SAE Aerodesign West 2021

Deliverable

Schulich Aerodesign

Software Sub team

XFLR5 Automation Script

CODED BY:

Zeeshan Salim Chougle

Submitted: **March 7, 2021**

TABLE OF CONTENTS Page No

1. Introduction

1.1 Overview of the project

XFLR5 is a great tool for low reynolds number analysis of UAVs but iterating through designs is a cumbersome process. We are developing a python based automation script which not only automates the fundamental features of XFLR5 namely: Creating airfoils, Airfoil analysis, Creating plane designs, Flight analysis and Stability analysis, but is also able to run multiple design iterations efficiently. The goal of the project is to allow the members of the Wing and Stability teams to compare through countless possible plane designs and narrow down on the most ideal one to be manufactured for the SAE competition.

The script is coded entirely in python and makes use of Pyautogui package which is freely available in the python library, the package primarily allows the developer to control the mouse and keyboard functions using python code which can then be used to automate various web applications. The script automates the processes by following a sequence of pre-coded mouse and keyboard instructions which execute differently during each iteration depending on the inputs provided by the user in order to obtain the desired outcome. Additionally, Tkinter python package is used to develop a user friendly GUI in order to receive the required inputs from the user.

1.2 Need for the project

Manual iterations of plane design and analysis is a cumbersome task especially when you have to perform a large batch of analysis to choose the most optimal plane design.

The script provides a solution to this problem by providing automatically generated statistical results of all analysis for each different plane design to the user which makes the process much more efficient and faster.

1.3 Objectives of the project

- Make the automation script as modular as possible

  In order to keep the code as modular as possible we created 5 individual modules to automate the 5 fundamental features of XFLR5 software. These modules include:

  1. Creating_airfoil
  2. Airfoil_analysis
  3. Plane_design
  4. Plane_analysis

5. Stability_analysis

Each of the module possesses the ability to function independently which allows the user to freely jump back and forth between different processes rather than performing all of them in a sequence. In addition to this, each module contains several functions and data types which can be imported from one module to another, allowing for reusability of data along with a sophisticated code structure, making the process of debugging much more easier. All these modules are finally called in the “Menu module” in a particular sequence depending on the users input.

- Set up well defined Fail safes.

This automation script works by following a sequence of clicks and keyboard-typewrites which requires all the windows and drop boxes to be placed exactly in the same location during each iteration. However, it is possible that in certain situations the windows might not be in full screen or placed in the proper spot due to which the code might be clicking and typing commands in unexpected ways and thus requires a fail safe.

This issue has been resolved by employing two fail-safe strategies:

(i) Automatic code termination: The code has been structed in a way such that it automatically terminates the program if it does not find the intended menu/window open during run-time.

(ii) Manual code termination: The code can be manually terminated by moving the cursor to any of the four corners of the window.

- Implement GUI for user friendly input.

In order to make the code as user friendly as possible even for users inexperienced with programming languages, GUI has been setup as the primary method for obtaining inputs from the user. Sperate GUIs have been implemented for all the individual modules with text boxes and radio buttons in order to simplify the process of receiving inputs from the user. A pre-formatted text file has been attached with the script which receives all the input from the GUI and acts as a database for storing those inputs for future iterations. Any changes to the inputs made in the GUI is directly read by the python script and implemented during the automation process.

- Maintain simple code structure for future improvements.

  We have tried to maintain a modular code structure to allow for easy debugging of code. Apart from this, proper indentations have been made for each line of code and sufficient comments have been added describing each section of the code which would make it much easier for future improvement/changes.

- Develop optimization script for performing multiple iteration

  The goal is to call the 5 fundamental modules, n number of times with certain automatic variations (incrementations/decrementations) to the parameters in the main module in order to perform a large batch of automated analysis for different plane designs.

- Implement the ability to save all analysis results to users PC for more efficient decision making.

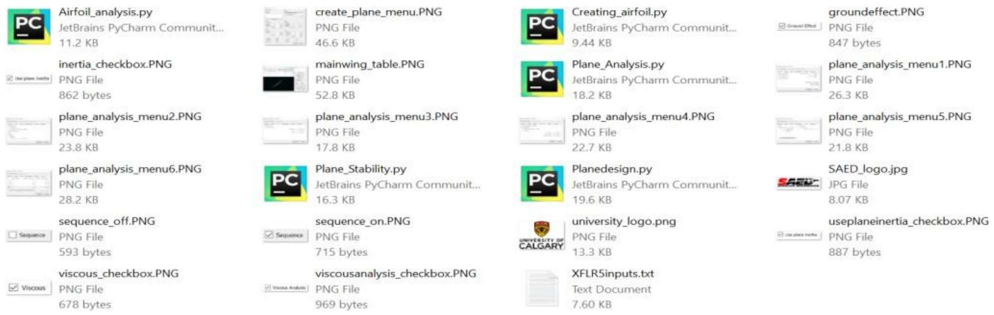
  The goal is to make use of the screenshot feature provided by pyautogui to take screenshots of all the statistical/non-statistical results and directly save them on the users PC allowing them to compare the results of all the analysis simultaneously at their own convenience.

1.4 Overview of existing systems and technology

Initially, when we began thinking of ideas to come up with an automation script for XFLR5, we thought of using C++ to automate the process by manipulating the source code or using python to develop our own version of XFLR5 wherein we would code all the formulas required to perform the calculation for different parameters and develop the resulting statistical outcomes for a large batch of plane designs. However, both approaches would take a significant amount of work and had reduced chances of success as we are not so familiar in those fields. So, we finalized on the concept of using pyautogui package of python to automate XFLR5, which we had discovered while researching any existing automation modules.

2. Setup

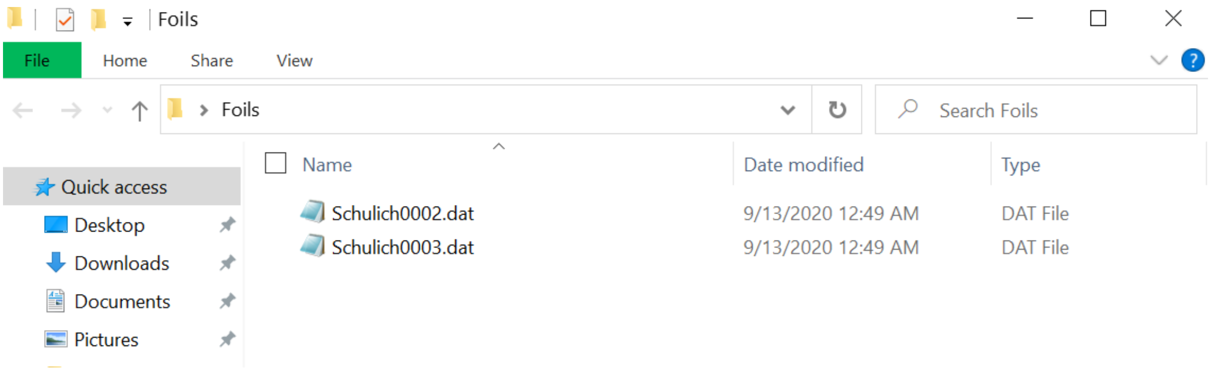
1. Download the code from GitHub and transfer all the files in a single folder. Having separate folders for images and text file even within the same root directory would make them unreachable to the script.



Airfoil_analysis.py JetBrains PyCharm Communit... 11.2 KB	create_plane_menu.PNG PNG File 46.6 KB	Creating_airfoil.py JetBrains PyCharm Communit... 9.44 KB	groundeffect.PNG PNG File 847 bytes
inertia_checkbox.PNG PNG File 862 bytes	mainwing_table.PNG PNG File 52.8 KB	Plane_Analysis.py JetBrains PyCharm Communit... 18.2 KB	plane_analysis_menu1.PNG PNG File 26.3 KB
plane_analysis_menu2.PNG PNG File 23.8 KB	plane_analysis_menu3.PNG PNG File 17.8 KB	plane_analysis_menu4.PNG PNG File 22.7 KB	plane_analysis_menu5.PNG PNG File 21.8 KB
plane_analysis_menu6.PNG PNG File 28.2 KB	Plane_Stability.py JetBrains PyCharm Communit... 16.3 KB	Planedesign.py JetBrains PyCharm Communit... 19.6 KB	SAED_logo.jpg JPG File 8.07 KB
sequence_off.PNG PNG File 593 bytes	sequence_on.PNG PNG File 715 bytes	university_logo.png PNG File 13.3 KB	useplaneinertia_checkbox.PNG PNG File 887 bytes
viscous_checkbox.PNG PNG File 678 bytes	viscousanalysis_checkbox.PNG PNG File 969 bytes	XFLR5inputs.txt Text Document 7.60 KB	

Screenshot showing all files placed in a single folder

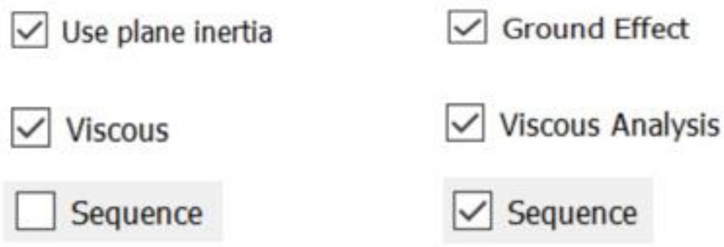
2. Create a Folder on desktop which stores all the air foils you want to import in XFLR5. This folder should contain only .dat files and every file placed in this folder will be opened in XFLR5 so add/remove the foils accordingly.



□ Name	Date modified	Type
Schulich0002.dat	9/13/2020 12:49 AM	DAT File
Schulich0003.dat	9/13/2020 12:49 AM	DAT File

Screenshot showing a sample folder containing foils

3. Take screenshots of the following checkboxes (the ON or OFF status must exactly be as shown in the images) from the XFLR5 menu and store them in the same folder as the rest of the code:



☑ Use plane inertia

☑ Ground Effect

☑ Viscous

☑ Viscous Analysis

□ Sequence

☑ Sequence

Images of all checkboxes to be screenshot

3. Instruction for usage

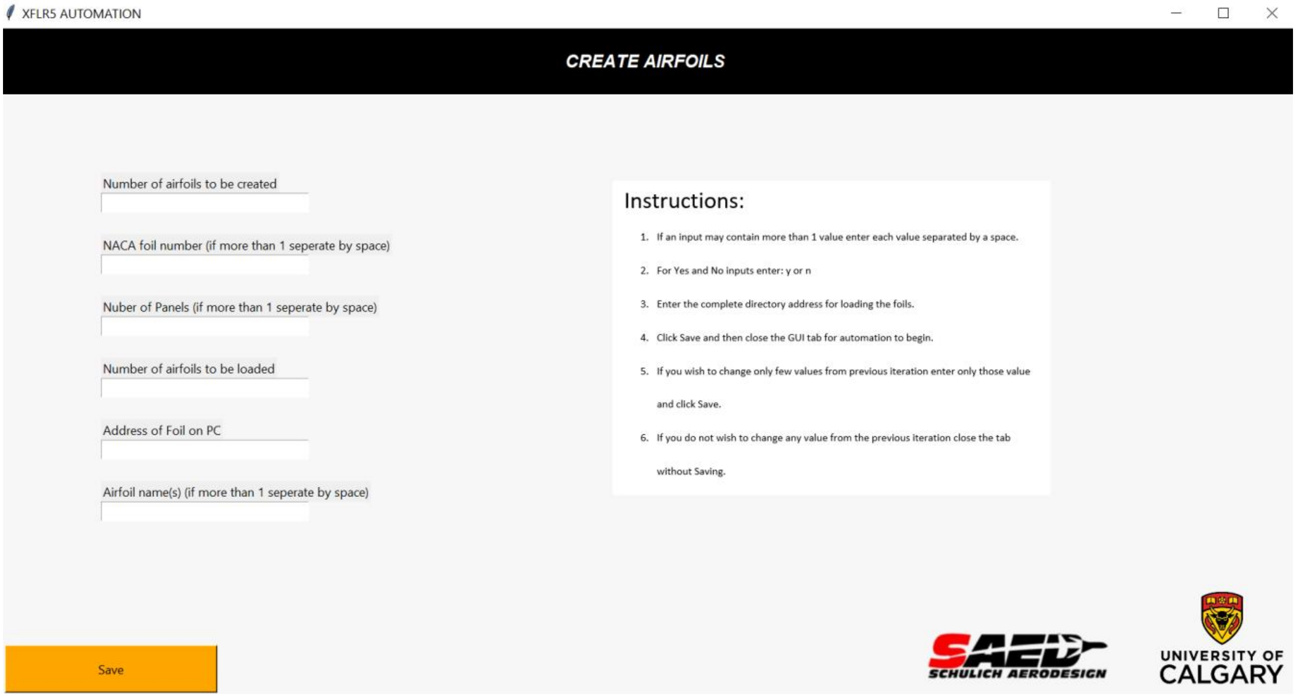
It is recommended to use an IDE to run the script due to their easy to use play/pause options.

1. When you run the code make sure XFLR5 is open in the background and is minimized.

2. There are two ways to input data to the script:

(i) GUI (preferred way):

As soon as you run the code a GUI will pop up enter the data and make sure to click “SAVE”. Once your data is saved close the tab and your automation process will begin. You do not need to switch to XFLR5 window, if the XFLR5 window was minimized in the background the code will automatically open the XFLR5 window as soon as you close the GUI tab. If you do not wish to change all the parameters from your previous run and only a few parameters are to be changed, enter only those parameters in the GUI and click Save , all the other parameters will directly receive the values of your last run/iteration of the script. An example has been provided below:



Number of airfoils to be created

NACA foil number (if more than 1 seperate by space)

Nuber of Panels (if more than 1 seperate by space)

Number of airfoils to be loaded

Address of Foil on PC

Airfoil name(s) (if more than 1 seperate by space)

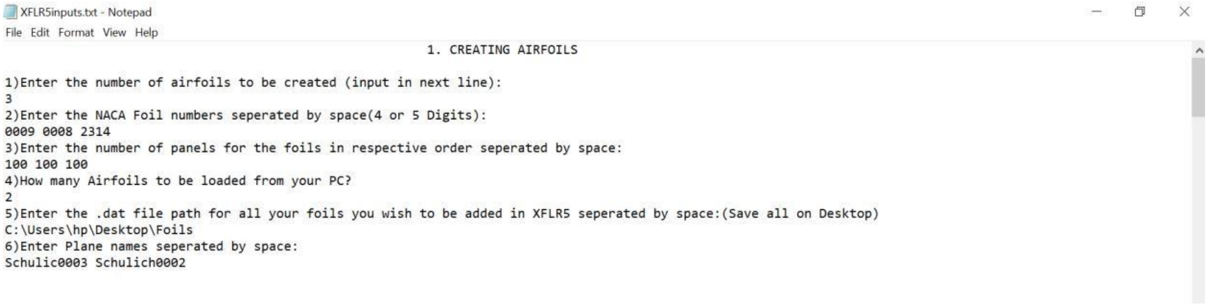
Instructions:

1. If an input may contain more than 1 value enter each value separated by a space.
2. For Yes and No inputs enter: y or n
3. Enter the complete directory address for loading the foils.
4. Click Save and then close the GUI tab for automation to begin.
5. If you wish to change only few values from previous iteration enter only those value and click Save.
6. If you do not wish to change any value from the previous iteration close the tab without Saving.

The blank parameters will take the values from the last time user ran the code

(ii) Text File:

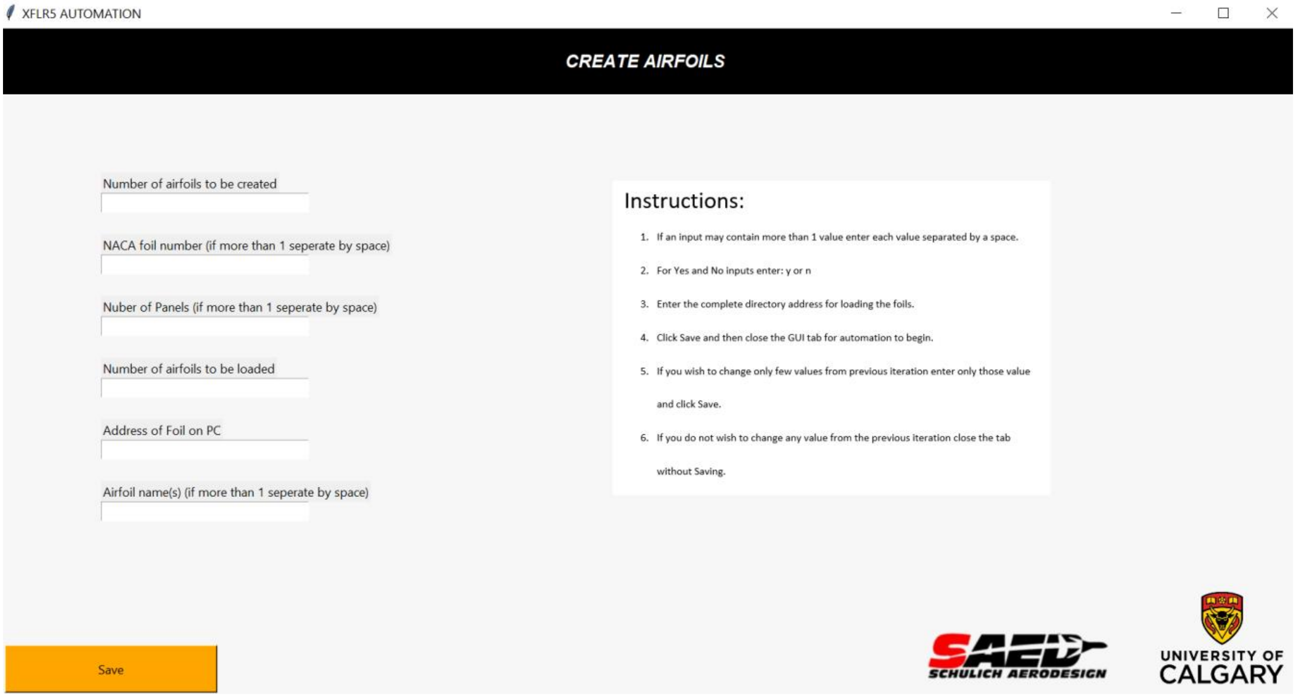
Another way to enter data to the script is by using the text file provided with the script. In order to use the text file, open the text file and change all the parameters you want to change and make sure to save the text file in the end. Now, when you run the code and the GUI pops up close the window without entering anything and the automation should begin with the data you entered in the text file. An example has been provided below:



```
                                                1. CREATING AIRFOILS

1)Enter the number of airfoils to be created (input in next line):
3
2)Enter the NACA Foil numbers seperated by space(4 or 5 Digits):
0009 0008 2314
3)Enter the number of panels for the foils in respective order seperated by space:
100 100 100
4)How many Airfoils to be loaded from your PC?
2
5)Enter the .dat file path for all your foils you wish to be added in XFLR5 seperated by space:(Save all on Desktop)
C:\Users\hp\Desktop\Foils
6)Enter Plane names seperated by space:
Schulic0003 Schulich0002
```

Change the parameters according to your requirement and save



CREATE AIRFOILS

Number of airfoils to be created

NACA foil number (if more than 1 seperate by space)

Nuber of Panels (if more than 1 seperate by space)

Number of airfoils to be loaded

Address of Foil on PC

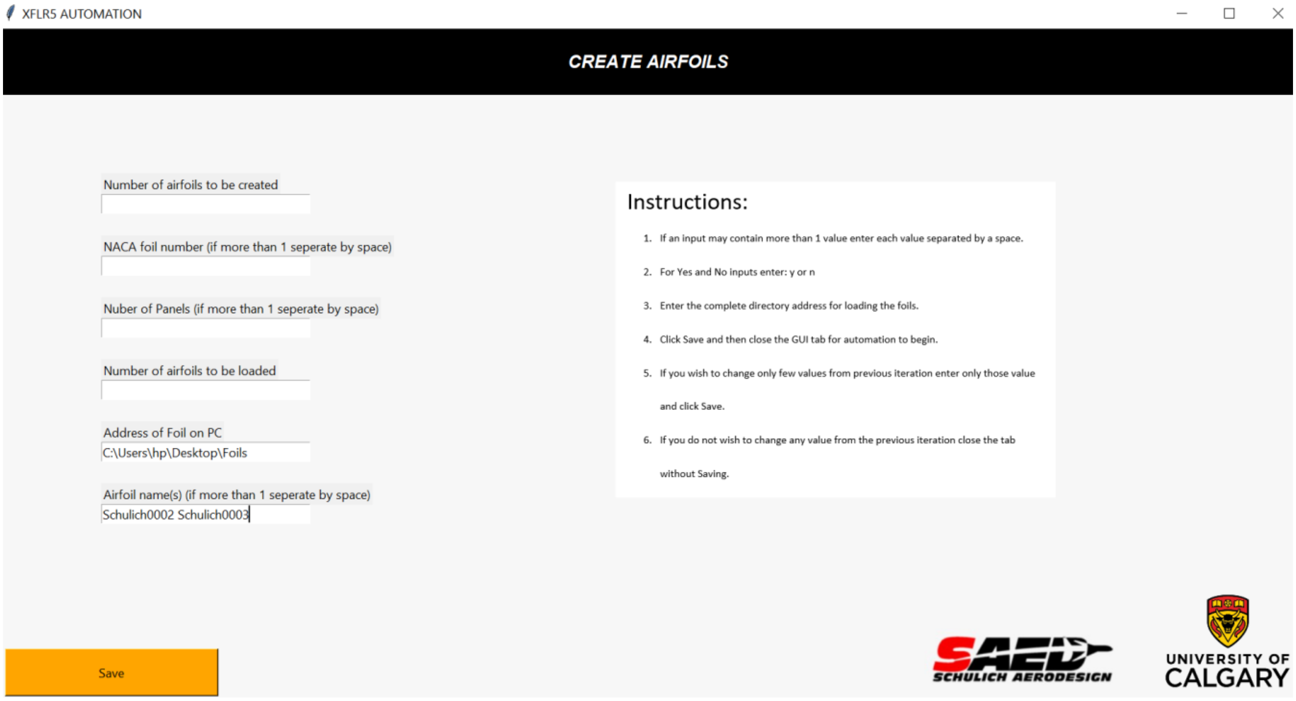
Airfoil name(s) (if more than 1 seperate by space)

Instructions:

1. If an input may contain more than 1 value enter each value separated by a space.
2. For Yes and No inputs enter: y or n
3. Enter the complete directory address for loading the foils.
4. Click Save and then close the GUI tab for automation to begin.
5. If you wish to change only few values from previous iteration enter only those value and click Save.
6. If you do not wish to change any value from the previous iteration close the tab without Saving.

Close the GUI blank without entering any parameters

3. When using the script for the first time make sure to add the “Foils” folder address inthe create foil module as shown below:



CREATE AIRFOILS

Number of airfoils to be created

NACA foil number (if more than 1 seperate by space)

Nuber of Panels (if more than 1 seperate by space)

Number of airfoils to be loaded

Address of Foil on PC
C:\Users\hp\Desktop\Foils

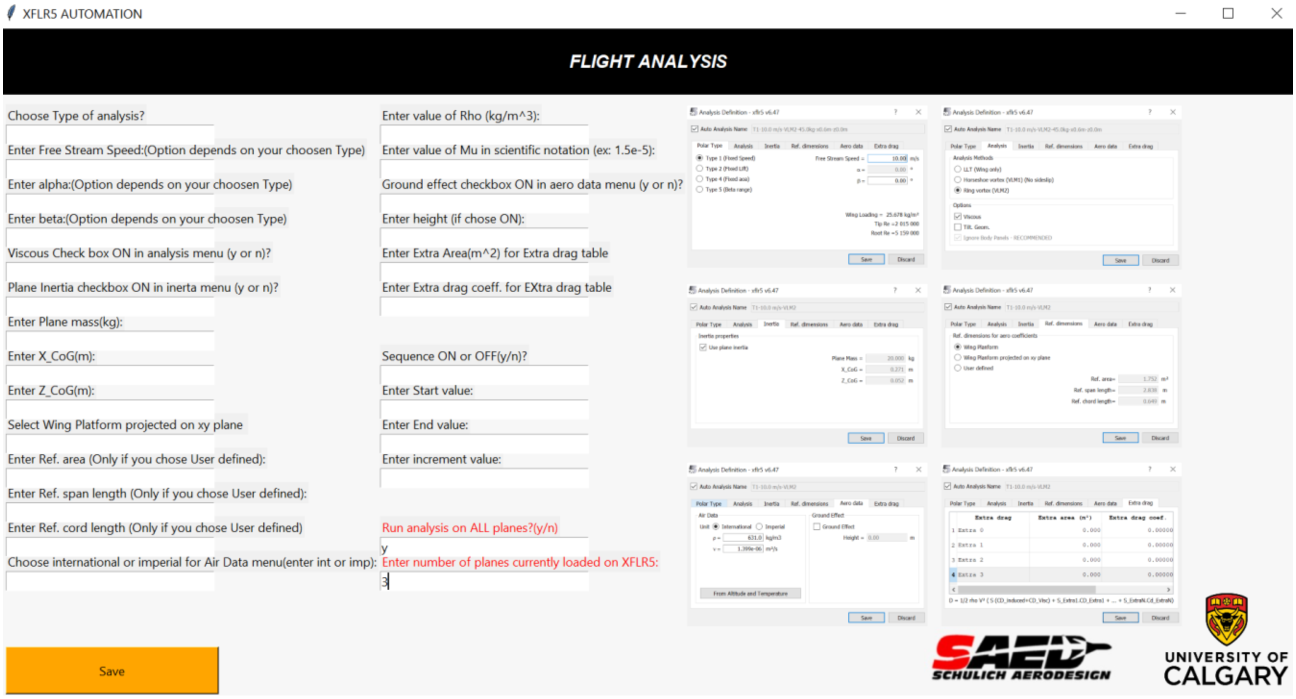
Airfoil name(s) (if more than 1 seperate by space)
Schulich0002 Schulich0003

Instructions:

1. If an input may contain more than 1 value enter each value separated by a space.
2. For Yes and No inputs enter: y or n
3. Enter the complete directory address for loading the foils.
4. Click Save and then close the GUI tab for automation to begin.
5. If you wish to change only few values from previous iteration enter only those value and click Save.
6. If you do not wish to change any value from the previous iteration close the tab without Saving.

This needs to be done at least once so the code can access your foils to be loaded in XFLR5

4. To utilize the optimization part of the script, the red highlighted text boxes in flight and stability analysis have to be filled. The instructions have been demonstrated below :


XFLR5 AUTOMATION

FLIGHT ANALYSIS

Choose Type of analysis?

Enter Free Stream Speed:(Option depends on your choosen Type)

Enter alpha:(Option depends on your choosen Type)

Enter beta:(Option depends on your choosen Type)

Viscous Check box ON in analysis menu (y or n)?

Plane Inertia checkbox ON in inerta menu (y or n)?

Enter Plane mass(kg):

Enter X_CoG(m):

Enter Z_CoG(m):

Select Wing Platform projected on xy plane

Enter Ref. area (Only if you chose User defined):

Enter Ref. span length (Only if you chose User defined):

Enter Ref. cord length (Only if you chose User defined)

Choose international or imperial for Air Data menu(enter int or imp):

Enter value of Rho (kg/m^3):

Enter value of Mu in scientific notation (ex: 1.5e-5):

Ground effect checkbox ON in aero data menu (y or n)?

Enter height (if chose ON):

Enter Extra Area(m^2) for Extra drag table

Enter Extra drag coeff. for EXtra drag table

Sequence ON or OFF(y/n)?

Enter Start value:

Enter End value:

Enter increment value:

Run analysis on ALL planes?(y/n)

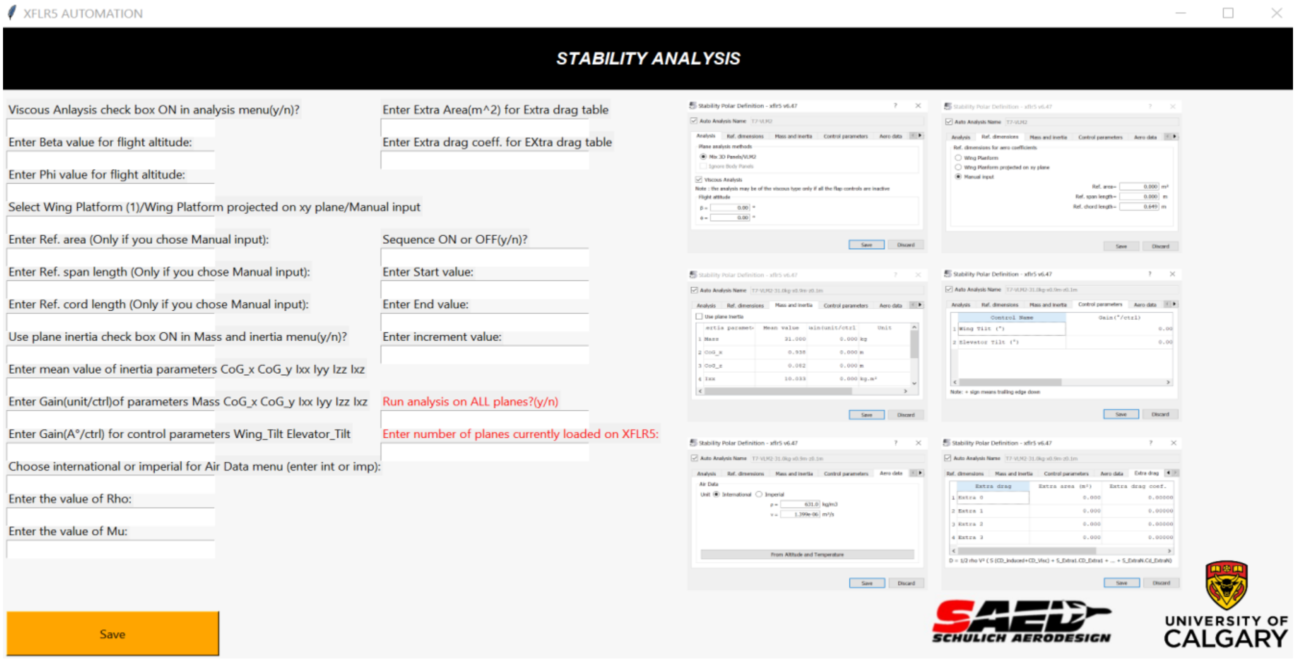
y

Enter number of planes currently loaded on XFLR5:

3

Save

The red highlighted text box have to be filled which would run the flight analysis for all the planes loaded on XFLR5


XFLR5 AUTOMATION

STABILITY ANALYSIS

Viscous Anlaysis check box ON in analysis menu(y/n)?

Enter Beta value for flight altitude:

Enter Phi value for flight altitude:

Select Wing Platform (1)/Wing Platform projected on xy plane/Manual input

Enter Ref. area (Only if you chose Manual input):

Enter Ref. span length (Only if you chose Manual input):

Enter Ref. cord length (Only if you chose Manual input):

Use plane inertia check box ON in Mass and inertia menu(y/n)?

Enter mean value of inertia parameters CoG_x CoG_y Ixx Iyy Izz Ixz

Enter Gain(unit/ctrl)of parameters Mass CoG_x CoG_y Ixx Iyy Izz Ixz

Enter Gain(A°/ctrl) for control parameters Wing_Tilt Elevator_Tilt

Choose international or imperial for Air Data menu (enter int or imp):

Enter the value of Rho:

Enter the value of Mu:

Enter Extra Area(m^2) for Extra drag table

Enter Extra drag coeff. for EXtra drag table

Sequence ON or OFF(y/n)?

Enter Start value:

Enter End value:

Enter increment value:

Run analysis on ALL planes?(y/n)

Enter number of planes currently loaded on XFLR5:

Save

The red highlighted text box have to be filled which would run the stability analysis for all the planes loaded on XFLR5

5. Lastly, Failsafe has been implemented in all the 5 modules. If any part of the code does not appear to be functioning as expected by the user, fail safe can be activatedby moving the cursor to any of the 4 corners of the window which will instantly terminate the code.

4. Feasibility

4.1 Technical feasibility

The script is completely python based and uses the pyautogui package which are freely available online and the technical skills required are manageable. Time limitations of the script development and the ease of implementing using these technologies are synchronized.

4.2 Resources and time feasibility:

Resources that are required for the Script development include:

1. Programming device (Laptop):

   No high specification computer device required neither to program nor to run the script.

2. Programming tools (freely available):

   This Script requires only XFLR5 and python (with pyautogui) which are freely available online to download.

3. Programming individuals.

   Individuals proficient in python and with decent knowledge of XFLR5 are required for this project.

Currently these are the existing system and technologies we have incorporated with our script:

- XFLR5 : This is the software used for plane design/analysis.
- Python : Coding script in python while using pyautogui package.
- Python : GUI development using Tkinter package for inputs.
- Notepad : For Text file based input database.

Automating XFLR5 by its GUI prohibits us from running anything on large computer clusters without significant additional modification, and makes it difficult to parallelize the code, other than the built-in process decomposition that XFLR5 contains. However, XFLR5 runs quickly and is very computationally inexpensive. The speedup we would stand to gain from a more complicated but more closely integrated software package would be of little benefit. Thus, by taking advantage of pyautogui's ability to allow the developer to automate all the processes efficiently without the need to integrate the script with the fundamental XFLR5 software, we hope to speedup each process of plane design/analysis at least 5x faster than manually possible.

So it's clear that the project has the required resource and time feasibility.

5. Considerations

5.1 Performance

Sufficient time delay has been provided between each automation process keeping in mind the hardware capabilities of different users. The Script should work smoothly for all Windows users. Testing is underway for macOS users. The script does not require high specs computer systems to run the automation process, should function smoothly for any system with at least 4 GB RAM.

5.2 Safety measures

This automation script works by following a sequence of clicks and keyboard-typewrites which requires certain windows and drop boxes to be placed exactly in the same location during each iteration. However, it is possible that in certain situations the windows might not be in full screen or placed in the proper spot due to which the code might operate in unexpected ways and thus requires a fail safe. To resolve this issue fail safe has been setup in the entirety of the code using the inbuilt fail safe feature provided by the pyautogui package. In case of failure of functioning of the script following instructions are to be followed by the user:

- In case the code does not behave as expected by the user, moving the cursor to any of the 4 corners of the screen should terminate the code at any point of the runtime.

Additionally, code has been implemented to check whether the intended window is open in each part of the processes. If the code fails to detect the window the script automatically terminates to avoid any form of unexpected behaviour.

5.3 Usability and ease of use:

Users will be provided with a complete user manual as a pdf. The script is designed to make it easy for any potential user to get familiar with the script within 10 minutes. GUI have been setup with instructions indicating where to enter the parameters for different processes along with the images of the menu where the user would supposedly be entering the data in XFLR5, In order to make it convenient to use the script even without any prior knowledge of programming. No additional training is required to use the script.

6. Testing Phase result and improvements

6.1 Issues detected and fixes:

1. While running the script on different systems having different screen sizes and resolutions it was detected that the pre-defined click functions were highly unreliable as the position of the buttons and textboxes varied depending on the screen resolution and size which caused the code to behave unexpectedly as it would click on unintended portions of the window for certain screen resolutions.

   Fix: This issue was resolved by replacing all the click functions with hot-key combinations as their functionality does not rely on the users screen resolution or size. Additionally, for a few processes where the use of mouse click was mandatory all the windows were adjusted to be placed at a certain specified region of the screen with a specified window size in order to counter the pixel variation.

2. The image recognition feature implemented to detect whether certain checkboxes were ticked on or off caused unnecessary delay as the code consumed a substantial amount of time to detect each checkbox which conflicted with our goal of making the automation process as efficient as possible.

   Fix: The time delay caused by image recognition was reduced by about 80% by specifying a region of the window where the checkbox can be expected to be found.

3. Certain XFLR5 processes opened windows that were placed on different sides of the monitor for different users which the script was unable to track in a few cases.

   Fix: This issue was resolved by coding all the windows to be placed exactly at a certain specified pixel region and to have the same size for all the users, in order to counteract the random positioning of the windows.

4. On creating an Air foil or a Plane with the same name, a overwrite menu popped up which the code did not account for and thus behaved unexpectedly.

   Fix: This issue was resolved by simply coding a few additional [‘enter’] presses at the end of those process, which now allows the users to freely overwrite existing Air foils and Planes.

5. The images of the checkboxes downloaded from GitHub for image recognition were not identified by systems with differed screen resolution.

   Fix: This issue can be resolved if the user takes a screenshot of those checkboxes as a part of the setup process.

6. The plane design module entered unexpected values in the tables in a few iterations randomly.

   Fix: The root cause of this issue was found to be the dysfunctionality of a few [‘enter’] presses which did not press when expected and thus entered parameters which were supposed to be entered in the next window. Adding sufficient time delay resolved this issue.

7. A few processes like the Multi-threaded batch analysis, flight analysis and stability analysis required the script to press enter after analyzing whether the process was completed or not.

   Fix: There are two possible fixes to this issue:

   (i) Using image recognition to observe when the process is completed and then press enter.

   (ii) Making a timely estimate of the process give a sufficient delay accordingly before pressing enter.

6.2 Further improvements that can be made:

1. A more efficient and reliable method of image recognition which does not require the user to manually take screenshots of the checkboxes.

2. A method to be able to deactivate a minimized active window .

3. A better way to keep track of the completion time for a XFLR5 process.

8.Conclusion

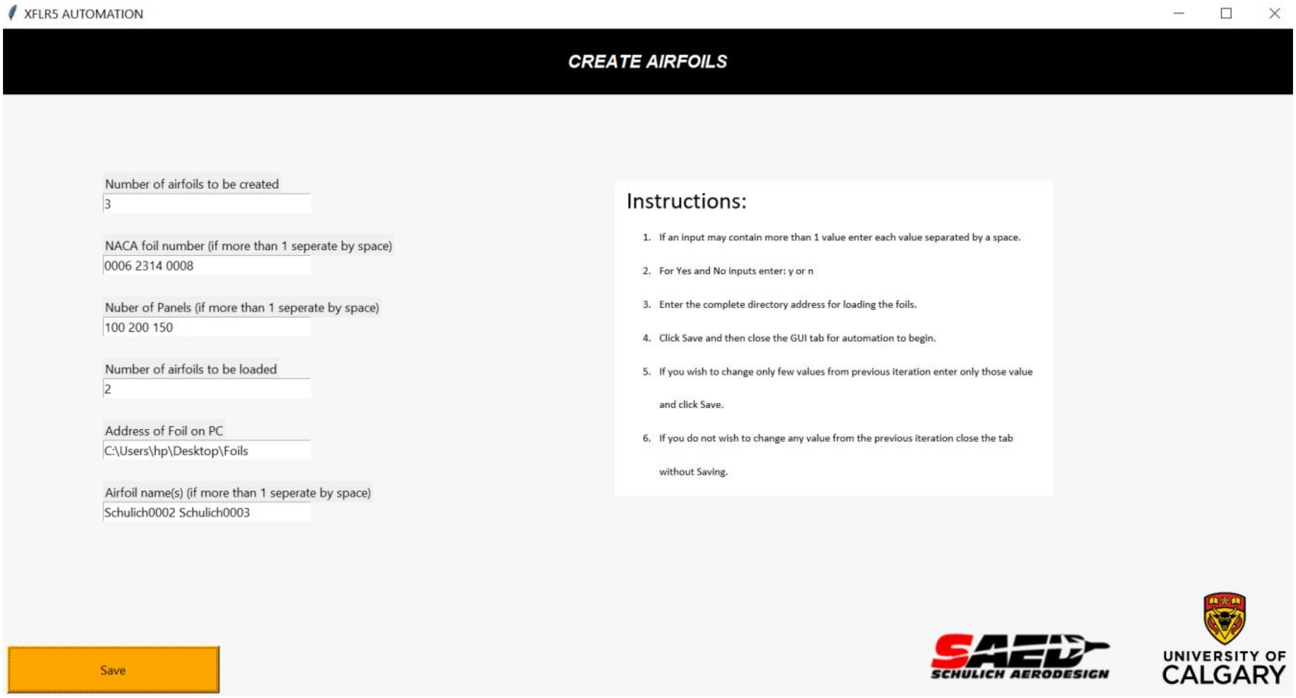
8.1 Summary

The main goal for this project was to come up with the most simplistic but efficient way to develop an automation script. After a lot of trial and error we finalized on developing an automation script using pyautogui package available in python library. To facilitate ease of user input for non-programming users, GUI has been implemented as the primary method for receiving input using Python’s Tkinter package. We have successfully developed a user-friendly script capable of automating all the fundamental features of XFLR5 with a significantly low margin of error during runtime. The optimization part of the script enables the users to run each analysis on all the planes loaded on XFLR5 which significantly lowers the time it takes for each individual plane’s flight and stability analysis.

8.2 Overview and interpretation of results obtained

1.Creating_airfoil:

(i) Input:



Number of airfoils to be created
3

NACA foil number (if more than 1 seperate by space)
0006 2314 0008

Nuber of Panels (if more than 1 seperate by space)
100 200 150

Number of airfoils to be loaded
2

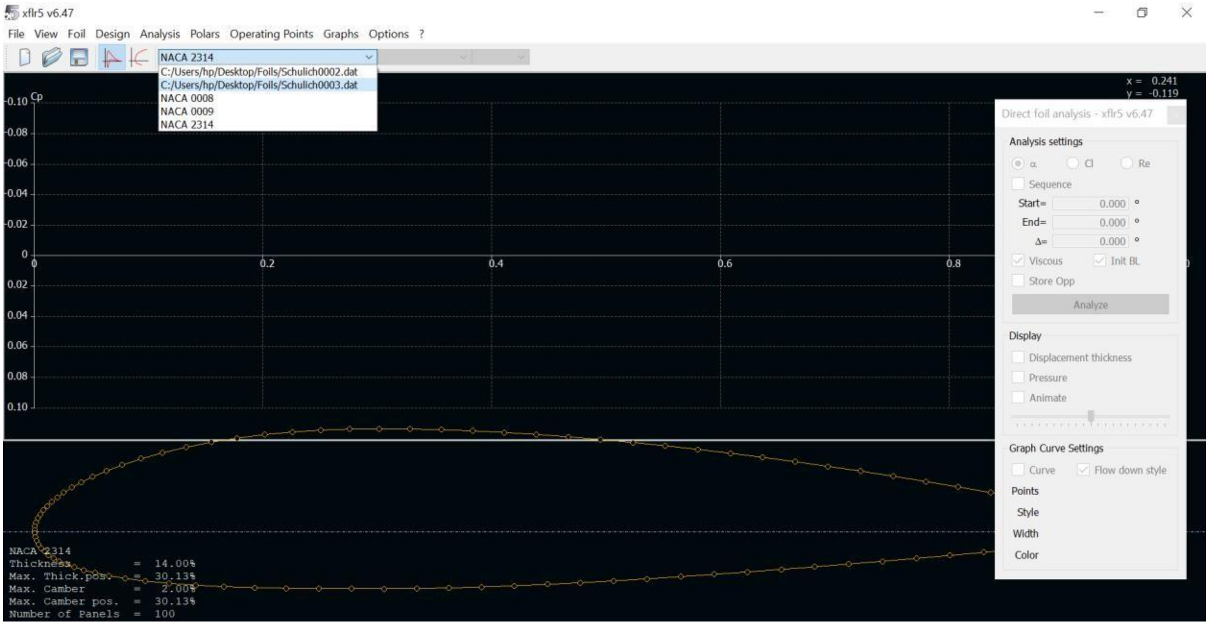
Address of Foil on PC
C:\Users\hp\Desktop\Foils

Airfoil name(s) (if more than 1 seperate by space)
Schulich0002 Schulich0003

Instructions:

1. If an input may contain more than 1 value enter each value separated by a space.
2. For Yes and No inputs enter: y or n
3. Enter the complete directory address for loading the foils.
4. Click Save and then close the GUI tab for automation to begin.
5. If you wish to change only few values from previous iteration enter only those value and click Save.
6. If you do not wish to change any value from the previous iteration close the tab without Saving.

(ii) Output (After automation):



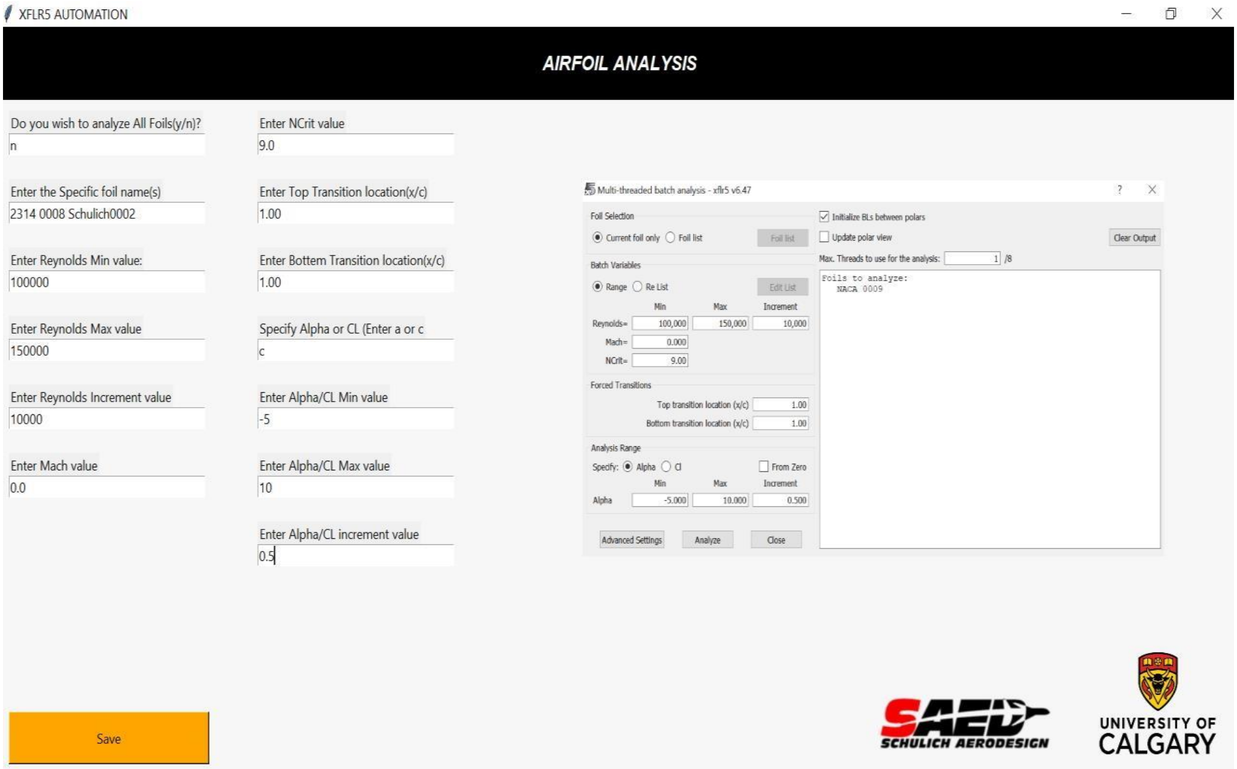
C:/Users/hp/Desktop/Foils/Schulich0002.dat
C:/Users/hp/Desktop/Foils/Schulich0003.dat
NACA 0008
NACA 0009
NACA 2314

□ Sequence
Start= 0.000 °
End= 0.000 °
Δ= 0.000 °
☑ Viscous ☑ Init BL
□ Store Opp

```
NACA 2314
Thickness          =  14.00%
Max. Thick.pos.    =  30.13%
Max. Camber        =   2.00%
Max. Camber pos.   =  30.13%
Number of Panels   =  100
```

2. Analysing_foil:

(i) input (GUI):



XFLR5 AUTOMATION

AIRFOIL ANALYSIS

Do you wish to analyze All Foils(y/n)?
n

Enter the Specific foil name(s)
2314 0008 Schulich0002

Enter Reynolds Min value:
100000

Enter Reynolds Max value
150000

Enter Reynolds Increment value
10000

Enter Mach value
0.0

Enter NCrit value
9.0

Enter Top Transition location(x/c)
1.00

Enter Bottom Transition location(x/c)
1.00

Specify Alpha or CL (Enter a or c
c

Enter Alpha/CL Min value
-5

Enter Alpha/CL Max value
10

Enter Alpha/CL increment value
0.5

Multi-threaded batch analysis - xflr5 v6.47

Foil Selection
● Current foil only ○ Foil list

Batch Variables
● Range ○ Re List

	Min	Max	Increment
Reynolds=	100,000	150,000	10,000
Mach=	0.000		
NCrit=	9.00		

Forced Transitions
Top transition location (x/c) 1.00
Bottom transition location (x/c) 1.00

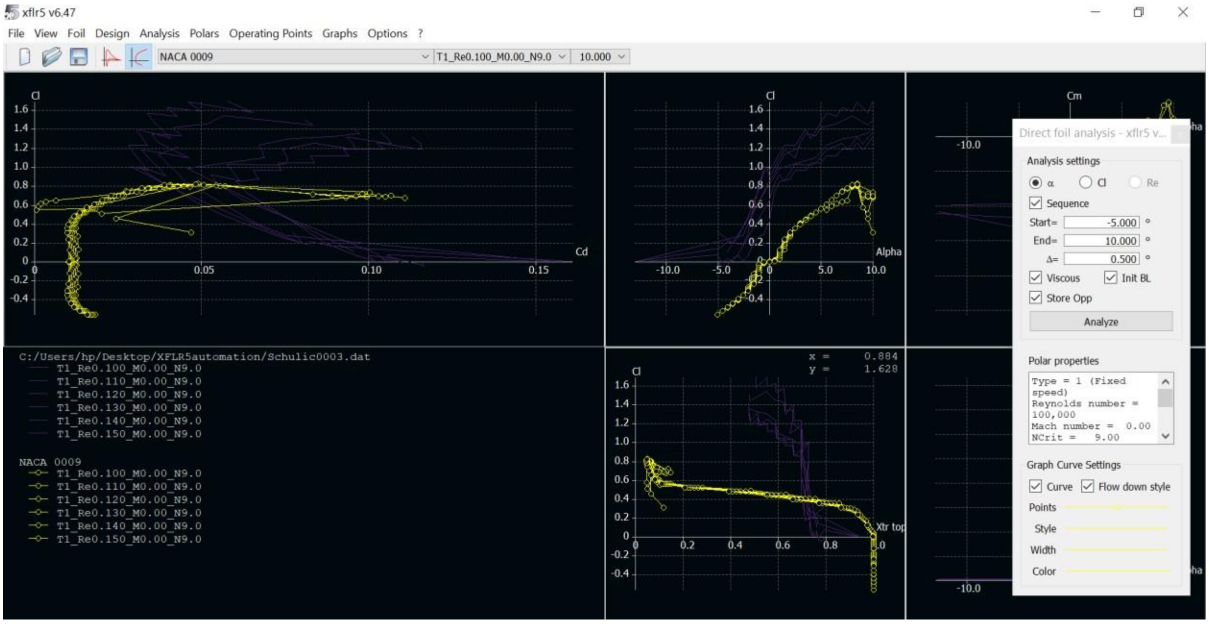
Analysis Range
Specify: ● Alpha ○ Cl □ From Zero

	Min	Max	Increment
Alpha	-5.000	10.000	0.500

☑ Initialize BLs between polars
□ Update polar view
Max. Threads to use for the analysis: 1 /8

```
Foils to analyze:
   NACA 0009
```

(ii)Output (After automation):



xflr5 v6.47

NACA 0009 | T1_Re0.100_M0.00_N9.0 | 10.000

```
C:/Users/hp/Desktop/XFLR5automation/Schulic0003.dat
     T1_Re0.100_M0.00_N9.0
     T1_Re0.110_M0.00_N9.0
     T1_Re0.120_M0.00_N9.0
     T1_Re0.130_M0.00_N9.0
     T1_Re0.140_M0.00_N9.0
     T1_Re0.150_M0.00_N9.0

NACA 0009
     T1_Re0.100_M0.00_N9.0
     T1_Re0.110_M0.00_N9.0
     T1_Re0.120_M0.00_N9.0
     T1_Re0.130_M0.00_N9.0
     T1_Re0.140_M0.00_N9.0
     T1_Re0.150_M0.00_N9.0
```

x = 0.884
y = 1.628

Direct foil analysis - xflr5 v...

Analysis settings
● α ○ Cl ○ Re
☑ Sequence
Start= -5.000 °
End= 10.000 °
Δ= 0.500 °
☑ Viscous ☑ Init BL
☑ Store Opp

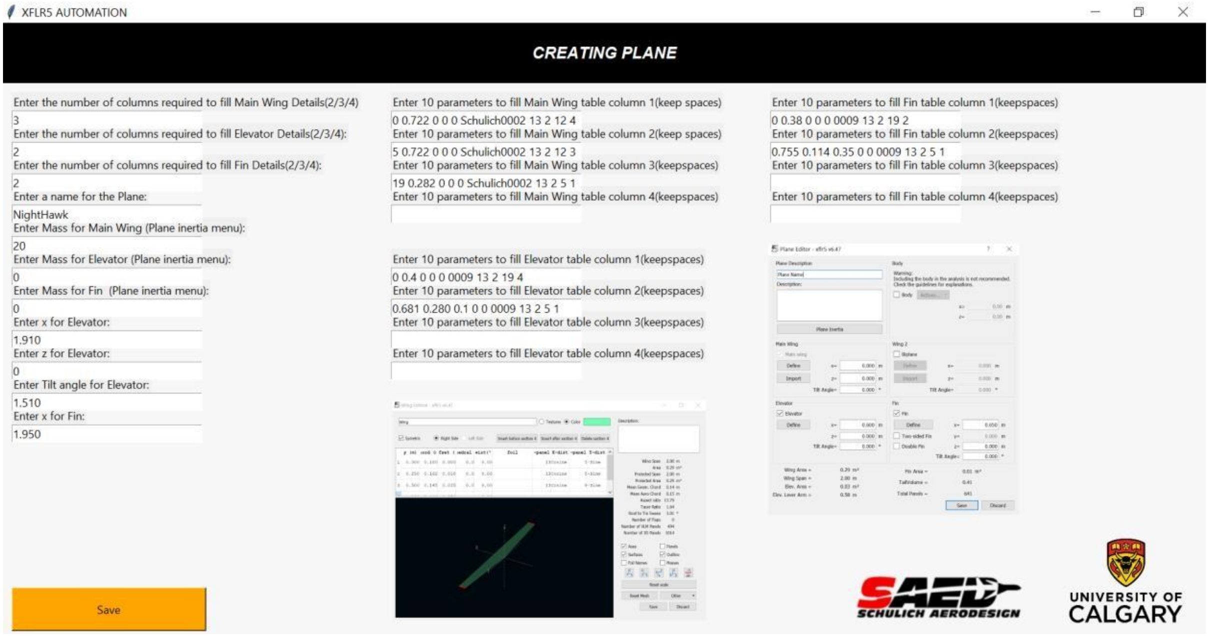
Polar properties

```
Type = 1 (Fixed
speed)
Reynolds number =
100,000
Mach number =  0.00
NCrit =   9.00
```

Graph Curve Settings
☑ Curve ☑ Flow down style
Points
Style
Width
Color

3. Plane_design:

(i) input (GUI):



XFLR5 AUTOMATION

CREATING PLANE

Enter the number of columns required to fill Main Wing Details(2/3/4)
3
Enter the number of columns required to fill Elevator Details(2/3/4):
2
Enter the number of columns required to fill Fin Details(2/3/4):
2
Enter a name for the Plane:
NightHawk
Enter Mass for Main Wing (Plane inertia menu):
20
Enter Mass for Elevator (Plane inertia menu):
0
Enter Mass for Fin (Plane inertia menu):
0
Enter x for Elevator:
1.910
Enter z for Elevator:
0
Enter Tilt angle for Elevator:
1.510
Enter x for Fin:
1.950

Enter 10 parameters to fill Main Wing table column 1(keep spaces)
0 0.722 0 0 0 Schulich0002 13 2 12 4
Enter 10 parameters to fill Main Wing table column 2(keep spaces)
5 0.722 0 0 0 Schulich0002 13 2 12 3
Enter 10 parameters to fill Main Wing table column 3(keepspaces)
19 0.282 0 0 0 Schulich0002 13 2 5 1
Enter 10 parameters to fill Main Wing table column 4(keepspaces)

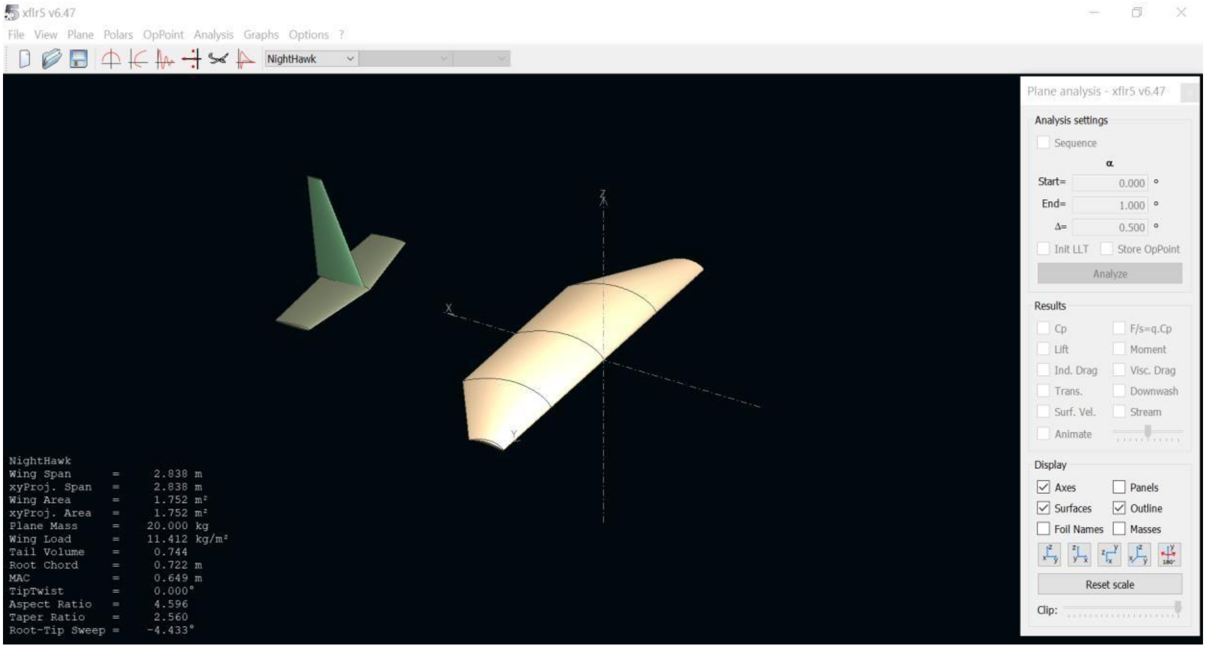
Enter 10 parameters to fill Elevator table column 1(keepspaces)
0 0.4 0 0 0 0009 13 2 19 4
Enter 10 parameters to fill Elevator table column 2(keepspaces)
0.681 0.280 0.1 0 0 0009 13 2 5 1
Enter 10 parameters to fill Elevator table column 3(keepspaces)

Enter 10 parameters to fill Elevator table column 4(keepspaces)

Enter 10 parameters to fill Fin table column 1(keepspaces)
0 0.38 0 0 0 0009 13 2 19 2
Enter 10 parameters to fill Fin table column 2(keepspaces)
0.755 0.114 0.35 0 0 0009 13 2 5 1
Enter 10 parameters to fill Fin table column 3(keepspaces)

Enter 10 parameters to fill Fin table column 4(keepspaces)

(ii) Output (After automation):



xflr5 v6.47
File View Plane Polars OpPoint Analysis Graphs Options ?
NightHawk
Plane analysis - xflr5 v6.47
Analysis settings
Results
Display

□ Sequence
α
Start= 0.000 °
End= 1.000 °
Δ= 0.500 °
□ Init LLT □ Store OpPoint
Analyze

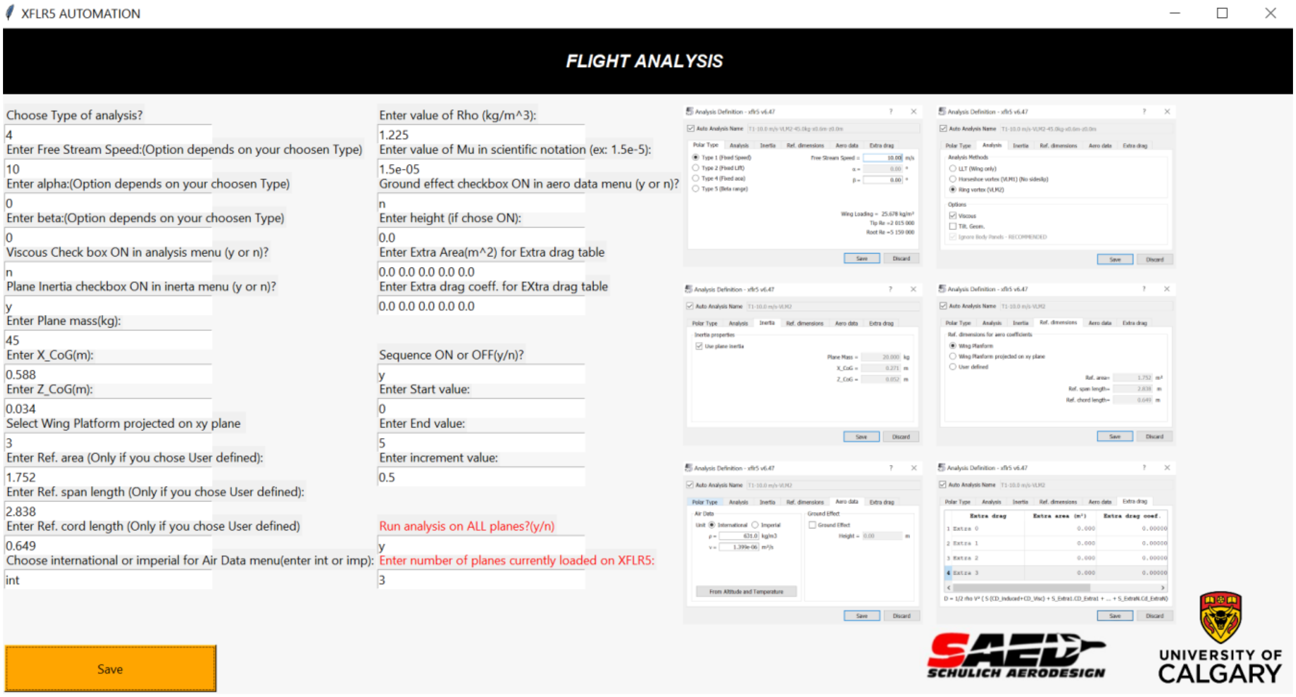
□ Cp □ F/s=q.Cp
□ Lift □ Moment
□ Ind. Drag □ Visc. Drag
□ Trans. □ Downwash
□ Surf. Vel. □ Stream
□ Animate

☑ Axes □ Panels
☑ Surfaces ☑ Outline
□ Foil Names □ Masses
Reset scale
Clip:

```
NightHawk
Wing Span      =     2.838 m
xyProj. Span   =     2.838 m
Wing Area      =     1.752 m²
xyProj. Area   =     1.752 m²
Plane Mass     =    20.000 kg
Wing Load      =    11.412 kg/m²
Tail Volume    =     0.744
Root Chord     =     0.722 m
MAC            =     0.649 m
TipTwist       =     0.000°
Aspect Ratio   =     4.596
Taper Ratio    =     2.560
Root-Tip Sweep =    -4.433°
```

4. Plane_analysis:

(i) Input (GUI):



XFLR5 AUTOMATION

FLIGHT ANALYSIS

Choose Type of analysis?
4

Enter Free Stream Speed:(Option depends on your choosen Type)
10

Enter alpha:(Option depends on your choosen Type)
0

Enter beta:(Option depends on your choosen Type)
0

Viscous Check box ON in analysis menu (y or n)?
n

Plane Inertia checkbox ON in inerta menu (y or n)?
y

Enter Plane mass(kg):
45

Enter X_CoG(m):
0.588

Enter Z_CoG(m):
0.034

Select Wing Platform projected on xy plane
3

Enter Ref. area (Only if you chose User defined):
1.752

Enter Ref. span length (Only if you chose User defined):
2.838

Enter Ref. cord length (Only if you chose User defined)
0.649

Choose international or imperial for Air Data menu(enter int or imp):
int

Enter value of Rho (kg/m^3):
1.225

Enter value of Mu in scientific notation (ex: 1.5e-5):
1.5e-05

Ground effect checkbox ON in aero data menu (y or n)?
n

Enter height (if chose ON):
0.0

Enter Extra Area(m^2) for Extra drag table
0.0 0.0 0.0 0.0 0.0

Enter Extra drag coeff. for EXtra drag table
0.0 0.0 0.0 0.0 0.0

Sequence ON or OFF(y/n)?
y

Enter Start value:
0

Enter End value:
5

Enter increment value:
0.5

Run analysis on ALL planes?(y/n)
y

Enter number of planes currently loaded on XFLR5:
3

Save

Analysis Definition - xflr5 v6.47
☑ Auto Analysis Name T1-10.0 m/s-VLM2-45.0kg-x0.0m-z0.0m
Polar Type | Analysis | Inertia | Ref. dimensions | Aero data | Extra drag
● Type 1 (Fixed Speed)
○ Type 2 (Fixed Lift)
○ Type 4 (Fixed aoa)
○ Type 5 (Beta range)
Free Stream Speed = 10.00 m/s
α = 0.00 °
β = 0.00 °
Wing Loading = 25.678 kg/m²
Tip Re =2 015 000
Root Re =5 159 000
Save Discard

Analysis Definition - xflr5 v6.47
Analysis Methods
○ LLT (Wing only)
○ Horseshoe vortex (VLM1) (No sideslip)
● Ring vortex (VLM2)
Options
☑ Viscous
□ Tilt. Geom.
☑ Ignore Body Panels - RECOMMENDED

Analysis Definition - xflr5 v6.47
Inertia properties
☑ Use plane inertia
Plane Mass = 20.000 kg
X_CoG = 0.271 m
Z_CoG = 0.052 m

Analysis Definition - xflr5 v6.47
Ref. dimensions for aero coefficients
● Wing Planform
○ Wing Planform projected on xy plane
○ User defined
Ref. area= 1.752 m²
Ref. span length= 2.838 m
Ref. chord length= 0.649 m

Analysis Definition - xflr5 v6.47
Air Data
Unit ● International ○ Imperial
ρ = 631.0 kg/m3
ν = 1.399e-06 m²/s
From Altitude and Temperature
Ground Effect
□ Ground Effect
Height = 0.00 m

Analysis Definition - xflr5 v6.47

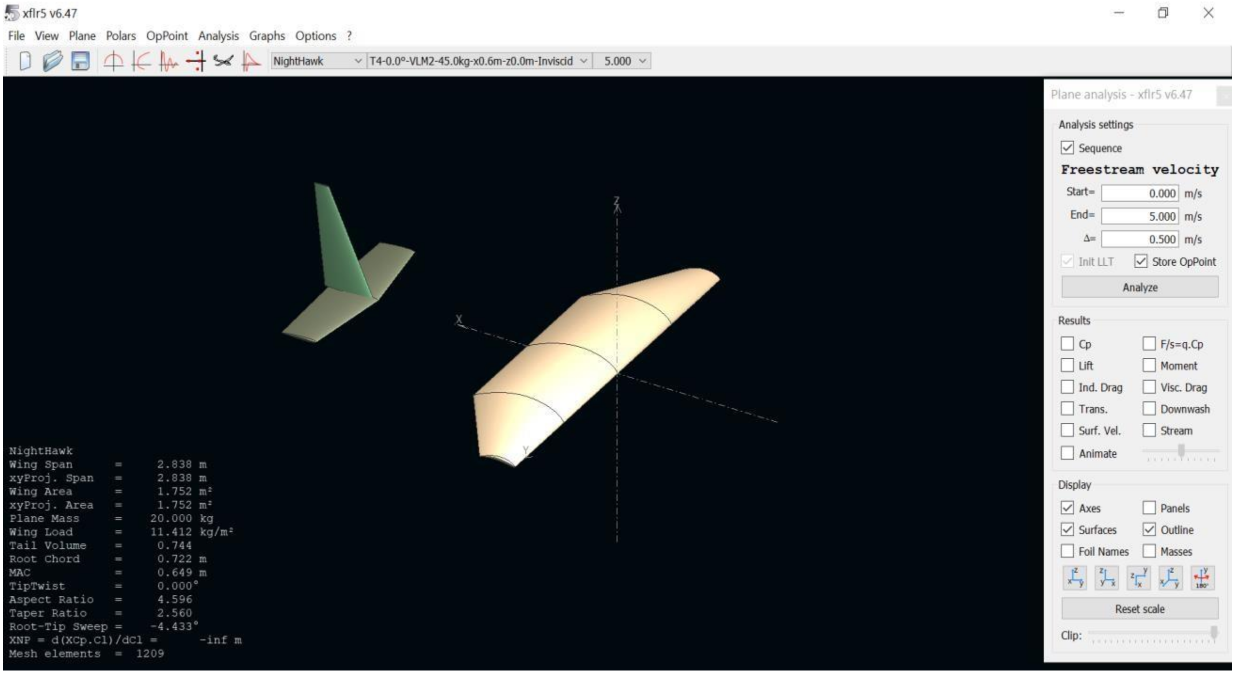
	Extra drag	Extra area (m²)	Extra drag coef.
1	Extra 0	0.000	0.00000
2	Extra 1	0.000	0.00000
3	Extra 2	0.000	0.00000
4	Extra 3	0.000	0.00000

D = 1/2 rho V² (S (CD_Induced+CD_Visc) + S_Extra1.CD_Extra1 + ... + S_ExtraN.Cd_ExtraN)

SAED SCHULICH AERODESIGN

UNIVERSITY OF CALGARY

(ii)Output (After automation):



xflr5 v6.47
File View Plane Polars OpPoint Analysis Graphs Options ?
NightHawk
T4-0.0°-VLM2-45.0kg-x0.6m-z0.0m-Inviscid
5.000
x
y
z

```
NightHawk
Wing Span       =     2.838 m
xyProj. Span    =     2.838 m
Wing Area       =     1.752 m²
xyProj. Area    =     1.752 m²
Plane Mass      =    20.000 kg
Wing Load       =    11.412 kg/m²
Tail Volume     =     0.744
Root Chord      =     0.722 m
MAC             =     0.649 m
TipTwist        =     0.000°
Aspect Ratio    =     4.596
Taper Ratio     =     2.560
Root-Tip Sweep =     -4.433°
XNP = d(XCp.Cl)/dCl =       -inf m
Mesh elements  =   1209
```

Plane analysis - xflr5 v6.47

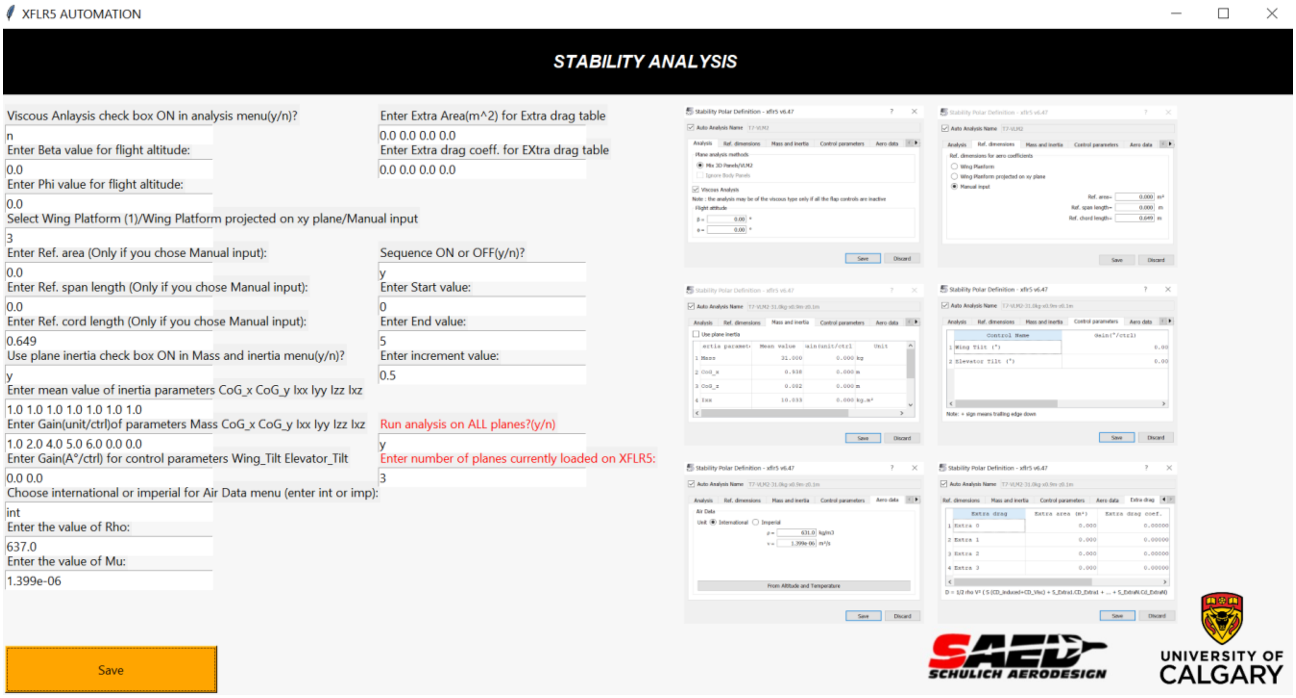
Analysis settings
☑ Sequence
Freestream velocity
Start= 0.000 m/s
End= 5.000 m/s
Δ= 0.500 m/s
☑ Init LLT ☑ Store OpPoint
Analyze

Results
□ Cp □ F/s=q.Cp
□ Lift □ Moment
□ Ind. Drag □ Visc. Drag
□ Trans. □ Downwash
□ Surf. Vel. □ Stream
□ Animate

Display
☑ Axes □ Panels
☑ Surfaces ☑ Outline
□ Foil Names □ Masses
Reset scale
Clip:

5. Stability_analysis:

(i) input (GUI):



XFLR5 AUTOMATION

STABILITY ANALYSIS

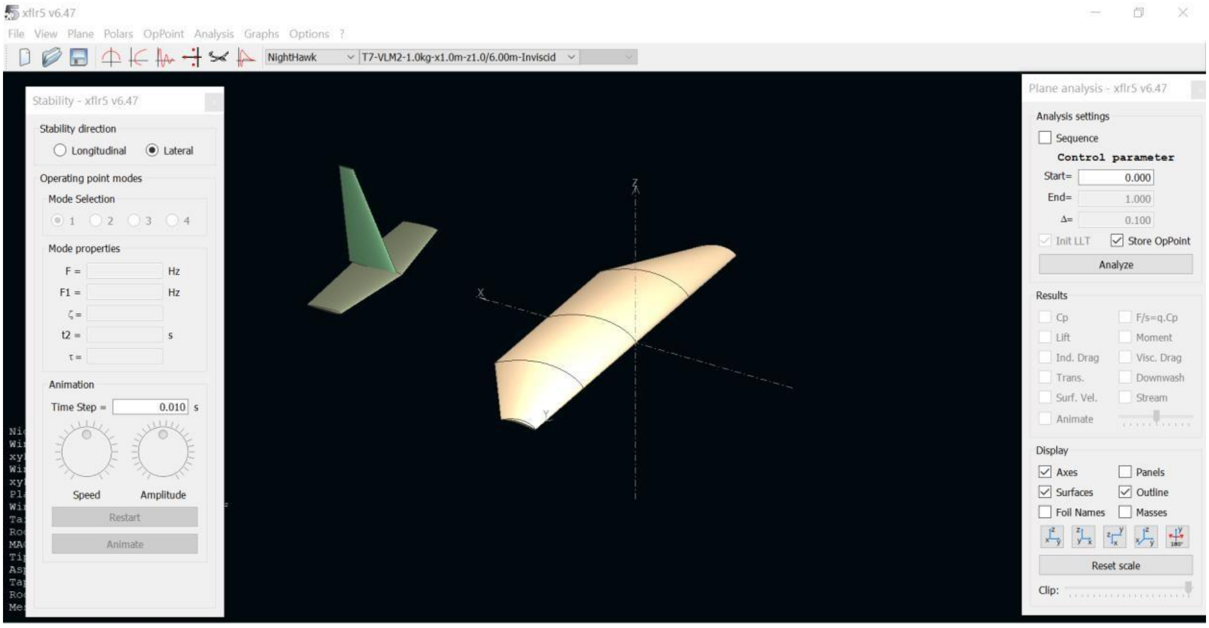
Viscous Anlaysis check box ON in analysis menu(y/n)?
n
Enter Beta value for flight altitude:
0.0
Enter Phi value for flight altitude:
0.0
Select Wing Platform (1)/Wing Platform projected on xy plane/Manual input
3
Enter Ref. area (Only if you chose Manual input):
0.0
Enter Ref. span length (Only if you chose Manual input):
0.0
Enter Ref. cord length (Only if you chose Manual input):
0.649
Use plane inertia check box ON in Mass and inertia menu(y/n)?
y
Enter mean value of inertia parameters CoG_x CoG_y Ixx Iyy Izz Ixz
1.0 1.0 1.0 1.0 1.0 1.0 1.0
Enter Gain(unit/ctrl)of parameters Mass CoG_x CoG_y Ixx Iyy Izz Ixz
1.0 2.0 4.0 5.0 6.0 0.0 0.0
Enter Gain(A°/ctrl) for control parameters Wing_Tilt Elevator_Tilt
0.0 0.0
Choose international or imperial for Air Data menu (enter int or imp):
int
Enter the value of Rho:
637.0
Enter the value of Mu:
1.399e-06

Enter Extra Area(m^2) for Extra drag table
0.0 0.0 0.0 0.0
Enter Extra drag coeff. for EXtra drag table
0.0 0.0 0.0 0.0

Sequence ON or OFF(y/n)?
y
Enter Start value:
0
Enter End value:
5
Enter increment value:
0.5

Run analysis on ALL planes?(y/n)
y
Enter number of planes currently loaded on XFLR5:
3

(ii) Output (After automation):



xflr5 v6.47

NightHawk
T7-VLM2-1.0kg-x1.0m-z1.0/6.00m-Inviscid

Stability - xflr5 v6.47

Stability direction: ○ Longitudinal ● Lateral

Operating point modes
Mode Selection: ○ 1 ○ 2 ○ 3 ○ 4

Animation
Time Step = 0.010 s

Plane analysis - xflr5 v6.47

Analysis settings
□ Sequence
Control parameter
Start= 0.000
End= 1.000
Δ= 0.100
☑ Init LLT ☑ Store OpPoint

Results
□ Cp □ F/s=q.Cp
□ Lift □ Moment
□ Ind. Drag □ Visc. Drag
□ Trans. □ Downwash
□ Surf. Vel. □ Stream
□ Animate

Display
☑ Axes □ Panels
☑ Surfaces ☑ Outline
□ Foil Names □ Masses

Thus, its evident from the screenshots that the code is working efficiently in automating all the 5 fundamental features of XFLR5.

7.3 Recommendation on future improvement

1. Utilizing a more efficient image recognition technique which would help significantly lower the duration of each code iteration.

2. Develop a more efficient and reliable method to keep track of the completion time of different XFLR5 processes.

3. Develop a log which keeps track of all the processes which are being completed step-by-step.

4. Develop a method which Is able to detect open non-minimized background windows which would allow us pop up the XFLR5 window during the beginning of code run-time .

5. Improving the GUI by adding Integer Fields, Decimal fields and radio buttons to make it more user friendly.

6. Discovering a method to run the script in background when XFLR5 is minimized which would enable us to run multiple version of the script simultaneously.