Subsonic Linearized Flow Analysis: Critical Mach Number Calculator with Python

Hafizh Renanto Akhmad
13621060
K01 AE3110 Aerodynamics I
Bandung, Indonesia
13621060@mahasiswa.itb.ac.id

I. Introduction

In this report, we will try to make a Python/MATLAB subroutine/code that calculates the critical Mach number, based on the critical pressure coefficient eq. and compressibility correction (Prandtl Glauert, Laitone, and Karman-Tsien).

II. Linearized Flow

During the 1940s, when engineers were tasked with achieving precise calculations for the aerodynamic forces affecting fighter planes operating within the range of airspeed where compressibility becomes a significant factor, they encountered a pivotal challenge. At the time, digital computers had not yet been invented, making it unfeasible to employ numerical solutions. Faced with this limitation, engineers seek assumptions about the fundamental physics of the flow, which allowed them to linearize the governing equations. The linearization solution can be obtained through the assumption of small perturbation.

A. Linearized Pressure Coefficient

Consider a slender body immersed in a uniform flow. In the uniform flow, the velocity is V_∞ and is oriented in the x-direction. In the perturbed flow, the local velocity is $\mathbf{V}$, where $\mathbf{V} = V_x\mathbf{i} + V_y\mathbf{j} + V_z\mathbf{k}$. The *perturbations* in each axis are denoted by u', v', and w', where $V_x = V_\infty + u'$, $V_y = v'$, and $V_z = w'$.

The pressure coefficient C_p is defined as

$$C_p \equiv \frac{p - p_\infty}{\frac{1}{2}\rho_\infty V_\infty^2} \tag{1}$$

Using the assumption of small perturbation, we may obtain the *linearized pressure coefficient*:

$$C_p \approx -\frac{2u'}{V_\infty} \tag{2}$$

B. Linearized Subsonic Flow

For the compressible subsonic flow over a thin airfoil at small angle of attack (which results in small perturbations), (2) can be rewrote as

$$C_p = \frac{C_{p_o}}{\sqrt{1 - M_\infty^2}} \tag{3}$$

which is called *Prandtl-Glauert Rule*, and is exceptionally practical correction approximation. However, this approximation does not fully recognize changes in the local regions of the flow. To improve the correction, Laitone apply (2) locally in the flow, i.e.,

$$C_p = \frac{C_{p_o}}{\sqrt{1 - M^2}} \tag{4}$$

which could be rewritten as

$$C_p = \frac{C_{p_o}}{\sqrt{1 - M_\infty^2} + \frac{M_\infty^2\left(1+\frac{\gamma-1}{2}M_\infty^2\right)}{\sqrt{1-M_\infty^2}}\frac{C_{p_o}}{2}} \tag{5}$$

and this correction is called as *Laitone rule* Another compressibility correction that has been used widely is the *Karman-Tsien rule*, by von Karman and Tsien who utilizes hodograph solution fo the nonlinear equations of motion along with a simplified "tangent gas" equation of state, which result in

$$C_p = \frac{C_{p_o}}{\sqrt{1 - M_\infty^2} + \frac{M_\infty^2}{1+\sqrt{1-M_\infty^2}}\frac{C_{p_o}}{2}} \tag{6}$$

C. Critical Mach Number

Consider an airfoil at a subsonic speed. The flow going through the top airfoil expands, dropping the static pressure to a minimum value at a certain point. There are a case when the freestream Mach number is increased enough such that the Mach number on the point of minimum pressure point reaches sonic speed. The corresponding freestream Mach number is called the *critical Mach number*, M_{cr}. The equations governing the critical Mach number and the compressible pressure coefficient at the minimum pressure point is

$$C_{p_{\text{cr}}} = \frac{2}{\gamma M_{\text{cr}}^2}\left[\left(\frac{1+\frac{\gamma-1}{2}M_{\text{cr}}^2}{1+\frac{\gamma-1}{2}}\right)^{\gamma/(\gamma-1)} - 1\right] \tag{7}$$

III. Methods

We need to find the critical Mach number for a given data of incompressible pressure coefficient at the minimum pressure point C_{p_o} of a certain airfoil. The procedure is easily determined: just find the intersections between the curve of (3), (5), and (6) with the curve of (7), as seen in Figure 1. However, we know that the listed compressibility rules are applied for subsonic flow $M < 1$, and thus C_{p_o} has to be negative. The correct compressibility curve are shown in 2.

The intersections are obtained by finding the root of the difference between each compressibility rule with the critical

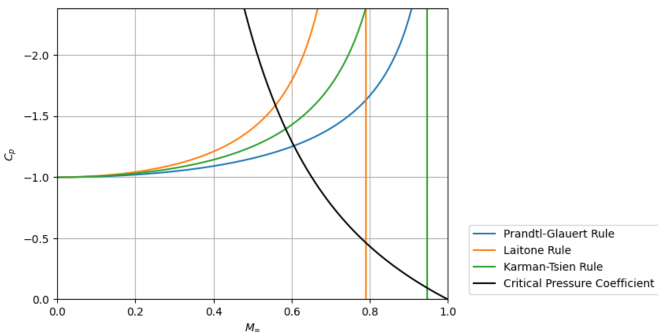

Fig. 1. Example of the curve of compressibility rules and critical pressure coefficient for $C_{p_o} = -1$

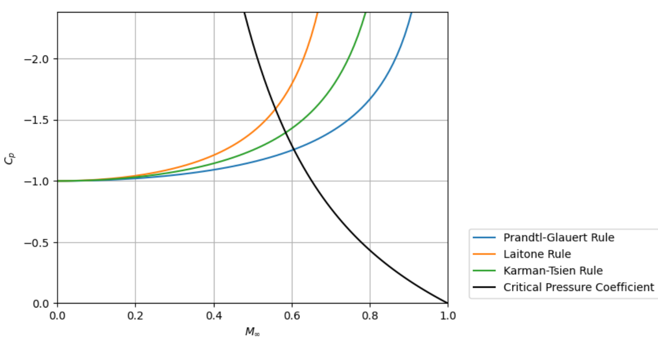

Fig. 2. The correct curve of compressibility rules and critical pressure coefficient for $C_{p_o} = -1$

mach number equation. However, we may observe that the equations are complex and explicit solution of M_{cr} can't be easily obtained. Thus, we may use numerical solution as an alternative.

To conduct the numerical solution, we utilize *Python* as the programming language and certain libraries: *numpy* (for mathematical operations) and *matplotlib* (for graphical interpretation). We will also use *tkinter* to make the GUI of the calculator. For this report, we use numerical solution method for root finding of *bisection method*. Bisection method involves the intermediate value theorem, that is, for a continuous function f on $[a, b]$ with $f(a) < L < f(b)$, there must be a value of c where $f(c) = L$. For root finding of the function f, $L = 0$. Firstly, we need an initial interval guess (x_l, x_u) where $f(x_l) \cdot f(x_u) < 0$. Next, we calculate the value at the center of the interval x_m. If $f(x_l) \cdot f(x_m) < 0$, then the root must be in the interval of (x_l, x_m), otherwise, in the the root must be in the interval of (x_m, x_u). Thus, the new center of interval value is calculated, and the process is repeated until the interval center value meets a certain error toleration, or until the number iteration reaches a certain limit.

To determine the 'good' initial interval, we may choose the lower limit of $M_\infty = 0$ and upper limit of M_∞ where the compressibility correction get closer and closer to infinity, that is, when the denominator of the compressibility correction equation becomes 0. The corresponding M_∞ for Prandtl-Glauert rule is $M_\infty = 1$, for Laitone rule is

$$M_\infty = \sqrt{\frac{2 - C_{p_o} - \sqrt{(2 - C_{p,o})^2 - 4(\gamma - 1)\, C_{p_o}}}{(\gamma - 1)\, C_{p_o}}} \tag{8}$$

and for Karman-Tsien rule is

$$M_\infty = \sqrt{1 - \left(\frac{1 - \sqrt{1 + C_{p_o}\,(C_{p_o} - 2)}}{C_{p_o} - 2}\right)^2} \tag{9}$$

A figure to show the location of the intersections may be made to give more clarity and visualization, and a GUI may be implemented to make the calculator feel lively.

IV. Results

The GUI of the calculator are as shown in Figure 3. When a value of specific heat ratio and incompressible pressure coefficient is inserted and the *Calculate* button is pressed, values of critical Mach number for each compressibility rule and the illustrating graph are generated, as shown in 4, 5, and 6. The critical Mach number values and the graph can be overwrited by inserting new values of specific heat ratio and/or incompressible pressure coefficient.

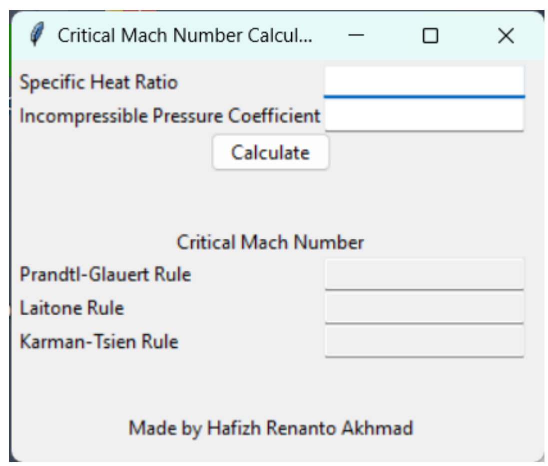

Fig. 3. The GUI of the calculator at start

V. Conclusion

The calculator works perfectly for the freestream Mach number in the interval of speed below sonic, except on certain extreme values, that is, when C_{p_o} gets very very close to the value of 0 or 1, due to limitation of computational calculation (precision of floating points).

Appendix

The source code of the program can be found here.

References

[1] J. Anderson, ISE Modern Compressible Flow: With Historical Perspective. New York: McGraw Hill, 2020.

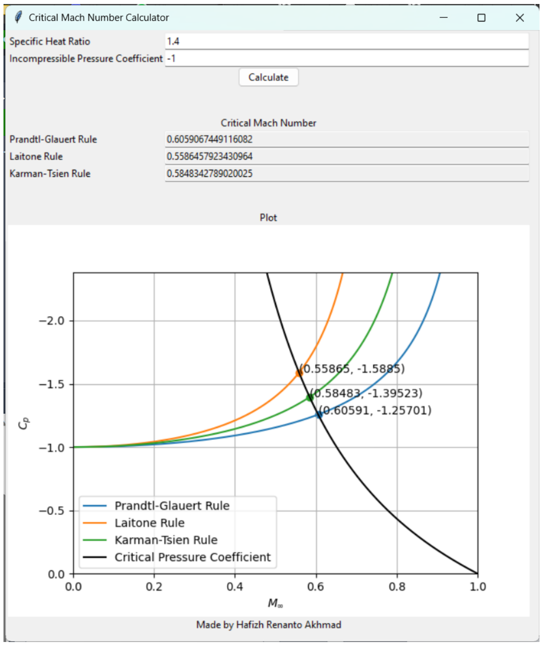

Fig. 4. The GUI of the calculator after calculating values of M_{cr} for $C_{p_o} = -1$

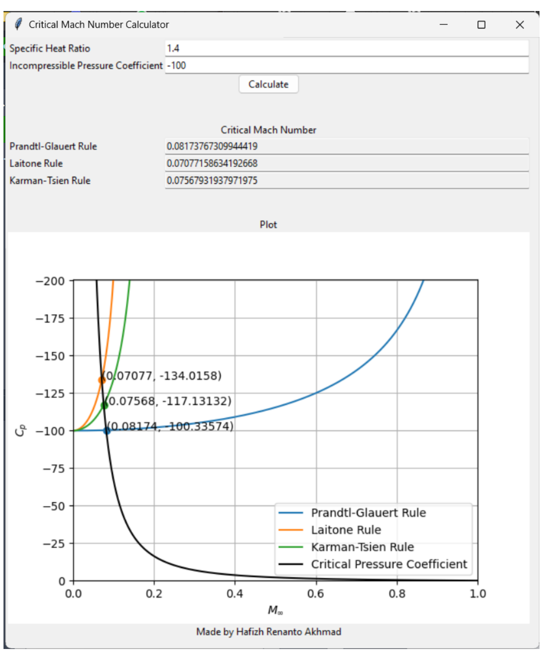

Fig. 5. The GUI of the calculator after calculating values of M_{cr} for $C_{p_o} = -100$

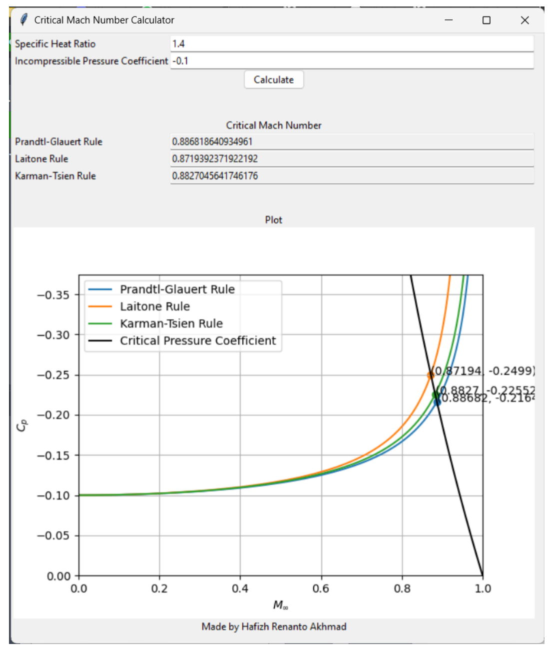

Fig. 6. The GUI of the calculator after calculating values of M_{cr} for $C_{p_o} = -0.1$