A MATLAB GUI-Based Approach to Wing Design and Aerodynamic Performance Evaluation

Suhas K S,[a)] Rohit Krishna,[b)] Kanak Agarwal,[c)] and Manikandan Murugaiah[d)]

Department of Aeronautical and Automobile Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India

[a)†]suhas.s5@learner.manipal.edu
[b)‡]rohit.krishna1@learner.manipal.edu
[c)§]kanak.agarwal@learner.manipal.edu
[d)]Corresponding author: manikandan.m@manipal.edu

Abstract. This study presents a MATLAB® based application developed for the design and aerodynamic analysis of varying wing geometries using the NACA 4 and 5-digit series airfoils. It presents an innovative framework that enables the efficient generation of intricate wing surfaces, doing away with cumbersome and skill-intensive wing design using CAD software. Further, the framework can generate complex geometrical features such as twist, taper, and sweep. With meticulous attention to the user perspective and user-friendliness, special emphasis has been given to the position, color, and usability of elements within the interface. It integrates the horizontal wind model (HWM) routine developed by the U.S. Naval Research Laboratory. This reinforces its practical application by computing the wind speed using location and time-specific data. The XFOIL solver integrated into the framework uses this initial wind speed data to conduct an initial aerodynamic analysis using the vortex panel method.

INTRODUCTION

Wing design is an integral aspect of aircraft design. Software like XFLR5 [1], AVL [2], and Tornado [3] can be used for wing design but are primarily aircraft design software. Due to their extensive nature, these software can seem daunting to an absolute beginner. Hence, aiming to simplify the wing design process, the developed MATLAB® based GUI application [4] presents an integrated platform for preliminary wing design. It incorporates functionality from the ground up, from airfoil parametrization to complex wing geometries and their aerodynamic analyses. Using a traditional approach would involve manually modifying the input parameters for each configuration, which would be cumbersome and inefficient, and this is especially relevant to wing design since it is primarily an iterative process. Furthermore, understanding the workflow can present a significant challenge for students new to these techniques. Addressing these limitations, this paper introduces a pioneering framework tailored to new students and primarily for academic use. The application aims to enhance understanding, accessibility, and efficiency in the context of wing generation and analysis. This application marks a significant advancement in simplifying the preliminary wing design process and facilitating accessibility for users at all levels of expertise.

This application offers unparalleled functionality at every step of the wing design process. Users can effortlessly determine airfoil lift and drag coefficients at user-specified angles of attack and Reynolds numbers, enhancing the tool's versatility. Additionally, generating plots depicting airfoil lift, drag, and moment coefficients across various angles of attack showcases its multitasking capabilities. The real-time visualization of the wing surface augments the efficiency of the design process by providing an intuitive understanding of the wing parameters at play. Through these features, our MATLAB® GUI-based application empowers users with the capability to conduct an initial comprehensive aerodynamic analysis, revolutionizing the wing design process. Figure 1 gives a comprehensive overview of the framework developed.

OVERVIEW OF THE FRAMEWORK DEVELOPED

Introduction

A diverse array of Graphical User Interface (GUI) components are incorporated to facilitate user interaction and enhance functionality. The main elements required to create MATLAB® GUI are as follows [5].

* **Components:** In MATLAB® GUIs, components like pushbuttons, labels, and edit boxes are graphical elements. They come in various types, such as graphical controls (e.g., push buttons, sliders), static elements (frames,

text), menus, and axes for displaying data.

- **Figures:** GUI components must be organized within a figure representing a computer screen window. Empty figures are created and can accommodate various combinations of components.

- **Callbacks:** For a MATLAB® GUI to function, it must be able to respond to user interactions like mouse clicks or keyboard inputs, which are events. Each event triggers a specific action in the program, known as a callback. Essentially, clicking a button executes a corresponding block of code. Hence, every graphical component requires a callback to implement its intended function.

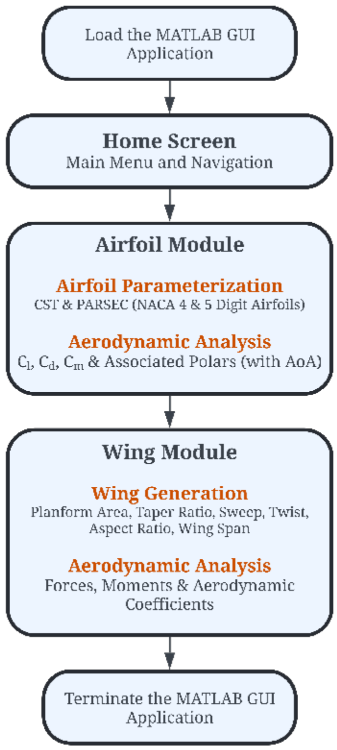


FIGURE 1: Outline of the Developed Application

The User Window

To optimize the user experience and maintain the interface's simplicity and flexibility, this application features three distinct windows, the home screen, the airfoil parametrization, and the wing windows [6]. Each window is carefully crafted to serve specific functions and provide users with a focused and intuitive environment. By dividing the application into distinct windows, each section remains focused and clutter-free, allowing users to concentrate solely on their specific objectives. This approach also enhances flexibility, as users can seamlessly transition between different tasks and workflows as needed. The following sections discuss the features, functionalities, and contributions to the overall user experience of each window.

The Home Section

The home section shown in Figure 2 serves as the central hub of the application, providing users with a seamless point of entry to access key functionalities such as airfoil parametrization and aerodynamic analysis. Designed with accessibility in mind, the home screen offers intuitive navigation options, allowing users to transition between different sections of the application effortlessly. To ensure optimal user experience, the dimensions of the home screen window are dynamically adjusted based on the user's screen size. By setting the width and height of the window as a proportion of the screen size, the application's utility is enhanced across various devices and display resolutions.

A callback has been implemented such that upon clicking the 'Airfoil Parametrization' button, users are directed to the relevant section of the application, where they can input parameters and initiate the airfoil parametrization and selection process [7]. Additionally, a 'Quit' button is incorporated on the home screen, offering a convenient way to exit the application. To maintain interface consistency and usability, the screen maximizing functionality has been disabled, ensuring that the interface remains fixed and predictable. Furthermore, meticulous attention has been given to the positioning, colors, and font sizes of all elements on the home screen, enhancing their readability and visual appeal.

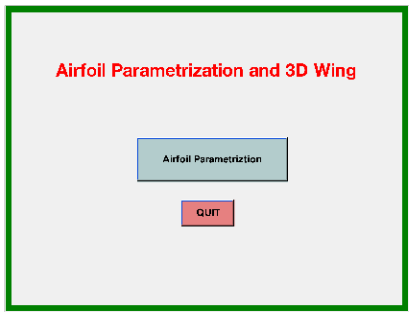


FIGURE 2: Home Section

Airfoil Parametrization and Selection

This window given by Figure 3 is meticulously designed to cater specifically to the parametrization of airfoils, focusing on integrating the CST and PARSEC methodologies seamlessly into the user interface. This section empowers users to input parameters, initiate parametrization processes, and visualize parametrization results with ease and clarity. To enhance usability and organization, separate panels within the airfoil parametrization window were implemented to delineate the various tasks carried out in this section.

The Class Shape Transformation (CST) method was developed by Brenda Kulfan, and the formulation incorporated into the framework was sourced from Reference [8], building on the work of Pramudita [9]. The CST method uses class and shape functions to define the airfoil geometry. The class function is used to define general classes of geometries, whereas the shape function is used to define specific shapes within a class of geometries. The airfoil is then defined using the Bernstein polynomials. The first term of the polynomial defines the leading-edge radius, and the last term is the boat-tail angle. The other terms in between are the "shaping terms". The drag predictions and pressure distribution for an eighth-order Bernstein's polynomial agree exactly with experimental data. Hence, a Bernstein polynomial of the eighth order was chosen, and eight weighted coefficients were obtained at the end of the CST process.

The PARSEC method uses a linear combination of shape functions to describe the airfoil. It defines eleven parameters that can represent the geometry of any airfoil. These parameters include the trailing edge thickness (ΔZ_{TE}), wedge angle (β_{TE}), coordinate (Z_{TE}) and direction (α_{TE}); the upper and lower crest locations [(X_{UP}, Z_{UP}) and (X_{LO}, Z_{LO})], the upper and lower curvatures [(Z_{xxLO}, Z_{xxUP})] and the leading edge radius (r_{LE}). The PARSEC parametrization method can effectively control the curvature of the airfoil's upper and lower surfaces, but it doesn't provide sufficient authority over the airfoil's trailing edge [10]. This method joins the upper and lower maximum points through smooth curves with the trailing edge. This could potentially compromise the highly critical regions around the trailing edge, where critical flow phenomena occur.

Users can select a directory containing data files of NACA 5 and NACA 4 series airfoils using the 'BROWSE' button. These airfoil '.dat' files were generated using the standard NACA equations. All the available airfoils are dynamically loaded using a drop-down menu for user selection [11]. The interface plots the CST and PARSEC representations of the selected airfoil, providing visual insights into the efficiency of the parametrization method. The error percentage associated with the parametrization process is displayed as a label, offering real-time feedback on

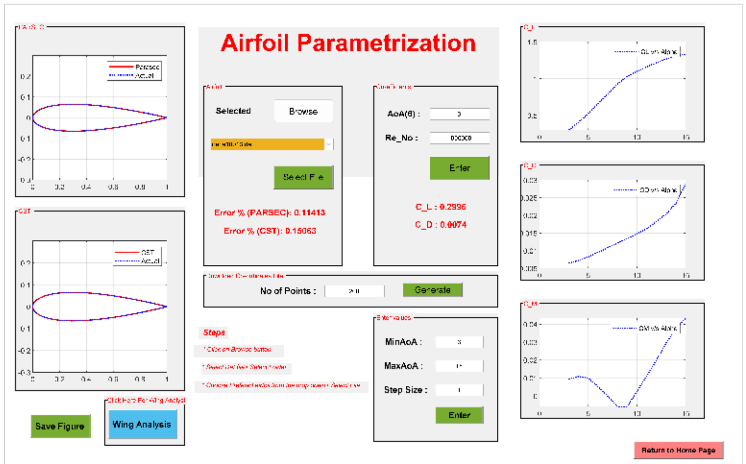


FIGURE 3: Airfoil Parametrization

accuracy. The Root Mean Square (RMS) method is used to calculate the error percentage for both parametrization techniques.

This application also includes a "Reverse CST and PARSEC" feature that generates customized coordinates for selected airfoil profiles. Users input the desired number of coordinates, and the program compares error percentages between CST and PARSEC methods to select the one with lower error. The coordinate file is automatically saved in the same folder as the NACA 4 and NACA 5 series airfoils. A standardized naming convention helps the user identify the generated file with ease. A message box appears on the screen to notify users when the coordinate file is successfully generated.

This application enables users to compute lift and drag coefficients for a selected airfoil at specific angles of attack and Reynolds numbers by integrating XFOIL [12] within the airfoil parametrization window. Users can input desired values for the angle of attack and Reynolds number in editable fields. The application parses commands to XFOIL via an input file, and XFOIL processes the commands and generates aerodynamic data captured in an output file. Leveraging the XFOIL solver facilitates rapid computation of lift and drag coefficients, providing users with valuable aerodynamic data in real time.

An application enables users to analyze airfoil behavior across angles of attack, considering lift, drag, and moment coefficients. Users specify the minimum, maximum, and step size for the angle of attack range. The application runs the XFOIL solver and plots coefficients visually, aiding in airfoil analysis and parametrization. This tool enhances usability and effectiveness for making informed design decisions.

This application offers additional features to enhance user experience and workflow efficiency. Users can save the entire screen as a PNG image, enabling easy documentation. Descriptive labels have been strategically placed to guide users through the interface. Seamless transitions between screens ensure continuity in workflow, allowing users to effortlessly switch to 3D aerodynamic analysis or return to the home screen via dedicated buttons. These supplementary features enable an intuitive and user-friendly application.

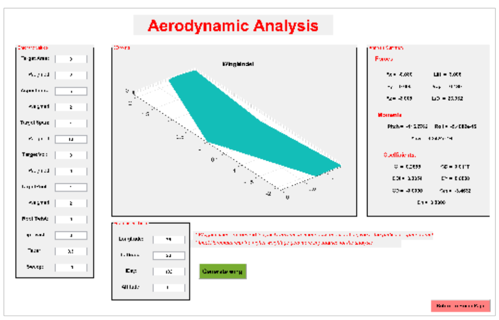
Wing Design

The wing window focuses on the generation and analysis of wing configurations. Users can input wing parameters, perform aerodynamic analyses, and explore various configurations within this dedicated workspace. The interface is tailored to accommodate the complexities of wing design and analysis, offering specialized tools and visualizations to aid users in their tasks.

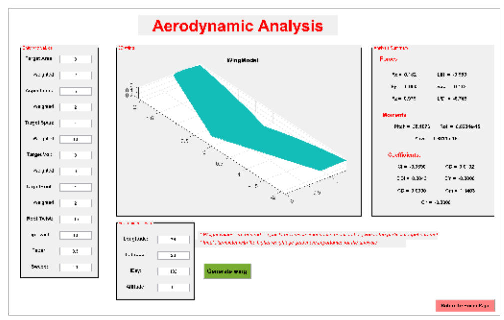
The wing is generated using Bézier surfaces based on the formulation by Sóbester and Forrester [13]. The definition of the spanwise vector ε is essential to this method. The vector ε defines the spanwise distance along which the airfoils are to be generated. The curvilinear axis is fixed to the wing's leading edge (LE), and its scale is normalized. The values of ε are 0 and 1, respectively, at the root and tip sections of the wing. Using the vector ε, the spanwise distribution of various geometric properties like twist, chord (taper), and sweep can be controlled. This particular method lends the ability to generate complex wing geometries with ease. Figure 4 represents different wing geometries generated using the framework.

The input parameters used to generate the wing are elucidated below:

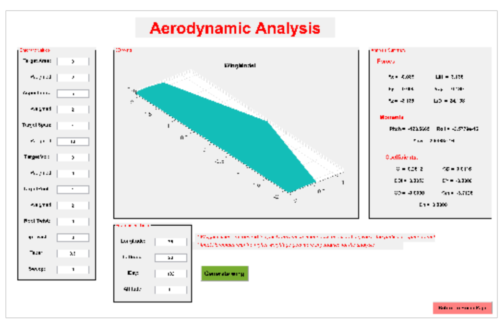
- **Target Area:** The desired area of the wing as specified by the user.
- **Aspect Ratio:** Defined as the ratio of the length of the wing to the square of its span area. A higher aspect ratio results in a narrower and longer wing, while a lower aspect ratio yields a broader and shorter wing design.
- **Target Span:** The designated length measured from tip to tip of the wing. A higher span corresponds to a longer wing, while a lower span results in a shorter wing.
- **Target Volume:** The specified volume that the wing is intended to occupy.
- **Target Root:** The specified root chord of the wing.
- **Twist:** The twist angle of a wing refers to the variation in the incidence angle along the wing's span. In other words, it describes how the angle of the wing changes from the root to the tip. Varying root and tip twist angles can be inputted.
- **Taper:** The taper ratio of a wing is a dimensionless parameter that describes the ratio of the wing's root chord (the chord at the wing's base, closest to the fuselage) to its tip chord (the chord at the wing's outermost point).
- **Sweep angle:** It is the angle between the leading edge and the apex of the wing.



(a) Swept Wing



(b) Wing with Twist Angle



(c) Tapered Wing

FIGURE 4: Different Wing Geometries Generated using the Framework

The atmospheric data is obtained using the International Standard Atmosphere (ISA) model [14]. At any given altitude (less than 32 km), pressure, temperature, density, and the speed of sound can be calculated. The wind speed data is obtained through the implementation of the U.S. Naval Research Laboratory Horizontal Wind Model (HWM) [15, 16] routine using built-in MATLAB® functions. The meridional and zonal wind components of the local wind velocity vector are calculated using the following inputs:

- **Latitude:** The desired latitude in °.

* **Longitude:** The desired longitude in $^{\circ}$.
* **IDAY:** Operational day of the year. The value lies between 1 and 365.
* **Altitude:** The desired altitude in *km*.

The approach formulated by Matthew Brown [17] was used to estimate the aerodynamic coefficients of the wing. The cranked-wing approach is used to define the arbitrary shapes of the planform. Each panel is defined by four distinct characteristics, namely, the airfoil section used, the length of the panel root and tip chords, the quarter chord location of the root and tip panel airfoil sections, and the incidence angle. Each point in the panel has a characteristic sweep angle. The incidence angles of the airfoil sections and their properties are assumed to vary linearly along the panel.

XFOIL was used to obtain the chordwise loading of the airfoil sections of the wing. Once the chordwise loading is obtained, the spanwise loading can be estimated using the lifting line theory. The determination of the distributed aerodynamic loads in compressible subsonic flow [17] was done using the method of lifting line theory described by Phillips [18]. This method is based on the 3D vortex lifting law. Essentially, a series of discrete horseshoe vortices are placed along the wing surface. This method can accurately predict the lift for complex geometries with twist and sweep angles. Further, the aerodynamic effects of airfoil thickness, control surfaces, and flaps can also be modeled using the chordwise airfoil data.

Upon user input of all necessary parameters and subsequent activation of the generate button, a callback function is triggered that automatically saves the airfoil coordinate file as two separate files for the root and tip. Simultaneously, point analysis is conducted in the background, computing all the required aerodynamic coefficients. The culmination of this process results in the generation of a 3D wing displayed seamlessly within the designated panel.

Labeled displays are incorporated within the analysis summary panel to provide users with a comprehensive overview of the analysis. The analysis summary includes the forces, moments, and coefficients of the wing surface at the specified operating conditions.

Conclusion

The paper presents a MATLAB® GUI for wing design and preliminary aerodynamic analysis. The primary aim of this application is to cater to the academic needs of students who are encountering parametrization techniques for the first time. With a focus on accessibility and simplicity, this application provides a valuable resource for students seeking to grasp these concepts effectively. By integrating the CST and PARSEC parametrization techniques with MATLAB® GUI capabilities, the tool empowers users with a powerful yet accessible platform for preliminary wing design. Moving forward, we are committed to continually upgrading the application with enhanced features.

ACKNOWLEDGMENTS

We would like to acknowledge the support of the Department of Aeronautical and Automobile Engineering at the Manipal Institute of Technology, MAHE, Manipal, Karnataka, India, for the facilities and resources provided that were essential to the completion of this work.

REFERENCES

1. XFLR5, "Xflr5," `http://www.xflr5.tech/xflr5.htm`, [Online, Accessed: 2024-06-21].
2. AVL, "Avl," `https://web.mit.edu/drela/Public/web/avl/`, [Online, Accessed: 2024-06-21].
3. Tornado, "Tornado," `http://tornado.redhammer.se/`, [Online, Accessed: 2024-06-21].
4. P. Marchand and O. Holland, *Graphics and GUIs with MATLAB*, 3rd ed. (Chapman and Hall/CRC, 2002).
5. M. Sharma, S. Sharma, and G. Sahu, "Designing matlab gui for various analog and digital communication systems," International Journal of Trend in Scientific Research and Development , 1397–1405 (2017).
6. E. C. Reuben and L. Y. Meng, "Gui composer: A lightweight gui software development framework for wsn, i.o.t and control system application," in *2022 5th International Conference on Mechatronics, Robotics and Automation (ICMRA)* (2022) pp. 69–73.
7. T. Li and H. Zhang, "Digital image enhancement system based on matlab gui," in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)* (2017) pp. 296–299.

8. B. M. Kulfan, "Universal parametric geometry representation method," Journal of Aircraft **45** (2008), 10.2514/1.29958.
9. P. S. Palar, "How to fit your airfoil with cst parameters," `https://pramsatriapalar.wordpress.com/2017/02/24/how-to-fit-your-airfoil-with-cst-parameters` (2017), [Online, Accessed: 2024-05-15].
10. N. P. Salunke, R. Ahamad, and S. Channiwala, "Airfoil parameterization techniques: A review," American journal of mechanical engineering **2**, 99–102 (2014).
11. K. Y. You, M. S. Sim, F. Hamid, K. Y. Lee, E. M. Cheng, and F. Esa, "Simple matlab-based eda for 5g phased-array antenna design," in *2022 IEEE International RF and Microwave Conference (RFM)* (2022) pp. 1–4.
12. M. Drela, "Xfoil: An analysis and design system for low reynolds number airfoils," in *Low Reynolds Number Aerodynamics*, edited by T. J. Mueller (Springer Berlin Heidelberg, 1989) pp. 1–12.
13. A. Sóbester and A. I. J. Forrester, *Aircraft aerodynamic design: geometry and optimization* (John Wiley & Sons, 2014).
14. U. Government, "U.s. standard atmosphere, 1976," Tech. Rep. (U.S. Government Printing Office, 1976).
15. A. E. Hedin, "Horizontal wind model (hwm)(1990)," Planetary and Space Science **40**, 556–557 (1992).
16. D. P. Drob, J. T. Emmert, J. W. Meriwether, J. J. Makela, E. Doornbos, M. Conde, G. Hernandez, J. Noto, K. A. Zawdie, S. E. McDonald, *et al.*, "An update to the horizontal wind model (hwm): The quiet time thermosphere," Earth and Space Science **2**, 301–319 (2015).
17. M. A. Brown, *A Computational Method for Determining Distributed Aerodynamic Loads on Planforms of Arbitrary Shape in Compressible Subsonic Flow*, Ph.D. thesis, University of Kansas (2013).
18. W. F. Phillips and D. O. Snyder, "Modern adaptation of prandtl's classic lifting-line theory," Journal of Aircraft **37**, 662–670 (2000).