Aircraft Stability Analysis
Architecture/Design Document

Kevin Parlak, Evan Donkus, David Gao, Tasnuva Mahjabin

Table of Contents

1. Introduction

1.1 Overview

A Microsoft Foundation Class (MFC) application which utilizes airfoil data provided by the third-party program, XFOIL, was created to generate flight parameters for certain aircraft properties. This Graphical User Interface (GUI) provides a platform that allows users to analyze aircraft static stability parameters. The purpose of this application was to create a program that can be used to quickly gather information about flight parameters of an aircraft by inputting values for the wing, tail, elevator, airspeed, and atmosphere. The target audience for this program was other aerospace engineering students who frequently use XFOIL and need to quickly calculate aircraft parameters.

1.2 Assumptions

Many assumptions were made in association with this program. Due to the complexity of aircraft stability, this program only looks at longitudinal static stability. Aircraft Stability Analysis shows trimmed capabilities of the aircraft being analyzed. As well, aircraft in this program are assumed to be in steady-level flight to reduce the complexity that drag presents in the equations. Propulsion and flap effects as well as effects from a trim tab were ignored to simplify deflection calculations. Finally, aircraft being examined in the program are assumed to have a rectangular wing to reduce the complexity of the downwash effect from the wing.

1.3 Requirements

Aircraft Stability Analysis had requirements that enabled the user to quickly navigate the software.

- Dialog file open only .txt files
- Dynamically change input
- User friendly interface
- Ability to dive into complex analysis
- Use of classes and inheritance to simplify calculations
- Dynamically change output

2. Design Goals

The goal of Aircraft Stability Analysis was to make computations of basic stability characteristics easier for students and professors. This program was created to reduce the time it takes to analyze an aircraft in certain conditions. Aircraft Stability Analysis shows how airspeed and airfoil data directly affect parameters associated with assumptions and calculations. This MFC application allows users to input actual airfoil data from XFOIL.

XFOIL is a common open source, text based program widely used in the aerospace industry to analyze two dimensional airfoils. XFOIL is a downloadable executable file that can be found at web.mit.edu/drela/Public/web/xfoil/. This subsonic airfoil development system outputs parameters such as angle-of-attack, lift coefficient, drag coefficient, and pitching moment coefficient for a specific airfoil. Airfoil dimensions and profiles are readily available on the Internet. A common website used to gather airfoil profiles is airfoiltools.com. Profiles found at this site can be loaded into XFOIL for analysis. XFOIL data is then used in association with Aircraft Stability Analysis to output relevant aircraft data to the user with the ease of a few button clicks.

3. System Behavior

Properly using this program involves a multitude of steps the user needs to follow. First, the user needs to collect an airfoil profile of their choosing. This file represents the shape of the specific airfoil. Next, XFOIL will be used to generate a text file containing data the user wishes to utilize in their calculations. This text file can be located anywhere on the user's computer. Finally, the user will be able to use the program. Aircraft Stability Analysis comes as a .exe file and can be opened with a double-click. After opening the program the user will first generate their wing from the XFOIL text file generated earlier. The user will then generate the atmosphere for which they will be conducting an analysis. The aircraft is built next by inputting all aircraft parameters. After completing all preceding steps, the user can hit "Fly" to gather their data.

4. Logical View

Aircraft Stability Analysis is divided into multiple sections to help guide the user. Output, Aircraft Properties, Atmospheric Properties, Wing Input, Tail and Elevator Input, and Airspeed Input define the sections of the GUI. These sections are governed by buttons and scrollbars. Fly, Build Aircraft, Generate, Generate Wing, Angle-of-Attack scrollbar, and Airspeed scrollbar rule each section respectively.

4.1 Buttons

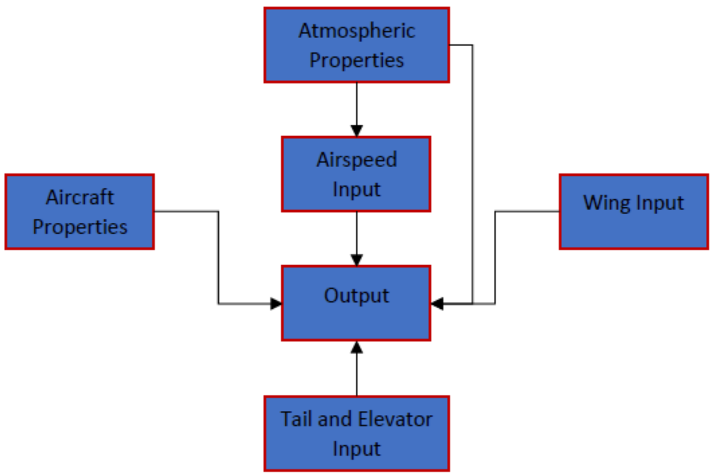
Each button corresponds to an executable within the program. The buttons also generate the various classes involved in the stability calculations. A data exchange occurs whenever a button is pressed by the user. "DoDataExchange" of the dialog header file connects the input from each edit control box to the associated variable to be used in calculations. Variables for each edit control box were added and initialized in the dialog .cpp file. "Generate Wing" connects to the user's computer, "Generate" links atmospheric data to the Atmosphere class, "Build Aircraft" inputs aircraft parameters into the Aircraft and Tail and Elevator classes, and "Fly" calls the Detailed Output and Preliminary Output classes.

4.2 Scrollbars

The vertical and horizontal scrollbars in the program use switch statements to adjust output as the user scrolls or clicks. This switch statement takes in defined limits for the scrollbar and increments or decrements depending on which way the user moves. The airspeed scrollbar sets scroll limits based on the atmosphere generated by the user. The equations used in this program are invalid when the aircraft approaches the transonic region. Therefore, based on atmosphere input, a Mach number of .8 is the highest the airspeed may approach. The angle-of-attack scrollbar derives from the generated wing input file. This file contains data from lines 13 until the end of the file. This scrollbar was programmed to only scroll from line 13 to where the file ends using a counter each time a line was read in.

4.3 High-Level Design

The high-level view or architecture consists of six major components: Atmospheric Properties, Airspeed Input, Wing Input, Aircraft Properties, Tail and Elevator Input, and

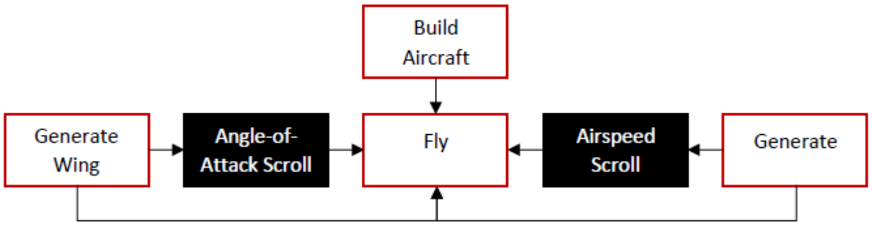


Output. These sections of the GUI define the high-level architecture of the software. This program was designed so the user could see how airspeed and airfoil data affect the output. Aircraft Stability Analysis also shows the user how different sections of the interface interact with each other.

- **Atmospheric Properties** generate data that communicate with how the airspeed input is configured.
- **Airspeed Input** develops rule sets based on **Atmospheric Properties** and affects the overall output.
- **Wing Input** uses third-party software and data to develop parameters that directly affect the output to the user.
- **Aircraft Properties** directly affect the overall output via user input.
- **Tail and Elevator Input** also directly affects overall output via user input.
- **Output** takes parameters from each section of the GUI.

4.4 Mid-Level Design

The mid-level design of the GUI consisted of buttons the user presses to execute various commands: Build Aircraft, Generate, Airspeed Scroll, Generate Wing, Angle-of-Attack Scroll, and Fly. These buttons are held within each section of the high-level design.

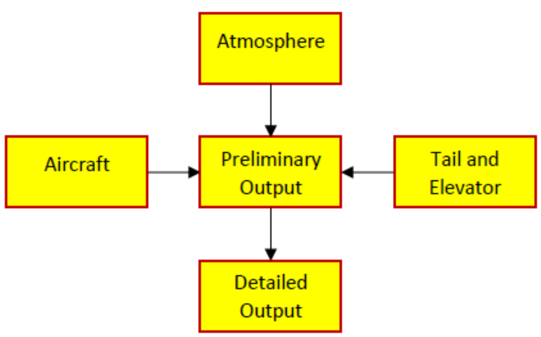


Each button executes each of the associated sections data to communicate with each other.

- **Build Aircraft** links the user input from the edit control boxes to variables describing each parameter.
- **Generate** takes atmospheric data and computes a max range for the airspeed scroll bar.
- **Airspeed Scroll** takes rules set by **Generate** and allows the user to dynamically adjust their airspeed.
- **Generate Wing** opens the user's computer and allows them to search for an airfoil text file to import.
- **Angle-of-Attack Scroll** takes limits from the .txt file imported and allows the user to scroll through the angles-of-attack within that file.
- **Fly** takes all parameters input into edit control boxes along with scroll bar values and computes various aircraft parameters.

4.5 Detailed Class Design

Aircraft Stability Analysis uses five main classes in order to use parameters input by the user. These five main classes include: Atmosphere, Aircraft, Tail and Elevator, Preliminary Output, and Detailed Output. Atmosphere, Aircraft, and Tail and Elevator



are the input classes that communicate with the output classes, Preliminary Output and Detailed Output.

- **Atmosphere** takes in altitude and temperature input along with a predefined Mach number, specific heat value for air, and Universal Gas Constant.
- **Aircraft** imports input from the Aircraft Properties section and ties the variables to the Build Aircraft button.

- **Tail and Elevator** tie input from the Tail and Elevator Input section to the Build Aircraft button.
- **Preliminary Output** derives from **Aircraft** and **Tail and Elevator**. This class derives input and computes next level variables due to the complexity of the equations used in Aircraft Stability Analysis. The Fly button controls this class.
- **Detailed Output** derives from **Preliminary Output**. This class derives output and computes final level variables which are displayed to the user in the Output section of the GUI.

5. Process View

Not Required.

6. Development View

Not Required.

7. Physical View

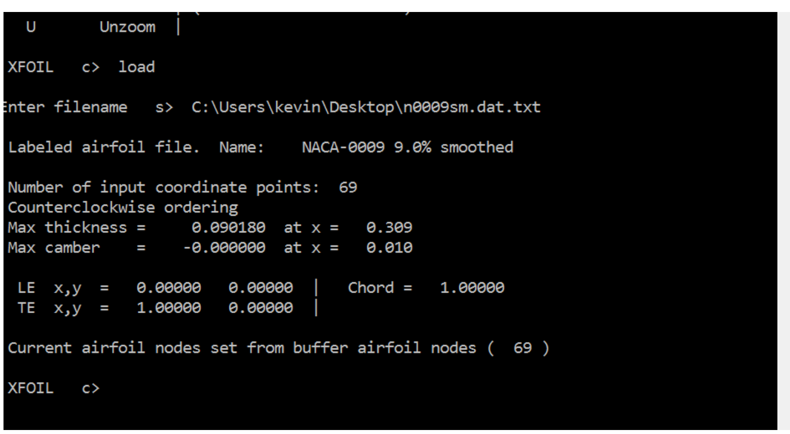
Not Required.

8. Use Case View

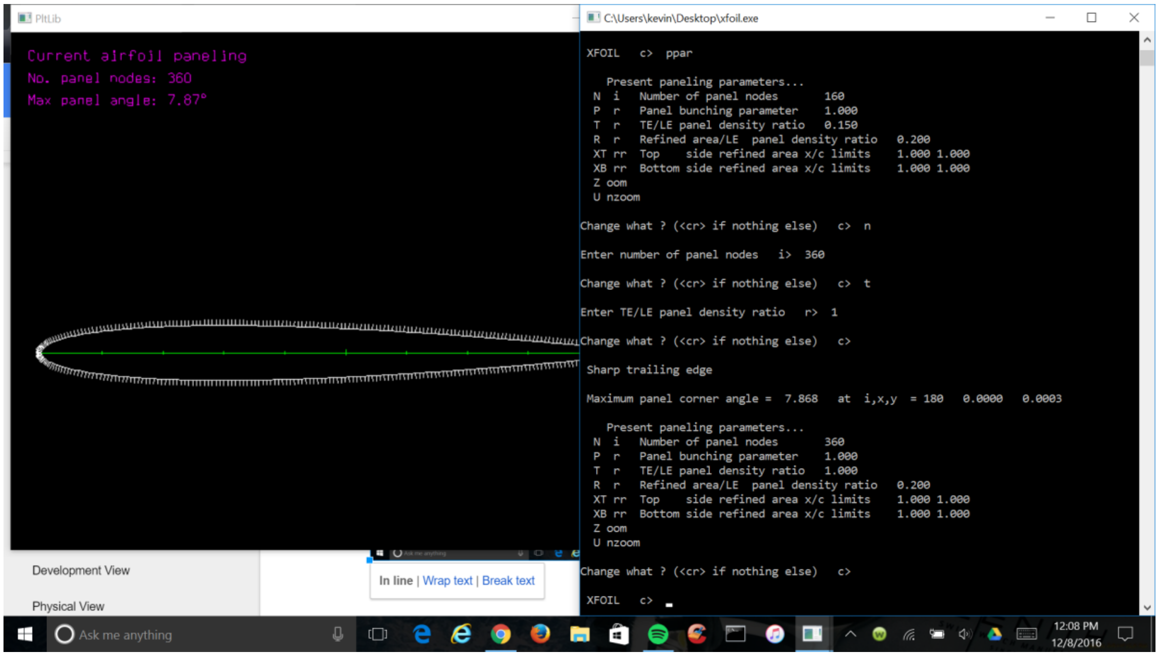
The user first needs to download an airfoil profile from an open source website or utilize one already available.

8.1 Third-Party Software – XFOIL

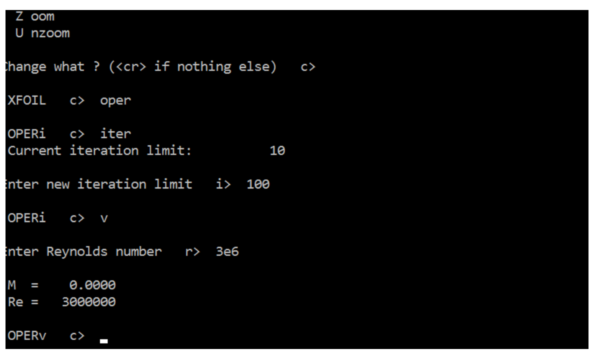
Next, XFOIL will be opened. The user will input their airfoil profile using the “load” command and hitting ENTER. The user will then import or physically drag their profile file into the interface. NOTE: If importing the file, one must enter the entire path name for XFOIL to recognize the file. The user will now change the iteration state of XFOIL



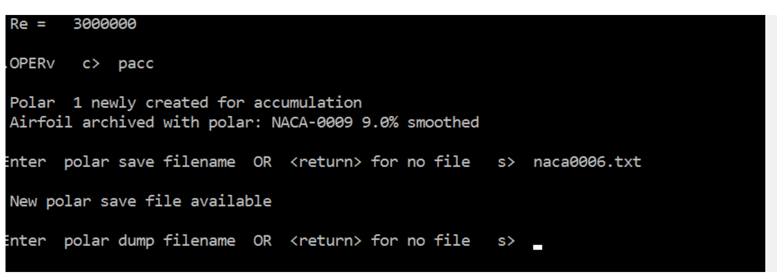
to achieve more accurate results from the XFOIL analysis. “ppar” will be entered in the command line which will display a 2D view of the airfoil. The user will then enter “n”, ENTER, “360”, ENTER, “t”, ENTER, “1”, ENTER, ENTER in that specific order in order to change the node size, paneling, and return to the operation interface. After



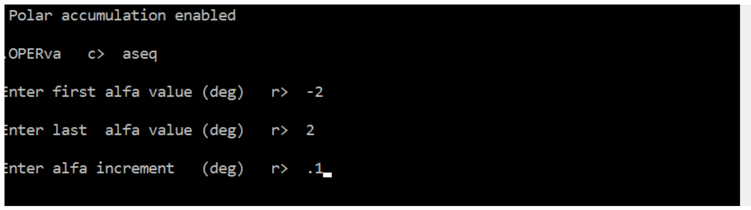
changing the paneling and node size, the user will enter “oper”, ENTER, “iter”, “100”, ENTER, “v”, ENTER to change the iteration size and enter a Reynolds Number they



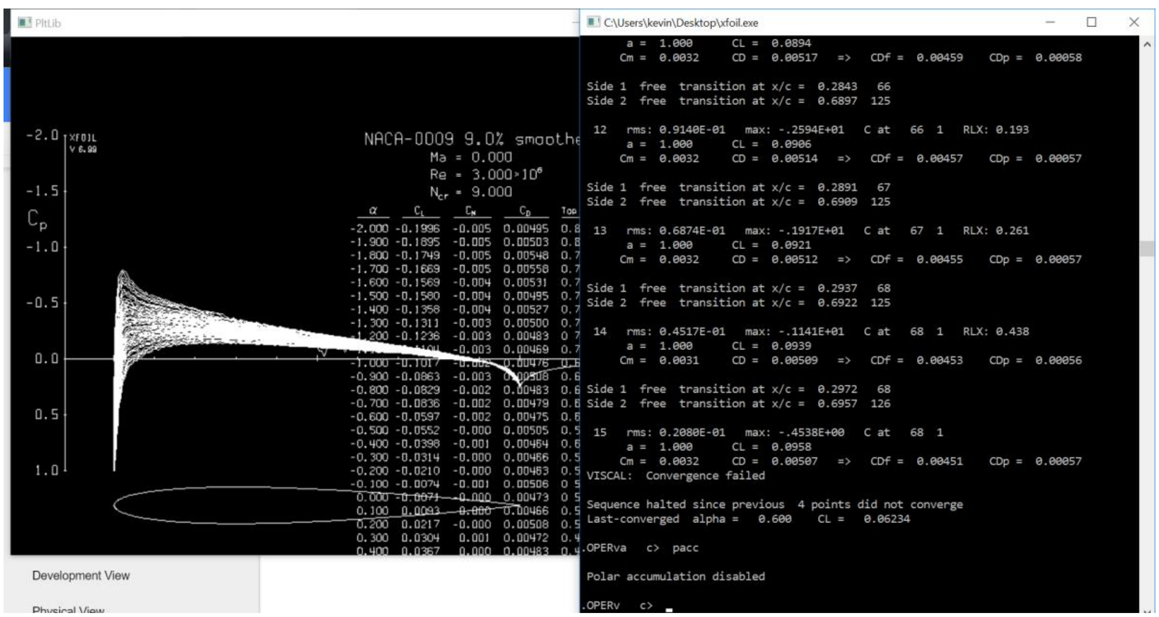
would like to analyze the airfoil for. Common Reynolds Number used are between 1,000,000 and 4,000,000 for cruising conditions. The user is now in viscous mode and can begin their polar accumulation. “pacc”, ENTER will then be entered. Following, XFOIL will prompt for a polar save filename. IMPORTANT: The user MUST enter a



filename with a .txt extension for the MFC application to read it in. ENTER must be hit again to save the file. Once the polar accumulation has been started the user can then enter the sweep of angles-of-attack they would like data for. This is done by executing "aseq", ENTER. XFOIL will ask for the first, last, and increment alpha. The user will input their desired values and lock each in by hitting ENTER. As soon as ENTER is hit

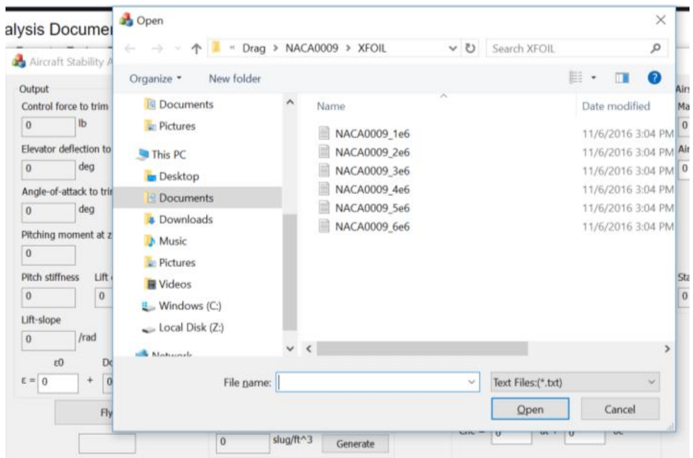


after the increment alpha is input, XFOIL will start its analysis. The user will let XFOIL run until finished, where a message displaying "Sequence halted" or "Sequence complete" will display at the end of the script. This signifies XFOIL has completed the analysis. The user will then type "pacc", ENTER to close and save the .txt polar file. This is the file which will be input into the MFC application.

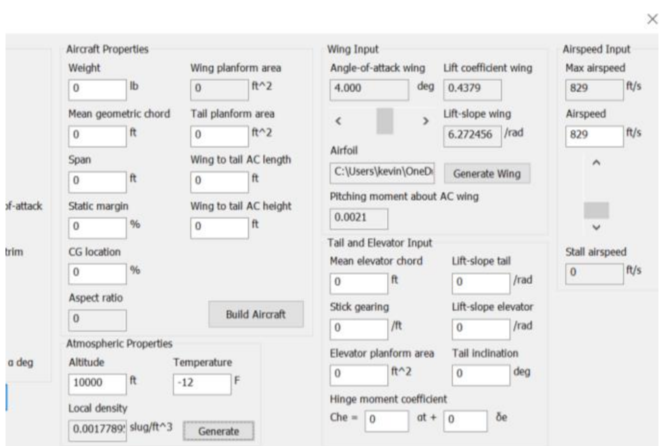


8.2 Aircraft Stability Analysis

The Aircraft Stability Analysis program can then be utilized. To use this software, the user needs to follow a direction of inputs for the program to run properly. First, the user opens the GUI and clicks the "Generate Wing" button. This button opens the computer directory and allows users to search for text files only. The user will choose the post-

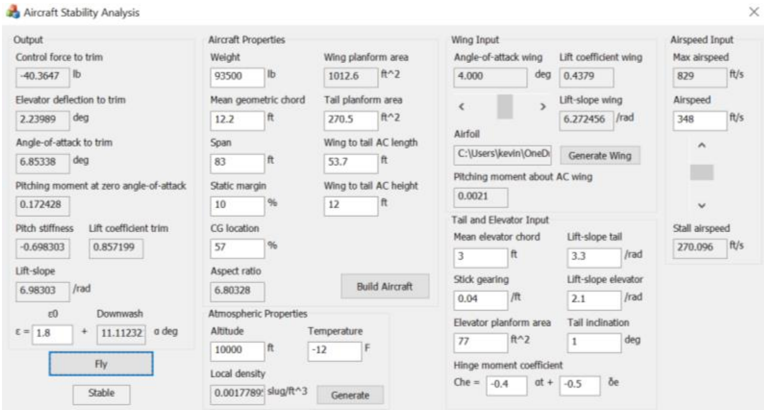


generated XFOIL polar accumulation file created and hit "Open". This now allows the user to scroll through angles-of-attack in the text file along with associated parameters. After choosing an angle-of-attack via the horizontal scrollbar the user will now generate the atmosphere in which their aircraft will be flying. Altitude and temperature will be input and "Generate" will be hit by the user. This will show the local density at which they will be flying and set max airspeed. After the user generates the atmosphere, the airspeed scrollbar will be enabled and the user can scroll to the airspeed at which they will be flying. The scrollbar will only scroll as far as max airspeed reads. After airspeed



is chosen, the user will input all aircraft parameters into Aircraft Properties and Tail and Elevator Input and click "Build Aircraft". This will generate all properties associated with the aircraft and calculate the aspect ratio along with wing planform area. After building the aircraft the user is ready to fly. The user will enter ε_0 and click "Fly".

Output for various aircraft parameters will be displayed to the user. These parameters can be used in future dynamic calculations.



9. Appendix

9.1 *Nomenclature*

Aircraft Properties

$Weight = W$

$Mean\ geometric\ chord = \bar{c}$

$Span = b$

$Static\ margin = SM = h_n - h$

$CG\ location = h$

$Aspect\ ratio = AR$

$Wing\ planform\ area = S_w = bc$

$Tail\ planform\ area = S_t$

$Wing\ to\ tail\ AC\ length = \bar{l_t}$

$Wing\ to\ tail\ AC\ height = h_H$

$Planform\ area = S$

Atmospheric Properties

$Altitude = PA$

$Temperature = OAT$

$Local\ density = \rho$

$Mach = M = .8$

$Specific\ heat = \gamma = 1.4$

$Universal\ Gas\ Constant = R = 1716$

$Pressure = P$

$Lapse\ rate = T_{lapse}$

Wing Input

$Angle - of - attack\ wing = \alpha_{wb}$

$Lift\ coefficient\ wing = C_{Lwb}$

$Lift - slope\ wing = a_{wb} = \dfrac{C_{Lwb}}{\alpha_{wb}}$
$Pitching\ moment\ about\ AC\ wing = C_{macwb}$
$Lift\ coefficient\ max = C_{Lmax}$

Tail and Elevator Input
$Mean\ elevator\ chord = \overline{c_e}$
$Stick\ gearing = G$
$Elevator\ planform\ area = S_e$
$Lift - slope\ tail = a_t$
$Lift - slope\ elevator = a_e$
$Tail\ inclination = i_t$
$Hinge\ moment\ coefficient = C_{he} = b_1\alpha_t + b_2\delta_e$

Airspeed Input
$Max\ airspeed = V_{max}$
$Airspeed = V$
$Stall\ airspeed = V_{stall}$

Output
$Control\ force\ to\ trim = P$
$Elevator\ deflection\ to\ trim = \delta_{etrim}$
$Angle - of - attack\ to\ trim = \alpha_{trim}$
$Pitching\ moment\ at\ zero\ angle - of - attack = C_{m0}$
$Pitch\ stiffness = C_{m\alpha}$
$Lift\ coefficient\ trim = C_{Ltrim}$
$Lift - slope = a$
$Downwash\ constant = k_A\ and\ k_H$
$Downwash = \dfrac{\partial\varepsilon}{\partial\alpha}$
$Downwash\ effect = \varepsilon_0$
$Neutral\ point = h_n$
$Neutral\ point\ wing = h_{nwb}$
$Determinant = det$
$Free\ elevator\ factor = F$
$Effect\ of\ elevator\ on\ lift = C_{L\delta_e}$
$Effect\ of\ elevator\ on\ moment = C_{m\delta_e}$
$Lift - slope\ free\ elevator = a'$
$Control\ free\ neutral\ point = h_n'$
$Hinge\ stiffness = C_{he_\alpha}$
$Hinge\ moment\ at\ zero\ angle - of - attack = C_{he0}$

9.2 Equations

Generate

$$T_{lapse} = (OAT + 460) - \left(\frac{3.57}{1000} PA\right)$$

$$P = 2116 \left(\frac{T_{lapse}}{(OAT + 460)}\right)^{5.256}$$

$$\rho = \frac{P}{\left(R(OAT + 460)\right)}$$

$$V_{max} = M\sqrt{\gamma R(OAT + 460)}$$

Build Aircraft

$$S = S_w + S_t$$

$$AR = \frac{b}{c}$$

Fly

$$V_{stall} = \sqrt{\frac{2W}{\rho C_{Lmax} S}}$$

$$C_{Ltrim} = \frac{2W}{\rho V^2 S}$$

$$k_A = \frac{1}{AR} - \frac{1}{1 + AR^{1.17}}$$

$$k_H = \frac{1 - \left|\frac{h_H}{b}\right|}{\sqrt[3]{\frac{2\overline{l_t}}{b}}}$$

$$\frac{\partial \varepsilon}{\partial \alpha} = 4.44 k_A k_H$$

$$a = C_{L\alpha} = a_{wb}\left[1 + \frac{a_t S_t}{a_{wb} S}\left(1 - \frac{\partial \varepsilon}{\partial \alpha}\right)\right]$$

$$\overline{V_H} = \frac{S_t \overline{l_t}}{S\bar{c}}$$

$$h_n = SM + h$$

$$h_{nwb} = h_n + \frac{a_t}{a}\overline{V_H}\left(1 - \frac{\partial \varepsilon}{\partial \alpha}\right)$$

$$C_{m\alpha} = a(h - h_{nwb}) - a_t \overline{V_H}\left(1 - \frac{\partial \varepsilon}{\partial \alpha}\right)$$

$$det = -a a_e \frac{S_t}{S}\left(\frac{\overline{l_t}}{\bar{c}} + h_{nwb} - h_n\right)$$

$$F = \left(1 - \frac{a_e b_1}{a_t b_2}\right)$$

$$C_{L\delta_e} = a_e \frac{S_t}{S}$$
$$C_{m\delta_e} = C_{L\delta_e}(h - h_{nwb}) - \overline{V_H} a_e$$
$$a' = a_{wb}\left[1 + \frac{F a_t S_t}{a_{wb} S}\left(1 - \frac{\partial \varepsilon}{\partial \alpha}\right)\right]$$
$$h_n' = h_{nwb} + \frac{F a_t}{a'}\overline{V_H}\left(1 - \frac{\partial \varepsilon}{\partial \alpha}\right)$$
$$C_{m0} = C_{macwb} + a_t \overline{V_H}(\varepsilon_0 + i_t)\left[1 - \frac{a_t S_t}{a S}\left(1 - \frac{\partial \varepsilon}{\partial \alpha}\right)\right]$$
$$C_{he_\alpha} = b_1\left(1 - \frac{\partial \varepsilon}{\partial \alpha}\right)$$
$$\delta_{etrim} = -\frac{(C_{m0} C_{L\alpha} + C_{m\alpha} C_{Ltrim})}{det}$$
$$\alpha_{trim} = \frac{C_{m0} C_{L\delta_e} + C_{m\delta_e} C_{Ltrim}}{det}$$
$$C_{he0} = -b_1(i_t + \varepsilon_0)\left[1 - \frac{a_t S_t}{a S}\left(1 - \frac{\partial \varepsilon}{\partial \alpha}\right)\right]$$
$$A = -G S_e \overline{c_e} \frac{W}{S} \frac{a' b_2}{det}(h - h_n')$$
$$B = G S_e \overline{c_e}\left[C_{he0} + \frac{C_{m0}}{det}\left(C_{he_\alpha} C_{L\delta_e} - b_2 C_{L\alpha}\right)\right]$$
$$P = A + B \frac{1}{2}\rho V^2$$