

AERO40003 Computing & Numerical Methods 1

Dr Elnaz Naghibi
Dr Maria Ribera-Vicent

2021-22 Coursework brief

Part 1: MATLAB

The aim of this coursework is for you to write a few more complex programs than what you typically have encountered so far in tutorials. Unlike most tutorial questions, you will have to come up with the structure of the program yourself. This is what programming will involve for the rest of your career after all.

The following brief presents the background information and instructions necessary for you to complete this part of the coursework. You must submit all the files (the `.m` files for these scripts/functions and figures) through this submission box in a `.zip` file.

The coursework will be marked based on completeness, readability, elegance, consistency, correctness and obviously on whether the programs run!

Panel Codes

1 Theoretical Background

The flow field around an arbitrary lifting body in potential flow (incompressible, inviscid and irrotational) can be predicted computationally using a Panel Code. Panel codes work by breaking up the surface of an arbitrary body into a number of panels, as seen in figure 1.

In order to apply a Panel Code to an airfoil, first we need to discretise its surface into N panels. The NACA 4-series of airfoils is a well known family of sections that can be generated using the parameters defined by the digits of their numerical code.

The first digit represents the maximum camber (m), in percent of chord; the second digit indicates the location of maximum camber (p), in tenths of chord; and the final two digits correspond to the maximum aerofoil thickness (t), which for this family occurs at 30% of the chord from the leading edge.

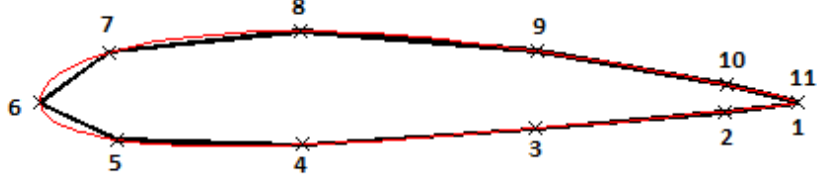


Figure 1: Discretization of a NACA 2415 airfoil using 10 panels on its surface. The numbers shown correspond to the order in which the panel endpoints are generated and stored in the arrays `x` and `z` of the function `panelgen()`.

The NACA 2415 example would therefore have a maximum of 2% camber at 0.4 of the chord, and a maximum thickness of 15% of the chord. Symmetrical airfoils will start with 00, as they have no camber.

Aerofoil sections can therefore be constructed by adding a thickness function over a mean camber line. The mean camber line for this family of aerofoils can then be obtained from the following expression (note, x is the non-dimensional coordinate, 0 at the leading edge and 1 at the trailing edge):

$$y_c = \begin{cases} \frac{m}{p^2}(2px - x^2), & 0 \leq x \leq p, \\ \frac{m}{(1-p)^2}((1-2p) + 2px - x^2), & p \leq x \leq 1 \end{cases} \quad (1)$$

Similarly, the thickness function for this family is given by:

$$y_t = 5t(0.2969\sqrt{x} - 0.126x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4) \quad (2)$$

The thickness needs to be apply perpendicular to the camber line, therefore the coordinates of any point on the airfoil surface become:

$$x_U = x - y_t \sin \theta, \quad z_U = y_c + y_t \cos \theta \quad (3)$$

$$x_L = x + y_t \sin \theta, \quad z_L = y_c - y_t \cos \theta \quad (4)$$

where θ is given by:

$$\theta = \arctan \frac{dy_c}{dx} \quad (5)$$

with

$$\frac{dy_c}{dx} = \begin{cases} \frac{2m}{p^2}(p - x), & 0 \leq x \leq p, \\ \frac{2m}{(1-p)^2}((p - x)), & p \leq x \leq 1 \end{cases} \quad (6)$$

The panel end point x can be calculated using a cosine distribution, which gives the first value at the trailing edge:

$$x = 1 - 0.5 \left(1 - \cos 2\pi \frac{i-1}{N} \right) \quad (7)$$

An additional panel will be placed starting at the trailing edge and extending to infinity (some large number) in the direction of the flow to represent the wake of the airfoil.

Each 2-dimensional panel can be made up of sources, sinks, doublets or vortices, hence affecting the flowfield around it. In this case we will be using doublet elements, each of which will affect the flowfield around it by imparting a horizontal velocity u and a vertical velocity v , proportional to the element's strength μ . Panel codes work by finding what strength each panel around the body should have so that a certain condition is met. The condition we will be using, is that the velocity normal to the wall and therefore also at the centre of each panel must be zero (non-permeability).

Therefore if we can find the total velocities U_i and V_i imparted by the freestream flow and all the panels at the centre of the i^{th} panel, we can ensure that the total velocity normal to it is zero. Mathematically the total velocity at the centre of the i^{th} panel can be expressed as:

$$U_i = U_\infty \cos \alpha + \sum_{j=1}^{N+1} \mu_j u_{ij} \quad (8a)$$

$$V_i = U_\infty \sin \alpha + \sum_{j=1}^{N+1} \mu_j v_{ij} \quad (8b)$$

where U_∞ is the speed of the freestream airflow, α is the airfoil's angle of attack, μ_j is the strength of the j^{th} panel and N is the total number of panels.

The velocities imparted at the centre of the i^{th} panel by the j^{th} panel, u_{ij} and v_{ij} , can be found using the function `cdoublet()`.

We can also find that the velocity component normal to the i^{th} panel at its midpoint is:

$$(U_t)_i = -U_i \sin \beta_i + V_i \cos \beta_i \quad (9)$$

where β_i is the panel angle given by

$$\beta_i = \tan^{-1} \left(\frac{dz}{dx} \right) = \tan^{-1} \left(\frac{z_{i+1} - z_i}{x_{i+1} - x_i} \right) \quad (10)$$

where $[x_i, z_i]$ and $[x_{i+1}, z_{i+1}]$ are the coordinates of the i^{th} panel's endpoints. Note that β_i should be estimated in such a way that $0 \leq \beta_i \leq 2\pi$, depending on the sign of dx and dz . (you may wish to use the `atan2()` function such that $0 \leq \beta_i \leq 2\pi$).

Combining equations (8) and (9), a system of $N + 1$ linear equations, with $N + 1$ unknowns is created. The first N relations are given by:

$$(U_t)_i = U_\infty \sin(\alpha - \beta_i) + \sum_{j=1}^{N+1} \mu_j (v_{ij} \cos \beta_i - u_{ij} \sin \beta_i) = 0 \quad (11)$$

and therefore

$$\sum_{j=1}^{N+1} \mu_j (v_{ij} \cos \beta_i - u_{ij} \sin \beta_i) = -U_\infty \sin(\alpha - \beta_i) \quad (12)$$

The final relation comes from the Kutta condition and simply states that circulation at the trailing edge must be zero and therefore:

$$\mu_{N+1} + \mu_1 - \mu_N = 0 \quad (13)$$

The system of equations can be solved computationally by posing the problem in the form:

$$\mathbf{A}\mathbf{x} = \mathbf{B} \quad (14)$$

where \mathbf{A} is an $(N+1) \times (N+1)$ square matrix and \mathbf{x} and \mathbf{B} are column vectors of length $(N+1)$.

Once the system has been solved and the unknown panel strengths (μ_1 to μ_{N+1}) have been found, equation (8) can be used to estimate what the velocity field around the airfoil is. The lift coefficient of the airfoil can also be crudely estimated using the strength of the wake panel doublet:

$$C_L = -\frac{2\mu_{N+1}}{U_\infty} \quad (15)$$

2 Implementation

First write a function `panelgen()` that takes the airfoil code as a string (i.e. '1410'), the number of panels to be used N and the angle of attack AoA . The function should use the method described above to calculate the coordinates of all the panel end points. The function should return two one dimensional arrays x and z .

Write a script that asks the user which airfoil to use, the freestream velocity U_∞ , the angle of attack α and the number of panels to be used N . The code should:

- Use the function `panelgen()` to generate and discretize the airfoil.
- Generate the matrices \mathbf{A} and \mathbf{B} and solve the system of equations to find the unknown panel strengths. You may find that the matrix \mathbf{A} is close to singular, but that is common for this kind of problem.
- Output the airfoil's lift coefficient.
- Plot the velocity field vectors (arrows) and streamlines around the airfoil in the domain where $-0.2 \leq x \leq 1.2$ and $-0.7 \leq z \leq 0.7$. To achieve that you will have to evaluate the velocities U and V not on the airfoil body but at various points in the domain surrounding the airfoil. Note that you will need a much finer grid to accurately draw the streamlines than to draw vectors. The `inpolygon()` function may also prove useful for identifying which points are inside your airfoil and should thus not be plotted.

- Save the plots programmatically.

However, if the airfoil is **NACA 2412**, the script should:

- Calculate the lift coefficient for a range of angles of attack between 0° and 10° and for different values of N : 50, 100 and 200.
- Read the XFOIL data file for this airfoil provided.
- Plot the C_L vs. α for the different cases of N against the XFOIL data for the range of angles calculated.
- Plot the velocity field vectors and streamlines for the last case.
- Save your plots programmatically.

3 Sample case

Try your program for different NACA 4-series airfoils, at various angles of attack. Check your results against the theoretical lift curve slope of 2π for those airfoils and the flowfields seen in figures 2 and 3.

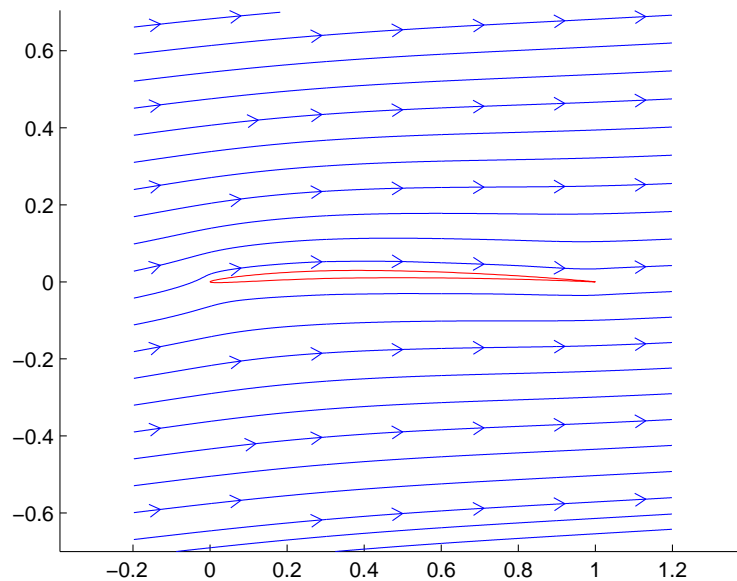


Figure 2: Streamlines around a NACA 2405 at $\alpha = 5^\circ$ and using $N = 250$.

4 What to submit

You should submit the following items in a folder called `CW_Pt1_username`, replacing username with your own, and zip the folder before uploading:

- Your `panelgen()` function.

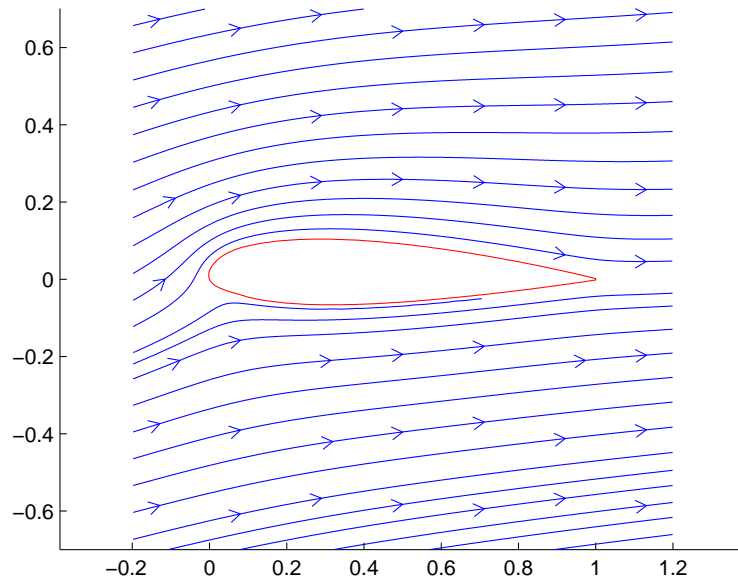


Figure 3: Streamlines around a NACA 2117 at $\alpha = 10^\circ$ and using $N = 250$.

- Your script running the panel code you have written.
- Arrow and streamline plots for two different NACA 4-series airfoils of your choice, each at two different angles of attack and at the speed and discretization of your choice (but label accordingly).
- C_L vs. α plot for NACA 2412 and arrow and streamline plots at $15m/s$, $\alpha = 10^\circ$ and $N = 200$.

5 Formatting guidelines

As you have more time for this project, it is expected that you will put some care into presenting both your code and plots neatly and professionally. Please refer to the mark scheme to make sure your code complies with the expectations we will be looking when marking. Correctness is indeed important. But other criteria such as readability or elegance are also important.

For this exercise, it is recommended that you work on an .m file. Save the file with some meaningful name. Parts of your code can be provided in functions, which should also be named meaningfully so the markers can identify them. Functions might make your code more elegant and readable, but only one is required.

Additionally, the following criteria are expected for the formatting of the graphs:

- Label your axes and add titles to your plots.
- For line plots with multiple lines, add a legend.

- For the airfoil plots, use a line thickness of 2 for the airfoils and 1 for the arrows and streamlines. Use different colours for each.
- Make sure your airfoils are not distorted.
- For the C_L vs. α plot, use a line thickness of 1.5. Plot the XFOIL line in black and the rest in different colours.
- Adjust the font size in the plots so that they are easy to read.