EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

An aerodynamic panel method and mesh generator for educational and research applications in aerodynamics of aircraft design

Kantinan Phuekpan, Natee Panagant*, Sujin Bureerat and Nantiwat Pholdee

Sustainable and Infrastructure Research and Development Center, Department of Mechanical Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand

Received 2 April 2026
Revised 4 September 2026
Accepted 8 October 2026

Abstract

The open-source aerodynamic panel tool is a rapid and efficient tool for estimating aerodynamic characteristics, particularly during several phases of aircraft design; however, the accessibility of transparent, adaptable, user-friendly, and open-source aerodynamic tools suitable for educational and research environments remains challenging. This paper introduces Aircraft Panel Generation and Aerodynamic Analysis, an educational open-source software designed to provide easy access to aerodynamic analysis tools for aircraft studies. The software incorporates implementations of the Vortex Lattice Method and Doublet Lattice Method to compute steady and unsteady loads using the Aerodynamic Influence Coefficient matrices for compressible subsonic flows. The framework features advanced panel generation that discretizes lifting surface configurations into structured chordwise and spanwise trapezoidal elements. The accuracy of the code is demonstrated by validation against data from a published German Aerospace Center technical report across both steady and unsteady aerodynamic cases. The results show that the computed pressure coefficients are in good agreement with reference solutions. The example aircraft models are analyzed and visualized as illustrative examples. This software could serve as a valuable resource for academics and researchers, providing a foundation for further aeroelastic studies.

Keywords: Vortex lattice method, Doublet lattice method, Aircraft designs, Aerodynamic panel

1. Introduction

Aerodynamic analysis is foundational to aircraft design and development, enabling engineers to predict how an aircraft will interact with airflow under varied flight conditions. This understanding is crucial for assessing performance, stability, and efficiency in aircraft, all of which drive innovation in the field. With the expansion of aeronautic applications such as unmanned aerial vehicles (UAVs) and modern commercial aircraft [1-6], the need for accessible aerodynamic analysis has grown significantly.

For aerodynamic analyses, medium-fidelity methods such as the Vortex Lattice Method (VLM) [7-10] and Doublet Lattice Method (DLM) [11, 12] are highly valued for their simplicity and effectiveness across several design phases. These are aerodynamic panel methods that compute the inviscid, attached flow about an aircraft's lifting surfaces in the subsonic regime. These methods are fast in the sense that a matrix of Aerodynamic Influence Coefficients (AICs) can be precomputed, so that the calculation of aerodynamic lift reduces to a simple matrix multiplication. Also, the methods provide distributed aerodynamic pressures or forces, which are ideal for applications involving fluid-structure interaction. Notably, the VLM is generally used for steady-state problems, whereas the DLM is employed in the frequency domain to determine the unsteady aerodynamic response to harmonic motion, which is important for flutter and gust load predictions. Accordingly, both methods are frequently employed in preliminary design and are required for many aeroelastic analyses [13-15].

Over the past decades, open-source software has played an increasingly important role in supporting many applications, especially in aerodynamic analysis, by providing accessible and modifiable platforms for both academic and professional use [16-19]. These tools support a wide range of applications, from conceptual aircraft design to aeroelastic studies. However, existing open-access platforms often present operational trade-offs between solver versatility, preprocessing capabilities, and ease of verification. Many open-source software packages [20-22] provide computationally efficient aerodynamic analyses suitable for conceptual design and academic instruction. On the other hand, many advanced multidisciplinary suites [23-26] offer comprehensive aero-structural and system-level modeling capabilities. Nevertheless, their high-level integration introduces operational overhead, complex dependencies, steep learning curves, and functional constraints that can limit their utility for dedicated aerodynamic and aeroelastic investigations.

Furthermore, in comprehensive aeroelastic aircraft design, evaluating steady-state aerodynamic behavior alone is insufficient. A robust low-order framework requires a dual formulation: the VLM for static trim, load distributions, and stability derivatives, and the DLM to compute unsteady Generalized Aerodynamic Force (GAF) matrices, phase lags, and harmonic downwash, which are essential for flutter and gust response analyses. Relying on disparate software packages for both steady and unsteady regimes introduce geometric discrepancies and mesh interpolation errors when interfacing with structural finite element models. Therefore, integrating

*Corresponding author.
Email address: natepa@kku.ac.th
https://doi.org/10.64960/easr.2026.267018

both VLM and DLM within a single, unified aerodynamic framework is critical to guarantee geometric consistency and ensure a seamless pipeline for multidisciplinary extensions.

There are several well-known open-source aerodynamic simulation software including Tornado [20], Athena Vortex Lattice (AVL) [21], DLMPro [22], ASWING [23], Stanford University Aerospace Vehicle Environment (SUAVE) [24], Next generation Conceptual Aeroelastic Structural Sizer (NeoCASS) [25], Simulation of High Aspect Ratio aeroplanes in Python (SHARPy) [26], and PanelAero [27]. Although several aerodynamic solvers are available, each software package lacks some features. Tornado, AVL, ASWING, and SUAVE are only capable of solving static aerodynamics, whereas the other software can support both static and dynamic aerodynamics. However, DLMpro and PanelAero are not capable of analyzing realistic wing geometry with multiple wing sections and control surfaces. Regarding the AIC data extraction, Tornado, AVL, and SUAVE cannot export AIC matrices. While ASWING, NeoCass, SHARPy, and PanelAero can generate AIC matrices, it requires additional user scripting and manual routines, which are quite complicated. Only the DLMPro can provide support for exporting the AIC matrix directly.

Accessible, lightweight aerodynamic tools that are integrated with both static and dynamic aerodynamic solvers, can export AICs and GAFs, and have user-friendly automated panel generation with a simple input format, are lacking in all available software. Since many software packages are available with varying features and capabilities, a detailed comparison between the proposed software and existing software is presented in Table 1.

Addressing this gap is critical for accelerating early-stage aircraft conceptualization, aeroelastic evaluations, and academic instruction. Delivers rapid results with low effort, and provides students and researchers with an open, modifiable foundation for aircraft design and aeroelastic analysis. To address these challenges, this paper presents APGAA (Aerodynamic Panel and Geometry Analysis Application), an open-source computational framework designed for educational and research environments. The contributions of this study are as follows:

1. Develop an open-source MATLAB or GNU Octave-based educational tool incorporating improved VLM and DLM implementation, and the software is integrated with advanced automated panel generation. Even without a MATLAB license, the source code can be used in the open-source GNU Octave.
2. Demonstrate software capability and usability through validation examples and comparison with existing tools.
3. The user interface is designed with user-friendliness in mind, using only a simple input file and a few execution commands that are required to accomplish steady and unsteady aerodynamic analyses.

These contributions are intended to address the gap between theoretical aerodynamic principles and their practical implementation, particularly within academic and research contexts where accessibility and clarity are crucial. By integrating validated aerodynamic solvers with advanced yet user-friendly modeling capabilities, APGAA serves as a robust platform for both instructional applications and preliminary aircraft design analyses. Moreover, the combined implementation of VLM and DLM panel methods significantly extends the software's applicability to aeroelastic phenomena and maneuver load evaluations. This open-source tool offers conclusively substantial value to students, educators, and researchers engaged in aerodynamics and flight dynamics studies.

2. Methodology

2.1 Software accessibility details

APGAA is MATLAB-based software designed for 3D aerodynamic analyses of aircraft using VLM and DLM. To facilitate open science and reproducibility, the software is permanently archived on Zenodo repository (https://doi.org/10.5281/zenodo.16931471) and maintained on Github repository (https://github.com/Natee-Panagant/APGAA). The following subsections outline the advances of the APGAA software framework, which comprises two major contributions: panel methods developed for aerodynamic analysis and panel generation, as explained in more detail in this section.

2.2 Literature review

Low- to medium-fidelity open-source aerodynamic tools are foundational to aircraft conceptual design and dynamic aeroelastic verification. However, their existing open-source frameworks present notable trade-offs across key functional domains: aerodynamic mesh generation, numerical solver formulations, control surface modeling, influence matrix export capabilities, and computational availability. The scoped software includes Tornado [20], AVL [21], DLMPro [22], ASWING [23], SUAVE [24], NeoCASS [25], SHARPy [26], and PanelAero [27].

Regarding aerodynamic mesh generation and geometric flexibility, standalone potential-flow tools such as Tornado rely on automated parametric multi-surface definitions, whereas AVL uses script-driven text inputs. Conversely, DLMPro requires manual coordinate definitions, restricting rapid parametric updates. Higher-level suites integrate geometry into broader structural or mission workflows, with distinct limitations: ASWING uses structural beam-like discretizations, SUAVE relies on built-in parametric wing builders, and PanelAero depends on external scripting or bulk data inputs from NASTRAN. Evidently, most existing meshing routines strictly constrain panel definitions to planar wing planforms aligned with the freestream flow, complicating the handling of arbitrary surface orientations. With respect to aerodynamic solver formulations, Tornado and AVL implement the classical VLM, which is restricted to steady or quasi-steady conditions, whereas DLMPro implements the DLM for unsteady oscillatory flows. Although SUAVE incorporates VLM, it lacks native unsteady aerodynamic solvers. PanelAero and NeoCASS can perform both updated VLM and DLM formulations, but require complex Nastran command card input style. Notably, SHARPy employs the time-domain Unsteady Vortex Lattice Method (UVLM) [28], which incurs substantial runtime overhead compared with frequency-domain solutions. Among other tools, ASWING relies on non-linear lifting-line formulations and strip theory, which fail to capture three-dimensional panel pressure distributions on low-aspect-ratio surfaces. Consequently, unified platforms that support both steady and unsteady aerodynamic formulations within a single environment remain rare. Control surface deflection modeling varies considerably across existing tools. AVL, DLMPro, and NeoCASS apply boundary-condition downwash modifications without altering the planar mesh. In contrast, Tornado and PanelAero use geometric rotations of hinged panel sub-elements. At a reduced geometric level, ASWING incorporates deflections directly into the structural beam as control points, while SUAVE relies primarily on empirical sectional wing corrections. At a higher fidelity level, SHARPy resolves non-linear hinge rotations directly within its coupled structural-aerodynamic grid. Consequently, a unified low-fidelity approach that handles robust panel-level control kinematics remains an open challenge. AIC matrix matrix export capabilities differ widely across existing solvers. AVL, ASWING, Tornado, and SUAVE do not offer direct

programmatic extraction of the AIC matrix. Instead, they provide only stability derivatives or internally coupled solutions. In contrast, DLMPro, NeoCASS, and PanelAero provide explicit AIC matrix outputs, while SHARPy derives equivalent state-space matrices internally. Consequently, the lack of direct, standardized AIC matrix export in most low-fidelity tools restricts their modular integration into external pipelines. Lastly, regarding programming languages and software availability, existing code is fragmented across Fortran, C++, Python, and MATLAB. In scientific computing and aerospace engineering education, Python and MATLAB have mostly emerged as the two primary competing platforms [28]. While Python is widely praised for its open-source library ecosystem, recent empirical research on student learning performance and technical adoption has revealed that MATLAB and GNU Octave offer better usability and higher execution efficiency for engineering students working on complex numerical implementations [29].

To address these functional gaps, APGAA is developed natively in the MATLAB and GNU Octave programming environment, leveraging its optimized linear algebra routines for dense matrix inversion, native complex-valued arithmetic for calculations, and built-in multi-core parallel processing [30-33]. Since the languages of MATLAB and GNU Octave are almost identical, with minimal exceptions. The proposed APGAA source code can be used in both environments. This can eliminate software licensing barriers for academic institutions and independent researchers. The APGAA codebase avoids proprietary toolboxes and maintains full compatibility for execution with GNU Octave [34], a free, open-source platform. This ensures transparent, uncoupled access computation resources without requiring MATLAB code modification or commercial license fees.

Beyond computational speed, MATLAB and GNU Octave provide a simpler coding language, an accessible interface, and an advanced graphical engine, allowing students and researchers to efficiently debug code and visualize aerodynamic models and results. To systematically position the proposed APGAA framework within this aerodynamic computational landscape, Table 1 summarizes the evaluated tools across their numerical formulations, preprocessing features, and software architectures.

Table 1 Functional comparison of open-source aerodynamic frameworks

Software	**Aerodynamic mesh generation**	**Aerodynamic solvers**	**Control surface deflection**	**AIC matrix export**	**Programming language**
Tornado [20]	Parametric multi-surface	VLM	Hinged panel rotation	No (Loads outputs)	MATLAB
AVL [21]	Script-based geometry	VLM	Surface deflection angles	No (Trim outputs)	Fortran / C
DLMPro [22]	Manual panel input	VLM/DLM	Not supported	Direct	MATLAB
ASWING [23]	Coupled Structural beams	Non-linear Lifting-Line	Integrated control points	Internal coupling only	Fortran
SUAVE [24]	Built-in wing builders	VLM	Empirical/panel factors	No (Loads outputs)	Python
NeoCass [25]	Semi-automated modules	VLM/DLM	Control surface modes	Internal coupling only	MATLAB / C
SHARPy [26]	Structural beam-to-surface grid	VLM/UVLM	Arbitrary kinematic inputs	State-space realization	Python / C++
PanelAero [27]	NASTRAN bulk data / script	VLM/DLM	NASTRAN control cards	NASTRAN control cards	Python
APGAA (proposed)	**Parametric multi-trapezoidal surfaces discretization**	**VLM/DLM**	**Parametric hinge mapping**	Direct	**MATLAB**

2.3 Formulation of panel methods

The VLM and DLM used here are implemented based on the DLR report [27]. These methods are used to calculate the pressure distribution on lifting surfaces. They rely on the computation of an AIC matrix, which depends on the geometry of the aircraft, Mach number (M_a) and the reduced frequency (k), a non-dimensional parameter representing the oscillatory motion of the structure, defined by

$$k = \omega/V, \tag{1}$$

Where ω is the frequency of oscillation and V is the freestream velocity. Some implementations refer to the Nastran definition, in which the reduced frequency includes a cord reference c_{ref}, leading to

$$k = c_{\mathrm{ref}}\,\omega/2V, \tag{2}$$

The geometry is discretized into a mesh of aerodynamic panels that represent the lifting surfaces. The AIC matrix then relates a normalized induced downwash (w_j) on each aerodynamic panel to a circulation strength (Γ_j), and these values are translated into a complex pressure coefficient (c_p) as expressed in

$$\Delta \boldsymbol{c}_{\mathrm{p}} = \mathbf{AIC}(M_a, k) \cdot w_j, \tag{3}$$

The formulation of the VLM is described and derived by Katz and Plotkin [7] and the full derivation of the DLM is given by Albano and Rodden [11] with improvements and extensions to the DLM [35-39]. Both methods model the lifting surfaces using horseshoe vortices. To address the compressibility effects, the Prandtl-Glauert transformation is applied to geometry as suggested by Hedman [39]. The induced velocities at control points are computed based on the orientation of each panel in its local coordinate system.

To calculate and unite these induced velocities, the VLM uses the Biot–Savart Law to evaluate these velocities as projected onto the surface normal to enforce the bounds, while the DLM applies the unsteady kernel functions at each panel, and then numerical integration is used to evaluate them. Quadratic integration is employed using the Desmarais method to approximate the coefficients. These induced velocities are projected onto the surface, resulting in induced normal velocity, and are related to the coordinate system. These computations yield the normal velocity at each control point, which is assembled into the AIC matrix. The VLM produces a real-valued AIC matrix representing steady aerodynamic effects, whereas the DLM outputs produce a complex-valued matrix that captures both the magnitude and phase of the unsteady response across varying reduced frequencies.

This software emphasizes simplicity and adaptability in both VLM and DLM processes for users. At the $k = 0$ state for the quasi-static case, it is noted that the solutions of the DLM should be equivalent to the VLM due to unsteady effects vanishing. However, since DLM involves frequency-domain integration, it is computationally more intensive and introduces numerical approximations. To address this, the steady (real) component of the DLM solution is replaced by the more accurate VLM solution, and the DLM solver returns only the unsteady increment. Figure 1 illustrates the internal workflow of the VLM and DLM implementations. In this software, both solvers can run simultaneously and perform vectorized operations during calculations to provide fast, accurate solutions for aerodynamic analysis. During panel generation, each panel is defined by four key control points: the vortex point (S_m), the collocation point (S_c), and two edge points of the vortex (S_i and S_o), as illustrated in Figure 2a. The DLM uses all four points in its calculations, whereas the VLM requires only the vortex point and the collocation point. It should be noted that any transonic effects, like compression shocks, are not captured. AIC correction techniques that use high-fidelity computational fluid dynamics (CFD) results are necessary to improve AIC accuracy in these cases, but these are not included in the proposed software. APGAA, using VLM and DLM methods as aerodynamic solvers, is only capable of analyzing in the subsonic regime. Since AICs are mandatory for aeroelastic analysis and may not be obtained directly via CFD, AICs for transonic and supersonic are commonly obtained with some correction techniques instead [40, 41]. However, this is beyond the scope of this study; general software for all speed regimes that require significant extensive study will be developed in the future.

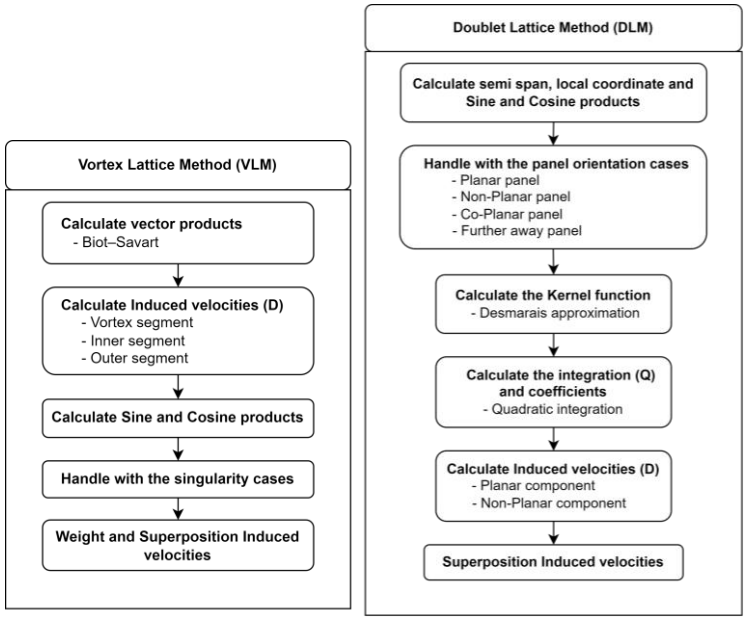


Figure 1 Flowcharts of VLM (a) and DLM (b)

2.4 Panel mesh generation

The panel modeling concept is based on a lifting-surface definition, and aerodynamic analysis is performed using either VLM or DLM. Because panel methods represent aerodynamic elements via singularity distributions on planar or mean-camber surfaces, the geometric input is defined by chord lines rather than by full airfoil contours. Thus, all lifting surfaces are modeled as flat panels corresponding to symmetric airfoil geometries. The main surface data (Surf), which includes lifting and control surfaces, and the control surface data (Sub-Surf) are designated as inputs. As shown in the Figure 2b, the Surf represents the entire surface outlined in blue, while the Sub-Surfs are highlighted in blue and green. The remaining pink surfaces represent the lifting surfaces. An advanced panel generation technique is introduced that automatically and simply generates multiple panel sections at different resolutions based on the defined surface positions. As shown in Figure 2b, modeling defines three regions.

The Surf is specified by providing the leading-edge points at the root and tip of the wing, along with the root chord length and tip chord length. For defining the Sub-Surfs, the locations are assigned relative to the Surf dimensions. In the chordwise direction, the start and end positions are defined by a percentage of the local chord length at each spanwise location. In the spanwise direction, the start and end positions are defined as a percentage of the total span length. This approach can enable more realistic modeling of complex configurations, including non-rectangular planforms such as tapered or swept wings. It could also support defining multiple Surf regions and capturing geometry with higher fidelity, particularly around control surfaces.

The meshing resolutions of the Surf and Sub-Surf regions are defined by specifying the number of divisions along the chordwise and spanwise directions, and they can be set independently. This enables local mesh refinement in critical areas, improving the accuracy and convergence of aerodynamic solutions. Typically, modeling control surfaces requires users to define them manually and separately from the lifting surfaces due to geometric discontinuities, especially at the junction between lifting and control surfaces. It often leads to gaps, overlaps, or misalignment in panel layouts, reducing simulation quality and increasing user effort.

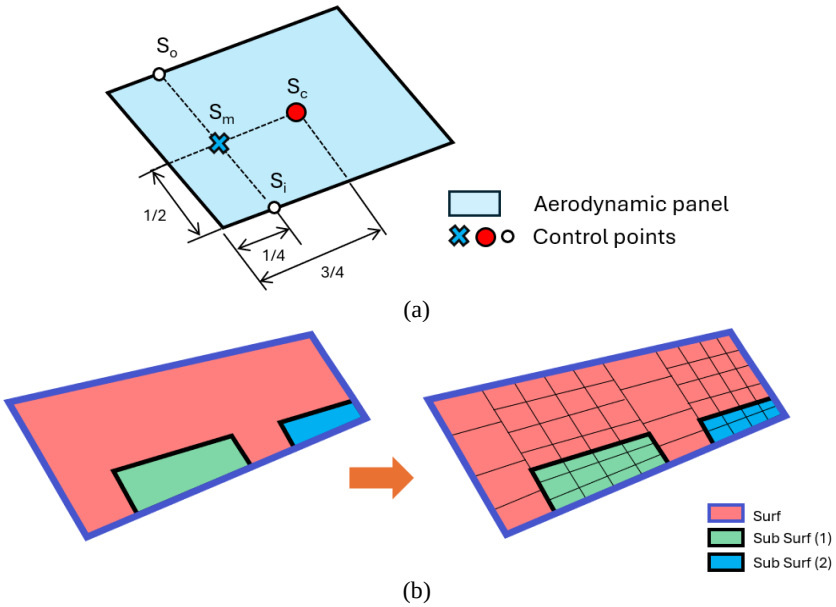


Figure 2 Creation of panels (a) and Main surface (Surf) and Sub-Surfaces (Sub-Surfs) (b)

In particular, many other tools require users to manually create multiple panel sections to avoid trailing-edge misalignments caused by control-surface cuts or deflections. In contrast, APGAA's panel generation ensures continuous and well-aligned paneling across all surfaces, eliminating the need for manual adjustments. To clarify this modeling approach, consider an example aircraft: lifting surfaces, such as the wing and empennage (comprising the horizontal and vertical tails), are defined as Surf, while control surfaces, such as flaps, ailerons, rudder, and elevator, are defined as Sub-Surf.

2.5 Software architecture and functionality

The framework of APGAA is organized as shown in Figure 3. The software workflow comprises three main processes: 1) Reading the input file, 2) Generating aerodynamic panels, and 3) Solving with VLM and DLM. To perform the analysis, the user must prepare an input file that contains the flight conditions and the aircraft geometry/configuration. Detailed instructions and examples for preparing these input files are available. After the input file is prepared, the analysis can be executed. In the first step of the analysis, the panels and control points are automatically generated. The aerodynamic analysis is consequently conducted using VLM and DLM. Several examples of input files are provided at the following directory: APGAA/tree/main/examples.

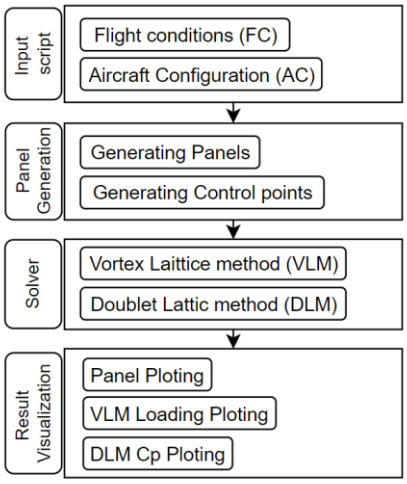


Figure 3 The framework of the APGAA software

The input script is divided into two sections: Flight Conditions (FC) and Aircraft Configuration (AC). The FC section defines the operational conditions under which the aircraft flies, including variables related to both the aircraft and the airflow. The AC section specifies the relevant geometric configuration of the aircraft, including the definitions of the Surf and Sub-Surf surfaces based on fundamental aircraft design principles. These dimensional specifications are sufficient to construct the aircraft model and generate the panel mesh. Additionally, airfoil profiles and symmetry definitions can be provided for inclusion in the model.

Once the user completes the input script, the variables are passed to the panel generation process. For each Surf defined, the surface is split into lifting surfaces and control surfaces. Each Sub-Surf mesh is generated separately relative to its parent Surf, and hinge-line data for the control surfaces are automatically created in this step. Specified dihedral and incidence angles are applied to correctly orient the panels, while the rotation axes for control surfaces are also defined according to the panel groupings. This automated panel generator simplifies parts of the panel orientation and the placement processes, which are often challenging in aircraft modeling. After

all Surfs are processed, the panels and associated control points are generated. The aerodynamic conditions and control points are then imported into the VLM and DLM solvers to compute the aerodynamic solutions. Each solver can also be used independently, if required by the user. The essential matrices are calculated to determine the pressure coefficient distribution and the AIC matrices. It should be noted that internal calculations within the VLM and DLM solvers do not directly output aerodynamic forces such as lift, drag, or side force.

However, these forces can be derived from the computed circulation and pressure distributions produced by the solvers. Finally, the software provides visualization tools that allow users to plot panel layouts and aerodynamic force distributions, aiding in analysis and verification.

3. Results

3.1 Solver validation

To ensure the numerical reliability and predictive accuracy of both the VLM and DLM solvers, thorough verification and validation were conducted. The solver implementations in APGAA are derived and refined from the PanelAero framework developed by the German Aerospace Center (DLR) [27], against which the APGAA results are benchmarked. The validation process involves comparing the phases of the pressure coefficient results across several demonstrated cases, including 1) a full aircraft (FAC), 2) closely coplanar surfaces (CCS), and 3) distantly positioned surfaces (DPS). The FAC case validates the accuracy of the solvers using a full-aircraft model based on the standard DLR Allegra research-aircraft geometry configuration. The CCS and DPS cases specifically assess the correctness of wing alignment relative to the horizontal tail. In the CCS case, the wing and tail are positioned nearly in the same plane, while in the DPS case, the horizontal tail is placed offset above the wing. All benchmark test cases were computed at a subsonic compressible Mach number ($M_a = 0.80$) and a freestream air density ($\rho_\infty = 1.225$). Unsteady dynamic responses were evaluated across three reduced frequencies (k) involving 0.001, 0.6, and 1.4, respectively. These values represent quasi-steady approximation and high-frequency unsteady motion. All lifting surfaces are modeled as planar flat panels, which are aerodynamically equivalent to a symmetric airfoil profile with zero camber. Complete numerical scripts, geometric panel meshes, and extended validation datasets are documented and accessible in the repository.

The computed pressure coefficient (C_p) distributions across all benchmark cases are illustrated in Figure 4 to Figure 6. In the steady flow regime, the expected aerodynamic behavior is observed: maximum pressure coefficient occurs along the leading edge and decreases monotonically chordwise toward the trailing edge. For the FAC case shown in Figure 4, the real component of C_p obtained from both VLM and DLM at $k = 0.001$, exhibits strong agreement; maximum values reach approximately 1.6 at the wing-root leading edge and decrease spanwise to 1.1 at the wing tip. The horizontal tail surfaces generate secondary peak values ranging between 1.0 and 1.2. For dynamic unsteady cases evaluated via DLM at $k = 0.6$ and $k = 1.4$. Both the real and imaginary components were investigated simultaneously, capturing the expected phase shifts that arise as the reduced frequency increases. For the CCS and DPS cases (Figure 5 and Figure 6), the aerodynamic panel alignment yields smooth numerical behavior free of nonphysical singularities. Wing peak C_p values range from 0.65 to 0.70. Downstream on the horizontal tail, peak values reach approximately 0.50 in the DPS case and 0.2 in the CCS case due to downwash and wake interaction. Across all evaluated configurations, the spatial distributions predicted by APGAA show good agreement with the benchmark solutions from PanelAero.

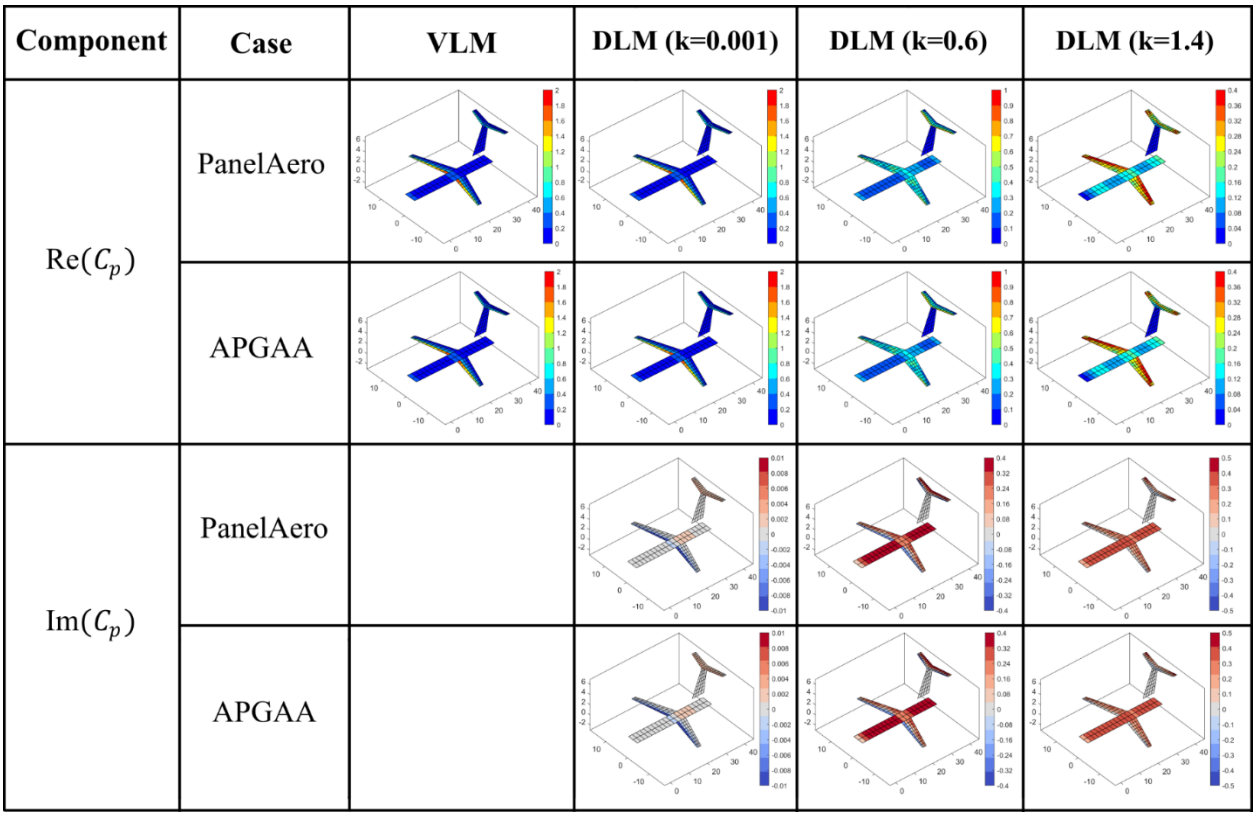


Figure 4 Pressure coefficient distribution in validation: APGAA vs. PanelAero – FAC case

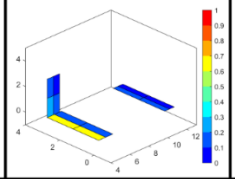
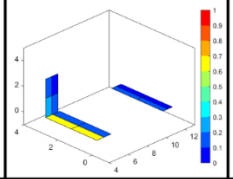
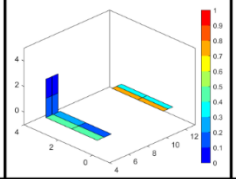
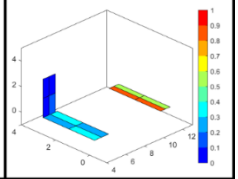
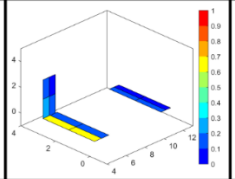
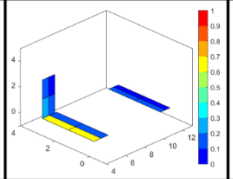
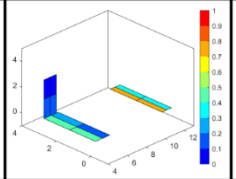
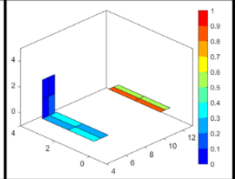
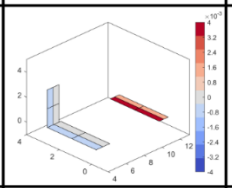
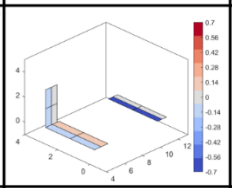
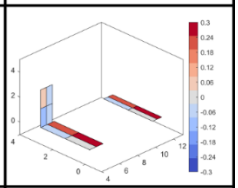
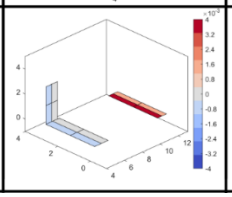
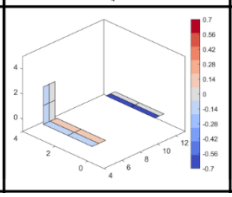
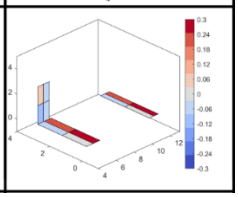
Component	Case	VLM	DLM (k=0.001)	DLM (k=0.6)	DLM (k=1.4)
$\mathrm{Re}(C_p)$	PanelAero				
	APGAA				
$\mathrm{Im}(C_p)$	PanelAero				
	APGAA				

Figure 5 Pressure coefficient distribution in validation: APGAA vs. PanelAero – CCS case

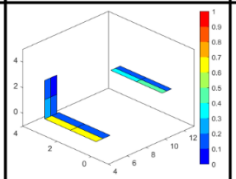
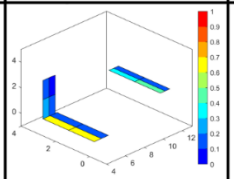
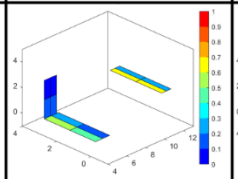
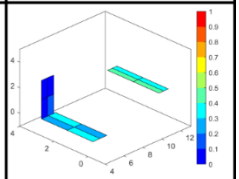
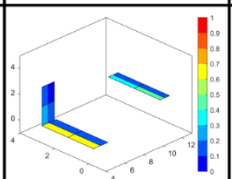
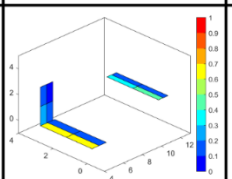
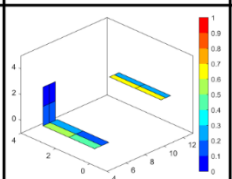
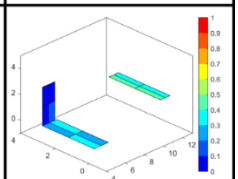
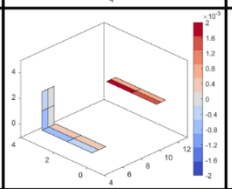
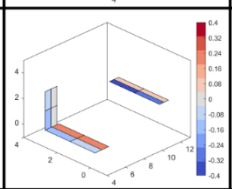
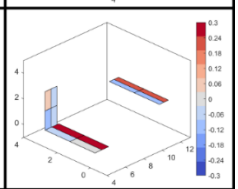
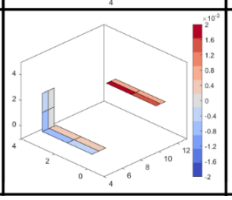
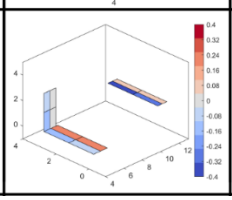
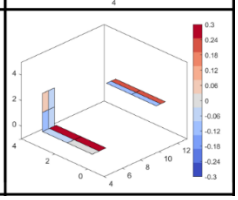
Component	Case	VLM	DLM (k=0.001)	DLM (k=0.6)	DLM (k=1.4)
$\mathrm{Re}(C_p)$	PanelAero				
	APGAA				
$\mathrm{Im}(C_p)$	PanelAero				
	APGAA				

Figure 6 Pressure coefficient distribution in validation: APGAA vs. PanelAero – DPS case

To clarify the different pressure coefficients, using absolute (a_{tol}) and relative (r_{tol}) tolerances of 0.0001 and 0.001, respectively, as implemented in the DLR report [27]. The computed error criterion is defined by

$$\left|C_{p,\mathrm{APGAA}} - C_{p,\mathrm{PanelAero}}\right| \leq a_{\mathrm{tol}} + r_{\mathrm{tol}}\left|C_{p,\mathrm{PanelAero}}\right| \tag{4}$$

where $C_{\mathrm{p,APGAA}}$ and $C_{\mathrm{p,PanelAero}}$ represent the pressure coefficients obtained from the proposed APGAA and PanelAero.

The comparison results are illustrated in Figure 7 and Figure 8. Each Figure represents the phases of the pressure coefficient at various reduced frequencies, where Figure 7 displays the real part ($\mathrm{Re}(C_p)$) and Figure 8 displays the imaginary part ($\mathrm{Im}(C_p)$). To identify the highlighted converging issue, converged results appear in white, while non-converged results are in red. The colored pressure-coefficient results are displayed on a panel-by-panel basis. The results obtained by APGAA coincide with those from PanelAero across all panels and cases, as evidenced by the absence of red panels.

To quantitatively assess solver consistency, statistical error metrics between APGAA and PanelAero are summarized in Table 2 for the real and imaginary components across the FAC, CCS, and DPS test cases. Across all configurations and flow conditions, the evaluated error metrics, including maximum (MAX), minimum (MIN), mean absolute error (MEAN), and standard deviation (STD), remain consistently below 10^{-10} or lower. This close equivalence in the resulting pressure coefficient distributions verifies the accuracy of the calculation and the inversion of the AIC matrices.

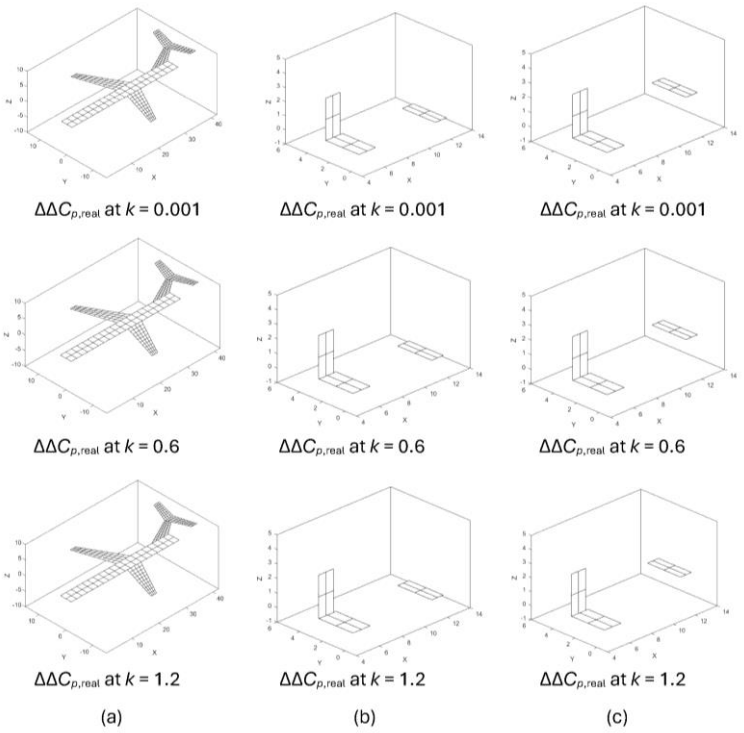


Figure 7 Error of Pressure coefficient in $\mathrm{Re}(C_p)$ validation: APGAA vs. PanelAero – (a) FAC, (b) CCS and (c) DPS cases

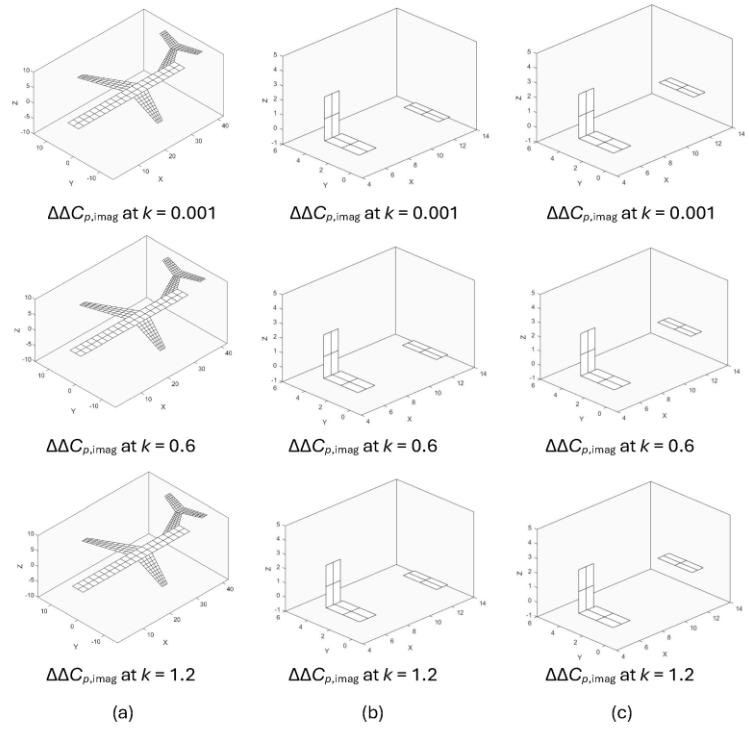


Figure 8 Error of pressure coefficient in $\mathrm{Im}(C_p)$ validation: APGAA vs. PanelAero – (a) FAC, (b) CCS and (c) DPS cases

Table 2 Statistical comparison of error data in cases

a) Full aircraft case

Complex number	Statistical Data	VLM	DLM		
			k=0.001	k=0.6	k=1.4
$\mathrm{Re}(C_p)$	MIN	0	0	0	9.714E-17
	MAX	1.155E-14	1.354E-14	1.049E-14	4.426E-13
	MEAN	1.095E-15	1.208E-15	1.515E-15	2.918E-14
	STD	2.133E-15	2.340E-15	2.186E-15	6.237E-14
$\mathrm{Im}(C_p)$	MIN	N/A	0	0	5.814E-17
	MAX	N/A	7.459E-17	1.496E-14	4.251E-13
	MEAN	N/A	7.247E-18	1.254E-15	2.784E-14
	STD	N/A	1.290E-17	1.959E-15	5.585E-14

b) Closely coplanar surfaces case

Complex number	Statistical Data	VLM	DLM		
			k=0.001	k=0.6	k=1.4
$\mathrm{Re}(C_p)$	MIN	0	0	0	0
	MAX	1.388E-16	2.220E-16	2.220E-16	2.220E-16
	MEAN	2.486E-17	7.286E-17	3.874E-17	5.692E-17
	STD	4.158E-17	7.876E-17	6.424E-17	6.602E-17
$\mathrm{Im}(C_p)$	MIN	N/A	6.776E-21	0	0
	MAX	N/A	8.674E-19	1.527E-16	2.220E-16
	MEAN	N/A	2.535E-19	2.739E-17	7.170E-17
	STD	N/A	2.967E-19	4.446E-17	7.085E-17

c) Distantly positioned surfaces case

Complex number	Statistical Data	VLM	DLM		
			k=0.001	k=0.6	k=1.4
$\mathrm{Re}(C_p)$	MIN	0	0	0	0
	MAX	5.551E-17	5.551E-17	1.110E-16	1.110E-16
	MEAN	7.517E-18	2.486E-17	3.600E-17	2.891E-17
	STD	1.711E-17	2.604E-17	4.190E-17	3.436E-17
$\mathrm{Im}(C_p)$	MIN	N/A	0	0	0
	MAX	N/A	2.168E-19	1.665E-16	5.551E-17
	MEAN	N/A	4.630E-20	3.556E-17	2.139E-17
	STD	N/A	6.342E-20	4.803E-17	2.262E-17

3.2 Illustrative examples

An example of an input script for aircraft definition is shown in Table 3. The example aircraft configuration has a backward-swept, dihedral wing and a conventional tail with backward sweep and no dihedral. The lifting surface creation consists of three main components: the wing, horizontal tail, and vertical tail. Symmetrical airfoil profiles (zero-camber flat panels) are applied across all lifting surfaces. Each component includes corresponding control surfaces (CS). The 0° inflow at 100 m/s and a flight speed of 0.5 Mach are aircraft flight conditions within the subsonic flow regime. Two load cases are considered: a) symmetrical on flow condition, b) 5.0° deflection of the second control surface (CS.2) in the wing, which demonstrates the effects of control surface deflection. The above-mentioned aircraft results generated by APGAA are shown in Figures 9-11.

In Figure 9, the aerodynamic model of the aircraft is displayed with different colors representing the various components and the generated panel mesh. Moreover, the axis arrows are visible for visualizing the rotation hinge of the control surfaces. The aerodynamic load resultants in the x, y, and z directions are shown in Figure 10a and Figure 10b, indicating the quantity and distribution of load across panels for the two load cases. In the symmetrical load case (Figure 10a), the load distribution on the left and right is symmetric, and the high lift forces at the leading edge gradually decrease chordwise until the low lift forces are located at the trailing edge. In the 5.0° deflection load case, Figure 10b represents asymmetric load distribution. The positive force pattern is on the right wing and the negative force pattern on the left wing. The lifting load pattern of the tail is similar to that of the wings; however, sideslip forces appear on the vertical tail. These sideslip forces reflect the force imbalance from the control surface deflection.

The pressure coefficients on each aerodynamic panel in the top view of the aircraft are computed by the VLM solver and illustrated in Figure 10c and 10d. Under symmetric flight conditions, the high C_p occurs roughly 0.60 along the wing leading edge and gradually diminishes chordwise to about 0.025 near the trailing edge. In the deflected control surface case, the local load distribution shifts noticeably, where the active control panel exhibits a localized positive maximum C_p nearly 0.80, and the opposite antisymmetrically deflected surface exhibits a negative minimum C_p near -0.50. This demonstrates the solver's ability to capture asymmetric control deflections and local aerodynamic loads.

As unsteady effects are computed by the DLM solver, Figure 11 presents the magnitudes and phases of the pressure coefficients across the aerodynamic panels. As the reduced frequency varies, both the magnitude and the phases of the pressure coefficients change accordingly. As k increases from 0.001 to 1.4, the magnitude distributions (Figures 11a and 11b) become progressively non-uniform, reflecting stronger dynamic unsteadiness. In the symmetric flow case at closely quasi-steady conditions ($k = 0.001$), the peak real component remains concentrated along the leading edge, ranging from approximately 0.30 to 0.60 in terms of $\mathrm{Re}(C_p)$ At a high reduced frequency of $k = 1.4$, while the general chordwise load profile is preserved, the peak magnitudes decrease as unsteady aerodynamic damping redistributes the dynamic load into out-of-phase components. For the control surface deflection case at $k = 0.001$, the pressure distribution indicates the steady localized load concentration induced by surface deflection. As the frequency increases to $k =$

1.4, these localized peak magnitudes reduce smoothly while their values preserve the overall characteristic deflection profile. Similarly, the phase distributions (Figures 11c and 11d) show greater phase change, reflecting larger phase differences across the panels as the frequency increases. Regarding the imaginary component, the values remain close to zero at $k = 0.001$. At high reduced frequency ($k = 1.4$), the imaginary component exhibits negative values along the wing leading edge, spanning roughly -0.06 to -0.10 for $\mathrm{Im}(C_p)$. After that, these values cross zero in chordwise and reach positive values near 0.06 to 0.10 at the trailing edge.

Consequently, it should be emphasized that accurately capturing these spatial phase variations is critical, as they provide essential physical insight into the dynamic lag and damping characteristics across the lifting surfaces. Furthermore, these trends confirm that the numerical formulation remains robust, physically consistent, and free of nonphysical singularities across coplanar and interfering panels. Consequently, this ensures the reliable computation of generalized unsteady aerodynamic forces required for aeroelastic stability and flutter boundaries. Furthermore, the extracted phase behavior can be extended to support unsteady aerodynamic analyses for other aircraft models under different operating conditions. At this point, the results demonstrate the APGAA software's capabilities for both research and educational purposes. The APGAA repository provides multiple examples and tutorials implemented in MATLAB live scripts, each offering a detailed explanation of the input data format and modeling procedures. For users seeking deeper insights, additional resources and extended examples are also available in the repository.

Table 3 Brief of flow conditions and specifications of an example model aircraft.

Flight conditions (FC)	Units	Value
Onflow Velocity of Air	m/s	100
Density of Air	kg/m^3	1.2
Mach number	-	0.5
Aircraft specification (AC)	Units	Value
Wings (Surf$_1$)		
Root Chord	m	1.25
Tip Chord	m	1.2
Span	m	6
Dihedral angle	°	5
Incidence angle	°	3
Control surface 1 (Sub-Surf$_{1,1}$)		
Span interval of starting	%	10
Span interval of ending	%	50
Chord interval of starting	%	70
Chord interval of ending	%	100
Rotation angle	°	-10
Control surface 2 (Sub-Surf$_{1,2}$)		
Span interval of starting	%	60
Span interval of ending	%	100
Chord interval of starting	%	70
Chord interval of ending	%	100
Rotation angle	°	-10
Horizontal tail (Surf$_2$)		
Root Chord	m	0.75
Tip Chord	m	0.5
Span	m	2.5
Dihedral angle	°	-5
Incidence angle	°	-2
Control surface 3 (Sub-Surf$_{2,1}$)		
Span interval of starting	%	0
Span interval of ending	%	100
Chord interval of starting	%	60
Chord interval of ending	%	100
Rotation angle	°	-10
Vertical tail (Surf$_3$)		
Root Chord	m	0.75
Tip Chord	m	0.5
Span	m	2.5
Dihedral angle	°	90
Incidence angle	°	0
Control surface 4 (Sub-Surf$_{3,1}$)		
Span interval of starting	%	50
Span interval of ending	%	100
Chord interval of starting	%	60
Chord interval of ending	%	100
Rotation angle	°	-10

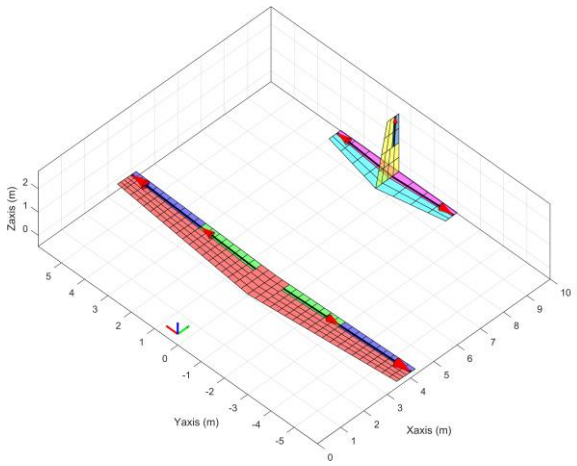


Figure 9 Aerodynamic panel model of example aircraft

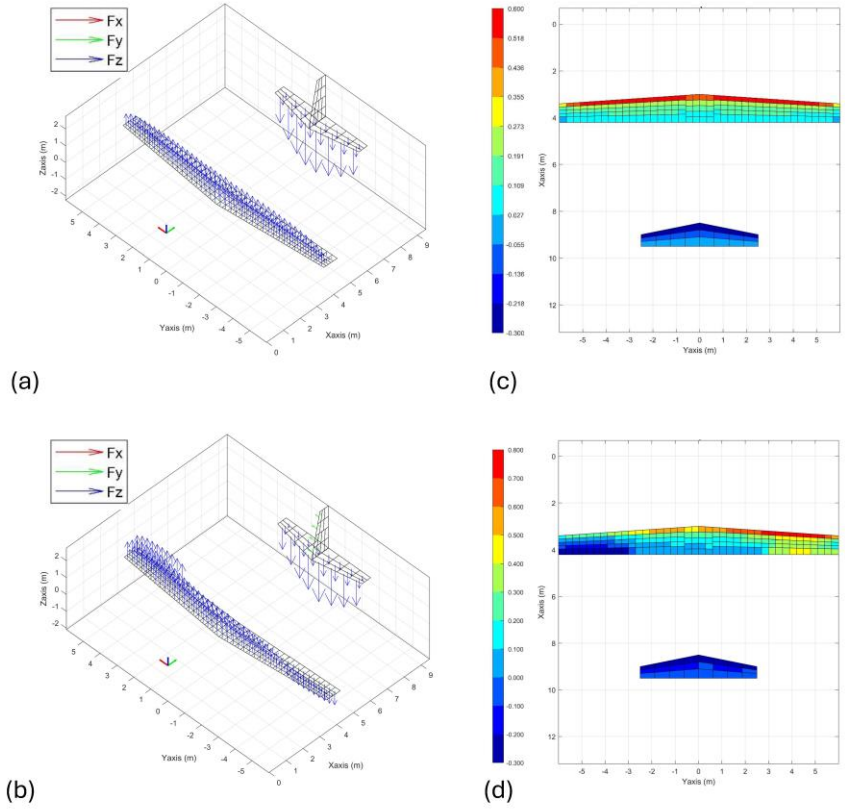


Figure 10 Aerodynamic loading model: (a) symmetry and (b) 5.0° deflection of CS.2 and the magnitudes of pressure coefficient VLM results (k=0) with (c) symmetry and (d) 5.0° deflection of CS.2 load cases

4. Conclusion

Aerodynamic analysis software based on panel methods, including both the VLM and DLM, has been developed in MATLAB. The software features advanced panel-generation capabilities, enabling the automatic creation of complete aerodynamic models of aircraft, including detailed control-surface modeling. The accuracy of the solvers was validated by comparison with results from the PanelAero tool, and the proposed software demonstrated acceptable agreement across a variety of test cases. The example cases demonstrated the software's ability to accurately capture critical aerodynamic phenomena, including the effects of control surface deflections on load distributions, pressure variations, and force balance, under both steady and unsteady conditions across a range of reduced frequencies. While APGAA currently provides robust, steady, and unsteady aerodynamic load calculations, several avenues remain for future development. First, the scope will be expanded beyond basic load prediction to encompass aircraft control maneuverability, static stability, and trim analysis, providing a more comprehensive design and analysis suite. Second, integrating alternative or complementary aerodynamic solvers, such as the Vortex Ring Method (VRM), is planned to offer greater modeling flexibility and potentially enhanced computational efficiency for specific aerodynamic configurations. Third, coupling APGAA with structural solvers to enable high-fidelity aeroelastic flutter and gust response analysis remains a primary long-term goal for extending

its applicability to flexible aircraft structures. Ultimately, as an open-source initiative, APGAA emphasizes transparency, computational flexibility, and broad accessibility. It serves as a practical, high-impact computational tool well-suited for academic research in preliminary aircraft design, control surface effectiveness, and aeroelastic studies. Furthermore, the inclusion of extensive benchmark examples and step-by-step tutorials makes it an ideal pedagogical resource for aerospace education, providing students and researchers with valuable hands-on experience in numerical aerodynamic modeling and unsteady flight mechanics.

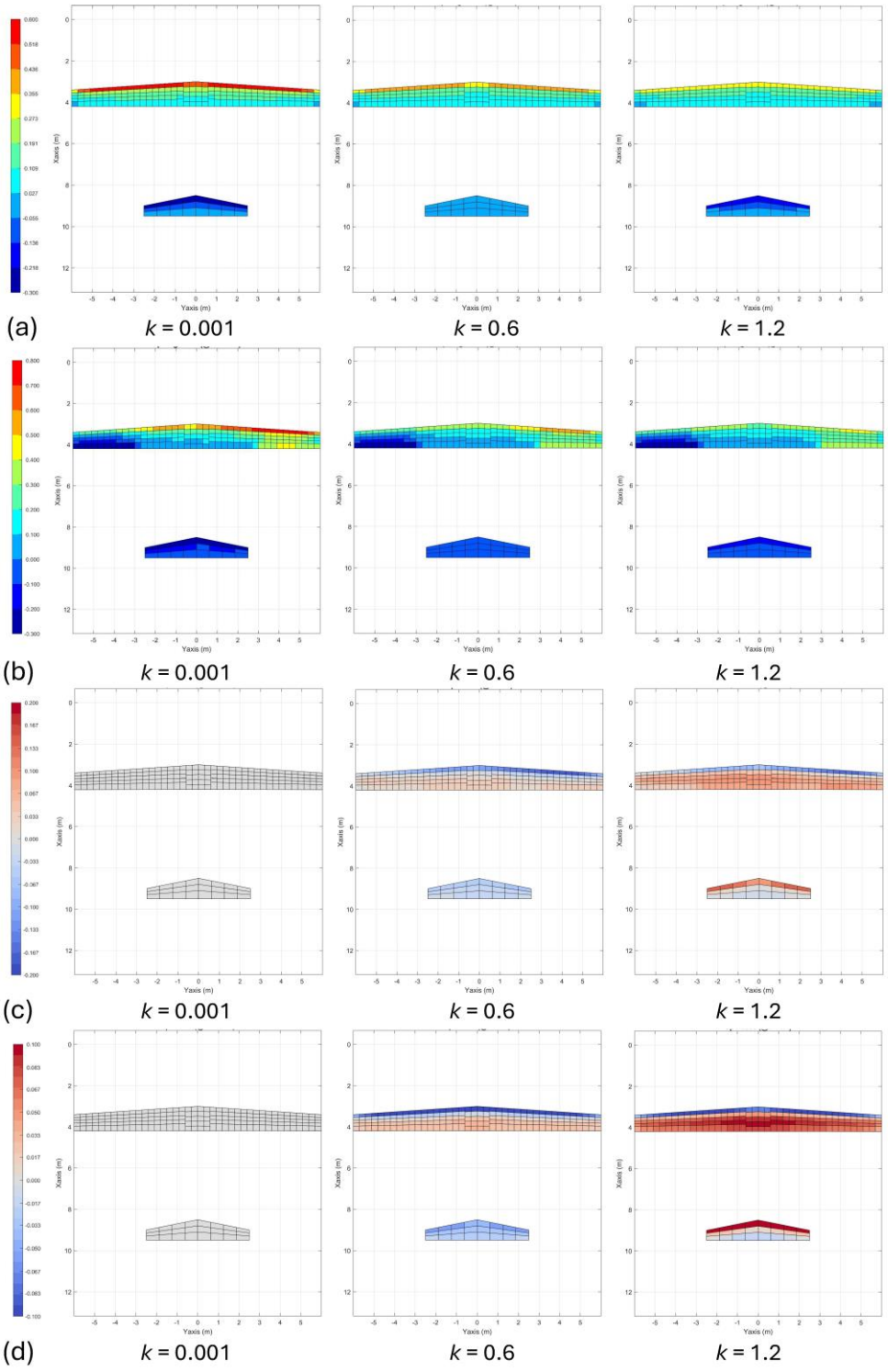


Figure 11 The magnitudes of pressure coefficient DLM results with different load cases: (a) symmetry and (b) 5.0° deflection of CS.2 cases and the phases of pressure coefficient DLM results with different load cases: (c) symmetry and (d) 5.0° deflection of CS.2 case

6. Acknowledgements

This project is funded by the National Research Council of Thailand (NRCT) through a Research Team Promotion Grant (Grant No. N42A650549) and the Royal Golden Jubilee Ph.D. Program (Grant No. N41A661144).

7. References

[1] Panagiotou P, Yakinthos K. Aerodynamic efficiency and performance enhancement of fixed-wing UAVs. Aerosp Sci Technol. 2020;99:105575.
[2] Belardo M, Beretta J, Marano AD, Diodati G, Paletta N, Di Palma L. On the preliminary structural design strategy of the wing of the next-generation civil tiltrotor technology demonstrator. Int J Aeronaut Space Sci. 2021;22:613-24.
[3] Bravo-Mosquera PD, Cerón-Muñoz HD, Catalano FM. Design, aerodynamic analysis and optimization of a next-generation commercial airliner. J Braz Soc Mech Sci Eng. 2022;44:609.
[4] Liu Y, Mo D, Nalianda D, Li Y, Roumeliotis I. Review of more electric engines for civil aircraft. Int J Aeronaut Space Sci. 2022;23:784-93.
[5] Kanokmedhakul Y, Pholdee N. Follow-up study of aircraft parameter estimation for lateral dynamics using self-adaptive teaching-learning based optimization with acceptance probability. Eng Appl Sci Res. 2025;52(3):251-9.
[6] Sadraey MH. Aircraft design: a systems engineering approach. 2nd ed. Hoboken: Wiley; 2024.
[7] Katz J, Plotkin A. Low-Speed aerodynamics. 2nd ed. Cambridge: Cambridge University Press; 2001.
[8] Kang WR, Park SH, Lee DJ. Simulation of counter-rotating vortex pairs on a triangular-flapped wing with flap-tip free vortex modeling. Int J Aeronaut Space Sci. 2018;19:805-15.
[9] Nguyen AT, Han JH, Vu TT. The effects of wing mass asymmetry on low-speed flight characteristics of an insect model. Int J Aeronaut Space Sci. 2019;20:940-52.
[10] Ishibashi H, Imamura T, Rinoie K. Estimation method for pitching moment non-linearities of cranked arrow wing at low speed. Int J Aeronaut Space Sci. 2022;23:680-91.
[11] Albano E, Rodden WP. A doublet lattice method for calculating lift distributions on oscillation surfaces in subsonic flows. AIAA J. 1969;7(2):279-85.
[12] Been Y, Kang Y, Kim S, Shin S, Jang S. Development of the snapshot method for six degree-of-freedom flight dynamics simulation of a high aspect ratio wing aircraft. Int J Aeronaut Space Sci. 2018;19:904-19.
[13] Wansasueb K, Bureerat S, Kumar S. Ensemble of four metaheuristic using a weighted sum technique for aircraft wing design. Eng Appl Sci Res. 2021;48(4):385-96.
[14] Rafiee R, Farsadi T, Tehrani MA, Sharifi P. Linkage learning optimization of aeroelastic and structural behavior of composite wings. Int J Aeronaut Space Sci. 2023;24:1187-98.
[15] Jeong I, Lee SL, Lee H, Kang W, Park JS, Cho H. Mid-Fidelity aeroelastic framework for main rotor load prediction in forward flight of medium utility helicopters. Int J Aeronaut Space Sci. 2026;27:2245-63.
[16] Higuchi H. Enhancement of aerodynamics and flight dynamics instruction with interactive computer visualizations. Comput Appl Eng Educ. 1994;2(2):87-95.
[17] Shankar P, Chung WT, Husman J, Wells V. A novel software framework for teaching aircraft dynamics and control. Comput Appl Eng Educ. 2015;23(1):63-71.
[18] Liu H, Jin X. Digital manufacturing course framework for senior aircraft manufacturing engineering undergraduates. Comput Appl Eng Educ. 2020;28(2):338-56.
[19] Ghosh S, Kundu S, Neogy S, Nandi A. An object-oriented approach toward two-dimensional engineering mechanics. Comput Appl Eng Educ. 2023;31(5):1372-97.
[20] Melin T. A vortex lattice MATLAB implementation for linear aerodynamic wing applications [thesis]. Sweden: Royal Institute of Technology; 2000.
[21] Budziak K. Aerodynamic analysis with athena vortex lattice (AVL). Hamburg: Aircraft Design and Systems Group (AERO), Department of Automotive and Aeronautical Engineering; 2015.
[22] Böhnisch N, Braun C, Koschel S, Marzocca P. Whirl flutter for distributed propulsion systems on a flexible wing. AIAA SCITECH 2022 Forum; 2022 Jan 3-7; San Diego: Virtual. San Diego: AIAA; 2021.
[23] Drela M. Integrated simulation model for preliminary aerodynamic, structural, and control-law design of aircraft. 40th Structures, Structural Dynamics, and Materials Conference and Exhibit; 1999 Apr 12-15; St. Louis, USA. St. Louis: AIAA; 1999.
[24] Wendorff A, Variyar A, Ilario C, Botero E, Capristan F, Smart J, et al. SUAVE: An aerospace vehicle environment for designing future aircraft 2020 [suavecode]. USA: GitHub; 2020.
[25] Fonte F, Ricci S. Recent developments of NeoCASS the open source suite for structural sizing and aeroelastic analysis. International Forum on Aeroelasticity and Structural Dynamics (IFASD 2019); 2019 Jun 10-13; Savannah, USA. p. 1-22.
[26] del Carre A, Muñoz-Simón A, Goizueta N, Palacios R. SHARPy: A dynamic aeroelastic simulation toolbox for very flexible aircraft and wind turbines. J Open Source Softw. 2019;4(44):1885.
[27] Voß A. An implementation of the vortex lattice and the doublet lattice method. Göttingen: Institute of Aeroelasticity; 2020.
[28] Fangohr H. A comparison of C, MATLAB, and Python as teaching languages in engineering. In: Bubak M, van Albada GD, Sloot PMA, Dongarra J, editors. Computational Science - ICCS 2004. Berlin: Springer; 2004. p. 1210-7.
[29] Cuevas EH, Zaldivar D, Perez M. Impact of programming languages on learning performance. Int J Inf Commun Technol Educ. 2025;21(1):1-17.
[30] Bilal A, Siddiqui F, Abbas M, Mansoor M. Aircraft wing structural design application in MATLAB app designer. Computer Meth Mater Sci. 2021;21(4):193-202.
[31] Horri N, Pietraszko M. A tutorial and review on flight control co-simulation using MATLAB/Simulink and flight simulators. Automation 2022;3(3):486-510.
[32] Iljukhin SN, Klishin AN, Chudinova ON. Approach to the presentation of aerodynamic characteristics in the MATLAB Simulink. AIP Conf Proc. 2022;2383:020002.
[33] Wani KUF, Nallasivam K. Dynamic simulation of aircraft vehicle-rigid pavements interaction on Pasternak foundation using finite element-based MATLAB algorithm. Innov Infrastruct Solut. 2025;10:277.
[34] Trejo JT, Pérez AF, Huamán MC, Quiroz JA, Huamán RT, Olivera VB. Experiences in the usage of octave on improving learning mathematics in an engineering faculty. 2021 IEEE World Conference on Engineering Education (EDUNINE); 2021 Mar 14-17; Guatemala City, Guatemala. USA: IEEE; 2021. p. 1-4.
[35] Rodden WP, Giesing JP, Kalman TP. New developments and applications of the subsonic doublet-lattice method for nonplanar configurations. AGARD Conference Proceedings; 1970 Nov 3-4; Tonsberg, Norway.

[36] Rodden WP, Giesing JP, Kalman TP. Refinement of the nonplanar aspects of the subsonic doublet-lattice lifting surface method. J Aircraft. 1972;9(1):69-73.
[37] Rodden WP, Taylor PF, McIntosh Jr SC. Further refinement of the subsonic doublet-lattice method. J Aircraft. 1998;35(5):720-7.
[38] Rodden WP, Taylor PF, McIntosh Jr SC. Improvements to the Doublet-Lattice Method in MSC/NASTRAN. 1999 MSC Aerospace Users' Conference Proceedings; 1999 Jun 7-10; Long Beach, California. p. 1-16.
[39] Hedman SG. Vortex lattice method for calculation of quasi steady state loadings on thin elastic wings in subsonic flow. Stockholm: Aeronautical Research Institute of Sweden; 1966.
[40] Thormann R, Dimitrov D. Correction of aerodynamic influence matrices for transonic flow. CEAS Aeronaut J. 2014;5:435-46.
[41] Thelen AS, Leifsson LT, Beran PS. Multifidelity flutter prediction using local corrections to the generalized AIC. Aerosp Sci Technol. 2020;106:106032.