READ 2024
RESEARCH & EDUCATION IN AIRCRAFT DESIGN
WARSAW, POLAND | 6-8 NOVEMBER 2024

CEASIOMPY: A MODULAR AND OPEN-SOURCE PYTHON ENVIRONMENT FOR AIRCRAFT DESIGN – A TOOL FOR RAPID EVALUATIONS

G. Benedetti[1], R. Gauthier[1], J. Oppelstrup[2], A. Rizzi[2], J.B. Vos[1] & M. Zhang[3]

[1]CFS Engineering, EPFL Innovation Park, Batiment A, 1015 Lausanne, Switzerland
[2]KTH Royal Institute of Technology
[3]Airinnova, Marövägen 26, 18249 Enebyberg, Stockholm

Abstract

In today's fast-paced engineering environment, the need for rapid evaluation is paramount. Aircraft design processes integrate numerous phases, including aerodynamic analysis, structural evaluation and optimisation. CEASIOMpy, an open source and modular Python environment, addresses this challenge by streamlining the aircraft design workflow. This paper outlines the architecture of CEASIOMpy, describes its main modules, and demonstrates its use in collaborative projects such as Low Boom Aircraft Common Research Model (LARM) and Collaborative System of Systems Exploration of Aviation Products, Services and Business Model (COLOSSUS). We show that CEASIOMpy significantly reduces simulation time while maintaining robust accuracy, proving that it is an essential tool for modern aerospace engineering challenges.

Keywords: CEASIOMpy, aircraft design, python

1. Introduction

The rapid pace of innovation in modern engineering disciplines, particularly in the aerospace industry, requires tools that can streamline complex design processes. Aircraft design, which requires the integration of multiple disciplines such as aerodynamics, structural mechanics, and system optimization, depends on software environments that can provide fast and accurate evaluations. Traditionally, these processes have relied on time-consuming simulations and manual integration of disparate systems, which slows down the design cycle process. CEASIOMpy aims to address these challenges by making workflows more agile and efficient.

CEASIOMpy is an open source, Python-based platform specifically designed to facilitate the rapid development and evaluation of aircraft concepts. It builds on the legacy of its predecessor, the *Computerized Environment for Aircraft Synthesis and Integrated Optimization Methods (CEASIOM) [2]*, which was originally developed in MATLAB as part of the European funded SimSAC project. By moving to Python, CEASIOMpy has adopted a modular structure, improved scalability, and enhanced integration capabilities, making it highly adaptable to modern design workflows.
This paper provides an overview of CEASIOMpy's capabilities and explores its use in real-world projects, demonstrating its efficiency and versatility. In addition, we highlight how CEASIOMpy contributes to the education of the next generation of engineers by providing them with a robust, hands-on tool for rapid aircraft design and analysis.

2. Methodology

2.1 Architecture of CEASIOMpy

CEASIOMpy is built on a modular architecture that allows users to seamlessly integrate different simulation and analysis tools into a unified workflow. This modularity is crucial in aircraft design,

where different domains such as aerodynamics, structures and optimization must interact efficiently to streamline the development process.
At the core of CEASIOMpy lies the Common Parametric Aircraft Configuration Schema (CPACS) [1], an XML-based data management system. CPACS acts as a comprehensive repository for storing and manipulating aircraft models and their associated data, facilitating the smooth exchange of information between different modules and ensuring consistency throughout the design process. The CPACS framework manages critical data such as:

- *Aircraft Geometry*: Defines the physical shape and layout of the aircraft, including the wings and fuselage.
- *Aerodynamic Properties*: Manage performance metrics such as lift, drag, and moments across different flight regimes.
- *Structural Parameters*: Material properties, stiffness, and load capacity.
- *Mission Analysis*: Supporting data for flight missions, performance evaluation, and fuel consumption analysis.

By centralizing these diverse data sets, CPACS enables the integration of aerodynamic, structural analysis, and performance evaluation tools within the CEASIOMpy environment, enabling automated and coherent aircraft design workflows.
The modularity of CEASIOMpy allows users to tailor their workflow to the specific requirements of their project. Key modules include

- **Aerodynamics Module**: Supports low- and medium-fidelity simulations using methods such as the *Vortex Lattice Method (VLM)*, providing quick aerodynamic assessments, and *SU2* [7] for medium-fidelity (Euler) simulations.
- **Geometry & Mesh Module**: Facilitates the automatic generation of high-quality meshes, which are essential for both aerodynamic and structural simulations.
- **Structural Module**: Handles *fluid-structure interaction (FSI)* analyses using simplified, low-fidelity models to provide rapid estimations of structural behavior under different load conditions.

These components work together to create a cohesive environment that reduces the complexity typically associated with integrating disparate analysis tools.

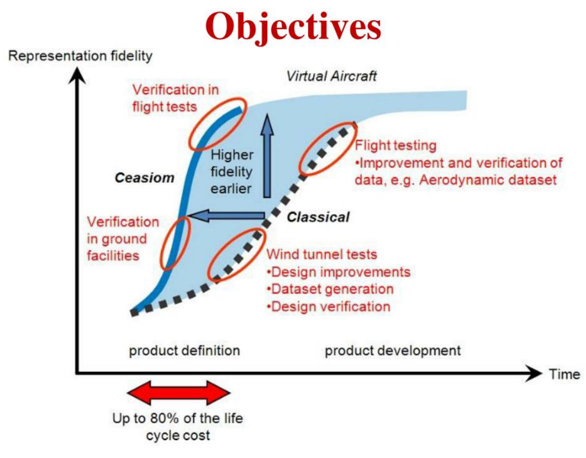


Figure 1 – CEASIOMpy contribution to aircraft design process.

By simplifying the workflow, CEASIOMpy enables faster and more efficient design iterations, especially in the conceptual and preliminary stages of aircraft design.

2.2 Key Features and Tools

One of the outstanding features of CEASIOMpy is its ability to deliver fast results when dealing with complex models, especially in scenarios where fast turnaround times are essential. This capability is particularly evident in the Mesh Generation module, the medium-fidelity CFD workflow, and the AeroFrame module for fluid-structure interaction (FSI).

The Mesh module is under continuous development with a focus on improving the robustness of hybrid mesh generation to achieve reliable and fast results. The primary tool used for mesh generation is *gmsh* [6], an open source and well maintained software widely recognized in the field. Integrated into the CEASIOMpy environment, *gmsh* allows users to generate high quality meshes suitable for Eulerian simulations, as well as hybrid meshes for RANS simulations, which are currently under development.

For the CFD workflow, the tools integrated into CEASIOMpy include AVL [9], the Vortex Lattice Method (VLM) solver developed at MIT, which allows users to automatically run low-fidelity simulations using a CPACS file. For medium and high fidelity simulations, CEASIOMpy uses *SU2* [7], developed at Stanford University. Thanks to its integration into CEASIOMpy, meshes can be generated by *gmsh* or by SUMO [8], another meshing tool included in the CEASIOMpy environment.
The AeroFrame module, the latest addition to CEASIOMpy, integrates AVL (the VLM) with a BEAM structural model, framAT [10], to produce aerodynamic results for fluid-structure interaction within minutes. This feature makes it ideal for early design iterations where rapid results are essential to evaluate a wide range of configurations without the computational burden of high-fidelity simulations.

CEASIOMpy's capabilities go beyond fast results generation and flexibility; it also excels at efficiently creating aerodynamic databases. This functionality is particularly useful for rapid prototyping, where the need to quickly explore numerous design configurations facilitates efficient decision-making during the conceptual design phase.

3. Results

3.1 COLOSSUS Project

CEASIOMpy plays an important role in the EU-funded Horizon Europe COLOSSUS project [3]. The COLOSSUS project is developing a System of Systems design framework and methodology which for the first time will enable the combined optimization of aircraft, fleet, operations and business models, see Fig. 2.

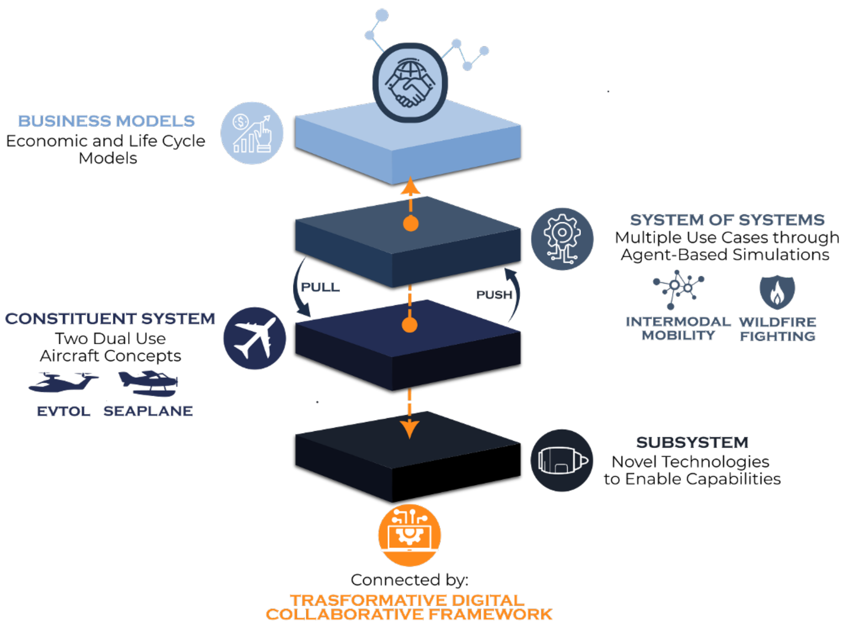


Figure 2 – Multi-level approach for the holistic design of aviation systems.

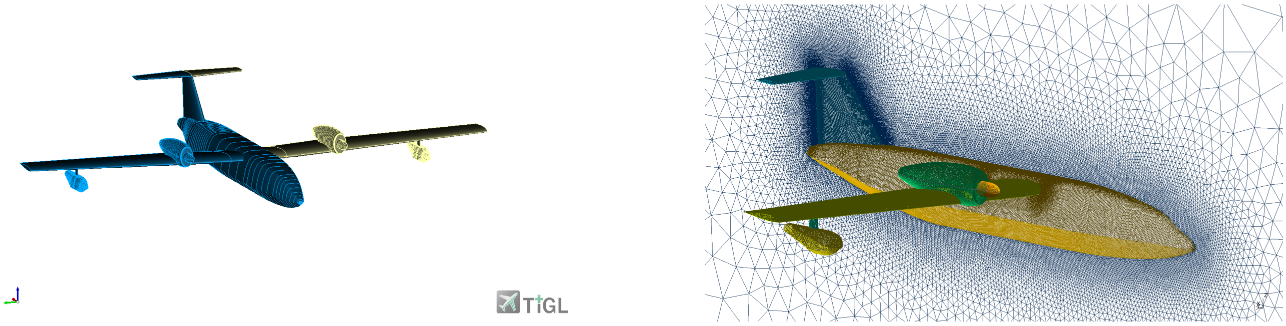
Two use cases are considered in the COLOSSUS project, an eVTOL-based advanced air mobility of passengers and goods, and a multi-role seaplane with hybrid propulsion that can be used for fire-fighting and passenger transport. In the COLOSSUS project CEASIOMpy is used for the design of the multi-role seaplane, and CFD analysis were made on different configurations. The flexibility of its Python-based environment enabled rapid modifications to aircraft geometry and re-analysis, making it an ideal tool for the iterative design processes required in conceptual studies.

3.1.1 Workflow set up

To generate an aerodynamic database, the following steps are required:

1. Import the CPACS file, which contains the geometry, into CEASIOMpy.
2. Select the appropriate workflow for the analysis.
3. Generate the mesh using the chosen meshing tool.
4. Set the calculation conditions (Mach number, angle of attack, side slip angle and altitude).
5. Execute the calculations.

In this case, the seaplane geometry was imported, and a medium-fidelity workflow, utilizing the modules *gmsh* (for mesh generation) and *SU2* (for CFD simulations), was selected.



(a) Seaplane CPACS file.

(b) Euler mesh of the Seaplane.

Figure 3 – Seaplane CPACS file and Euler mesh.

3.1.2 Results

A comparison of Euler simulations for the baseline seaplane configuration was performed with two workflows: one using CEASIOMpy, which employs *gmsh* for mesh generation and *SU2* for inviscid simulations, and another using ICEMCFD to generate a multi-block structured mesh alongside the corporate solver Navier-Stokes Multi Block (NSMB).
As shown in Table 1, there is a significant difference in the time required to complete the different workflows.

Simulation Type	CEASIOMpy (Time)	ICEMCFD + NSMB (Time)
Mesh generation	∼seconds	∼hours
CFD Analysis	∼hours	∼hours
Design iteration	<hour	∼days

Table 1 – Comparison of simulation times between the CEASIOMpy and ICEMCFD + NSMB workflows.

To obtain useful information for seaplane design, simulations were carried out at Mach 0.2 and an altitude of 5000 m. As an inviscid method was used, which cannot capture flow separation, the angle of attack was varied between 0° and 12°.

Figures 4 and 5 show the Mach number and pressure contours, respectively, at 0° angle of attack.

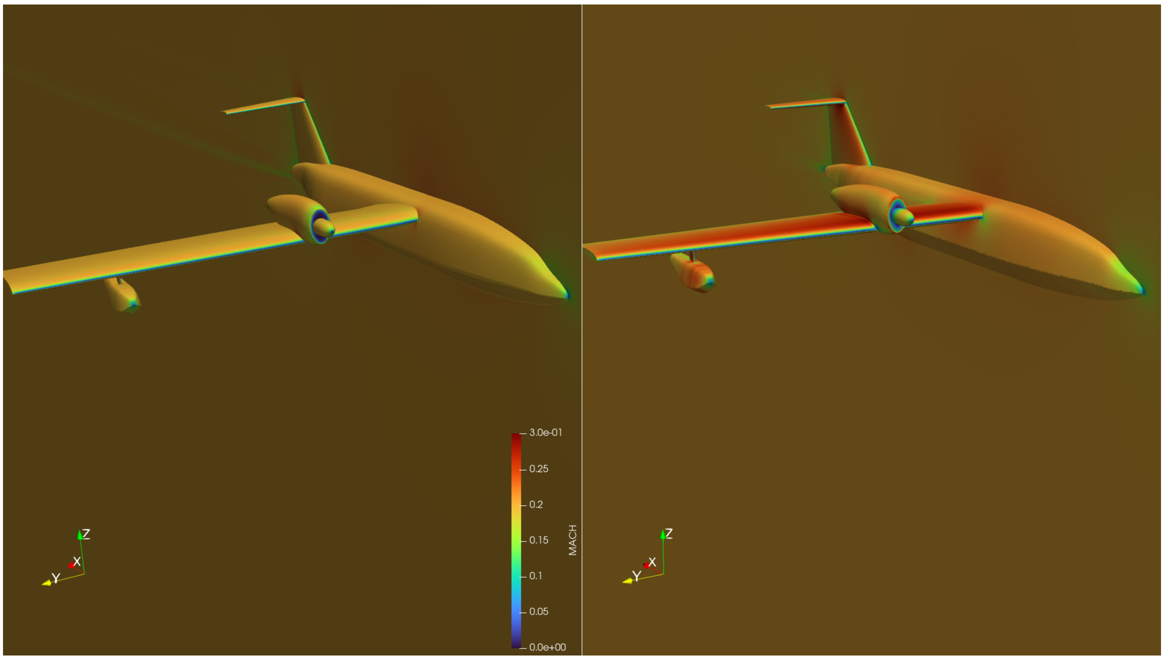


Figure 4 – Mach comparison between CEASIOMpy simulation (on the left) and NSMB at an angle of attack of 0°.

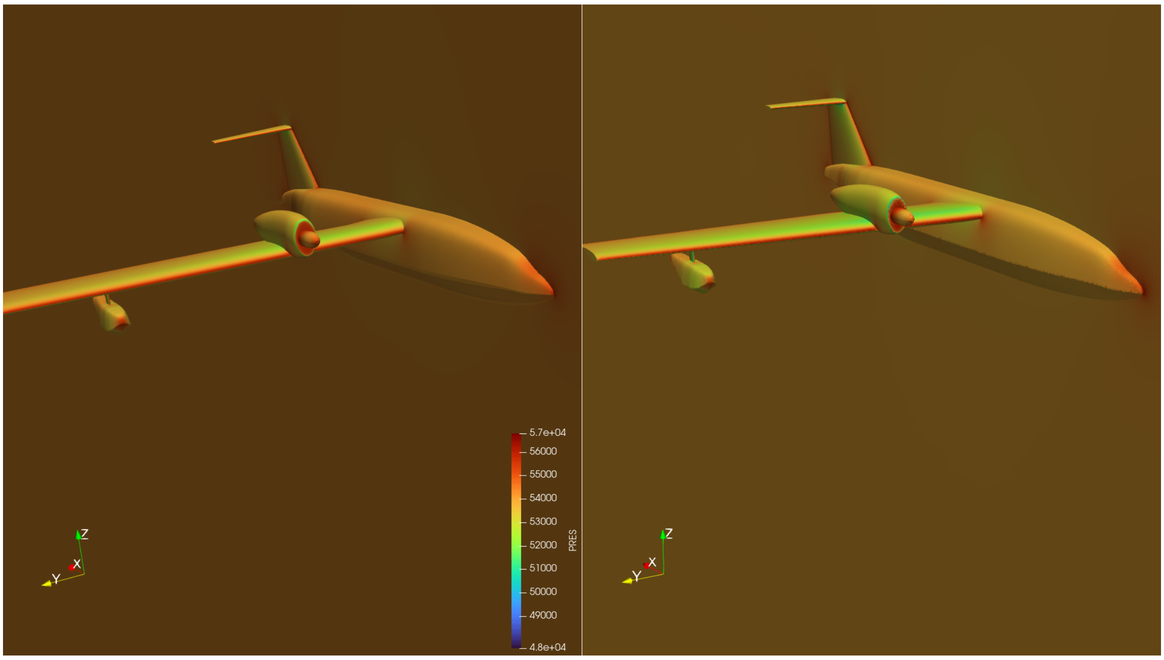


Figure 5 – Pressure comparison between CEASIOMpy simulation (on the left) and NSMB at an angle of attack of 0°.

To gain insight into the results at other angles of attack, Figure 6 shows that the two workflows produce very similar results. In particular, for smaller angles of attack, the results are almost identical.

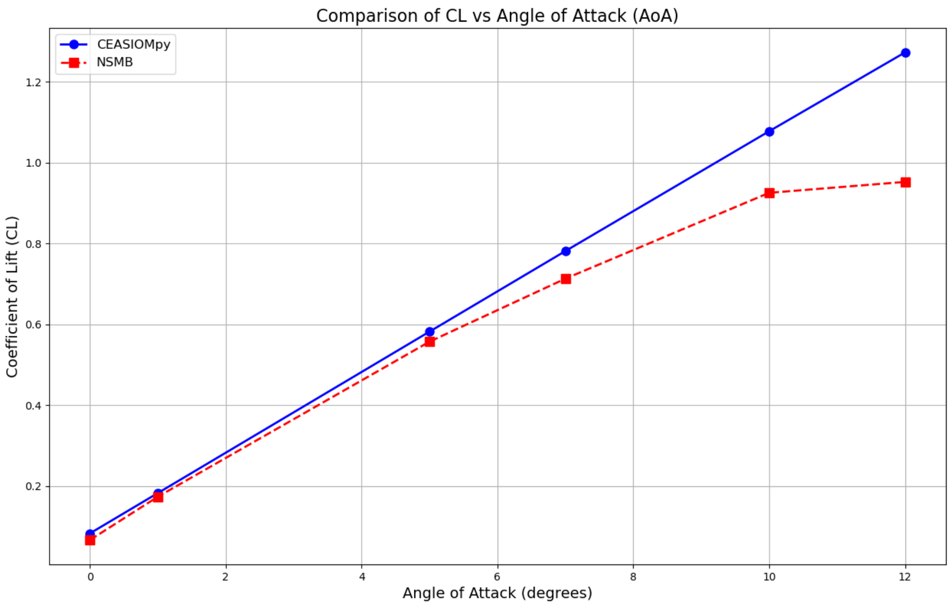


Figure 6 – C_L vs AoA CEASIOMpy and NSMB results comparison

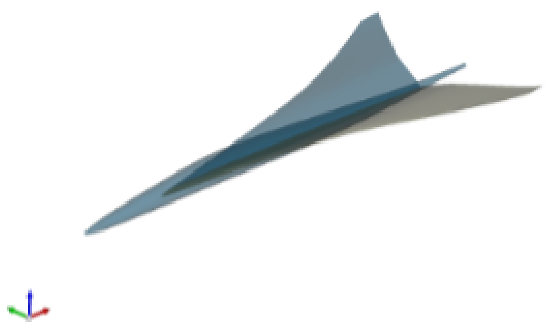
Despite the significant differences in the time required to complete each workflow, in particular for the mesh generation, the results are very satisfying.

3.2 Application of CEASIOMpy for the LARM Model

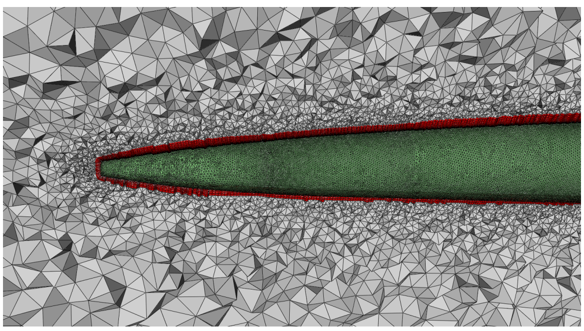
The Low Boom Aircraft Common Research Model (LARM) was developed by the Chinese Aeronautics Establishment (CAE). In a recent collaboration involving CAE, Airinnova, and CFS Engineering, CEASIOMpy was employed to analyze the model, proving particularly useful for quickly generating and managing a comprehensive workflow analysis.

The first step in the workflow involved generating the CPACS file, as shown in Figure 7a. Once this was accomplished, the mesh was generated (Figure 7b).

For mesh generation, *gmsh* was utilized to create the surface mesh. Subsequently, *pentagrow*[8] was employed to extrude the boundary layer from the surface mesh, and finally, *tetgen*[11] was used to generate the volume mesh between the boundary layer (BL) and the spherical farfield. Producing a high-quality hybrid mesh was crucial for enabling accurate Reynolds-Averaged Navier-Stokes (RANS) simulations within CEASIOMpy.



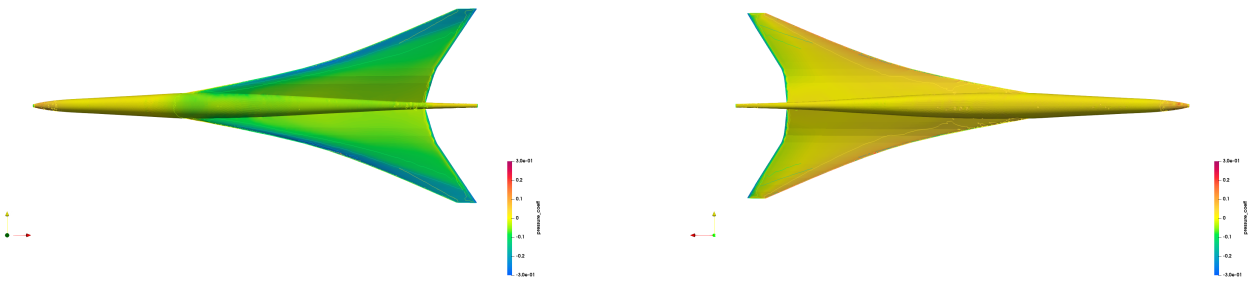
(a) LARM CPACS file.



(b) LARM hybrid mesh.

Figure 7 – LARM model setup.

Several simulations were conducted on this model, yielding promising results.
The pressure coefficient contours (Figure 8) provide a clear visualization of the pressure distribution over the surface of the LARM model. Areas of higher pressure can be identified near the leading edges and regions where shock formation occurs, indicating abrupt deceleration of the airflow.

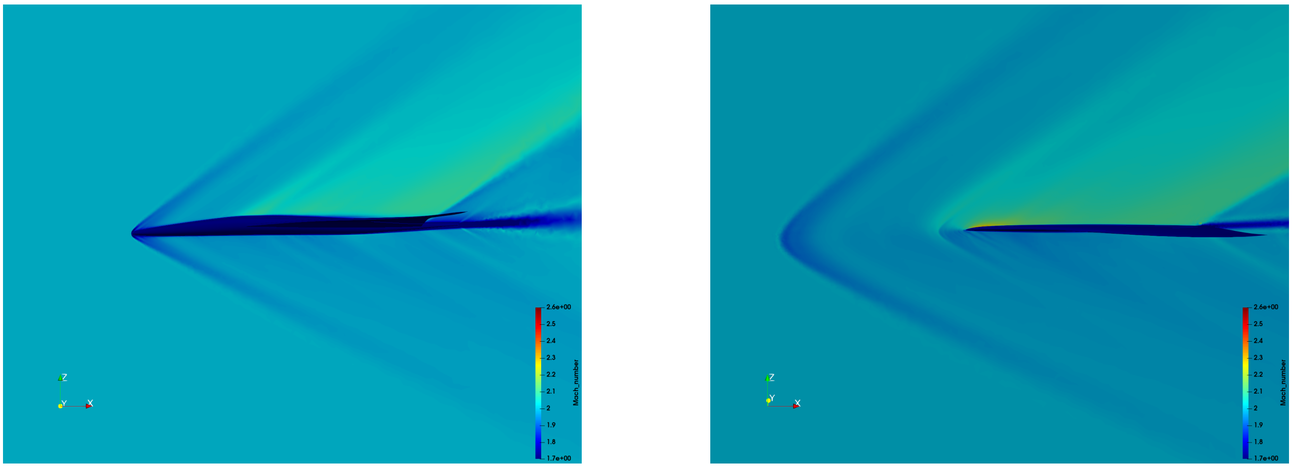


(a) LARM Mach=2, C_p contours on upper body.

(b) LARM C_p contours on lower body.

Figure 8 – LARM C_p contours on the body.

From the Mach number contours in Figure 9, the presence of oblique shocks due to the supersonic flow can be clearly seen.



(a) LARM Mach number contours in the symmetry plane.

(b) LARM Mach number contours in the plane y=3 m.

Figure 9 – LARM Mach number contours at different planes.

Figure 10 shows the pressure variation obtained from our Chinese partners during wind tunnel tests. The graph illustrates the characteristic pressure rise caused by the Mach cone generated in supersonic flow.

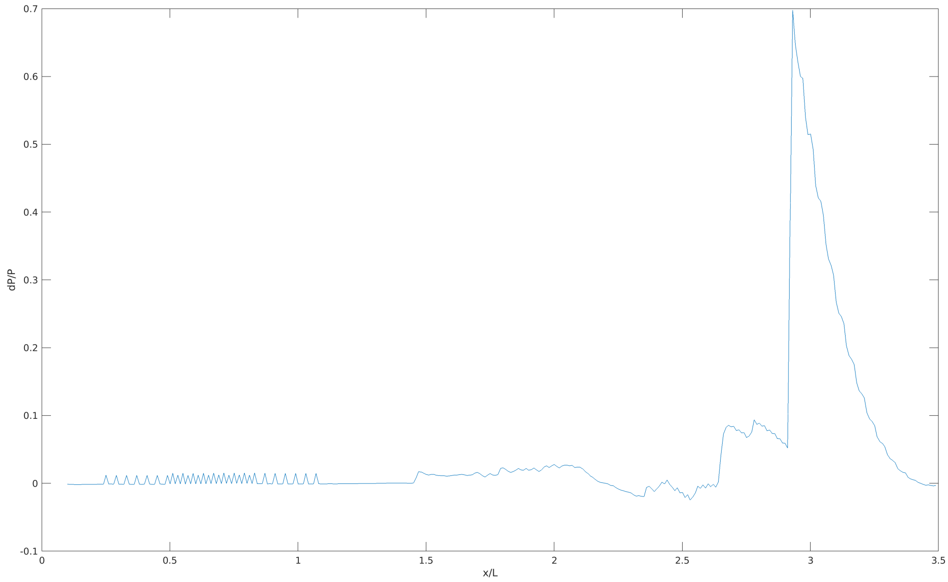


Figure 10 – The first column, X/L, represents the ratio of the axial station position to the characteristic length of the model, while the second column, $\Delta p/P$, corresponds to the ratio of the overpressure at the axial station to the free stream static pressure.

The sharp increase in pressure at certain axial locations corresponds to the regions where oblique shocks are observed in the Mach contour, providing a direct connection between the pressure variation and the shock wave phenomena.

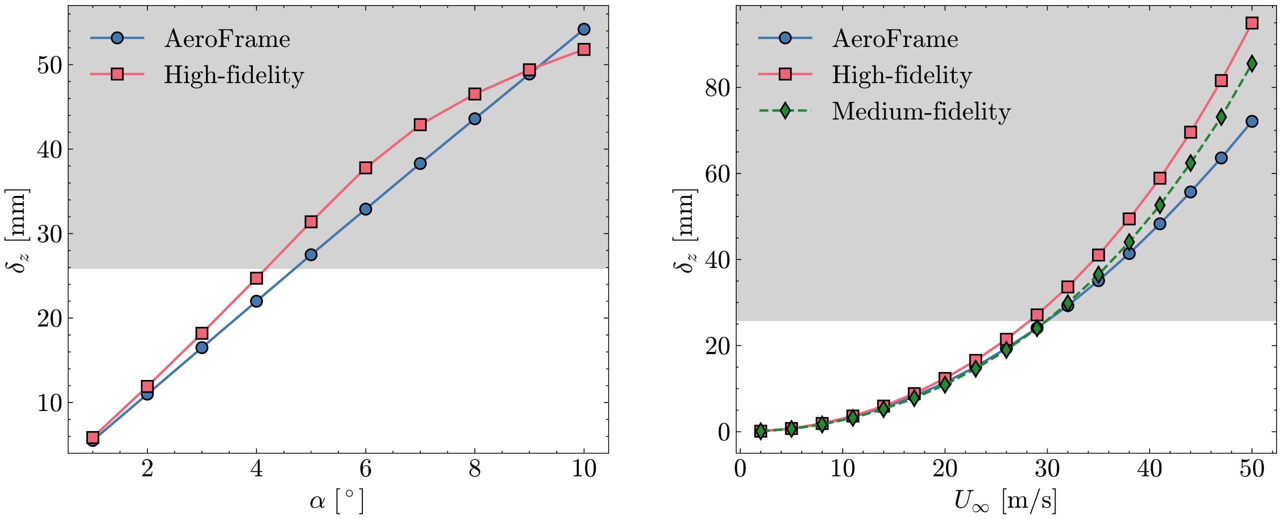
3.3 Aeroelastic Framework (AeroFrame)

AeroFrame is one of the last features introduced in CEASIOMpy, enabling aeroelastic computations. The module combines a finite element method (FEM) implementation of linear beam equations for structural calculations, with the vortex lattice method (VLM) for aerodynamic computations. AeroFrame is limited to static aeroelastic analyses. A partitioned approach is employed, using separate solvers for fluid and structural computations, respectively AVL and FramAT. The module maps the aerodynamic forces and the structural displacements between the two non-matching meshes (fluid and structural meshes) using nearest neighbor interpolation.

The advantage of AeroFrame is its low computational cost which is significantly smaller than that of high-fidelity methods, requiring several hours to produce results. Tab. 2 highlights the significant reduction in computation time for low-fidelity analyses, which can be completed in less than 30 minutes with CEASIOMpy, compared to several hours or even days with traditional high-fidelity methods.

Simulation Type	CEASIOMpy (Time)	Traditional Tools (Time)
Mesh generation	$\sim$ seconds	$\sim$ hours
FSI analysis	$\sim$minutes	$\sim$ hours
Design iteration	$<$ hour	$\sim$ days

Table 2 – Comparison of simulation times between CEASIOMpy and traditional tools.



(a) (b)

Figure 11 – AeroFrame results for tip vertical deflection δ_z of a rectangular wing, compared with medium- and high-fidelity computations (Euler/Navier-Stokes solution and nonlinear shell model). The white shaded area represents the region of validity of the linear beam model. The wing material has an elastic modulus $E = 325$ GPa and a shear modulus $G = 125$ GPa (for both beam and shell). (a) Influence of the angle of attack at free-stream velocity $U_\infty = 17\ \mathrm{m\,s^{-1}}$. (b) Influence of the free-stream velocity for an angle of attack $\alpha = 1.5°$.

AeroFrame shows consistent results for small deflections, in good agreement with literature data and medium to high-fidelity computations for both simple and complex wing geometries (Fig. 11). The white area indicates the range in which the results are considered valid as reported in the literature [12]. In particular, it highlights that nonlinear effects are negligible for deflections less than 1.62% of the half-span length, ensuring that the analysis remains in the linear regime.

The module is well-suited for preliminary aircraft design tasks, demonstrating high accuracy in small deformation scenarios and providing rough estimations when the large displacement regime is reached. Its low computational cost enables extensive analyses, facilitating iterative analyses across various material properties and wing geometries.

3.4 CEASIOMpy in Education

CEASIOMpy was not only developed for research and industrial applications but also aims to enhance the learning experience of aerospace engineering students. By providing a flexible and user-friendly tool, CEASIOMpy accelerates the educational process.

The role of CEASIOMpy in education is evident in university master's courses and educational literature. These two aspects often converge, as demonstrated in the master's level course *Computational Aerodynamic Design of Aircraft* at KTH Royal Institute of Technology. This course integrates aircraft design theory based on the textbook *Aircraft Aerodynamic Design with Computational Software* [4], where CEASIOMpy serves as one of the primary software tools employed throughout the curriculum.

The version of CEASIOMpy used in the training for this course at KTH is supported by the Swedish national research project DECADE (Digital Engineering for AirCraft Aerodynamic DEsign), part of the National Aeronautical Research Program Sweden (Nationella flygtekniska forskningsprogrammet, NFFP), funded by the Swedish Innovation Agency Vinnova (grant number 2023-01551). This funding has enabled the continued development and application of CEASIOMpy in the educational environment, providing students with access to state-of-the-art tools and methodologies.

During the course a clear workflow was set-up: start from the geometry (the CPACS file), generate a mesh and run different cases, see Figure 12. Running the different cases was done with the help of the High Performance Computing (HPC) center of KTH.

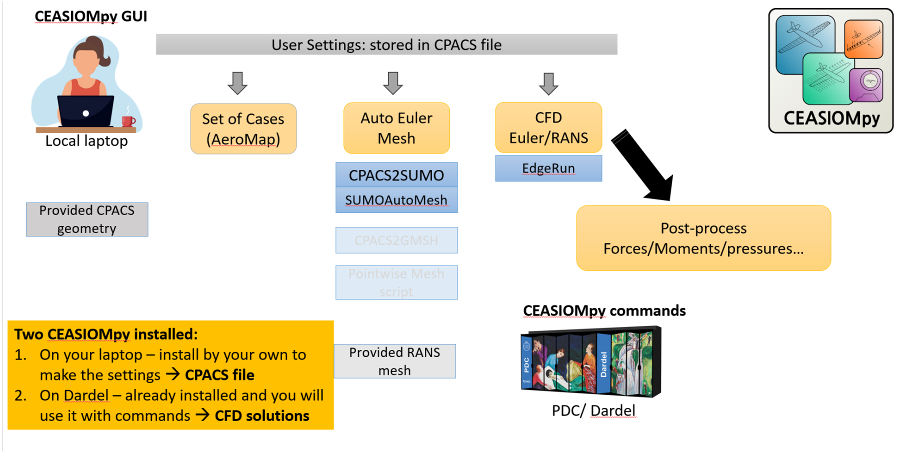


Figure 12 – Workflow employing CEASIOMpy as used in the Spring Term Aircraft design course at KTH 2024.

Throughout the course, students engage with a selection of fundamental geometries that vary in complexity, primarily concentrating on "clean" configurations that exclude intricate details such as weapon bays or control surfaces. This approach simplifies the learning process and allows for effective testing of meshing scripts and validation of tasks performed with CEASIOMpy using CPACS configurations.

The course includes mandatory computer labs where students perform CFD simulations using both CEASIOMpy and the Swedish national CFD code, M-Edge, which is pre-installed on KTH's Dardel HPC cluster. To adapt to KTH's educational infrastructure, CEASIOMpy has been adapted to support the execution of M-Edge instead of the open source tool SU2. This ensures compatibility with the tools and computing environments available to students. In addition, the CEASIOMpy modification allows students to create scripts locally on their machines that can later be executed on the HPC cluster, demonstrating the flexibility of the platform.

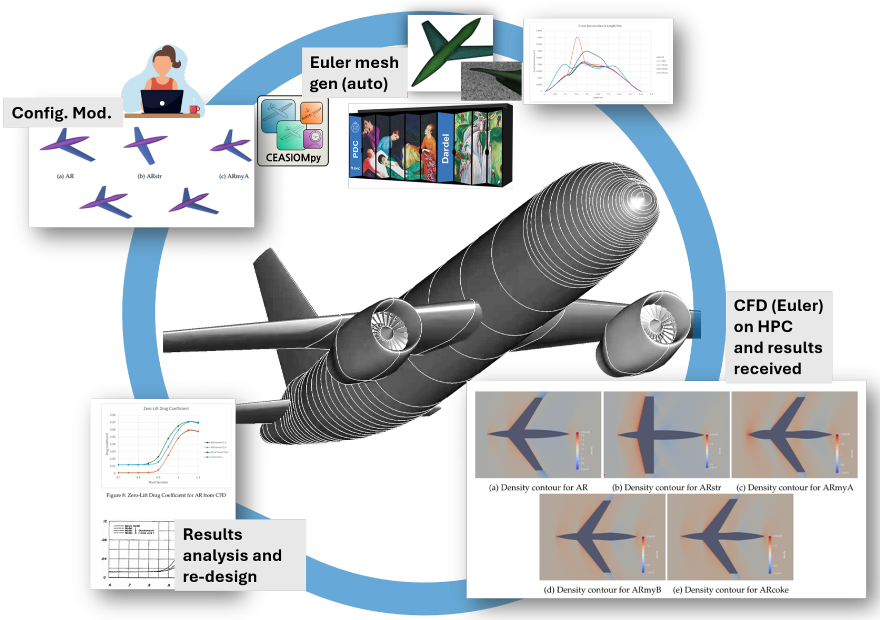


Figure 13 – CEASIOMpy workflow from geometry to aircraft aerodynamic performance, some pictures extracted from students' reports of LabAR (Lab Area-Ruling) 2024.

A key component of the course is the LabAR (Lab Area-Ruling): Design for Low Wave Drag, which focuses on designing aircraft to minimise wave drag. In this lab, students are tasked with calculating the Mach sweep of a baseline aircraft, modifying configurations and studying the effects on wave drag. They explore different designs by adjusting parameters such as wing sweep angle, wing thickness and fuselage shape based on the Sears-Haack body. The iterative process facilitated by CEASIOMpy helps students gain a deeper understanding of how these design variables affect wave drag. The automated CFD workflow provided by CEASIOMpy allows students to modify designs, run simulations on HPC resources, and efficiently visualise the results, greatly enhancing the overall learning experience.

This hybrid approach, combining local and remote computing, allows students to engage with the complete CFD workflow, providing them with a comprehensive understanding of both the software and the computing resources essential to modern aerospace engineering.

4. Conclusions

CEASIOMpy has proven to be a powerful and versatile tool for rapid aircraft design, offering significant advantages through its open source nature and modular architecture. These features make it highly adaptable to a wide range of aerospace applications, from conceptual studies to more advanced design tasks. Its successful use in collaborative projects such as AGILE, AGILE4.0, COLOSSUS and the LARM analysis has demonstrated CEASIOMpy's ability to significantly reduce the time and effort required for early design analysis.
In addition to its use in industry and research, CEASIOMpy has also shown great potential in education. By providing students with a hands-on, flexible aircraft design platform, it enhances their learning experience and helps them develop practical skills that are directly applicable to real-world engineering challenges. This educational component makes CEASIOMpy not only a tool for professionals, but also a valuable resource for educating the next generation of aerospace engineers.
As the complexity of aircraft design challenges continues to increase, the need for flexible and efficient tools becomes even more critical. The ongoing development of CEASIOMpy, driven by European projects and contributions from Master's students, ensures that the platform will continue to evolve and improve, making it a useful asset for the aerospace engineering community.

Acknowledgments

The research presented in this paper has been performed in the framework of the COLOSSUS project (Collaborative System of Systems Exploration of Aviation Products, Services and Business Models). Funded by the European Union under Grant Agreement no 101097120. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the european union or cinea. Neither the European union nor the granting authority can be held responsible for them.

The Swiss participation in the Colossus project was supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 22.00609.

References

[1] M. Alder, E. Moerland, J. Jepsen and B. Nagel. *Recent Advances in Establishing a Common Language for Aircraft Design with CPACS*. Aerospace Europe Conference 2020, Bordeaux, France, 2020.

[2] R. Rizzi and J.B. Vos. *Modeling and simulating aircraft stability and control - The SimSAC project*, Special Issue of Progress in Aerospace Sciences, Vol. 47, 2011.

[3] P.S. Prakasha, N. Naeem, K. Amadori, J. Akbari, F. Nicolosi, L.K. Franzen, M. Ruocco, B. Nagel,*COLOSSUS EU Project - Collaborative System of Systems Exploration of Aviation Products, Services and Business Models: Overview and Approach*, ICAS Conference, Florence, 2024.

[4] A. Rizzi and J. Oppelstrup, *Aircraft Aerodynamic Design with Computational Software* Cambridge: Cambridge University Press, 2021.

[5] A. Dettmann, Loosely coupled, modular framework for linear static aeroelastic analyses. 2019.

[6] "Gmsh", Christophe Geuzaine and Jean-François Remacle, 2009, `http://gmsh.info/`

[7] SU2, `https://su2code.github.io/`

[8] Surface Modeler SUMO, `https://www.larosterna.com/oss/`

[9] VLM solver AVL, `https://web.mit.edu/drela/Public/web/avl/`

[10] “airinnova/framat,” May 2024. original-date: 2019-08-13T13:15:55Z.

[11] Hang Si. 2015. "TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator". ACM Trans. on Mathematical Software. 41 (2), Article 11 (February 2015), 36 pages. DOI=10.1145/2629697

[12] Dang Huixue, Yang Zhichun, Li Yi "Accelerated loosely-coupled CFD/CSD method for nonlinear static aeroelasticity analysis" `https://doi.org/10.1016/j.ast.2010.01.004`