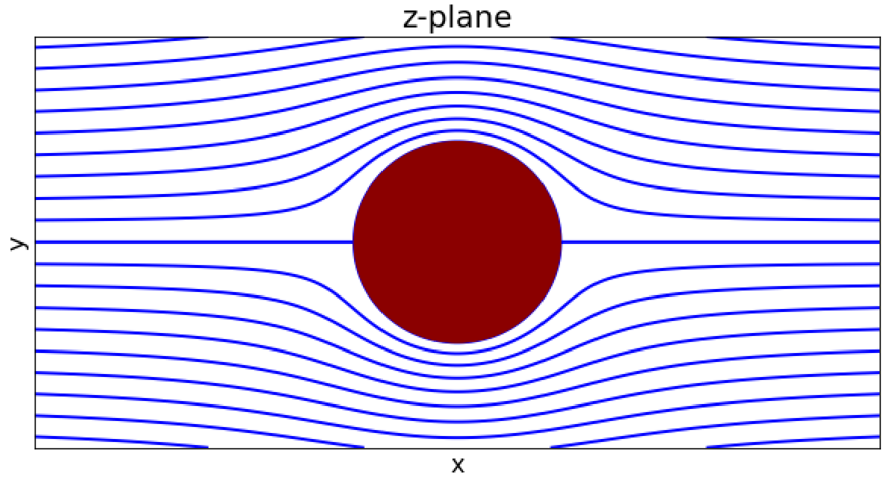


Figure 5.2: Uniform flow past a stationary cylinder that is centred at origin, with $U_0 = 1$ and $\mu = 6$.

5.2 Uniform Flow Past a Cylinder with Circulation

In Section 4.4, we saw how we can superimpose two potential flows to create a new flow and in this section we will be doing the same again. This time we will superimpose the line vortex from Section 4.3 onto the work that was done above in Section 5.1. From equations (4.16) and (5.1), we have the following velocity potentials

$$
\begin{aligned}
\Phi_1 &= U_0\left(r+\frac{a^2}{r}\right)\cos\theta\,; \\
\Phi_2 &= \frac{\Gamma}{2\pi}\theta\,,
\end{aligned}
$$

and since these are potential flows we can add them together to obtain

$$\Phi = \Phi_1 + \Phi_2 = U_0 \left(r + \frac{a^2}{r} \right) \cos\theta + \frac{\Gamma}{2\pi}\theta \,. \tag{5.11}$$

We can then calculate the velocity profile

$$\mathbf{v} = U_0 \left(1 - \frac{a^2}{r^2} \right) \cos\theta \, \mathbf{e}_r + \frac{1}{r} \left[-U_0 \left(r + \frac{a^2}{r} \right) \sin\theta + \frac{\Gamma}{2\pi} \right] \mathbf{e}_\theta$$
$$\Rightarrow \mathbf{v} = U_0 \left(1 - \frac{a^2}{r^2} \right) \cos\theta \, \mathbf{e}_r - \left[U_0 \left(1 + \frac{a^2}{r^2} \right) \sin\theta - \frac{\Gamma}{2\pi r} \right] \mathbf{e}_\theta \,,$$

and it follows that we can find the stream function

$$\frac{\partial \Psi}{\partial \theta} = U_0 \left(r - \frac{a^2}{r} \right) \cos\theta \tag{5.12}$$
$$\frac{\partial \Psi}{\partial r} = U_0 \left(1 + \frac{a^2}{r^2} \right) \sin\theta - \frac{\Gamma}{2\pi r} \tag{5.13}$$

by integrating (5.12) with respect to θ we get

$$\Psi_1 = U_0 \left(r - \frac{a^2}{r} \right) \sin\theta + f(r) \tag{5.14}$$

and then integrating (5.13) with respect to r, we obtain

$$\Psi_2 = U_0 \left(r - \frac{a^2}{r} \right) \sin\theta - \frac{\Gamma}{2\pi} \ln r + g(\theta) \tag{5.15}$$

thus, combining equations (5.14) and (5.15), we have the stream function

$$\Psi = U_0 \left(r - \frac{a^2}{r} \right) \sin\theta - \frac{\Gamma}{2\pi} \ln r + c \,. \tag{5.16}$$

Now, we have a complex potential

$$F(z) = U_0 \left(r + \frac{a^2}{r} \right) \cos\theta + \frac{\Gamma}{2\pi}\theta + i \left[U_0 \left(r - \frac{a^2}{r} \right) \sin\theta - \frac{\Gamma}{2\pi} \ln r \right] ,$$

then, we can perform some manipulation to obtain the following

$$\begin{aligned} F(z) &= U_0 \left(r + \frac{a^2}{r} \right) \cos\theta + i \left(r - \frac{a^2}{r} \right) \sin\theta + \frac{\Gamma}{2\pi}\theta - i\frac{\Gamma}{2\pi} \ln r \\ &= U_0 \, r \cos\theta + U_0 \frac{a^2}{r} \cos\theta + ir \sin\theta - i\frac{a^2}{r} \sin\theta + \frac{\Gamma}{2\pi}\theta - i\frac{\Gamma}{2\pi} \ln r \\ &= U_0 \, r(\cos\theta + i\sin\theta) + U_0 \frac{a^2}{r}(\cos\theta - i\sin\theta) + \frac{\Gamma}{2\pi}(\theta - i\ln r) \\ &= U_0 \, re^{i\theta} + U_0 \frac{a^2}{r} e^{-i\theta} - i\frac{\Gamma}{2\pi}(\ln r + i\theta) \end{aligned}$$
$$\Rightarrow F(z) = U_0 \left(z + \frac{a^2}{z} \right) - i\frac{\Gamma}{2\pi} \ln z \,. \tag{5.17}$$

This result now confirms a lot of what we have done up to this point. Since it is indeed the superposition of the complex potentials of uniform flow past a cylinder (5.1) and a line vortex (4.18). Which, as we have already covered, is actually the superposition of uniform flow, a dipole, and a line vortex.

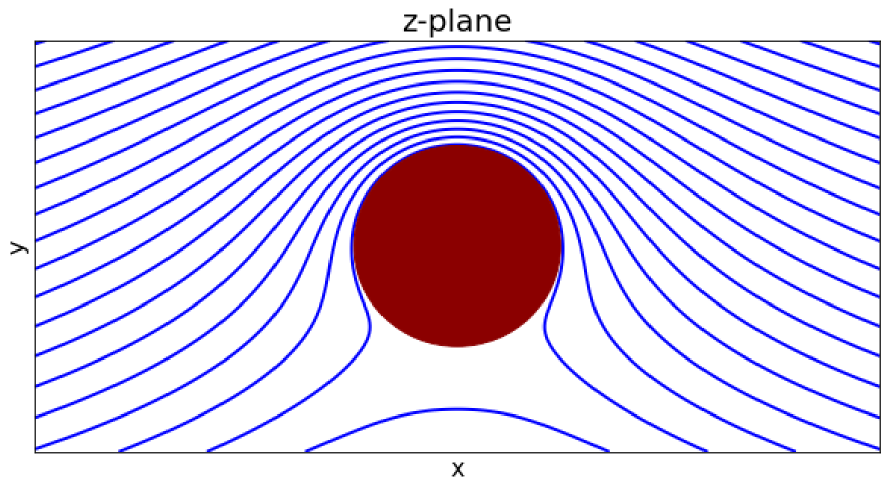


Figure 5.3: The superposition of uniform flow, dipole and line vortex to create flow around a cylinder with circulation. Uniform flow is parallel to horizontal axis, $\alpha = 0$, with the circulation $\Gamma = 10$.

Now we can find the complex velocity by differentiating with respect to z

$$\frac{dF}{dz} = U_0\left(1 - \frac{a^2}{z^2}\right) - i\frac{\Gamma}{2\pi z}\,. \tag{5.18}$$

5.3 Stagnation Points

Stagnation points are points in a flow where $\mathbf{v} = 0$. These are useful in finding velocity, and therefore pressure, at a point in the flow. We use stagnation points in order to find the velocity at one point of the flow, which can then be used to find the velocity at another point of the flow.

In this section we will be looking at the stagnation points when we have uniform flow and no circulation, then we will see the effect on the stagnation points when we introduce circulation. In Section 7 we will have a deeper look into the relevance of stagnation points.

Stagnation Points in Uniform Flow

In Section 5.1, we obtained the velocity profile for uniform flow past a stationary cylinder. Which is

$$\mathbf{v} = U_0\left(1 - \frac{a^2}{r^2}\right)\cos\theta\,\mathbf{e}_r - U_0\left(1 + \frac{a^2}{r^2}\right)\sin\theta\,\mathbf{e}_\theta$$

since stagnation points are when $\mathbf{v} = 0$, then we must have

$$v_r = U_0\left(1 - \frac{a^2}{r^2}\right)\cos\theta = 0 \tag{5.19a}$$

$$v_\theta = -U_0\left(1 + \frac{a^2}{r^2}\right)\sin\theta = 0 \tag{5.19b}$$

equation (5.19b) is satisfied when $\sin\theta = 0$, therefore when $\theta = n\pi$, $(n \in \mathbb{Z})$. Since we are dealing with a circle, then it is satisfied when $\theta = 0$ and $\theta = \pi$. When $\theta = 0$ and $\theta = \pi$, $\cos\theta \neq 0$. Thus equation (5.19a) is satisfied when

$$\begin{aligned}\left(1 - \frac{a^2}{r^2}\right) &= 0\\ \Rightarrow \frac{a^2}{r^2} &= 1\\ \Rightarrow r &= a\,.\end{aligned}$$

Hence, the stagnation points are at $(a, 0)$ and (a, π), where a is equal to the radius of the circle. In Figure 5.4, the stagnation points are marked by the green dots.

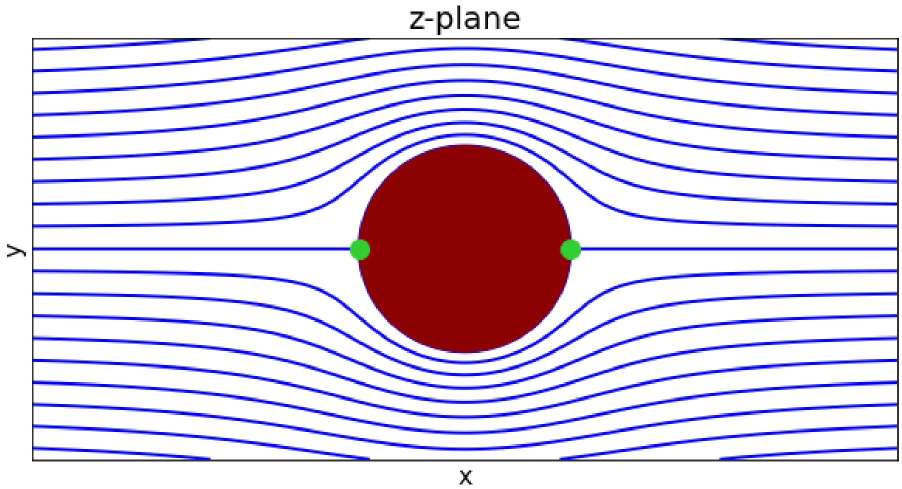


Figure 5.4: Stagnation points in uniform flow around a stationary cylinder.

Since the flow is a superposition of uniform flow and dipole. We need to calculate the radius of the dipole, which depends on the strength of both the uniform flow and the dipole. This can be solved using the same method as above. The velocity profile is simply the superposition of both velocity profiles. We therefore have

$$\mathbf{v} = \left(\frac{\mu}{2\pi r^2}\cos\theta + U_0\right)\mathbf{e}_r - \frac{\mu}{2\pi r^2}\sin\theta\,\mathbf{e}_\theta\,,$$

then, since stagnation points are when $\mathbf{v} = 0$, we need to satisfy the following two components

$$v_r = \left(\frac{\mu}{2\pi r^2}\cos\theta + U_0\right) = 0 \tag{5.20a}$$

$$v_\theta = -\frac{\mu}{2\pi r^2}\sin\theta = 0 \tag{5.20b}$$

again, equation (5.20b) is satisfied when $\theta = 0$ and $\theta = \pi$. Thus $\cos\theta = \pm 1$ in equation (5.20a), and using $\cos\theta = -1$ we have:

$$\begin{aligned} -\frac{\mu}{2\pi r^2} + U_0 &= 0 \\ U_0 &= \frac{\mu}{2\pi r^2} \\ r^2 &= \frac{\mu}{2\pi U_0} \\ \Rightarrow r &= \sqrt{\frac{\mu}{2\pi U_0}} \,. \end{aligned} \tag{5.21}$$

You may notice that equation (5.21) is the rearranged formula that we used to show that the superposition of uniform flow and a dipole is the solution to uniform flow around a cylinder back in Section 4.4.2. It is also the formula that is used to produce the dark red circles on all the illustrations with flow around a cylinder. Hence, the stagnation points in Figure 5.4 are at:

$$r = a = \pm\sqrt{\frac{\mu}{2\pi U_0}} \,.$$

Stagnation Points with Circulation

The previous method in using the velocity profile to find the stagnation points, was similar to the method used in the Fluid Dynamics module. This section follows the method used by Hughes and Brighton (1999), supported by calculations filling the gaps and confirming what had been produced. In fact, there is a typographical error in their work on this concerning equation (5.22), which was observed during the research.

We will solve it by using the complex velocity of flow around a cylinder with circulation (5.18). Which is

$$\frac{dF}{dz} = U_0\left(1 - \frac{a^2}{z^2}\right) - i\frac{\Gamma}{2\pi z} \,, \tag{5.22}$$

now, since a stagnation point is when $\mathbf{v} = 0$, we have

$$U_0\left(1 - \frac{a^2}{z^2}\right) - i\frac{\Gamma}{2\pi z} = 0 \,,$$

we can now expand the brackets and rearrange

$$\begin{aligned} 0 &= U_0\left(1 - \frac{a^2}{z^2}\right) - i\frac{\Gamma}{2\pi z} \\ &= U_0 - U_0\frac{a^2}{z^2} - i\frac{\Gamma}{2\pi z} \\ &= U_0 z^2 - U_0 a^2 - i\frac{\Gamma}{2\pi} z \\ \Rightarrow & z^2 - i\frac{\Gamma}{2\pi U_0} z - a^2 = 0 \,, \end{aligned}$$

which is in the form of a quadratic. We can use the quadratic formula to solve for z

$$
\begin{aligned}
z &= \frac{i\frac{\Gamma}{2\pi U_0} \pm \sqrt{\left(i\frac{\Gamma}{2\pi U_0}\right)^2 - 4(1)(-a^2)}}{2(1)} \\
&= i\frac{\Gamma}{4\pi U_0} \pm \frac{1}{2}\sqrt{-\frac{\Gamma^2}{4\pi^2 U_0^2} + 4a^2} \\
&= i\frac{\Gamma}{4\pi U_0} \pm \frac{1}{2}\sqrt{4a^2\left(1 - \frac{\Gamma^2}{16\pi^2 a^2 U_0^2}\right)} \\
&= i\frac{\Gamma}{4\pi U_0} \pm \frac{2a}{2}\sqrt{1 - \frac{\Gamma^2}{16\pi^2 a^2 U_0^2}} \\
\Rightarrow z &= a\left(i\frac{\Gamma}{4\pi a U_0} \pm \sqrt{1 - \frac{\Gamma^2}{16\pi^2 a^2 U_0^2}}\right). \qquad (5.23)
\end{aligned}
$$

which looks quite complicated, but has an important use when we come to look at lift calculations in Section 7. For now, this equation tells us the location of the two stagnation points when there is circulation. This can be seen in Figure 5.5.

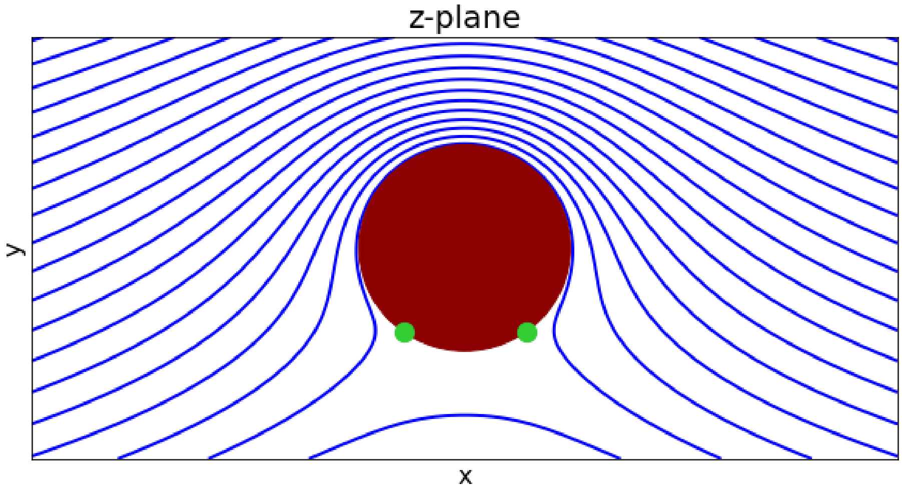


Figure 5.5: Stagnation points on a cylinder with circulation.

Similar to the previous section, since we are using the superposition of the complex potentials of uniform flow (4.3), a dipole (4.23), and a line vortex (4.18). We need to use the superposition of these to calculate the radius of the cylinder and therefore the locations of the stagnation points. We have

$$F(z) = U_0 z + \frac{\mu}{2\pi z} - \frac{\Gamma}{2\pi}\ln z$$

differentiating with respect to z, and recalling that stagnation points are when $\mathbf{v} = 0$, we see that

$$\frac{dF}{dz} = U_0 - \frac{\mu}{2\pi z^2} - \frac{\Gamma}{2\pi z} = 0$$

now rearranging we find

$$
\begin{aligned}
U_0 z^2 - i\frac{\Gamma}{2\pi} z - \frac{\mu}{2\pi} &= 0 \\
\Rightarrow z^2 - i\frac{\Gamma}{2\pi U_0} z - \frac{\mu}{2\pi U_0} &= 0
\end{aligned}
$$

which is another quadratic and we can use the quadratic formula to solve, again. As follows

$$
\begin{aligned}
z &= i\frac{\Gamma}{4\pi U_0} \pm \frac{1}{2}\sqrt{\left(-i\frac{\Gamma}{2\pi U_0}\right)^2 - 4(1)\left(-\frac{\mu}{2\pi U_0}\right)} \\
&= i\frac{\Gamma}{4\pi U_0} \pm \frac{1}{2}\sqrt{-\left(\frac{\Gamma}{2\pi U_0}\right)^2 + \left(\frac{4\mu}{2\pi U_0}\right)} \\
&= i\frac{\Gamma}{4\pi U_0} \pm \sqrt{\frac{1}{4}\left(\frac{4\mu}{2\pi U_0}\right) - \frac{1}{4}\left(\frac{\Gamma}{2\pi U_0}\right)^2} \\
\Rightarrow z &= \pm\sqrt{\left(\frac{\mu}{2\pi U_0}\right) - \left(\frac{\Gamma}{4\pi U_0}\right)^2} + i\frac{\Gamma}{4\pi U_0} \,,
\end{aligned}
$$

notice that the radius squared is the formula for the radius of the dipole in uniform flow (5.21). Therefore, we have

$$
z = \pm\sqrt{r^2 - \left(\frac{\Gamma}{4\pi U_0}\right)^2} + i\frac{\Gamma}{4\pi U_0}
$$

which is the equation used to find the stagnation points in the superposition of uniform flow, a dipole, and a line vortex. You may notice that this is the same as equation (5.23), when $r = a$, produced by Hughes and Brighton (1999). Thus, this equation has the coordinates:

$$
\begin{gathered}
\left(+\sqrt{r^2 - \left(\frac{\Gamma}{4\pi U_0}\right)^2}, \frac{\Gamma}{4\pi U_0}\right) ; \\
\left(-\sqrt{r^2 - \left(\frac{\Gamma}{4\pi U_0}\right)^2}, \frac{\Gamma}{4\pi U_0}\right) .
\end{gathered}
$$

6 Flow Around Joukowski Aerofoil

We have finally made it to aerofoils! We will begin with a very important complex variables topic for this project, called *conformal mapping*. We then will proceed onto to the *Joukowski Transformation*, looking at some basic transformations. Finishing the section with finally looking at flow around a cylinder, transformed into an aerofoil.

6.1 Conformal Mapping

The use of a conformal mapping is vital for this project. The main feature of conformal mapping is that they enable a geometric approach to complex analysis. A complex function

$$\zeta = f(z) = u(x,y) + iv(x,y)\,,$$

where $\zeta = \alpha + i\beta$, is called a mapping $\zeta = f(z)$. It maps from the complex z-plane into the complex ζ-plane. This enables some problems to be simplified; by having a problem with a simple solution in the z-plane, it can solve a complicated problem in the ζ-plane using a mapping.

If a mapping has the following two properties:

- the mapping preserves the magnitude of angles;
- the orientation of which the measurement is taken is consistent;

then the mapping is said to be *conformal.*

The following theory is fundamental to conformal mapping, and to this project, as stated in Spiegel *et al.* (2009):

Theorem 6.1. *If $f(z)$ is analytic and $f'(z) \neq 0$ in a given region, then the mapping $\zeta = f(z)$ is conformal at all points in that region.*

Which is another reason why analytic functions, shown in section 3.1, are so important to this project. In the next section, we will see that the Joukowski Transformation satisfies Theorem 6.1.

Another important aspect of conformal mapping is that if a function is harmonic in the z-plane, then the corresponding function in the ζ-plane is also harmonic (Kreyszig, 2011).

6.2 Joukowski Transformation

A special kind of conformal mapping is the *Joukowski Transformation.* It was discovered by Nikolai Joukowski at the beginning of the 20th Century. The transformation is defined as

$$\zeta = z + \frac{a^2}{z}\,, \tag{6.1}$$

where $\zeta = \alpha + i\beta$. We will eventually use equation (6.1) to easily transform the flow around a cylinder to flow around an aerofoil. We can prove that the Joukowski Transformation is a conformal mapping by checking that it satisfies Theorem 6.1.

Proof. We can easily see that $f'(z) \neq 0$. Since

$$\zeta'(z) = 1 - \frac{a^2}{z^2}\,,$$

Satisfying that condition. Except at the points $z = \pm 1$ (when $a = 1$, as I have used in this project), which are known as critical points. Critical points are the only points where the streamlines, $\Psi =$ constant, and the equipotential lines, $\Phi =$ constant, do not intersect at right angles (Kreyszig, 2011).

Now to prove that the mapping is analytic, using the Cauchy-Riemann equations (3.1). Firstly, we need to split the equation into its real and imaginary parts. As follows

$$
\begin{aligned}
\zeta &= (x+iy) + \frac{a^2}{(x+iy)} \\
&= x + iy + \frac{a^2}{x+iy} \cdot \frac{x-iy}{x-iy} \\
&= x + iy + \frac{a^2(x-iy)}{x^2+y^2} \\
&= \frac{x+iy}{1} + \frac{a^2(x-iy)}{x^2+y^2} \\
&= \frac{(x+iy)(x^2+y^2) + a^2(x-iy)}{x^2+y^2} \\
&= \frac{x^3 + xy^2 + ix^2y + iy^3 + a^2x - ia^2y}{x^2+y^2} \\
\Rightarrow \zeta &= \frac{x^3 + xy^2 + a^2x}{x^2+y^2} + i\frac{y^3 + x^2y - a^2y}{x^2+y^2}\,.
\end{aligned}
$$

now to satisfy the first part of the C-R equations

$$
\begin{aligned}
\frac{\partial u}{\partial x} &= \frac{(3x^2+y^2+a^2)(x^2+y^2) - (x^3+xy^2+a^2x)(2x)}{(x^2+y^2)^2} \\
&= \frac{[3x^4 + x^2y^2 + a^2x^2 + 3x^2y^2 + y^4 + a^2y^2] - [2x^4 + 2x^2y^2 + 2a^2x^2]}{(x^2+y^2)^2} \\
\Rightarrow \frac{\partial u}{\partial x} &= \frac{x^4 + y^4 + 2x^2y^2 + a^2y^2 - a^2x^2}{(x^2+y^2)^2}
\end{aligned}
$$

and

$$
\begin{aligned}
\frac{\partial v}{\partial y} &= \frac{(3y^2+x^2-a^2)(x^2+y^2) - (y^3+x^2y-a^2y)(2y)}{(x^2+y^2)^2} \\
&= \frac{[3y^4 + x^2y^2 - a^2y^2 - a^2x^2 + x^4 + 3y^2x^2] - [2y^4 + 2x^2y^2 - 2a^2y^2]}{(x^2+y^2)^2} \\
\Rightarrow \frac{\partial v}{\partial y} &= \frac{x^4 + y^4 + 2x^2y^2 + a^2y^2 - a^2x^2}{(x^2+y^2)^2}
\end{aligned}
$$

therefore

$$
\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}\,.
$$

Now, for the second part of the C-R equations

$$
\begin{aligned}
\frac{\partial v}{\partial x} &= \frac{(2xy)(x^2+y^2)-(y^3+x^2y-a^y)(2x)}{(x^2+y^2)^2} \\
&= \frac{2x^3+2xy^3-[2xy^3+2x^3y-2a^2xy]}{(x^2+y^2)^2} \\
\Rightarrow \frac{\partial v}{\partial x} &= \frac{2a^2xy}{(x^2+y^2)^2}
\end{aligned}
$$

and

$$
\begin{aligned}
\frac{\partial u}{\partial y} &= \frac{(2xy)(x^2+y^2)-(x^3+xy^2+a^2x)(2y)}{(x^2+y^2)^2} \\
&= \frac{2x^3y+2xy^3-[2x^3y+2xy^3+2a^2xy]}{(x^2+y^2)^2} \\
\Rightarrow \frac{\partial v}{\partial x} &= -\frac{2a^2xy}{(x^2+y^2)^2}
\end{aligned}
$$

therefore

$$
\frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}\,.
$$

Thus, the Cauchy-Riemann equations are satisfied and therefore both conditions of Theorem 6.1 are now satisfied. Hence, the Joukowski Transformation is indeed a conformal mapping. □

In the following section we will explore more about the Joukowski Transformation alongside python output to get a explicit demonstration of the transformation in action.

Joukowski Transformations in Python

The Joukowski Transformation is intended to be used on a circle. In the following examples, notice that when the circle goes through $z = \pm a$ the mapping at that point is a flat line; when the circle encloses $z = \pm a$, but does not touch the point, then the mapping results in a rounded edge. When $z = \pm a$, these are the critical points that were mentioned in the previous section, and are the points in which the mapping is not conformal.

In the following figures, there are a couple of items to look out for. On each image you will see two blue dots, these are set at $z = \pm a$ on the z-plane and at $\zeta = \pm 2a$ on the ζ-plane. This will give an excellent indication of the effects of the critical points. There is also a green dot plotted on each of the images of the z-plane, this dot marks the centre of the circle, showing what the effect the location of the circle has on the shape.

When $R = a$, in other words when the radius, R, of the circle is equal to the constant, a, in equation (6.1), and is centred at the origin, then the transformation creates a flat plate that goes from $-2a$ to $2a$ (Acheson, 2003). As shown in Figure 6.1, where the flat plate goes from -2 to 2. Noting that the circle goes through both $z = \pm a$ and the mapping results in two flat edges.

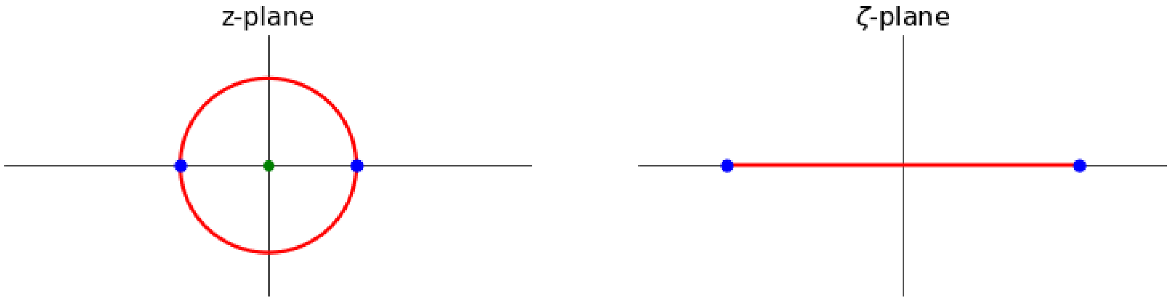


Figure 6.1: The unit circle in the z-plane mapped onto the ζ-plane, using the Joukowski Transformation, where $R = a = 1$.

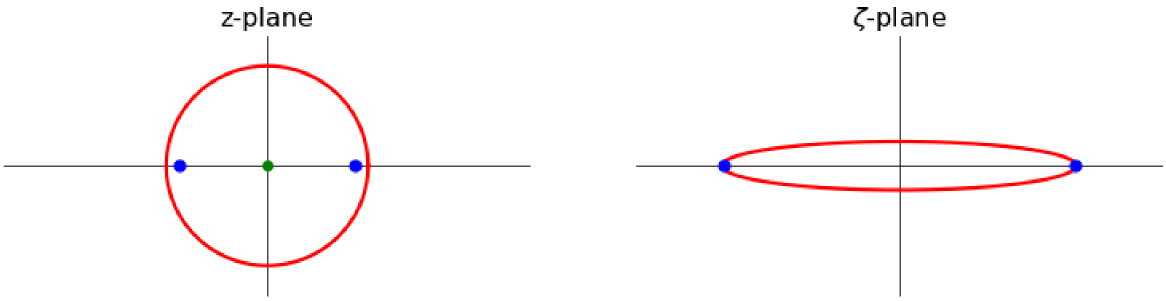


Figure 6.2: A circle centred at the origin, with radius $R = 1.15$ and $a = 1$, on the z-plane mapped into an ellipse on the ζ-plane.

If we make the radius of the circle larger than the constant in Joukowski's transformation ($R > a$), while still centred at the origin, it causes the circle to be mapped into an ellipse. Notice that the circle does not go through either point ($a = \pm 1$) and therefore the mapping has rounded edges. As shown in Figure 6.2.

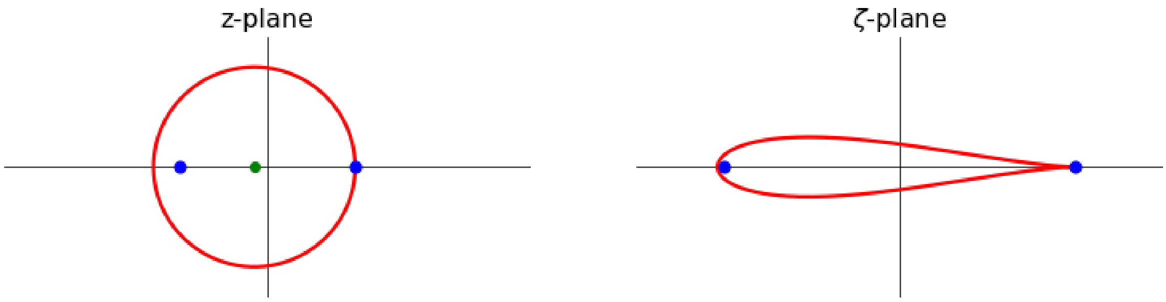


Figure 6.3: A Circled centred at $(-0.15, 0)$, with radius $R = 1.15$ and $a = 1$, on the z-plane mapped onto the ζ-plane creating a symmetric aerofoil.

Now, if we shift the centre of the circle along the real axis, then we will start to see the aerofoil shape being created. When the centre of the circle located on the real axis, this will transform into a symmetric aerofoil. In Figure 6.3 the circle has radius $R > a$, which has been shifted along the real axis by $x = a - R$. This ensures that the circle passes through the points $z = a$. Therefore enclosing $z = -a$, resulting in an aerofoil shape with a rounded nose and a sharp tail.

Finally, if we now move the circle slightly up so that the circle still enclosed $z = -a$ and goes through the point $z = a$, then we will see that the corresponding mapping is the cross-cut of a stereotypical aerofoil shape. The point where $\zeta = 2a$ is actually a singularity, as we will see in the following section, which can be removed using an appropriate amount of circulation (Hughes and Brighton, 1999).

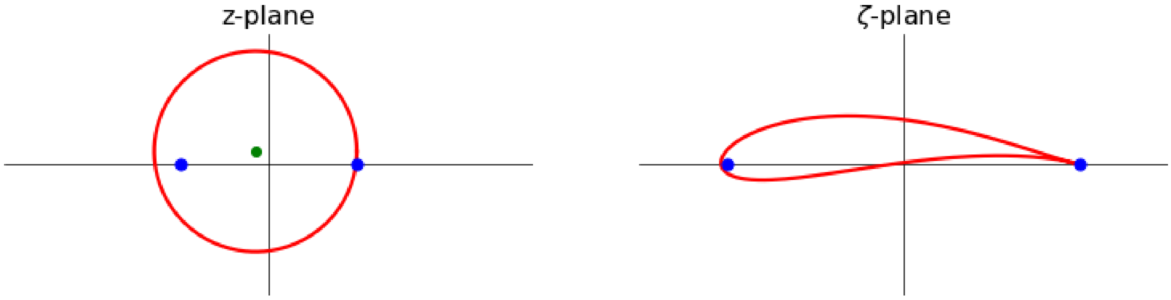


Figure 6.4: A Circled centred at $(-0.15, 0.1)$, with radius $r = 1.15$ and $a = 1$, on the z-plane mapped into a curved aerofoil on the ζ-plane.

If we had shifted the circle along the real axis in the positive direction, it would have also resulted in an aerofoil, but it would be facing the opposite direction. Provided the criteria mentioned above is still met. Since we are using flow that is flowing from left to right, we will continue to face the aerofoil in the direction of the incoming flow. See appendix A.2 to view the code used in creating these illustrations.

6.3 Flow Past Aerofoils

In this section we will see various flows around cylinders in the z-plane, which we saw in Section 5, and mapping them onto the ζ-plane using the Joukowski Transformation. We will see standard uniform flow around the objects, then add some circulation to see the effects of it, and finish the section with the introduction of angle of attack. Which, of course, is a vital aspect to an aeroplane taking off.

6.3.1 Flow Past an Aerofoil

Here we will show uniform flow past a cylinder and flow past a cylinder with circulation, from Section 5, mapped onto the ζ-plane. Since flow with circulation is just a superposition of line vortex, then we will solve both of them together. Using $\Gamma = 0$ for standard uniform flow past a cylinder. Recall equation (5.17), that uniform flow past a cylinder with circulation has the complex potential

$$F(z) = U_0 \left(z + \frac{a^2}{z} \right) - i\frac{\Gamma}{2\pi} \ln z \,. \tag{6.2}$$

This time, instead of using the already known stream function of the superposition of uniform flow (4.2), a dipole (4.22), and a line vortex (4.17), as we have done previously. Let us work backwards from (6.2). Therefore, using $z = re^{i\theta}$, we have

$$
\begin{aligned}
F(z) &= U_0\left(re^{i\theta} + \frac{a^2}{re^{i\theta}}\right) - i\frac{\Gamma}{2\pi}\ln re^{i\theta} \\
&= U_0\left(r(\cos\theta + i\sin\theta) + \frac{a^2}{r}(\cos\theta - i\sin\theta)\right) - i\frac{\Gamma}{2\pi}(\ln r + i\theta) \\
&= U_0\left(\left(r + \frac{a^2}{r}\right)\cos\theta + i\left(r - \frac{a^2}{r}\right)\sin\theta\right) - i\frac{\Gamma}{2\pi}\ln r + \frac{\Gamma}{2\pi}\theta \\
\Rightarrow F(z) &= U_0\left(r + \frac{a^2}{r}\right)\cos\theta + \frac{\Gamma}{2\pi}\theta + iU_0\left(r - \frac{a^2}{r}\right)\sin\theta - i\frac{\Gamma}{2\pi}\ln r
\end{aligned}
$$

thus, taking the imaginary components, we have the stream function

$$
\Psi = U_0 r\sin\theta - \frac{U_0 a^2}{r}\sin\theta - i\frac{\Gamma}{2\pi}\ln r\,.
$$

Recall the formula for the radius of a dipole in uniform flow (5.21). Squaring both sides we obtain: $a^2 = \mu/2\pi U_0$, then substituting this back into our stream function

$$
\Psi = U_0 r\sin\theta - \frac{\mu}{2\pi r}\sin\theta - i\frac{\Gamma}{2\pi}\ln r\,,
$$

which is our required superimposed stream function. In Figure 6.5 we have the circle centred at $(-0.15, 0)$ in uniform flow, transformed using the Joukowski transformation into a symmetric aerofoil. Since the transformation is a conformal mapping, then all angles are preserved, except for the trailing edge where there is a singularity. Thus the plot shows accurate streamlines around both objects.

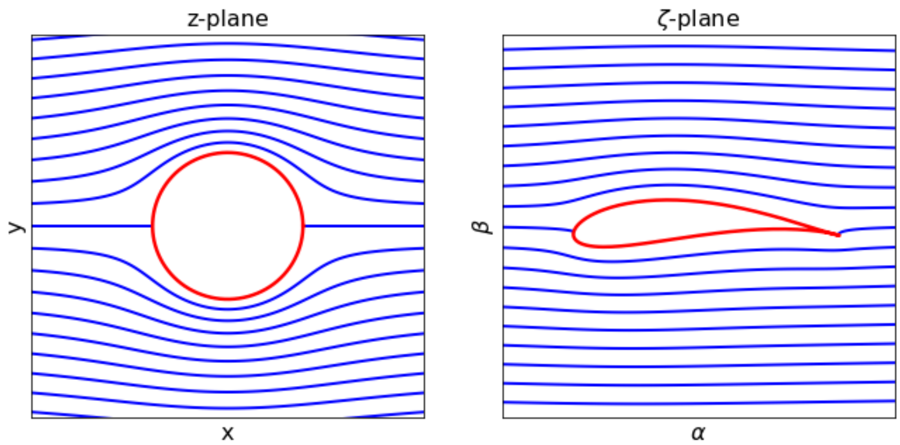


Figure 6.5: Joukowski transformation performed on a stationary cylinder in uniform flow ($\Gamma = 0$).

As mentioned in the previous section, adding sufficient circulation to the flow shifts the singularity off the trailing edge, to behind the aerofoil and therefore it becomes negligible. This can be seen in Figure 6.6, where it is obvious that where there was a streamline coming out of the trailing edge of Figure 6.5, it is no longer there with circulation. Negative circulation is required for lift to be positive in the y-direction (Hughes and Brighton, 1999).

This is essentially the Kutta-Joukowski Theorem, which is beyond the scope of this project but we will have a brief look at it in the Section 7. See appendix A.2 for the code used to produce these illustrations.

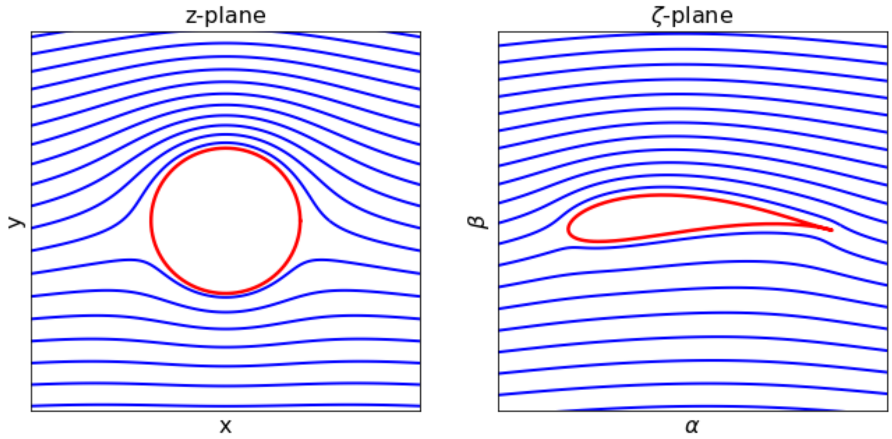


Figure 6.6: Joukowski transformation performed on a cylinder with circulation, $\Gamma = -4$.

6.3.2 Flow Around Aerofoil with Angle of Attack

Alongside circulation, angle of attack is a key ingredient of aerodynamics. In this section we will see the effects of angle of attack. Again, we will use the superposition of a line vortex and set $\Gamma = 0$ when we require no circulation. In Section 4.1 we found the complex potential of uniform flow at angle α to be

$$F(z) = U_0 z e^{-i\alpha}$$

and up to this point, we have been using the complex potential

$$F(z) = U_0 \left(z + \frac{a^2}{z} \right) - i\frac{\Gamma}{2\pi} \ln z$$

which is actually the simplified complex potential for uniform flow, parallel to the horizontal axis, around a cylinder. Therefore, if we want to deal with an angle of attack, we need to combine these and obtain

$$F(z) = U_0 \left(z e^{-i\alpha} + \frac{a^2}{z} e^{i\alpha} \right) - i\frac{\Gamma}{2\pi} \ln z e^{-i\alpha} \,. \tag{6.3}$$

To achieve the angle of attack, we can create another axis that is rotated by α and we can label this axis $z' = x' + iy'$ (Brennen, 2004). Thus $z' = ze^{-i\alpha}$, and a displaced circle is therefore

$$z' = (z - z_0)e^{-i\alpha} \,.$$

now to expand and find the real and imaginary parts, that were used in creating the illustrations

$$\begin{aligned} z' &= ze^{-i\alpha} - z_0 e^{-i\alpha} \\ &= z\cos\alpha - iz\sin\alpha - z_0\cos\alpha + iz_0\sin\alpha \\ &= (x+iy)\cos\alpha - i(x+iy)\sin\alpha - (x_0+iy_0)\cos\alpha + i(x_0+iy_0)\sin\alpha \\ \Rightarrow z' &= (x-x_0)\cos\alpha + (y-y_0)\sin\alpha - i(x-x_0)\sin\alpha + i(y-y_0)\cos\alpha \end{aligned}$$

thus, we have the components

$$
\begin{aligned}
x' &= (x - x_0)\cos\alpha + (y - y_0)\sin\alpha \\
y' &= -(x - x_0)\sin\alpha + (y - y_0)\cos\alpha
\end{aligned}
$$

which were used to create the angle of attack in Figures 6.7 and 6.8. Notice that there is a streamline at the trailing edge of the aerofoil in Figure 6.7. This is the singularity, again. So even with angle of attack, there is still a singularity. At this point, by potential flow theory, the velocity is infinite (Hughes and Brighton, 1999). Which obviously is not the case in real flows.

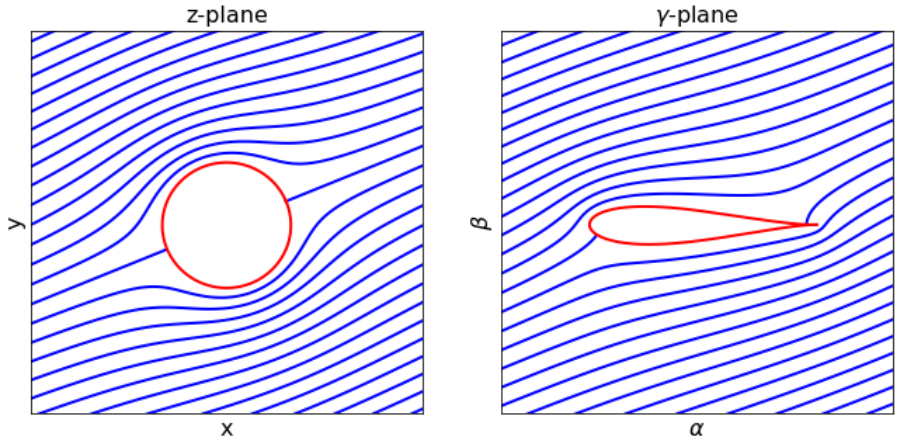


Figure 6.7: Flow around Joukowski aerofoil with angle of attack $\alpha = \frac{\pi}{8}$.

Thus circulation must play an important role in how aeroplanes are able to fly. With the introduction of circulation, the streamlines no longer have a singularity on the wing. As mentioned previously, circulation has resulted in the singularity being shifted behind the aerofoil. As seen in Figure 6.8, which has $\Gamma = -5$. Since there is angle of attack, the value for Γ can be different. Which will be explained briefly in Section 7. See appendix A.2 for code used in this section.

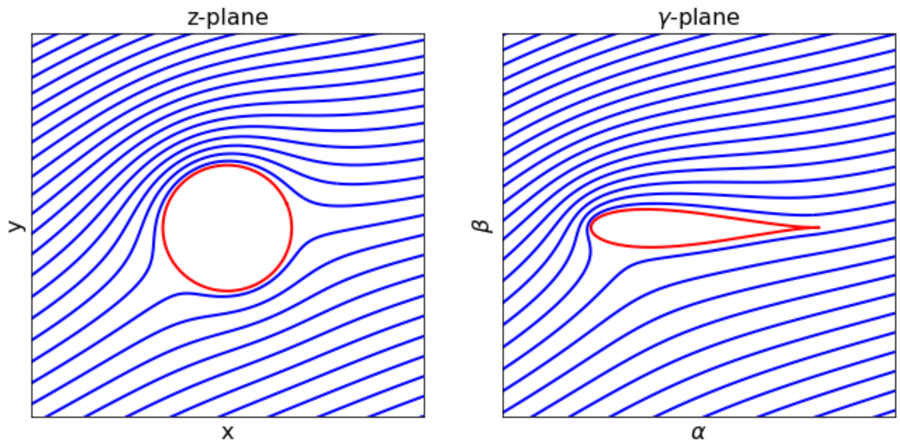


Figure 6.8: Flow around Joukowski aerofoil with circulation and angle of attack $\alpha = \frac{\pi}{8}$.

7 A Brief Look at Lift

This section is the 'icing on the cake' of the project. The main objective was to model flow around a cylinder, then use the Joukowski Transformation. Below is an introduction into lift calculations...

7.1 Kutta-Joukowski Lift Theorem

The full theorem is beyond the scope of this project, but it does tie in very nicely to some of the work done in this project, and would be an obvious next phase of research. In the previous section we saw that there was a singularity at the trailing edge which is removed through circulation. This is the basis of the Kutta-Joukowski Theorem.

In this section we will revisit the stagnation points of flow around a cylinder, calculating their locations with various strengths of circulation, and what connect the stagnation points have with lift.

Location of Stagnation Points

Back in Section 5.3 we found the stagnation points of various flows. Recall that stagnation points are the places where the $\mathbf{v} = 0$. This has important uses in doing lift calculations.

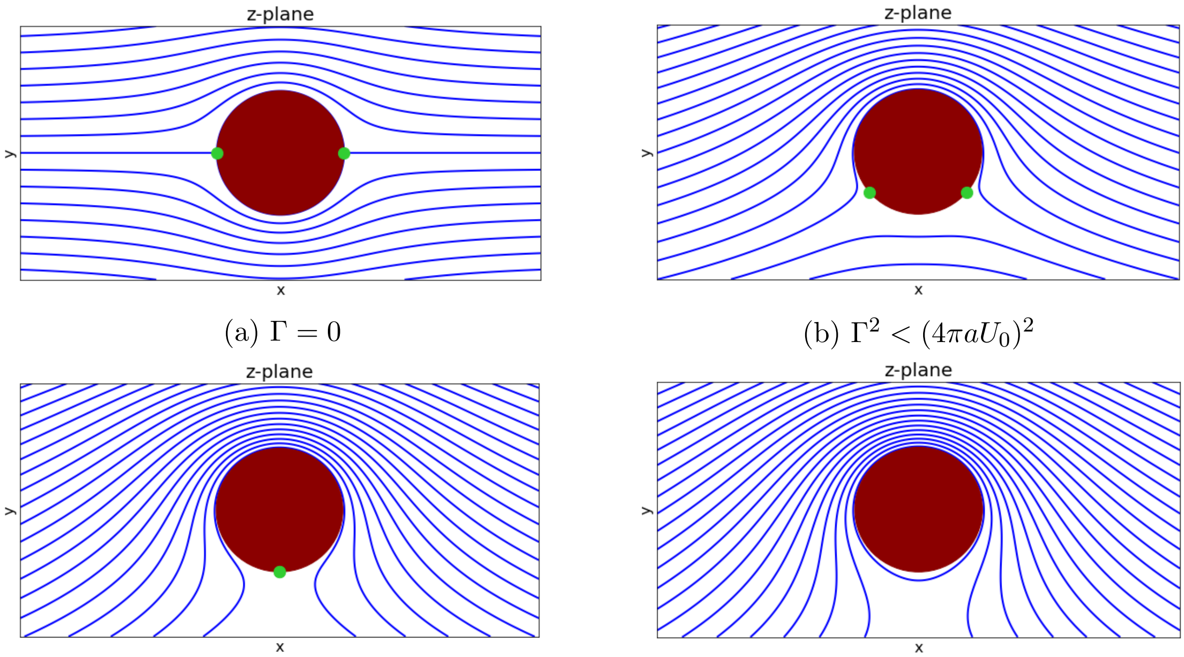
Section 5.3 looked at the stagnation points of a cylinder with circulation. We found the stagnation points to be at

$$z = a\left(i\frac{\Gamma}{4\pi a U_0} \pm \sqrt{1 - \frac{\Gamma^2}{16\pi^2 a^2 U_0^2}}\right).$$

There are four possible cases: (a) $\Gamma = 0$, (b) $\Gamma^2 < (4\pi a U_0)^2$, (c) $\Gamma^2 = (4\pi a U_0)^2$, and (d) $\Gamma^2 > (4\pi a U_0)^2$. Each case has a different pattern, affected by the condition of the stagnation points.

> **Case (a):** When $\Gamma = 0$, we have $z = \pm a$. This means there is no circulation in the flow, therefore the stagnation points are at $r = a$ at $\theta = 0$ and $\theta = \pi$, as shown in Figure 7.1a.
>
> **Case (b):** When $\Gamma^2 < (4\pi a U_0)^2$, we have two stagnation points located on the lower half of the cylinder, as shown in Figure 7.1b.
>
> **Case (c):** When $\Gamma^2 = (4\pi a U_0)^2$, we have one stagnation points located at $\theta = \frac{3\pi}{2}$, as shown in Figure 7.1c.
>
> **Case (d):** When $\Gamma^2 > (4\pi a U_0)^2$, we have a singular stagnation point, which moves down as Γ increases. My code could not produce the location of this stagnation point, but the stagnation point is located underneath the cylinder, not on the radius like the others. As seen in Figure 7.1d.

Hence, if $\Gamma^2 > (4\pi a U_0)^2$ then there is sufficient circulation to move the singularity off the radius of the cylinder and therefore off the trailing edge after the transformation. This is valid for cases where we have flow past a cylinder with circulation, but not with angle of attack. Recall in Section 6.3, we were able to use a different value for Γ when we had angle of attack.



(a) $\Gamma = 0$ (b) $\Gamma^2 < (4\pi a U_0)^2$

(c) $\Gamma^2 = (4\pi a U_0)^2$ (d) $\Gamma^2 > (4\pi a U_0)^2$

Figure 7.1: Different cases of stagnation points.

This is because the formula to find the required circulation when there is an angle of attack is (Brennen, 2004):

$$\Gamma = -4\pi a U_0 \sin\alpha + \beta\,. \tag{7.1}$$

where β is the camber of the foil. Which is not within the scope of this project. If we had more time and space I would have liked to investigate (7.1) further. For now, we will just accept that this is the formula used to find the required circulation to create lift when there is an angle of attack.

Lift on a Cylinder

In the previous section we established that when $\Gamma^2 > (4\pi a U_0)^2$, the singular stagnation point moves off the radius of the cylinder and infact moves downwards away from the cylinder. It is understood that Hughes and Brighton (1999) decided to square both sides because for lift we need $\Gamma < 0$, as stated in Acheson (2003). Thus, for lift to occur with no angle of attack, we require:

$$\Gamma = -4\pi a U_0\,, \tag{7.2}$$

which will stay negative due to a and U_0 both being positive values, by definition. In Figure 7.1, we can see that the combination of circulation with the uniform flow increases the speed over the top of the cylinder. Where there is high speed, the pressure is low, and visa versa. We can then see that below the cylinder, the circulation is opposing the uniform flow creating lower speeds and hence high pressure. High pressure "wants to go" where low pressure is, which then causes lift (Henderson, 2021). The equation for lift on a cylinder, L, is:

$$L = -i\rho U_0 \Gamma\,, \tag{7.3}$$

where ρ is the density of the fluid. Now that we are talking about pressure, this leads us onto *Bernoulli's Equation* for pressure. Which we are going to look at in the following section.

7.2 Pressure Coefficient

In this section we will be looking at the pressure coefficient which is derivable from the famous *Bernoulli's Equation*. The pressure coefficient for a steady, incompressible flow (Brennen, 2004) is defined as:

$$C_p = 1 - \frac{u^2 + v^2}{U_0^2}\,, \tag{7.4}$$

note that stagnation points are when $C_p = 1$. The following graphics show the heat map of pressure due to uniform flow with and without circulation, finishing with the pressure when we have angle of attack and circulation. The accuracy of the following images was mostly confirmed by theoretical information obtained from *Aerospace for Students* (University of Sydney, 2005). Accompanied with all the information obtained during the research for this project.

We start with looking at the case when there is uniform flow around a stationary cylinder. That is, $\alpha = 0$ and $\Gamma = 0$. Recall from Section 5.3 that the stagnation points are located at $\theta = 0$ and $\theta = \pi$ at $r = a$. In Figure 7.2, we can clearly see these stagnation points. We can also see that the pressure is lowest at the top and bottom of the cylinder, confirming that the velocity is highest in those locations.

It is then clear to see that the spread of pressure around the Joukowski aerofoil is quite even. This is what we would expect to see since there is no lift nor movement of the aerofoil.

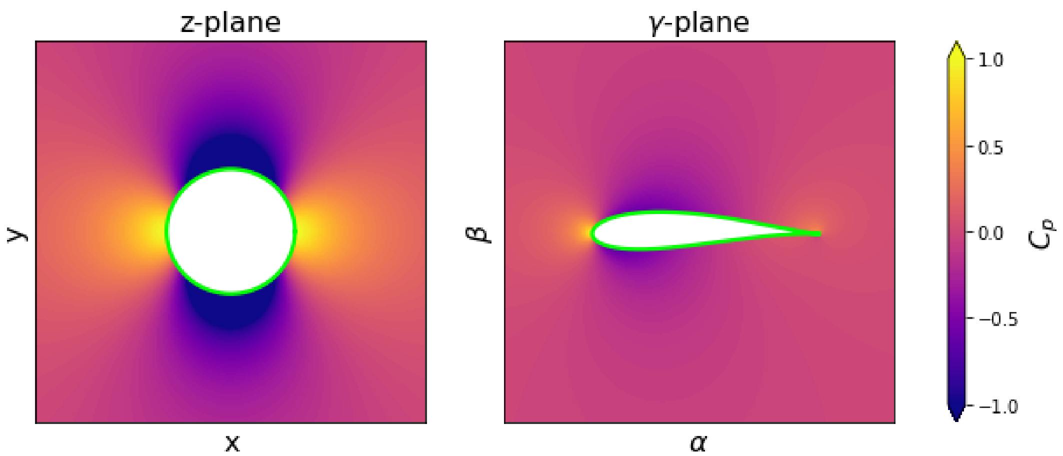


Figure 7.2: The pressure coefficient of uniform flow past a stationary cylinder and the corresponding Joukowski aerofoil.

The next case is when we have circulation but still no angle of attack. That is, $\Gamma = -6$ and $\alpha = 0$, as shown in Figure 7.3. We can straight away see that the stagnation points are where we expect to see them in flow with circulation. With a majority of the flow going over the top of the cylinder with high velocity, causing the huge patch of low pressure. In the aerofoil image, we can clearly see the low pressure above the aerofoil, as would be expected.

In our final illustration we have circulation and angle of attack, as shown in Figure 7.4. That is, with $\Gamma = -6$ and $\alpha = \frac{\pi}{8}$. We can see that the two images are very similar to our illustrations of circulation without angle of attack. The cylinder appears to have the same

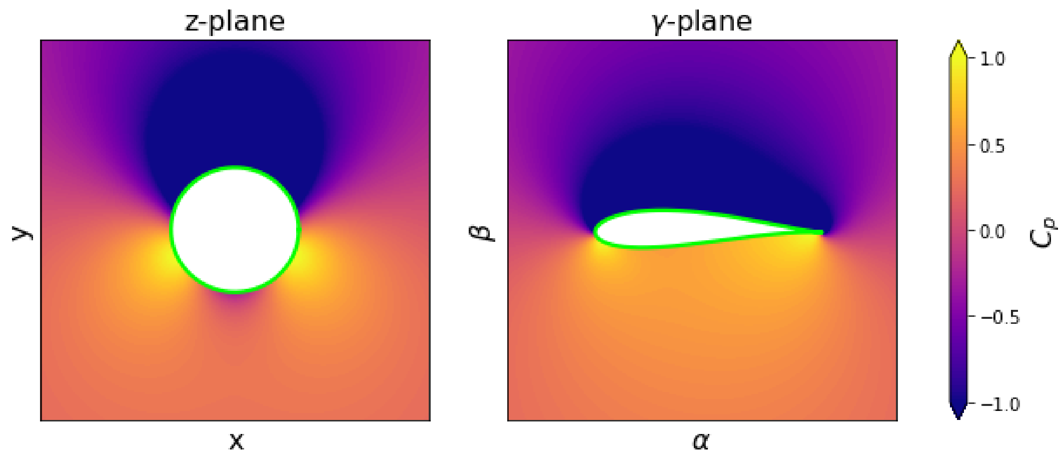


Figure 7.3: The pressure coefficient of uniform flow past a cylinder with circulation and the corresponding Joukowski aerofoil.

stagnation points and similar size area of low pressure above the cylinder. Which we expect to see, since it is just flow around a cylinder at another angle.

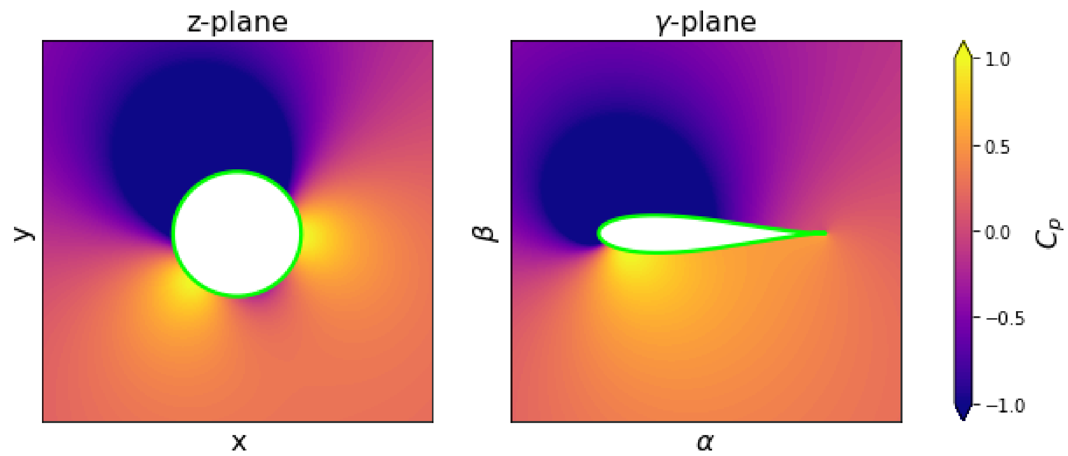


Figure 7.4: The pressure coefficient of uniform flow past a cylinder with circulation and angle of attack, with the corresponding Joukowski aerofoil.

In the pressure map of the aerofoil, it has varied slightly to the image with angle of attack. It seems to be more concentrated at the front of wing, where we would expect to see the lift of an aerofoil. The full code for all the figures produced in this section can be seen in appendix A.3

8 Discussion

In this final section I want to talk about a few aspects of my research and the writing of the project. I will start by concluding the project, discussing different aspects of the project and some of the things I have learned along the way. Followed by some options for further research, and finishing the section with my overall thoughts and reflection of the project.

Conclusion

I managed to complete everything I had set out to do with this project. As mentioned in the introduction, I wanted to cover the fundamentals of fluid dynamics and complex variable theory, then aimed to eventually model flow around a cylinder and transforming into an aerofoil. As well as covering a brief section on lift and pressure, which I planned to investigate if I had time at the end.

In regards to the main body of the project, I am delighted that it flows from one section to the next, with logical connections between them. I extensively covered the fundamental mathematics used throughout this project, and I expect the reader has been provided with sufficient basics to understand the main body.

In the end, I did not get as much done in the lift and pressure section, as I had originally hoped. But, I was very impressed that my code writing process resulted in me making the stunning illustrations with pressure. It was quite a small step with regards to writing the code, opposed to the required hand calculations. I do feel like there is so much information that I could have included, but I will save that conversation for the next section.

If I were to be critical of the project, I mainly wish I had performed more hand calculations with the Joukowski Transformation and most of the work in that section. As it transpired, I found it much easier to create plots that used the transformation rather than doing the hand calculation, which the pioneers of the subject would have done. I attempted a couple of them once I had completed the project, but time restraints did not allow me to get them to a desirable standard.

Further Research

As I have already mentioned a couple of times throughout the project, the next obvious phase of research would be lift calculations. I only very briefly touched on them, but there is so much more that could be covered with it. To begin with, I would derive some of the formulae in Section 7 which I only stated to give a general idea.

These equations for the circulation and lift appear in most of the books I did my research with. Given more time and space I would have researched and investigated further. I have seen that the distribution of pressure around an aerofoil is something that has been researched a lot. Which differs depending on angle of attack, the camber line, and the thickness of an aerofoil.

Another obvious next stage of research would be based on the Joukowski Transformation. I kept it as simple as I could mainly allowed python to perform the calculations. If performing hand calculations on the transformation, there are a few more aspects to take into account. Such as the thickness, which can be controlled the location of the circle in the z-plane; or the camber line, which is a line that stretches from the tail to the nose of the aerofoil, has some important effects when angle of attack is incorporated.

I could also go into a lot more detail but the Kutta-Joukowski Theorem. I barely even stated the theorem, just its use and effects to remove the singularity from the trailing edge.

Which is actually more of by-product of the theorem and an obvious condition to use, since real flows would not have infinite velocity. With the Kutta-Joukowski Theorem, I could have introduced Blasius' Theorem, which can be used to derive the equations for lift.

Something I have not even mentioned in this project, are the effects of viscosity and the boundary layer. Of course, we were using potential flow so I did not need to be discussed in the project, but aerofoil certainly have a boundary layer. Which, I suppose, would lead towards looking at drag calculations, another aspect that has barely been mentioned. I do not know too much more, but it would certainly be an area to investigate further.

On top of all of this, though, almost every topic I have covered within this project has only been the surface. There is so much more within topics such as potential flow theory, harmonic functions, conformal mapping, just to name a few. The opportunities for further research seem broad, but as does the opportunity for a deeper look into the topics we have seen through this project.

Thoughts

To begin with, I found it incredible throughout this project that so much tied together and there are multiple methods and routes I could take to end up with the same result. This became apparent when I discovered what the solution to flow around a cylinder actually is. I had completed the majority of my hand written work for this project when I discovered that it's the superposition of other complex potentials. It was only when it came to writing code that I figured this out! I had performed all hand calculations ignorant to that fact.

When writing up the project I would be performing some checking calculations on the side to confirm some of my results. And so often would I discover something that I had not originally. Such as, in section 5.3, when I calculated the location of stagnation points in the superposition of uniform flow, a dipole, and a line vortex, I noticed - on my first draft check through - that it was actually the same equation as the one I had mentioned above, that was acquired from Hughes and Brighton (1999).

At some points it seemed almost too convenient that some things tied together so nicely, or would intertwine with something else. The Cauchy-Riemann equation seem to show up everywhere! I figure this is an attribute of harmonic functions, that once you have one, they seem to connect with everything else.

I am very happy with my final year project choice. I thoroughly enjoyed the exploration of such an interesting area of mathematics. The way that fluid dynamics and complex variables are used in tandem to attempt to solve a very complex problem. I cannot begin to imagine how Joukowski, Kutta, and Lanchester were able to perform some of their calculations over 100 years ago.

I particularly enjoyed the coding aspect of my project - I was very nervous about it. I had been able to complete all coding work we had done so far at UWE when provided, and almost hand fed, to us by lecturers. But to take on this mammoth task on my own and then to come out with the amazing images I produced, I could not be more happy. Because of the work I have done with python throughout this project, it has given me confidence to pursue a career that involves coding.

A Appendix

A.1 Flow Code

This appendix will include all of the code I wrote for simple flows (Section 4) and flow around cylinders (Section 5). Throughout these sections, I was still learning and adapting my skills, I found it easier to write the code working in cartesians. Therefore I had to convert a lot of my calculations to cartesians. Which I will show in the following sections along with talking through some of my process.

All of the following use the same packages and have been set up for a cartesian mesh grid. Which can be seen below:

```
[1]: import math
     import numpy as np
     import matplotlib.pyplot as plt
     %matplotlib inline
```

```
[2]: # Mesh Set-up
     #========================================
     N = 200
     x_s, x_e = -4, 4
     y_s, y_e = -2, 2

     x = np.linspace(x_s, x_e, N)
     y = np.linspace(y_s, y_e, N)

     X, Y = np.meshgrid(x, y)
```

All plots, until otherwise stated, use the following code as its settings. For each section I will only shows the lines used for plotting.

```
[5]: # Plot
     #=========================================
     plt.figure(figsize=(10, 5))

     plt.xlabel('x', fontsize=16)
     plt.ylabel('y', fontsize=16)
     plt.title("z-plane", fontsize=20)
     plt.xlim(x_s, x_e)
     plt.ylim(y_s, y_e)
     plt.xticks([])
     plt.yticks([])
```

Uniform Flow Code

This section concerns the figures created in Section 4.1.

I began my process with creating a uniform flow plot in python. I was able to create a piece of code that enabled me to adapt the direction of flow. Simply, by editing the value of alpha I was able to create plots depicting uniform flow at any angle on the plane.

In these plots I used the matplotlib function `Streamplot`, for which I used the velocity components $u(x,y)$ and $v(x,y)$. Thus, we have

$$u = U_0 \cos\alpha\,; \quad v = U_0 \sin\alpha$$

In the following image I have defined the velocity components in a function, set the parameters for uniform flow and angle of attack, and finished with the call to function:

```
[3]: # Function Defining
     #=========================================
     def uniform_velocity(U, aoa):
         u = U * np.ones( (N,N), dtype=float ) * np.cos(aoa)
         v = U * np.ones( (N,N), dtype=float ) * np.sin(aoa)
         return u, v
```

```
[4]: # Parameters
     #=========================================
     U_0 = 1
     alpha = (1/4) * math.pi

     # Function Calling
     #=========================================
     u, v = uniform_velocity(U_0, alpha)
```

Here we see the streamplot settings:

```
# Streamline Plot
#=========================================
plt.streamplot(X, Y, u, v,
               density=0.75, linewidth=1.75, arrowsize=1.5, arrowstyle='-|>')
```

Source and Sink Code

To create the illustrations in Section 4.2 I needed to convert the equations to cartesian, as follows

$$\begin{aligned}\Psi(r,\theta) &= \frac{m}{2\pi}\theta \\ \Rightarrow \Psi(x,y) &= \frac{m}{2\pi}\arctan\left(\frac{y}{x}\right)\end{aligned}$$

now to differentiate, with respect to x

$$\frac{\partial \Psi}{\partial x} = \frac{m}{2\pi}\left(-\frac{y}{x^2+y^2}\right)$$

and with respect to y

$$\frac{\partial \Psi}{\partial y} = \frac{m}{2\pi}\left(\frac{x}{x^2+y^2}\right)$$

thus, giving us the velocity components:

$$\begin{aligned}u &= \frac{\partial \Psi}{\partial y} = \frac{m}{2\pi}\left(\frac{x}{x^2+y^2}\right) \\ v &= -\frac{\partial \Psi}{\partial x} = \frac{m}{2\pi}\left(\frac{y}{x^2+y^2}\right) .\end{aligned}$$

The following shows the code used to create a source and sink. I have defined the velocity components, set the source and sink strength, then called to the functions. As can be seen below:

```
[3]: # Function Defining
     #========================================
     def ss_velocity(m, X, Y):
         u = m / (2*math.pi) * (X) / ((X)**2 + (Y)**2)
         v = m / (2*math.pi) * (Y) / ((X)**2 + (Y)**2)
         return u,v
```

```
[4]: # Parameters
     #========================================
     mu_source = 5
     mu_sink = -5

     # Function Calling
     #========================================
     u1, v1 = ss_velocity(mu_source, X, Y)
     u2, v2 = ss_velocity(mu_sink, X, Y)
```

In the following screenshots we can see the streamplot and scatter settings for the source and sink:

```python
# Source Plot
#========================================
plt.streamplot(X, Y, u1, v1,
               density=1, linewidth=1.75, arrowsize=1.5, arrowstyle='-|>')
plt.scatter(0, 0, zorder=2,
            color='darkred', s=300, marker='o')
```

```python
# Sink Plot
#========================================
plt.streamplot(X, Y, u2, v2,
               density=1, linewidth=1.75, arrowsize=1.5, arrowstyle='-|>')
plt.scatter(0, 0, zorder=2,
            color='darkred', s=300, marker='o')
```

Line Vortex Code

This section concerns the figures created in Section 4.3. Again, I had to convert to cartesians. As follows:

$$\begin{aligned}\Psi(r,\theta) &= -\frac{\Gamma}{2\pi}\ln r\\ \Leftrightarrow \Psi(x,y) &= -\frac{\Gamma}{2\pi}\ln\left(x^2+y^2\right)^{\frac{1}{2}}\\ &= -\frac{\Gamma}{2\pi}\frac{1}{2}\ln\left(x^2+y^2\right)\\ \Rightarrow \Psi(x,y) &= -\frac{\Gamma}{4\pi}\ln\left(x^2+y^2\right)\end{aligned}$$

it was then straight forward to obtain the velocity components

$$\begin{aligned}u &= \frac{\partial\Psi}{\partial y} = -\frac{\Gamma}{4\pi}\left(\frac{2y}{x^2+y^2}\right)\\ v &= -\frac{\partial\Psi}{\partial x} = -\frac{\Gamma}{4\pi}\left(\frac{2x}{x^2+y^2}\right)\end{aligned}$$

thus, in their simplest form

$$\begin{aligned}u &= -\frac{\Gamma}{2\pi}\left(\frac{y}{x^2+y^2}\right)\\ v &= \frac{\Gamma}{2\pi}\left(\frac{x}{x^2+y^2}\right).\end{aligned}$$

The following screenshot shows the defining of the velocity components, setting the strength of circulation and finishes with calling to the function:

```
[3]: # Define Functions
     #========================================
     def vortex_velocity(gam, X, Y):
         u = ( - gam / (2*math.pi) ) * ( (Y) / ( (X)**2 + (Y)**2 ) )
         v = ( gam / (2*math.pi) ) * ( (X) / ( (X)**2 + (Y)**2 ) )
         return u, v
```

```
[4]: # Parameters
     #========================================
     gamma = 10

     # Function Calling
     #========================================
     u, v = vortex_velocity(gamma, X, Y)
```

This screenshot shows the settings for the streamplot function:

```
# Line Vortex Streamlines
#========================================
plt.streamplot(X, Y, u, v,
               density=1, linewidth=1.75, arrowsize=1.5, arrowstyle='-|>')
```

Superposition Code

This section concerns the figures created in Section 4.4. The conversion to cartesians can be seen as follows:

$$\begin{aligned}\Psi(r,\theta) &= -\frac{\mu}{2\pi}\frac{\sin\theta}{r} \\ \Rightarrow \Psi(x,y) &= -\frac{\mu}{2\pi}\frac{y}{x^2+y^2}\end{aligned}$$

then, using the quotient rule, we obtain the velocity components:

$$\begin{aligned}u &= -\frac{\mu}{2\pi}\left(\frac{x^2-y^2}{(x^2+y^2)^2}\right) \\ v &= -\frac{\mu}{2\pi}\left(\frac{2xy}{(x^2+y^2)^2}\right)\end{aligned}$$

The following screenshot shows the dipole velocity components being defined, followed by its strength and calling to the function:

```
[3]: # Define Functions
     #=======================================
     def dipole_velocity(mu, X, Y):
         u = ( - mu / (2*math.pi) ) * ( ((X)**2 - (Y)**2) / ((X)**2 + (Y)**2)**2 )
         v = ( - mu / (2*math.pi) ) * ( ( 2*(X)*(Y) ) / ((X)**2 + (Y)**2)**2 )
         return u,v
```

```
[4]: # Parameters
     #=======================================
     mu = 6

     # Function Calling
     #=======================================
     u, v = dipole_velocity(mu, X, Y)
```

This screenshot shows the settings for the streamplot function and a dot to mark the centre of the dipole:

```
# Dipole Plot
#=======================================

# Streamlines
plt.streamplot(X, Y, u, v,
               density=1, linewidth=1.75, arrowsize=1.5, arrowstyle='-|>')

# Dipole Centre
plt.scatter(0, 0, zorder=2,
            color='darkred', s=300, marker='o')
```

This is the code used to create the superposition of a dipole and uniform flow. It used the same functions as the dipole and uniform flow, both have been mentioned above. The code to plot uniform flow and a dipole is:

```
# Superposition Plot
#=======================================

#Streamlines
plt.streamplot(X, Y, u_cyl, v_cyl,
               density=1, linewidth=1.75, arrowsize=1.5, arrowstyle='-|>')

# Dipole centre
plt.scatter(0, 0, zorder=2,
            color='darkred', s=300, marker='o')
```

Uniform Flow Around Cylinder Code

In Section 5 I moved from showing the `streamplot` function to the `contour` function. As stated in the main body, I did this because I think the contour function is a better representation of streamlines. That meant that for this section I made functions using velocity components for the streamplot function and made functions using the stream function for the contour function.

The functions used for streamplot are the sames that have been used above. For the contour function I needed to convert the stream function to cartesians, which is:

$$\Psi(x, y) = U_0(y\cos\alpha - x\sin\alpha) - \frac{\mu}{2\pi}\left(\frac{y}{x^2 + y^2}\right)$$

This screenshot shows the defining of function for both the velocity components and the stream functions:

```
[3]: # Define Functions
     #========================================

     # Uniform Flow
     def uniform_velocity(U, aoa):
         u = U * np.ones( (N,N), dtype=float ) * np.cos(aoa)
         v = U * np.ones( (N,N), dtype=float ) * np.sin(aoa)
         return u, v

     def uniform_stream(U, aoa, X, Y):
         psi = ( U * Y * np.cos(aoa) ) - ( U * X * np.sin(aoa) )
         return psi

     # Dipole
     def dipole_velocity(mu, X, Y):
         u = ( - mu / (2*math.pi) ) * ( ((X)**2 - (Y)**2) / ((X)**2 + (Y)**2)**2 )
         v = ( - mu / (2*math.pi) ) * ( ( 2*(X)*(Y) ) / ((X)**2 + (Y)**2)**2 )
         return u, v

     def dipole_stream(mu, X, Y):
         psi = ( -mu / (2*math.pi) ) * ( (Y) / ( (X)**2 + (Y)**2 ) )
         return psi
```

This screenshot shows the adjustable parameters, the calling to functions, and the superposition of functions:

[4]:
```
# Parameters
#=======================================
U_0 = 1
alpha = (0) * math.pi
mu = 6

# Function Calling
#=======================================
u1, v1 = uniform_velocity(U_0, alpha)
u2, v2 = dipole_velocity(mu, X, Y)

psi1 = uniform_stream(U_0, alpha, X, Y)
psi2 = dipole_stream(mu, X, Y)

# Superposition
#=======================================
u, v = (u1 + u2), (v1 + v2)
psi = psi1 + psi2
```

The following screenshots show the settings for the streamplot and contour functions, with the lines of code used to create the red circle:

```
# Streamplot Plot
#======================================

#Streamlines
plt.streamplot(X, Y, u, v,
               density=1, linewidth=1.75, arrowsize=1.5, arrowstyle='-|>')

#Circle
rad = math.sqrt( mu / (2 * math.pi * U_0))
circle = plt.Circle( (0,0), radius=rad, color='darkred', zorder=2)
plt.gca().add_patch(circle)
```

```
# Contour Plot
#======================================

# Contour Lines
plt.contour(X, Y, psi,
            250, colors='blue', linewidths=2, linestyles='solid')

# Circle
rad = math.sqrt( mu / (2 * math.pi * U_0))
circle = plt.Circle( (0,0), radius=rad, color='darkred', zorder=2)
plt.gca().add_patch(circle)
```

Flow Around Cylinder with Circulation Code

In Section 5.2 we started to use only the contour function, therefore only needed the stream function. The stream function in cartesians is:

$$\Psi(x,y) = U_0(y\cos\alpha - x\sin\alpha) - \frac{\mu}{2\pi}\left(\frac{y}{x^2+y^2}\right) - \frac{\Gamma}{4\pi}\ln x^2 + y^2$$

Since the stream functions for uniform flow and dipole are shown above, here is a screenshot of the line vortex stream function:

```
# Line Vortex
def vortex_velocity(gam, X, Y):
    u = ( - gam / (2*math.pi) ) * ( (Y) / ( (X)**2 + (Y)**2 ) )
    v = ( gam / (2*math.pi) ) * ( (X) / ( (X)**2 + (Y)**2 ) )
    return u, v

def vortex_stream(gam, X, Y):
    psi = ( - gam / (4*math.pi) ) * ( np.log( (X)**2 + (Y)**2 ) )
    return psi
```

This screenshot shows the parameters, call to functions and the superposition of the functions:

```
[4]: # Parameters
     #========================================
     U_0 = 1
     alpha = (0) * math.pi

     mu = 6
     gamma = 10

     # Function Calling
     #========================================

     u1, v1 = uniform_velocity(U_0, alpha)
     u2, v2 = dipole_velocity(mu, X, Y)
     u3, v3 = vortex_velocity(gamma, X, Y)

     psi1 = uniform_stream(U_0, alpha, X, Y)
     psi2 = dipole_stream(mu, X, Y)
     psi3 = vortex_stream(gamma, X, Y)

     # Superposition
     #========================================

     u, v = (u1 + u2 - u3), (v1 + v2 - v3)
     psi = psi1 + psi2 - psi3
```

The following screenshot shows the settings for the contour function and the red circle:

```
# Contour Lines
plt.contour(X, Y, psi,
            200, colors='blue', linewidths=2, linestyles='solid')

# Circle
rad = math.sqrt( mu / (2 * math.pi * U_0) )
circle = plt.Circle( (0,0), radius=rad, color='darkred', zorder=2)
plt.gca().add_patch(circle)
```

Stagnation Points Code

In Section 5.3 we calculated the locations of stagnation points. The following lines show the locations. Note that when $\Gamma = 0$, then it just equals the same as equation (5.21).

```
# Stagnation Points
x3_stag, y3_stag = ( math.sqrt(rad**2 - (gamma / (4 * math.pi * U_0))**2),
                     + gamma / (4 * math.pi * U_0))
x4_stag, y4_stag = ( - math.sqrt(rad**2 - (gamma / (4 * math.pi * U_0))**2),
                     + gamma / (4 * math.pi * U_0))

plt.scatter([x3_stag, x4_stag], [y3_stag, y4_stag],
            color='limegreen', s=150, marker='o', zorder=3);
```

A.2 Joukowski Transformation Code

This is all the code created for the Joukowski Transformation, in Section 6.

Joukowski Transformation Code

This screenshot shows the code used to create basic Joukowski transformations, used in Section 6.2. It shows the packages used, the set up figures, defining of functions and the adjustable paramenters.

```
[1]: import math
     import numpy as np
     import matplotlib.pyplot as plt
     %matplotlib inline
```

```
[2]: # Set-up
     #========================================
     N = 200
     x_s, x_e = -3, 3
     y_s, y_e = -1.5, 1.5
```

```
[3]: # Define Functions
     #========================================

     # Circle
     def circle(xc, yc):
         t = np.linspace(0, 2*math.pi, N)

         x = r * np.cos(t)
         y = r * np.sin(t)

         z = (x+xc) + 1j * (y+yc)
         return z

     # Transformation
     def trans(zj, a):
         w = zj + a**2 / zj
         return w
```

```
[4]: # Parameters
     #========================================
     a = 1
     r = 1.15

     x_centre, y_centre = -0.15, 0.15
```

This screenshot shows the calling to functions and the many settings I used to create very pretty looking plots.

```
# Function Calling
#========================================
z = circle(x_centre, y_centre)
w = trans(z, a)

x, y = np.real(z), np.imag(z)
xx, yy = np.real(w), np.imag(w)
```

```
[5]: # PLOTS
#========================================
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(13,3))

# Circle subplot
#========================================
ax1.set_xlabel('x', fontsize=16)
ax1.set_ylabel('y', fontsize=16)
ax1.set_title("z-plane", fontsize=16)
ax1.set_xlim(x_s, x_e)
ax1.set_ylim(y_s, y_e)
ax1.axis('off')

ax1.axhline(y=0, linewidth=0.7, color='k', linestyle='-', zorder=1)
ax1.axvline(x=0, linewidth=0.7, color='k', linestyle='-', zorder=2)
ax1.plot(x, y, color='r', linewidth=2.25)
ax1.scatter(x_centre, y_centre, color='green', zorder=3)
ax1.scatter([-1*a, 1*a], [0, 0], color='b', s=50, zorder=4)

# Joukowski subplot
#========================================
ax2.set_xlabel(r'$\alpha$', fontsize=16)
ax2.set_ylabel(r'$\beta$', fontsize=16)
ax2.set_title(r"$\zeta$-plane", fontsize=16)
ax2.set_xlim(x_s, x_e)
ax2.set_ylim(y_s, y_e)
ax2.axis('off')

ax2.axhline(y=0, linewidth=0.7, color='k', linestyle='-')
ax2.axvline(x=0, linewidth=0.7, color='k', linestyle='-')
ax2.plot(xx, yy, color='r', linewidth=2.25)
ax2.scatter([-2*a, 2*a], [0, 0], color='b', s=50, zorder=4)
```

Aerofoil Code

This section concerns the figures created in Section 6.3.1. In all the remaining pieces of code, I used a circular polar mesh grid. The set-up for the grid is consistent and can be seen here, along with the packages used. And no more need to convert to cartesians!

```
[34]: import math
      import numpy as np
      import matplotlib.pyplot as plt
      %matplotlib inline
```

```
[35]: # Mesh Grid
      #========================================
      N = 200
      R_start, R_end = 1.15, 5

      t = np.linspace(0, 2*math.pi, N)
      r = np.linspace(R_start, R_end, N)

      R, T = np.meshgrid(r, t)
```

This screenshot shows the defined functions:

```
[36]: # Define Functions
      #========================================

      # Mesh
      def mesh(xc, yc):
          x = R * np.cos(T)
          y = R * np.sin(T)

          z = (x+xc) + 1j * (y+yc)
          return z

      # Transformation
      def trans(zj, a):
          w = zj + (a**2) / zj
          return w

      # Stream Function
      def stream(U, mu, gam, R, T):
          psi = U * R * np.sin(T) - ( mu/ (2*math.pi*R) ) * np.sin(T) - ( gam / (2*math.pi) ) * np.log(R)
          return psi

      # Redline
      def redline(rr, xc, yc):
          xx = rr * np.cos(t)
          yy = rr * np.sin(t)

          zz = (xx+xc) + 1j * (yy+yc)
          return zz
```

This screenshot shows the adjustable parameters and the calling to functions and separating the real and imaginary parts:

```
[37]: # Parameters
      #=======================================
      x_centre, y_centre = -0.15, 0.0

      U_0 = 1
      a = 1
      mu = 8.3
      gamma = -4

      # Function Calling
      #=======================================
      z = mesh(x_centre, y_centre)
      X, Y = np.real(z), np.imag(z)

      w = trans(z, a)
      XX, YY = np.real(w), np.imag(w)

      zz = redline(R_start, x_centre, y_centre)
      xx, yy = np.real(zz), np.imag(zz)

      ww = trans(zz, a)
      xxx, yyy = np.real(ww), np.imag(ww)

      psi = stream(U_0, mu, gamma, R, T)

      # Plot limits
      #=====================================
      x_s, x_e = x_centre - 3, x_centre + 3
      y_s, y_e = y_centre - 3, y_centre + 3
```

This shows the code used to produce the subfigures.

```
[5]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(11,5))

     # Circle subplot
     #============================================
     ax1.set_xlabel('x', fontsize=16)
     ax1.set_ylabel('y', fontsize=16)
     ax1.set_title("z-plane", fontsize=16)
     ax1.set_xlim(x_s, x_e)
     ax1.set_ylim(y_s, y_e)
     ax1.set_xticks([])
     ax1.set_yticks([])

     ax1.contour(X, Y, psi,
                 35, colors='blue', linewidths=2, linestyles='solid')
     ax1.plot(xx, yy, color='red', linewidth=2.5)


     # Aerofoil subplot
     #============================================
     ax2.set_xlabel(r'$\alpha$', fontsize=16)
     ax2.set_ylabel(r'$\beta$', fontsize=16)
     ax2.set_title(r"$\zeta$-plane", fontsize=16)
     ax2.set_xlim(x_s, x_e)
     ax2.set_ylim(y_s, y_e)
     ax2.set_xticks([])
     ax2.set_yticks([])

     ax2.contour(XX, YY, psi,
                 35, colors='blue', linewidths=2, linestyles='solid')
     ax2.plot(xxx, yyy, color='red', linewidth=2.5)
```

Angle of Attack Code

The following screenshots show the code I created to produce the illustrations for angle of attack. Firstly, we can see the defining of functions. Used in Section 6.3.2.

```
[3]: # Angle of Attack
#========================================
def off_axis(aoa, xc, yc, R, T):
    x = R * np.cos(T)
    y = R * np.sin(T)

    g = (x+xc) * np.cos(aoa) + (y+yc) * np.sin(aoa)
    h = - (x+xc) * np.sin(aoa) + (y+yc) * np.cos(aoa)

    k = g + 1j * h
    return k

# Joukowski Transformation
#========================================
def trans(zj, a):
    w = zj + a**2 / zj
    return w

# Stream Function
#========================================
def stream(U, mu, gam, R, T):
    psi = U * R * np.sin(T) - ( mu / (2*math.pi*R) ) * np.sin(T) - ( gam / (2*math.pi) ) * np.log(R)
    return psi

# Redline
#=======================================
def redline(aoa, rad, xc, yc):
    xx = rad * np.cos(t)
    yy = rad * np.sin(t)

    gg = (xx+xc) * np.cos(aoa) + (yy+yc) * np.sin(aoa)
    hh = - (xx+xc) * np.sin(aoa) + (yy+yc) * np.cos(aoa)

    zz = gg + 1j * hh
    return zz
```

The following screenshot shows the adjustable parameters along with the calls to functions and separating of real and imaginary parts:

```
[18]: # Settings
      #=======================================
      x_centre, y_centre = -0.15, 0.05

      U_0 = 1
      a = 1
      mu = 8.3
      gamma = -5

      radius = math.sqrt( mu / (2 * math.pi * U_0) )

      alpha = -(1/8) * math.pi

      # Function Calling
      #========================================
      k = off_axis(alpha, x_centre, y_centre, R, T)
      X, Y = np.real(k), np.imag(k)

      w = trans(k, a)
      XX, YY = np.real(w), np.imag(w)

      zz = redline(alpha, radius, x_centre, y_centre)
      xx, yy = np.real(zz), np.imag(zz)

      ww = trans(zz, a)
      xxx, yyy = np.real(ww), np.imag(ww)

      psi = stream(U_0, mu, gamma, R, T)

      # Plot limits
      #============================================
      x_s, x_e = x_centre - 3.5, x_centre + 3.5
      y_s, y_e = y_centre - 3.5, y_centre + 3.5
```

This screenshots shows the lines of code used to produce the subfigures:

```
[10]: # PLOTS
      #==========================================
      fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(11,5))


      # Circle subplot
      #==========================================
      ax1.set_xlabel('x', fontsize=16)
      ax1.set_ylabel('y', fontsize=16)
      ax1.set_title("z-plane", fontsize=16)
      ax1.set_xlim(x_s, x_e)
      ax1.set_ylim(y_s, y_e)
      ax1.set_xticks([])
      ax1.set_yticks([])

      ax1.contour(X, Y, psi,
                  35, colors='blue', linewidths=2, linestyles='solid')
      ax1.plot(xx, yy, color='red', linewidth=2)


      # Aerofoil subplot
      #==========================================
      ax2.set_xlabel(r'$\alpha$', fontsize=16)
      ax2.set_ylabel(r'$\beta$', fontsize=16)
      ax2.set_title(r"$\gamma$-plane", fontsize=16)
      ax2.set_xlim(x_s, x_e)
      ax2.set_ylim(y_s, y_e)
      ax2.set_xticks([])
      ax2.set_yticks([])

      ax2.contour(XX, YY, psi,
                  35, colors='blue', linewidths=2, linestyles='solid')
      ax2.plot(xxx, yyy, color='red', linewidth=2)
```

A.3 Pressure Code

In the final section of the appendix we will see the code used in Section 7. The following screenshots shows all the functions needed to define all transformations with all different settings.

```
[107]: # Angle of Attack
#======================================
def off_axis(aoa, xc, yc, R, T):
    x = R * np.cos(T)
    y = R * np.sin(T)

    g = (x+xc) * np.cos(aoa) + (y+yc) * np.sin(aoa)
    h = - (x+xc) * np.sin(aoa) + (y+yc) * np.cos(aoa)

    k = g + 1j * h
    return k

# Cylinder
#=========================================================
def velocity_components(U, rad, gam, R, T):
    u = U * np.cos(T) * ( 1 - rad**2 / R**2 )
    v = - U * np.sin(T) * ( 1 + rad**2 / R**2 ) - ( gam / (2 * math.pi * R) )
    return u, v

def complex_velocity(uf, vf):
    dF_z = uf - 1j * vf
    return dF_z

# Joukowski Transformation
#========================================================
def trans(zj, a):
    w = zj + a**2 / zj
    return w

def jt_diff(z, a):
    dw = 1 - (a**2 / z**2)
    return dw

def jt_velocity(df, dw):
    dF_w = df / dw
    return dF_w

# Redline
#=====================================
def redline(aoa, rad, xc, yc):
    xx = rad * np.cos(t)
    yy = rad * np.sin(t)

    gg = (xx+xc) * np.cos(aoa) + (yy+yc) * np.sin(aoa)
    hh = - (xx+xc) * np.sin(aoa) + (yy+yc) * np.cos(aoa)

    zz = gg + 1j * hh
    return zz
```

The following section shows the adjustable parameters of the code and the calls to functions.

[6]:
```
# Settings
#=======================================
x_centre, y_centre = -0.15, 0.05

U_0 = 1
a = 1
mu = 8.3
gamma = 4

alpha = -(1/8) * math.pi

radius = math.sqrt( mu / (2 * math.pi * U_0) )

# Function Calling
#========================================

k = off_axis(alpha, x_centre, y_centre, R, T)
X, Y = np.real(k), np.imag(k)
w = trans(k, a)
XX, YY = np.real(w), np.imag(w)

zz = redline(alpha, radius, x_centre, y_centre)
xx, yy = np.real(zz), np.imag(zz)
ww = trans(zz, a)
xxx, yyy = np.real(ww), np.imag(ww)


u, v = velocity_components(U_0, radius, gamma, R, T)

dF_z = complex_velocity(u, v)

dw = jt_diff(k, a)

dF_w = jt_velocity(dF_z, dw)
uu, vv = np.real(dF_w), np.imag(dF_w)

# Plot limits
#======================================
x_s, x_e = x_centre - 3.5, x_centre + 3.5
y_s, y_e = y_centre - 3.5, y_centre + 3.5
```

This very final piece of code shows the defined pressure coefficents and the lines of code used to created the plots, along with adding the colour bar to the two subfigures.

```
[7]: # Pressure coefficient
     #===========================================
     cp = 1.0 - (u**2 + v**2) / U_0**2
     cp2 = 1.0 - (uu**2 + vv**2) / U_0**2


     # Plot
     #===========================================
     fig1, (ax1, ax2) = plt.subplots(1, 2, figsize=(11,4))
     plt.set_cmap('plasma')

     # Circle subplot
     #===========================================
     ax1.set_xlabel('x', fontsize=16)
     ax1.set_ylabel('y', fontsize=16)
     ax1.set_title("z-plane", fontsize=16)
     ax1.set_xlim(x_s, x_e)
     ax1.set_ylim(y_s, y_e)
     ax1.set_xticks([])
     ax1.set_yticks([])

     ax1.contourf(X, Y, cp,
                  levels=np.linspace(-1, 1, 100), extend='both')
     ax1.plot(xx, yy, color='lime', linewidth=2.25)


     # Aerofoil subplot
     #===========================================
     ax2.set_xlabel(r'$\alpha$', fontsize=16)
     ax2.set_ylabel(r'$\beta$', fontsize=16)
     ax2.set_title(r"$\gamma$-plane", fontsize=16)
     ax2.set_xlim(x_s, x_e)
     ax2.set_ylim(y_s, y_e)
     ax2.set_xticks([])
     ax2.set_yticks([])

     cf = ax2.contourf(XX, YY, cp2,
                  levels=np.linspace(-1, 1, 100), extend='both')
     ax2.plot(xxx, yyy, color='lime', linewidth=2.25)


     # Plot colorbar
     #============================================
     cbar = plt.colorbar(cf, ax=[ax1, ax2])
     cbar.set_label(r'$C_p$', fontsize=16)
     cbar.set_ticks([-1.0, -0.5, 0.0, 0.5, 1.0])
```

References

[1] James, G. (2018) *Advanced Modern Engineering Mathematics*. 5th ed. London: Pearson Education.

[2] Moran, J. (2003) *An Introduction to Theoretical and Computational Aerodynamics*. New York: Dover Publications Inc.

[3] Hughes, W. and Brighton, J. (1999) *Schaum's Outlines: Fluid Dynamics*. 3rd ed. New York: McGraw-Hill Education.

[4] Spiegel, M., Lipschutz, S., Schiller, J. and Spellman, D. (2009) *Schaum's Outlines: Complex Variables*. 2nd ed. New York: McGraw-Hill Education.

[5] Acheson, D. (2003) *Elementary Fluid Dynamics*. New York: Oxford University Press.

[6] Kreyszig, E. (2011) *Advanced Engineering Mathematics*. 10th ed. New Jersey: Wiley.

[7] Henderson, K. (2021) Fluid Dynamics Lecture Notes. BSc Mathematics [online]. Available from: https://blackboard.uwe.ac.uk/uwenav/ultra/courses/_340634_1/cl/outline [Accessed 23 Apr 2022].

[8] Bridge, L. (2021) Complex Variables Lecture Notes. BSc Mathematics [online]. Available from: https://blackboard.uwe.ac.uk/uwenav/ultra/courses/_334881_1/cl/outline [Accessed 23 Apr 2022].

[9] Brennen, C. (2004) *An Internet Book on Fluid Dynamics*. Available from: http://brennen.caltech.edu/fluidbook/content.htm [Accessed 23 Apr 2022].

[10] University of Sydney (2005) *Aerodynamics for Students: Elements of Potential Flow*. Available from: http://www-mdp.eng.cam.ac.uk/web/library/enginfo/aerothermal_dvd_only/aero/fprops/poten/node4.html [Accessed 23 April 2022].

[11] Science World (2022) *Four Forces of Flight*. Available from: https://www.scienceworld.ca/resource/four-forces-flight/ [Accessed 23 Apr 2022].