

Airfoil Analysis

Joe Spencer

September 6, 2022

Methods

An airfoil redirects the air flowing past it to generate lift force. When it does this, its motion in the direction of the airstream is resisted by drag force. Lift force and drag force are controlled by the lift and drag equations, shown in Equation 1:

$$\begin{aligned} L &= \frac{1}{2} \rho u^2 c_L \\ D &= \frac{1}{2} \rho u^2 c_D \end{aligned} \tag{1}$$

It can be seen that lift and drag force are both proportional to the square of velocity, but that they are subjected to different coefficients, the [Lift Coefficient](#) and the [Drag Coefficient](#). These coefficients change depending on the [Reynolds Number](#) of the wing and its [Angle of Attack](#), also denoted by the greek letter α . In addition to affecting the lift and drag experienced by an airfoil, the angle of attack can also affect its [Coefficient of Moment](#).

This project sought to identify how adjusting an airfoil's [Chord](#), or c , and [Camber](#) can affect its performance, making it better suited for different environments. For this project, I downloaded the coordinates of a variety of different airfoils from the [UIUC Applied Aeronautics Group](#) and then imported them to a julia program to calculate their [Lift Coefficient](#), [Drag Coefficient](#), [Pressure Drag Coefficient](#), and [Moment Coefficient](#). As can be seen from the included figures, these coefficients each changed with different angles of attack.

I used the [Xfoil.jl](#) Julia wrapper to calculate the coefficients of lift, drag, and . of the Xfoil potential flow software. Xfoil is a potential-flow code that can calculate an airfoil's lift, drag, pressure drag, moment coefficients. A [Potential-flow Code](#) divides the wing into many small plates and calculates the lift, drag, pressure drag, and moment coefficients for each of a finite number of points along the length of the wing. This program is repeated with different starting estimates for each coefficient until it either converges or is run a set number of times. The user can adjust the limit of how many times the code can run before it gives up.

The [julia file](#) files created for this project and its accompanying [header file](#) could control separately for different input options. I included one function that used a default angle step size of 1 degree, with 100 iterations. It found the lowest and highest angles for which a solution converged, then plotted all four solutions overlaid on each other between these angles. A second function allowed the user to adjust the number of iterations, and a third function allowed the user to manually set the maximum and minimum angles of attack.

For this code, it also became necessary to create similar airfoils, only adjusted slightly for the size and position of their [Camber](#), and the maximum thickness as a percentage of the [Chord](#). To do this, I created a function in the previously mentioned [header file](#) to calculate the shape of an airfoil based on its [NACA Airfoil](#) (National Advisory Committee for Aeronautics) number. Although there are also other ways to describe an airfoil's profile, this seemed like a clear and efficient method.

Table 1: Initial Position, Velocity, and Circulation of Each Vortex

A link to a GIF for each vortex is in its far left column.

GIF	f_m (Hz)	$1^{st} [x_0, y_0, z_0]$	$1^{st} [\dot{x}_0, \dot{y}_0, \dot{z}_0]$	Γ_1	$2^{nd} [x_0, y_0, z_0]$	$2^{nd} [\dot{x}_0, \dot{y}_0, \dot{z}_0]$	Γ_2
1	10	[1.0,0,1.0]	[0,0,0]	1.0	[-1.0,0,1.0]	[0,0,0]	1.0
2	10	[2.0,0,2.0]	[0,0,0]	1.0	[-2.0,0,2.0]	[0,0,0]	1.0
3	100	[1.0,0,1.0]	[0,0,0]	1.0	[-1.0,0,1.0]	[0,0,0]	1.0
4	1	[1.0,0,1.0]	[0,0,0]	1.0	[-1.0,0,1.0]	[0,0,0]	1.0
5	10	[1.0,0,1.0]	[0,0,0]	1.0	[-1.0,0,1.0]	[0,0,0]	-1.0
6	10	[1.0,0,1.0]	[0,0,0]	2.0	[-1.0,0,1.0]	[0,0,0]	1.0
7	10	[0.75,0,0.75]	[0.2,0,0]	2.5	[-0.8,0,0.6]	[0,0,0]	2.5
8	10	[1.0,0,1.0]	[0.006,0,0.008]	1.0	[-1.0,0,1.0]	[0,0,0]	1.0
9	10	[1.0,0,1.0]	[0.2,0,0.1]	3.0	[-1.0,0,1.0]	[0.2,0,0.1]	3.0

Each of the digits in a [NACA Airfoil](#) has its own purpose in defining an airfoil. The first digit is the ratio of the maximum [Camber](#) to the [Chord](#), multiplied by ten. The second digit is the distance of the location of maximum camber from the leading edge of the wing, in tenths of the entire length of the wing. The third and fourth digit define the maximum wing thickness as a percentage of the Chord. The profiles of both the top and bottom of the wing can be described by Equations 2 and 3. In these Equations, x represents the distance from the front of the wing, y_t represents the wing thickness at x , y_c represents the location of the wing's center at x , t represents the maximum thickness as a decimal percentage of the [chord](#), m denotes the wing's maximum camber as a percent of the wing's chord length, and p is the distance of the maximum camber from the front of the wing, in tenths.

$$y_c = \frac{m}{p^2}(2px - x^2) \quad 0 \leq x \leq p$$

$$y_c = \frac{m}{(1-p)^2}((1-2p) + 2px - x^2) \quad p \leq x \leq 1$$

$$y_t = 5t[0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4] \quad (3)$$

The code to create an airfoil from its [4-digit NACA Number](#) was first checked against an airfoil file downloaded from the [UIUC Applied Aeronautics Group](#) website. Once these were found to be the same, this code was used to create similar airfoils that only differed slightly in either their maximum camber, maximum camber location, and maximum wing thickness. These results were then compared to measure the effect of each variable.

Results and Discussion

Pairs of vortex rings can move together along a line, appearing to play leapfrog with each other. This phenomenon can occur when the circulation of one vortex ring pulls a second vortex ring forward, and once the second vortex ring has its own inertia moving it forward it begins to pull the first vortex ring. For this assignment, a set of 4 points marking the top and bottom of 2 vortex rings was modeled to leapfrog off of each other, and different graphs and GIFs were created of this behavior. The code used to solve this problem, as well as output data and graphs from the nine drafts of this project, can be found in the LeapFrog branch (*JoeSpencer1-LeapFrog*) of this [GitHub Repository](#).

[Coefficient of Drag](#) [Coefficient of Lift](#) [Coefficient of Moment](#) [α Airfoil Polar](#) [Lift Coefficient](#) [Stall Chord](#) [Camber thickness](#) [Freestream Velocity](#) [Reynolds Number](#) [Mach Number](#) [NACA Airfoil](#)

I used the Julia programming language to model these leapfrogging vortices. Julia was especially handy for this project because it could perform calculations on large matrices quickly. Another feature of Julia that became especially handy for this project is its ability to publish figures easily for

help visualizing data. Initially, data was written to a file, but recording data to a file was later skipped in the final version to instead dedicate runtime to examining different initial conditions.

Vortices can be described in vector form by Equation 1. This equation shows that the velocity at a point around a vortex is proportional to its circulation, $\vec{\Gamma}$, divided by its distance from the center, \vec{r} . $\vec{\Gamma}$ and \vec{r} are crossed to obtain \vec{V} , meaning that if $\vec{\Gamma}$ is in the y-direction and \vec{r} is in the x-direction, then the velocity induced by the vortex will be in the y-direction. This also means that an initial velocity is not required for leapfrogging vortices, since the circulation itself will induce that. For my completion of this lab, I used a cross-section of the vortex rings and the x and z-axes with $\vec{\Gamma}$ entirely on the y-axis, pointing into or out of the page.

Although $\vec{\Gamma}$ and \vec{r} are crossed, the denominator of Equation 1 contains an r^2 term, which means that circulation has a reciprocal relationship with distance, as also expected intuitively. So, in a case where the vortices are further apart, they will be induced less motion in each other. If the vorticity is increased, the vortices will be induced more strongly and move more quickly in their leapfrogging pattern.

This project allowed some freedom to choose initial conditions. In the final version of the project, I have tried 9 different sets of initial conditions. Each vortex was centered around the x-axis. In Table 1, the starting conditions for each set of leapfrogging vortices are listed. This variety of initial conditions yielded a variety of different results, which will be discussed. Table 1 shows the initial position and velocity coordinates and the initial vorticity for each set of leapfrogging vortices. As shown in the table, differences in each of these initial conditions were tested.

The final program for modeling leapfrogging vortices also allowed the user to adjust the number of steps the program would simulate and the length of each time step, but these parameters were left at 0.1 seconds and 1,000 steps for most of the program. Shorter and longer time steps between measurements were tried, but the lower frequency shown in [GIF 8](#) diverged from the actual solution for the leapfrogging vortices, and the higher calculation frequency modeled in [GIF 9](#) only slowed down the program while measuring fewer vortices that were not substantially different.

Once I understood that I was to model the top and bottom points of 2 vortices, it was relatively smooth sailing to establish a working code. To make this model I used the `DelimitedFiles`, `Printf`, and `Plots` libraries in Julia. After I had created a good model, I decided to animate it. I eventually found that to be fairly simple. I used the `@animate` function to make a gif for each vortex. These can be viewed either by clicking the colored number links in the table, or by going to the LeapFrog branch of the [GitHub Repository](#) mentioned previously.

Results and Discussion

After I had created the functions mentioned in the Methods section, I put them to work analyzing different airfoils. I first completed the example in the [Xfoil.jl Documentation](#) provided with the lab, using [this code](#). After that, I created figures that measured the effect of angle of attack on lift, drag, and moment coefficients for various airfoils and conditions.

This lab examined how vortices can perpetuate each other. The following nine figures show the different paths followed by each vortex pair in this experiment. The results shown in these graphs provide insights about when the phenomenon of vortex leapfrogging occurs, and when it does not.

One interesting relationship to note in the vortex rings that this experiment found is that for vortex rings that go on the longest, the velocity is inversely proportional to the distance. This can be recalled from Equation 1, which showed that the magnitude of velocity is directly proportional to circulation and ultimately inversely proportional to distance. The [videos](#) included in this report show that for the vortices that continued longest, the highest velocity occurred when r

was smallest, and lowest velocity came when r between the top and bottom vortices of the ring was greatest.

1 Glossary

- Angle of Attack, α - The angle of attack, α is the angle between the motion of oncoming fluid and the chord line of the wing. a positive α corresponds to a wing tilted upwards.
- Camber - The Camber of an airfoil is represented by the Camber Line, which runs halfway between its top and bottom surfaces. This line represents the curvature of an airfoil. An airfoil with positive camber is slightly convex on top and slightly concave on its bottom.
- Chord, c - An airfoil's Chord is the imaginary line running straight from the tip of an airfoil to its tail. The chord line is used to find an airfoil's [Angle of Attack](#).
- Drag Coefficient (2D), c_D - The drag coefficient determines how much drag force opposing motion will be experienced. It comes from a combination of [Pressure Drag](#), also called Form Drag, and [Viscous Drag](#) or Skin Friction. The drag force equation describes it in this way:

$$D = \frac{1}{2}\rho u^2 c_D \quad (4)$$

- Freestream Velocity, V_∞ - The velocity of an oncoming air flow directly upstream from an airfoil, before it interacts with it.
- Lift Coefficient (2D), c_L - The lift coefficient is used in the equation below to define how much lift force acts perpendicular to the direction of the oncoming fluid flow.

$$L = \frac{1}{2}\rho u^2 c_L \quad (5)$$

- Lift Curve Slope - The lift curve plots the [Lift Coefficient](#) against the [Angle of Attack](#). This shows the effect that changing the angle of attack will have on the plane's total lift force.
- Mach Number, M - The Mach number is the velocity of an object in proportion to the speed of sound in its medium. When an airfoil is traveling near the speed of sound, its top portions can have fluid velocity above Mach 1. When an airfoil is traveling above Mach 1, the fluid on both sides of it also have velocities above mach 1 while there are points directly before and after it that are below Mach 1.

$$M = \frac{u}{c} \quad (6)$$

- Pressure Drag - Pressure Drag, also called Form Drag, comes from the the formation of a vacuum behind an object. The object experiences higher pressure ahead of it than behind it, so the pressure difference pushes it backwards.
- Pitching Moment Coefficient (2D), c_M - The moment coefficient is used to calculate the pitching moment a wing will experience from its dynamic pressure q , area S , and chord length c .

$$M = qSc c_M \quad (7)$$

- Polar - The Airfoil Polar is a plot showing the [Lift](#) and [Drag](#) coefficients corresponding with each [Angle of Attack](#) for an airfoil. Examining the ratios of lift to drag is instrumental in choosing the optimal Angle of Attack.
- - A potential-flow code calculates the [lift](#), [drag](#), and [moment](#) forces an airfoil experiences at many different points along it. Potential-flow theory assumes constant, incompressible, inviscid fluid flow. It calculates the coefficient for each of these small points and then combines them to find the coefficients along the entire wing.

- Reynolds Number, Re - The Reynolds Number is a unit-less number for fluid flow described by the equation below. It can be used to predict patterns in the fluid's flow, using its flow speed u , characteristic length L , and kinetic viscosity ν , or else by its density ρ , flow speed u , characteristic length L , and fluid density μ .

$$\begin{aligned} Re &= \frac{uL}{\nu} \\ &= \frac{\rho uL}{\mu} \end{aligned} \tag{8}$$

- Stall - Stall occurs when an airfoil's [Angle of Attack](#) is too great in magnitude. When the angle of attack is too great, flow separation occurs, reducing rather than augmenting the airfoil's [Lift Coefficient](#) as the Angle of Attack increases.
- Thickness - An airfoil's thickness can be measured in two different ways, either along its [Chord Line](#) or along its [Camber Line](#). Thickness measured perpendicular to the Camber Line is also called the American Convention, and thickness measured perpendicular to the Chord Line is also called the British Convention.
- Viscous Drag - Viscous Drag, also called Skin Friction, is drag caused by friction with the fluid particles flowing past an airfoil. Along with [Pressure Drag](#), it contributes to an airfoil's total [Drag Coefficient](#) airfoil.
- 4-Digit NACA Airfoil - A wing shape developed by the National Advisory Committee for Aeronautics (NACA). The first digit is the maximum [Camber](#) in tenths of the [Chord](#). The second digit is the distance in tenths of the maximum Camber from the leading edge, out of ten. the final two digits are the maximum wing thickness as a percentage of the Chord. Descriptions of other NACA numbers can also be found on this [Wikipedia Article](#).