

# Benchmarking New CEASIOM with CPACS adoption for aerodynamic analysis and flight simulation

*Aidan Jungo*

CFS Engineering, Lausanne, Switzerland

*Mengmeng Zhang*

Airinnova AB, Stockholm, Sweden

*Jan B. Vos*

CFS Engineering, Lausanne, Switzerland, and

*Arthur Rizzi*

Department of Aeronautical and Vehicle Engineering, Royal Institute of Technology (KTH), Stockholm, Sweden and Airinnova AB, Stockholm, Sweden

## Abstract

**Purpose** – The purpose of this paper is to present the status of the on-going development of the new computerized environment for aircraft synthesis and integrated optimization methods (CEASIOM) and to compare results of different aerodynamic tools. The concurrent design of aircraft is an extremely interdisciplinary activity incorporating simultaneous consideration of complex, tightly coupled systems, functions and requirements. The design task is to achieve an optimal integration of all components into an efficient, robust and reliable aircraft with high performance that can be manufactured with low technical and financial risks, and has an affordable life-cycle cost.

**Design/methodology/approach** – CEASIOM ([www.ceasiom.com](http://www.ceasiom.com)) is a framework that integrates discipline-specific tools like computer-aided design, mesh generation, computational fluid dynamics (CFD), stability and control analysis and structural analysis, all for the purpose of aircraft conceptual design.

**Findings** – A new CEASIOM version is under development within EU Project AGILE ([www.agile-project.eu](http://www.agile-project.eu)), by adopting the CPACS XML data-format for representation of all design data pertaining to the aircraft under development.

**Research limitations/implications** – Results obtained from different methods have been compared and analyzed. Some differences have been observed; however, they are mainly due to the different physical modelizations that are used by each of these methods.

**Originality/value** – This paper summarizes the current status of the development of the new CEASIOM software, in particular for the following modules: CPACS file visualizer and editor CPACSupdater (Matlab) Automatic unstructured (Euler) & hybrid (RANS) mesh generation by sumo Multi-fidelity CFD solvers: Digital Datcom (Empirical), Tornado (VLM), Edge-Euler & SU2-Euler, Edge-RANS & SU2-RANS Data fusion tool: aerodynamic coefficients fusion from variable fidelity CFD tools above to compile complete aero-table for flight analysis and simulation.

**Keywords** Aircraft design, Data fusion, Aerodynamic coefficients, Multi-fidelity

**Paper type** Research paper

## 1. Introduction & background

Designing an aircraft is a very complex engineering task. The complexity needs to be handled by decomposition, using a hierarchy of different levels. In general, the aircraft design process is divided into three phases: the conceptual design phase, the preliminary design phase and the detailed design phase. The product fidelity, the model complexity and the time needed for the design process increase exponentially from the previous phase to the next one.

In the conceptual design phase, the aircraft is defined at a system level. Many variants are studied, and the design selected is the one that best fulfills the mission to the specifications of the market or a customer. This determines a general aircraft configuration capable of performing its mission, together with first sizing estimates. In the preliminary design phase, the central

---

The current issue and full text archive of this journal is available on Emerald Insight at: [www.emeraldinsight.com/1748-8842.htm](http://www.emeraldinsight.com/1748-8842.htm)



Aircraft Engineering and Aerospace Technology  
90/4 (2018) 613–626  
Emerald Publishing Limited [ISSN 1748-8842]  
[DOI [10.1108/AEAT-11-2016-0204](https://doi.org/10.1108/AEAT-11-2016-0204)]

---

© Aidan Jungo, Mengmeng Zhang, Jan B. Vos and Arthur Rizzi. Published by Emerald Publishing Limited. This article is published under the Creative Commons Attribution (CC BY 4.0) licence. Anyone may reproduce, distribute, translate and create derivative works of this article (for both commercial and non-commercial purposes), subject to full attribution to the original publication and authors. The full terms of this licence may be seen at <http://creativecommons.org/licenses/by/4.0/legalcode>

Received 17 November 2016  
Revised 31 January 2017  
24 March 2017  
Accepted 24 March 2017

challenge is to perform design optimization in a distributed environment made up of distinct disciplinary design teams with individual solution strategies, locally defined variables and constraints, potentially costly computational analyses and interdisciplinary coupling. A variety of *multidisciplinary design optimization* (MDO) methods have been developed that enable a formalized design optimization process in the preliminary design phase. **Figure 1** indicates the major design process of the conceptual and early preliminary design work leading up to freezing the configuration, starting wind tunnel testing and then continuing to advance the preliminary design.

The methods used differ from each other in how they handle local feasibility, interdisciplinary compatibility and local design autonomy. One such method is *collaborative optimization*. The detailed design phase includes the manufacturing details, the detailed definitions of the product and performance data as well as all other required product information. The earliest design process is a crucial stage, which commits up to 80 per cent of the life-cycle cost. But the actual cost is incurred much later (Zhang, 2015), as many of the decisions taken in this phase are made with a great deal of uncertainty. Improvements in this design stage offer the greatest scope for innovation, and for this reason we focus on bringing more fidelity into the early design steps to reduce the uncertainty. In the SciTech 2015 Conference in Kissimmee, the keynote speech on January 5, 2015 with title “Technology Roadmaps Pave the Way to Our Future” spelled out that “the potential of making more use of computational fluid dynamics (CFD) in the conceptual design [...]” could be especially useful for unconventional and innovative configurations.

In this paper, we will focus on the early design stages, i.e. the conceptual design as well as the preliminary design, and explore ways to improve the prediction fidelity in these stages.

### 1.1 Data-centric scheme CPACS and workflow manager RCE

As discussed earlier in the text, aircraft design requires several different design groups (each having their own focus) that need

to exchange large amounts of data obtained from their analysis procedures and models. Managing the interconnections is complex and error-prone.

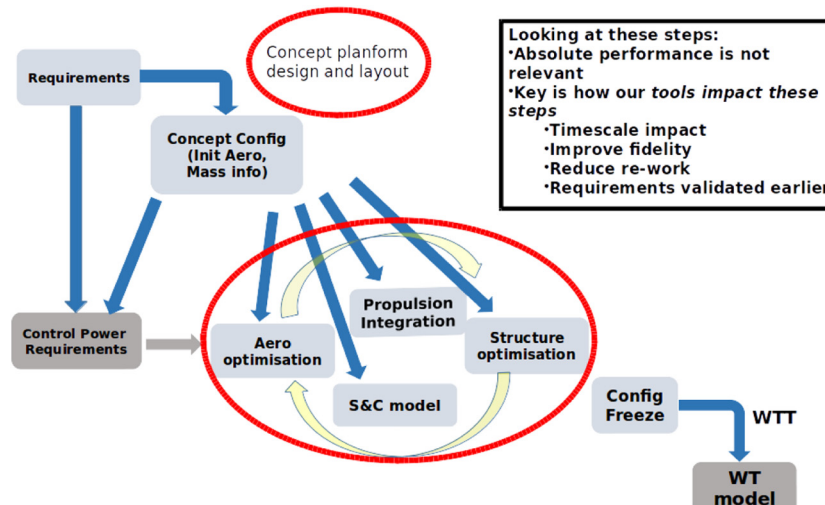
Adoption of a standardized, data-centric scheme for storage of all data improves consistency and reduces risks of misconceptions and errors in the process. It however requires an initial effort to make interfaces between analysis modules and the data archive. The Common Parametric Aircraft Configuration Schema (CPACS) (CPACS – A Common Language for Aircraft Design, 2015; CPACS Documentation, 2015), developed by Deutschen Zentrums für Luft- und Raumfahrt (DLR), was adopted for the New CEASIOM framework, and is described in the following section. A design loop runs several analysis modules in sequence. The remote component environment (RCE) integration environment and workflow manager records the sequence and manages the data transport and translation as well as logging the process. RCE, developed by DLR, makes it easy to set up and run a workflow also using modules in which the engineers are not discipline-experts.

### 1.2 Aerodynamic table for flight simulation

For the stability and control analysis as well as for the flight simulation, a large look-up table for aerodynamic forces and moments needs to be generated. There are different table/input formats required by different flight analysis tools. For example, the simulation and dynamic stability analyzer (SDSA) (Goetzendorf-Grabowski *et al.*, 2011) developed by Warsaw University of Technology requires a set of three-dimensional tables of force and moment coefficients with the standard three-channel control systems. Details of the table format and its applications can be found in Zhang (2015).

In the results discussed in this paper, all the aero-data are saved in CPACS. **Table I** shows the aerodata format defined in CPACS XML file. The force and moment coefficients are recorded in body-axes as  $c_f$  and  $c_m$ , aligning with x-, y- or z directions. The tables are four-dimensional with independent variables machNumber ( $Ma$ ), reynoldsNumber ( $Re$ ), angleOfYaw ( $\beta$ ) and angleOfAttack ( $\alpha$ ). The dependence on

**Figure 1** Major design process in the conceptual and early preliminary stage; conceptual stage yields the overall aircraft layout design and preliminary proceeds with the component design



**Table 1** Format of aerodynamic table aeroPerformanceMap defined in CPACS

$\alpha$	$\beta$	Re	Ma	cfx	cfy	cfz	cmx	cmz	cmz
x	x	x	x	x	x	x	x	x	x

rotation rates and control surface deflections are represented by dynamic and control derivatives, also recorded in four-dimensional tables. Detailed definitions can be found in [CPACS Documentation \(2015\)](#).

## 2. CEASIOM history and current status

The CEASIOM software framework for conceptual-preliminary design was created in the SimSAC – Simulating Aircraft Stability and Control Characteristics for Use in Conceptual Design – EU Framework 6 project. Its mission was to develop an integrated simulation environment to compute stability and control information with a quantifiable uncertainty. The resulting CEASIOM-100 tool programmed in MATLAB brought variable fidelity simulation to the conceptual-preliminary design process. CEASIOM-v4.0 is the latest version with each module up-to-date and is freely available from the CEASIOM website ([www.ceasiom.com](http://www.ceasiom.com)).

CEASIOM is a framework system that integrates discipline-specific tools such as computer-aided design (CAD), mesh generation, CFD, stability and control analysis, and structural analysis, all for the purpose of aircraft conceptual design ([Zhang, 2015](#)).

A new CEASIOM version is under development in the EU Project AGILE ([www.agile-project.eu](http://www.agile-project.eu)), by adopting the CPACS XML data-format for representation of all design data pertaining to the aircraft under development. All the CEASIOM modules will be integrated in the RCE framework because it provides special extensions suitable for optimization.

With all the analysis modules integrated in RCE, a complete workflow from can be set up starting from the CPACS aircraft configuration (obtained from e.g. overall design stage) until stability and control assessment and flight performance. During this process, data from the variable-fidelity aerodynamic analysis tools are created, compiled and fused into a coherent aero-data set. In this paper, we will present the new CEASIOM functionalities by running such an exercise using the DC1 aircraft from the AGILE project (see also further below) ([Figure 2](#)).

### 2.1 Geometry – CPACSupdater

The aircraft geometry for the “old” CEASIOM was defined as a CEASIOM type XML file, which came from QCARD system built by [Isikveren \(2002\)](#) in his PhD dissertation. The New CEASIOM adopts the CPACS format, and all the aircraft geometries are defined in CPACS. The CPACS aircraft definition will be handled in module CPACSupdater within New CEASIOM.

It is extremely difficult to correctly parametrize an aircraft defined in CPACS XML file without visual feedback. The CPACSupdater, based on the old CPACScreator, is a Matlab program which works as a robust XML visual editor by parsing and transforming the XML file into the Matlab structure. The aircraft can be viewed, modified and updated smoothly, not

only its external shapes and components but also its sub-components such as flaps and leading/trailing edge devices (TED) and its internal structures such as ribs, spars and fuel tanks. The updated aircraft definition is saved as a CPACS file for further use, ready to be delivered to other modules inside CEASIOM or to other external tools that are able to read the CPACS file format.

The CPACSupdater is a continuous work based on CPACScreator ([Zhang, 2013](#)), with algorithmic corrections and bugs-fixing. It is a pure geometry definition updater that computes the necessary geometric parameters calculated, such as the mass distributions and weight and balance information (under-development).

[Figure 3](#) shows the main Graphic User Interface (GUI) of CPACSupdater that can be used to view, add, remove or modify a CPACS aircraft geometry definition. It contains two parts, the left part is the User Control Panel consisting all *components*, *subcomponents*, their parameters and user actions. The right part is the instant graphic feedback which responds to the user actions made in the User Control Panel.

Currently, three sub-components are supported by CPACSupdater: TED; a set of spars and ribs (defined as one sub-component); and fuel tanks. The sub-components transformations and symmetry parameters are defined by their parent component and shared with all its sub-components. [Figure 4\(a\)](#) and [\(b\)](#) shows the GUI to modify the TED and sets of ribs and spars, respectively.

### 2.2 Mesh generator – sumo

Sumo is a graphic tool aimed at rapid creation of aircraft geometries and automatic surface mesh generation ([Tomac and Eller, 2011](#)). It is not a full-fledged CAD system, but rather an easy-to-use sketchpad, highly specialized towards aircraft configurations. It is actively developed to streamline the workflow as much as possible to the intended use: rapid surface modeling of aircraft configurations; automatically unstructured surface meshes generation without user intervention. The unstructured volume meshes can be generated from the surface mesh, using the tetrahedral mesh generator TetGen ([Si, 2013](#)). The following features are included in the new CEASIOM version:

- Automatic CPACS-sumo Python converter to read/convert CPACS format geometry into sumo native .smx file. CPACS very flexible user-defined node feature (cpacs.toolspecific) allows the related manual-tuning parameters (if not default) for mesh generation to be saved and imported by sumo in CPACS format.
- Automatic Euler (tetrahedron) and RANS (pentahedron) meshes generation. To generate the RANS mesh, pentagrow ([Tomac, 2014](#)) needs to be run before TetGen in batch mode by executing a configuration file with a list of user-defined parameters to set up the prismatic layers such as the first cell height, the total number of layers and the growth rate. A short user guide and a template for the configuration file with the recommendation setting of parameters are provided within CEASIOM package.
- The output file (the volume mesh) can be of variable formats, including CGNS, a bmesh file as CFD solver Edge ([Eliasson, 2002](#)) naive format, a su2 file as CFD solver SU2 format or TetGen’s plain ASCII format.

Figure 2 New CEASIOM workflow

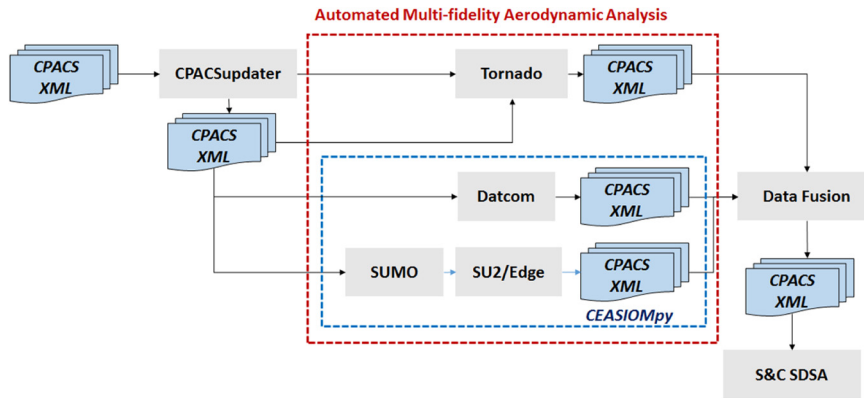


Figure 3 CPACSupdater GUI to view, modify and update a CPACS aircraft geometry definition

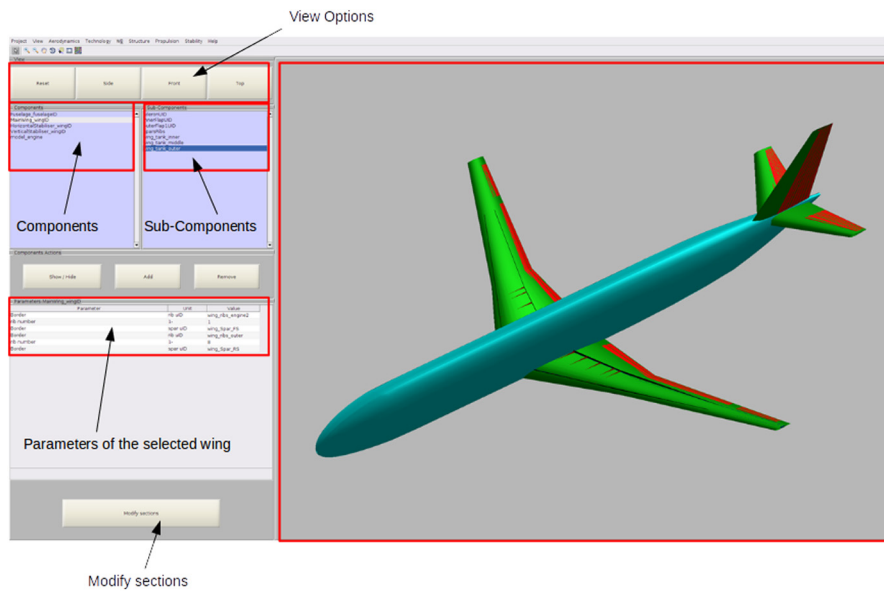
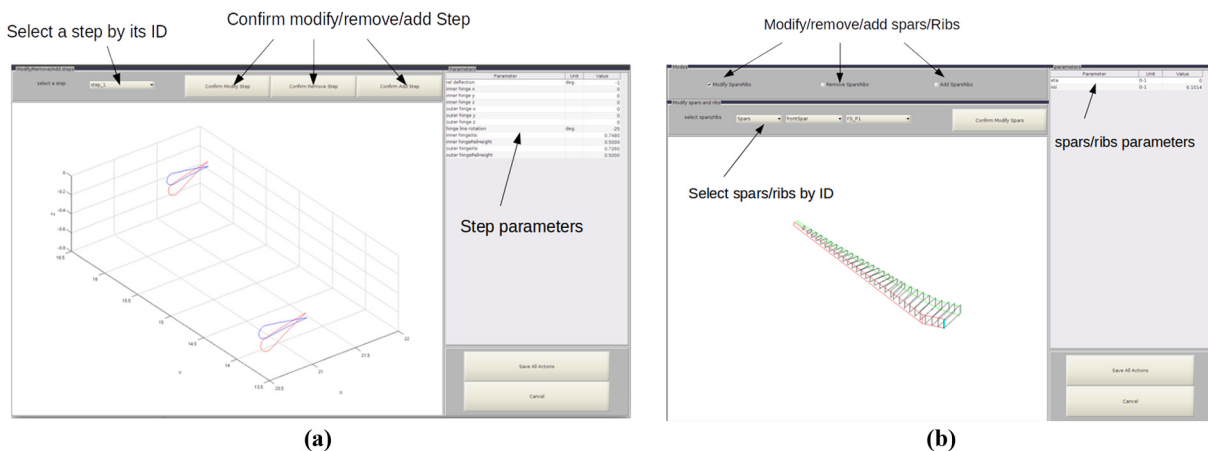


Figure 4 CPACSupdater GUI to modify the sub-components of a CPACS-defined aircraft



Notes: (a) To modify the TED hinge line and rotation steps; (b) to add, remove or modify the spars and ribs, with ribs selected

Figure 5 shows the geometry and RANS mesh of X-31 model made by sumo.

### 2.3 Aerodynamics

In old CEASIOM version, the aerodynamic module, called AMB (Aerodynamic Model Builder), has several solvers (using different fidelities) for the generation of the tables of aerodynamic forces and moments required for flight dynamics analysis. These functionalities have been kept and are continued in the new CEASIOM version. A new option is to use sampling and data fusion to generate the tables, and this will be further discussed in Section 2.5. In the new CEASIOM version, AMB is atomized and each aerodynamic solver can use the CPACS file directly, and the aero-data from each solver is saved individually in the CPACS format. This provides more flexibility because one can either use CEASIOM as a whole, or just use only a part of it.

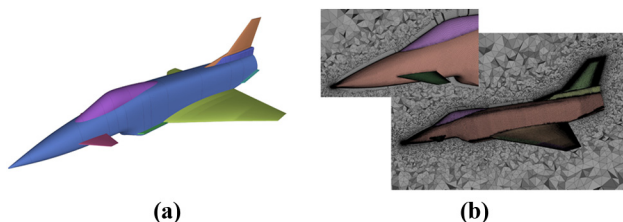
#### 2.3.1 Automated multi-fidelity aerodynamic analysis

The aerodynamic module in the new CEASIOM version, called automated multi-fidelity aerodynamic analysis (automated MAA), has the capacity to analyze aircraft (in CPACS format) with at least four levels (L)[1] of aerodynamic analysis tools:

- L0: empirical based (aerodynamic) tools, as typical in state-of-the-art textbook methods. In CEASIOM, the L0 aerodynamic tool is handbook method tool Datcom from USAF [(PDAS), P. D. A. S., 2013].
- L1: physical based analysis tools, based on a *simplified* representation of the physics phenomena, suitable for the design of a conventional aircraft. In CEASIOM, the L1 aerodynamic tool is the vortex-lattice method tool Tornado (Melin, 2003).
- L2: physical based analysis tools, based on a more detailed representation of the physics phenomena, suitable for the design of an un-conventional or a novel aircraft. In New CEASIOM, the L2 aerodynamic tools are the Euler-equation solvers SU2 (Palacios *et al.*, 2013, 2015) and/or Edge (Eliasson, 2002).
- L3: physical based analysis tools, similar as L2 tools but based on the most detailed representation of the physics phenomena. In the new CEASIOM version, the L3 aerodynamic tools are the Navier–Stokes equation solvers SU2 (Palacios *et al.*, 2013, 2015) and/or Edge (Eliasson, 2002).

The workflow for the automated-MAA model in CEASIOM is shown in Figure 6, which is a more detailed representation of

Figure 5 Representative geometry and mesh of the X-31 [ref] model in sumo



Notes: (a) Sumo geometry; (b) sumo-penta RANS mesh

the *Automated Multi-fidelity Aerodynamic Analysis* in red box shown in Figure 2. The I/O for all the tools are CPACS XML files, and a large portion of the automated MAA module is written in [copyright] Python and dubbed as CEASIOM<sub>py</sub>.

#### 2.3.2 Empirical methods digital Datcom

Digital Datcom (Stability and Control Data Compendium) [(PDAS), P. D. A. S., 2013] is based on an empirical database calculated for a lot of different wing geometries. This database has been created by the US Air Force. The advantage of this method is the short computation time (a few second to create an entire aerodynamic database). DATCOM can only give aerodynamic coefficients for conventional aircraft. DATCOM does not provide all the outputs for the transonic regime, but some data are available from experimental data or for certain configurations. The range of utilization is limited to a maximum Mach number of 0.6. DATCOM is integrated into CEASIOM through a converter which is able to create DATCOM input file directly from a CPACS file.

#### 2.3.3 Potential solver tornado

The vortex lattice method (VLM) Tornado (Melin, 2003) is a potential flow solver included in both the old and CEASIOM versions, and it provides the aerodynamic coefficients and first-order derivatives (a central difference calculation) over lifting surfaces at low speeds. The lifting surfaces are modeled at the camber lines, i.e. no thickness. The modification to the horseshoe vortices, namely, the vortex slings, which has seven segments instead of three, brings flexibility to model trailing edge movable surfaces. The leading edge movable surfaces can be modeled like-wise, but it is of less interest as it only changes the maximum lift which VLM cannot predict anyway. The steady wake can be chosen fixed in the body coordinate system or flowing the free stream. Overall effects of compressibility at high Mach number (<0.75) are included through the Prandtl–Glauert correction (Anderson, 2004). The induced drag can be calculated by both the Kutta–Joukowski law (default) and Trefftz-plane integration (Katz and Plotkin, 1991).

For the modeling of the fuselage, we tried several simple body models in the current CEASIOM version with no encouraging results, including the cruciform bodies, and the slender body theory (Katz and Plotkin, 1991). It is clear that to include body effects, we must move to higher-fidelity solvers. We decide to go to Euler solvers instead, having some panel codes in between. In the new CEASIOM version, some additional features of Tornado are included:

- directly import/export CPACS XML file format, including the automatic paneling for Tornado from the CPACS XML description;
- graphic aircraft configuration visualization including fuselage representation and control surfaces identifications;
- time saving by mex-version of core-functions for matrix computations; and
- all-moving surfaces and overlapped movable surfaces.

Figure 7 shows the visualization of the configuration and panel distributions in Tornado for the DC1-MDA aircraft, which will be described and used in the following section as a test example. The I/O for Tornado is in CPACS format, and the aircraft configuration in Figure 7(a) includes the fuselage read from CPACS which is only for visualization.

Figure 6 The Automated multi-fidelity aerodynamic analysis (MAA) workflow in new CEASIOM

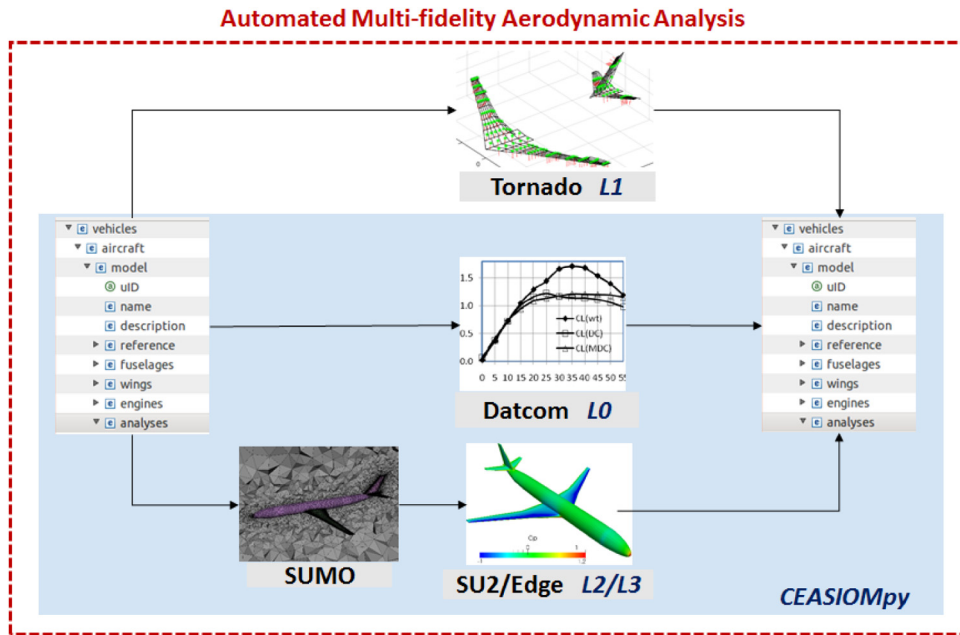
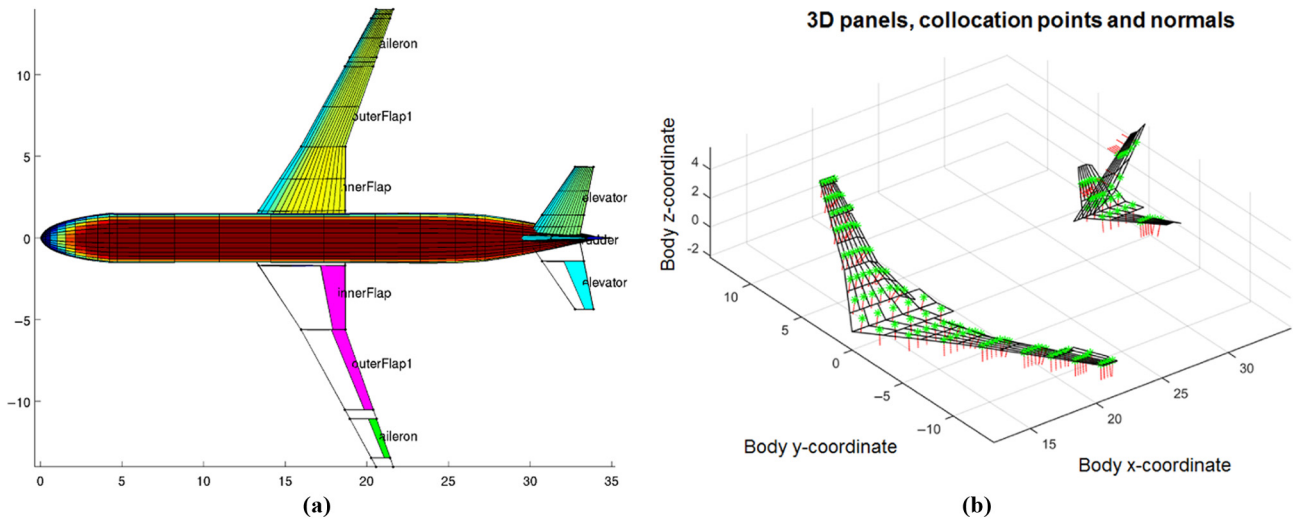


Figure 7 Geometry and panel distributions in tornado for DC1-MDA, imported from CPACS



Notes: (a) Configuration visualization; (b) panels and collocation points with control surfaces deflected

2.3.4 High-fidelity solvers: SU2 and Edge

The high-fidelity solvers SU2 (Palacios *et al.*, 2013, 2015) and Edge (Eliasson, 2002) are used to solve Euler and RANS equations. Edge is a Swedish national CFD code for solving 2D/3D viscous/inviscid compressible flow problems on unstructured grids with arbitrary elements, developed by Swedish Defense Research Agency (FOI). It can be used for both steady state and edge-based formulation which uses a node-centered finite-volume technique to solve the governing equations. A number of RANS-based turbulence models, as well as LES and DES, can be treated in Edge.

The SU2 (Palacios *et al.*, 2013) software suite from Stanford University, is an open-source, integrated analysis and design tool for solving complex, multi-disciplinary problems on unstructured computational grids. The built-in optimizer is a Sequential Least Squares Programming (SLSQP) algorithm (Griva *et al.*, 2009) from the SciPy Python scientific library. The gradient is calculated by continuous adjoint equations of the flow governing equations (Palacios *et al.*, 2013, 2015). SU2 is in continued development. Most examples pertain to not only inviscid flow but also RANS flow models with the Spalart–Allmaras and the Menter SST  $k-\omega$  turbulent models can be treated.

An important capability for the high-fidelity flow solvers (Euler-equation solvers) is the possibility to carry out the analysis for deflected control surfaces. The way Edge calculates the aerodynamics of control surface deflections is based on the use of transpiration boundary conditions. In this approach, instead of moving the grid, the wall velocity component normal to the actual deflected surface is prescribed, and this approach eliminates the need for mesh deformation. SU2 uses a mesh deformation by moving the control points of free-form deformation box; thus, all calculations can be run virtually on the clean configuration grid. But on the other hand, both methods impose a limitation on the amount of maximum and minimum deflections. An alternative approach would be the generation of a different grid for each new configuration of deflected control surfaces. This approach is not so feasible for the intended use of creating aerodynamic tables, given the number of possible combinations of control surfaces and corresponding deflection angles, especially if the details of the deflected control surfaces, such as gaps, are going to be modeled. A newly developed meshing technique for seamlessly automated generating Euler and RANS mesh based on sumo for the morphing trailing edge deflections (no gaps) is partly shown in [Zhang's \(2015\)](#) PhD thesis and is still under development and validation with more test cases. More details will be found in the coming EASN paper *Automated Meshing and Data Fusion Applied to Create Aerodataset for AGILE DC-1 and Beyond*.

#### 2.4 CEASIOMpy

CEASIOMpy is a continuous development interface that will cover the whole CEASIOM and will replace the [copyright] Matlab-coded CPACSupdater in the future. It has been written in [copyright] Python to permit a more flexible use and development on different platforms without license issues. It is natural to migrate CEASIOM from Matlab to Python with CPACS adoption, as the TIXI and TIGL Python libraries created by the DLR already exist to parse the CPACS XML files.

So far, CEASIOMpy has mainly covered the core module, the aerodynamic module in CEASIOM, which includes two different methods to calculate the aerodynamic coefficients, the empirical method Digital Datcom(L0) and the possibility to make Euler or Navier Stokes calculations using SU2(L2 or L3).

[Figure 8](#) shows the user interface of CEASIOMpy. As stated above, currently only the “Aero” tab is valid. User can import a CPACS aircraft geometry, define the flight conditions, select analysis tools and launch calculations.

After the calculation of one or several aerodynamic databases, the results can be plotted in an interactive way, with the possibility to draw different configurations in the same figure.

In the future, CEASIOMpy will also integrate a new version of Tornado and a Data Fusion module that will be developed in the AGILE project. The Data Fusion model (see also next Section) will be used to merge/fuse aerodynamic databases created with different methods, and to export the results to a stability and control analysis tool like SDSA.

#### 2.5 Data fusion

Assessment of maneuverability and agility in the conceptual design stage brings great challenges in the design process regarding the stability and control analysis over the entire flight envelope. A large lookup table of forces and moments must be constructed by CFD, and we have to address the computational cost: A useful look-up table for stability and control analysis, the so-called aerodynamic database, needs thousands of entries due to the high dimensionality of the parameter space.

The data fusion module is to construct surrogate models for the data production based on variable fidelity analysis tools in the MDO framework together with Kriging, co-Kriging and adaptive modeling techniques for fusing the outputs from each tool. The basic steps are as following:

- *Initialization.* Define the unknown aero-loads to be modeled; specify the parameter space by defining the independent variables and their range.
- *Sampling.* Two sets of sample points are used, one is from the dense sampling of parameter space by the low-fidelity method, and the other is for the sparser samples obtained using the high-fidelity method.
- *Co-Kriging surrogate model.* From the Kriging interpolation model, the co-Kriging model is constructed from the augmented samples both from the low- and high-fidelity methods. The (R)MSE (Root) Mean Square Error is computed to help choose the most beneficial updates to be added to improve the prediction.
- *Sampling updates.* Examine (R)MSE by the criterion for termination. The suggested updating hi-fi samples are added to the sampling space and evaluated by corresponding CFD tools.
- *Final surrogate model.* The final surrogate models based on the initial samples and the updates added to achieve some accuracy criterion.

The sampling procedures in the Data Fusion module is shown in [Figure 2](#), representing the above steps from 1 to 4.

The automated MAA module which has L0-L3 level tools are the default tools built in Data Fusion module to provide the (new) sampling data.

### 3. DC1-MDA aircraft example

The aircraft used to demonstrate the use of the New CEASIOM is the reference aircraft serving as Use Case in [AGILE\[2\]](#) project Design Campaign 1 (DC1). It is the summary and outcome from “Design Challenge 0”, using all the (semi-)empirical based tools, dubbed as DC0.

The aircraft is called DC1-MDA and is used in Design Campaign 1 (and beyond) for multi-disciplinary analysis (MDA) purpose. This aircraft does not correspond to an existing one, but it is in the range of an Airbus A320 or Boeing 737. It has been created within the AGILE framework using different aircraft conceptual design software tools ([Baseline Definition Reference Aircraft, 2016](#)). The main difficulty comes from the fact that no consortium data exist for this aircraft, so the comparisons are only possible between our different tools ([Baseline Definition Reference Aircraft, 2016](#)). The selected characteristics of DC1-MDA are shown in [Table II](#) and [Figure 9](#).

Figure 8 CEASIOMpy GUI to calculate aerodynamic database directly from a CPACS aircraft geometry

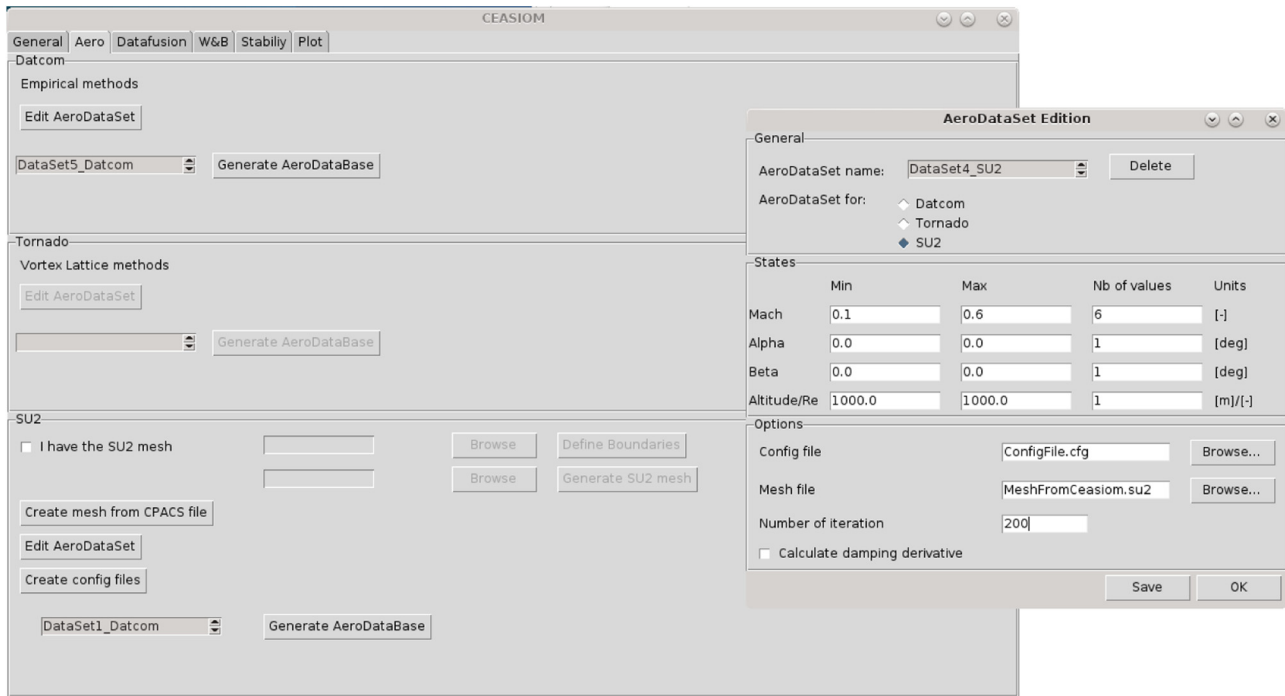
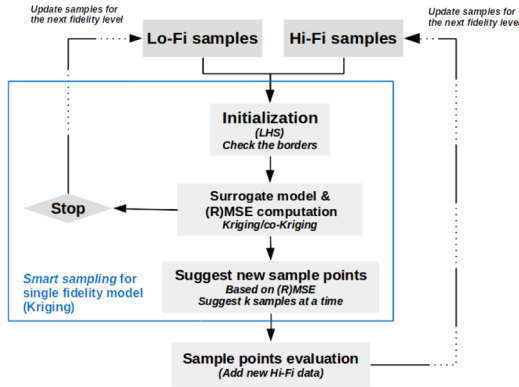


Figure 9 Sampling procedures in data fusion module



3.1 Model visualization and “repair” for computational fluid dynamics

The DC1-MDA model can be loaded and visualized in CPACSupdater, including the internal structures as well as the trailing edge control surfaces, as shown in Figure 4. Note that this configuration has a large rudder (from DC0), the following physical-based aerodynamic analysis will assess its rationality (Figure 10).

A successfully generated mesh is the key for CFD computation. Sumo can automatically generate unstructured meshes from geometry defined in CPACS file provided that the geometry is smooth enough to allow high-quality surface and volume meshes. A non-meshable geometry shall be modified, or “repaired” to produce a meshable CFD model. The DC1-MDA model is not necessarily meshable, as it is produced during DC0 by a number of the empirical-based tools. When

we visualized it in CPACSupdater, it was found that there were several un-intended almost grazing surface intersections, particularly in the center section of the wing just penetrating the lower side of the fuselage, see Figure 11(a). The surface geometry module in sumo does not compute “trimmed surface patches” from interactions based on the continuous geometry, as that would require additional user interaction to determine which parts to trim. The intersection lines cannot be flipped or split in the mesh refinement process. Although the surface intersections could be meshed, it would lead to a mesh of very low quality, and the computational results would be questionable. Minor modifications solved this problem easily, by changing the vertical position (z) of the main wing minutely in CPACSupdater etc.[3], and saved back as a (new) CPACS file which becomes the reference for this study.

This is a simple example that by importing and manipulating the CPACS-based geometry in CPACSupdater it can be made more suitable for meshing. The meshable model is automatically obtained by the CPACS-sumo converter applying on the modified DC1-MDA CPACS file (Figure 12).

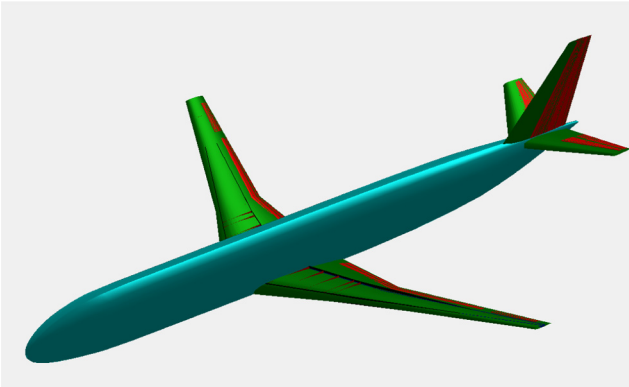
3.2 Automated MAA results

The automated MAA module in CEASIOM has a number of tools with different fidelities to provide aerodynamic coefficients. In this section, all the tools are applied to the DC1-MDA aircraft to collect the aerodynamic database from different tools, with or without the same flight conditions, trying to filling in the desired flight envelope and simulating its flight performance.

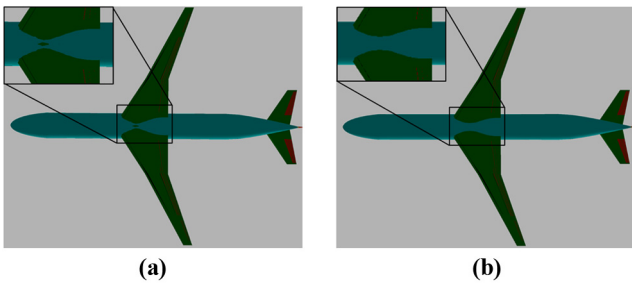
The computed flight conditions are imported (and stored in the output file) in the aeroPerformanceMap node of the XML file, which has “Angle of attack”, “Angle of side-slip”, “Mach number” and “Reynolds number” 4 children nodes, and if



**Figure 10** DC1-MDA Model visualized in CPACSupdater

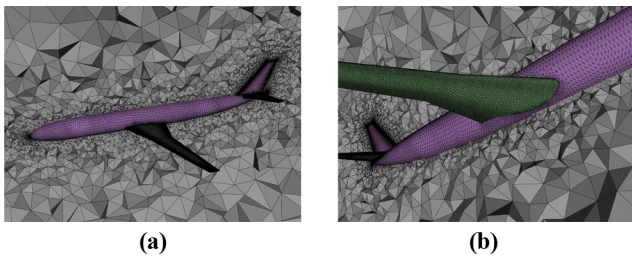


**Figure 11** DC1-MDA geometry visualization and rendering in CPACSupdater



**Notes:** (a) Before modifying the wing-body interaction in CPACSupdater; (b) after modifying the wing-body interaction in CPACSupdater

**Figure 12** Unstructured mesh created by sumo for DC1-MDA model



**Notes:** (a) DC1-MDA volume mesh, View 1; (b) DC1-MDA volume mesh, View 2

control surface calculations are involved, 2 more children nodes, “control surface name” and “relative deflection” applied. Details can be found in [CPACS Documentation \(2015\)](#).

Different flight conditions have been extracted from the aeroPerformanceMap and been tested for each tool, depending on their capability. For example, using VLM (Tornado) or

Euler method solvers, they do not need inputs as Reynolds number, but the altitude. In aeroPerformanceMap node in CPACS, the altitude is not defined, so it has been calculated either with corresponding Mach number and Reynolds number, or by user definition.

VLM Tornado is used as the tool to compute as many as the coefficients and derivatives over the cruise flight conditions at low speeds because it gives a good compromise between the computational time and accuracy. Euler solvers SU2 and Edge are used for transonic speed calculations. The following paragraphs show and discuss:

- the comparison of aerodynamic coefficients with elevator deflections at low speed computed by Tornado for predicting trimmed in pitch;
- a grid sensitivity study for the model imported automatically from the CPACS XML file, using two different Euler solvers SU2 and Edge; and
- the comparison of the L0,L1,L2 tools in the CEASIO-MAA module.

### 3.2.1 Aero-coefficients for trimming analysis – tornado

Tornado can compute all the static and quasi-static aerodynamic coefficients including the first-order derivatives for TED deflections. [Figure 13](#) shows the  $C_L$  and  $C_m$  coefficients for different elevator deflection angles at Mach = 0.2. It can be easily computed that  $C_{L,\delta e} = 0.0077/^\circ$  and  $C_{m,\delta e} = -0.0322/^\circ$ .

At this stage, the horizontal trim for different altitudes at low speed range can be estimated by knowing the mass of the aircraft and its reference values ([Table II](#)). The motion of equations for the steady and level flight are:

$$L = C_L \frac{1}{2} \rho(alt) V^2 \cdot S = W \quad (1)$$

$$M = C_m \frac{1}{2} \rho(alt) V^2 \cdot S = 0 \quad (2)$$

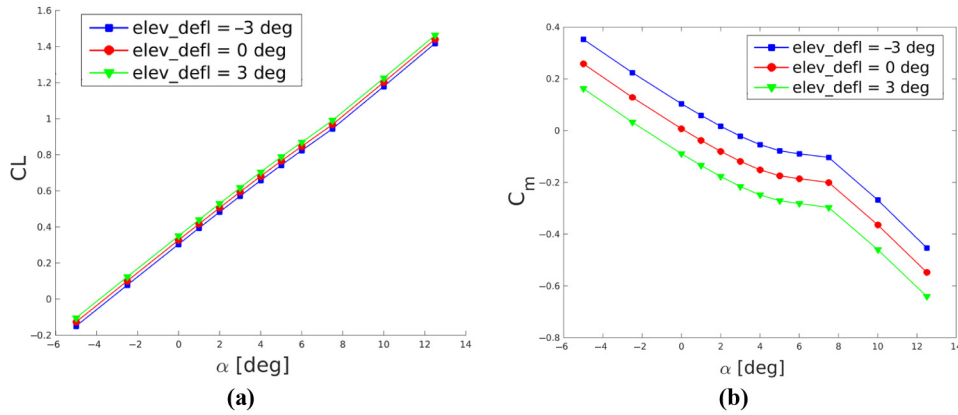
where  $W$  is the total weight of the aircraft,  $L$  is the lift force and  $M$  is the pitching moment around its Center of Gravity (CG) which is already defined in the CPACS file. Air density  $\rho$  is a function of flight altitude.

[Figure 14](#) shows the horizontal trim conditions from Tornado computed according to [equations \(1\)](#) and [\(2\)](#). We can see that the trim angles of attack and elevator deflections for the altitude above sea level are a bit too large, which is prone to stall. The main wing and the elevator need to be re-designed for trim. Also, the static margin is around 44 per cent calculated by SDSA; this indicates that the aircraft is too stiff in steering, and it needs to be re-design for re-distributing the mass to get CG shifted a bit afterward to reduce the static margin.

### 3.2.2 Grid sensitivity study

The validity of the Euler calculations has been tested by analyzing the sensitivity of the mesh choice. The default conditions for this sensitivity study were a Mach number of 0.7 with an angle of attack of  $1^\circ$  at an altitude of 1000 meter above the sea level with ISA atmospheric model. At the beginning, the default mesh generated by sumo has been used, and then the minimum and maximum length of tetrahedra has been divided by 1.2 for each

**Figure 13** Coefficients comparison for different elevator deflection angles, computed from Tornado, at Mach = 0.2, altitude = 5000 m



**Notes:** (a) Lift coefficient  $C_L$  and (b) pitching moment coefficient  $C_m$  vs the angle of attack

**Table II** The DC1-MDA selected characteristics

Parameter	DC1-MDA	Unit
Wing span	28.1	m
Wing area	82.8	m <sup>2</sup>
MAC	3.73	m
MTOW	39750	kg

Source: [Baseline Definition Reference Aircraft \(2016\)](#)

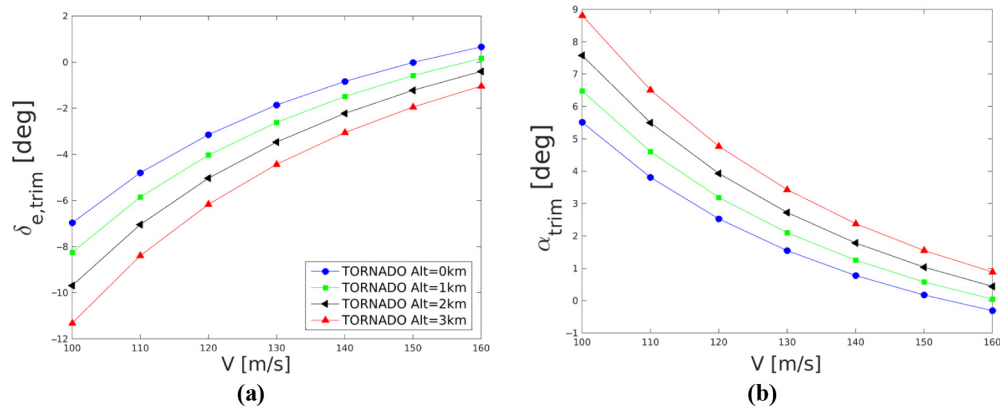
new step, which lead to an increase of the total number of tetrahedra shown in [Figure 15](#). Besides, a second type of mesh was created, the so called “R” refine mesh (with leading and trailing edge refinement increased from the default value 2 to 3 and 4, respectively) in opposition to the “D” default mesh.

[Figure 16](#) shows the evolution of lift coefficient, drag coefficient, moment coefficient as well as the computational time for the different meshes. [Figure 16\(b\)](#) shows that the artificial drag can be reduced by using a finer grid, and both solvers converge to the same value. The increase in computation time was obviously exponential, and the calculations by both

solvers were launched on a 4-core [Xeon@3.70GHz](#) workstation. The computational time by SU2 is significantly longer than Edge, and one reason can be that the coding languages to implement the numerical schemes are different, and that SU2 as a C code mentality which is not that suited for high performance codes. Edge is coded in Fortran, which is naturally suited for numerical programming. Fortran was carefully designed to allow the compiler to recognize most spots for optimization, due to the language features. On the other hand, C/C suits system related development. For the CFD problems, a well-coded Fortran tool has some advantages than a C code regarding to the scientific computing efficiency and the capacities of parallelism.

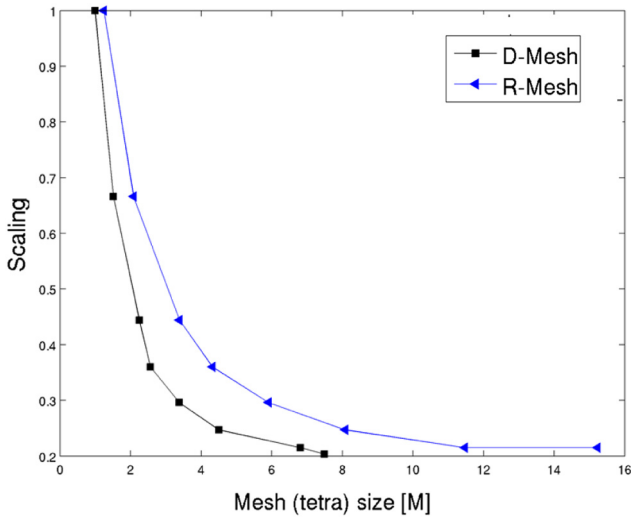
It was decided to use the mesh that was a good compromise between a reasonable computation time and converged coefficients value that are close to what we would obtain with the better mesh. The mesh chosen is the R-mesh with around 6 million cells (called working mesh). If we examine [Figures 15](#) and [16](#) carefully, it can be seen that the working mesh has relatively low computation time while the computed aero-coefficients are close to the converged values.

**Figure 14** Trim results from tornado



**Notes:** (a) Trim elevator deflection angle; (b) trim angle of attack

Figure 15 Mesh scaling



3.2.3 L0-L2 tools comparison

The aerodynamic coefficients obtained with the different available tool have been compared. A mach number of 0.6 was used to avoid the transonic effects that are not well

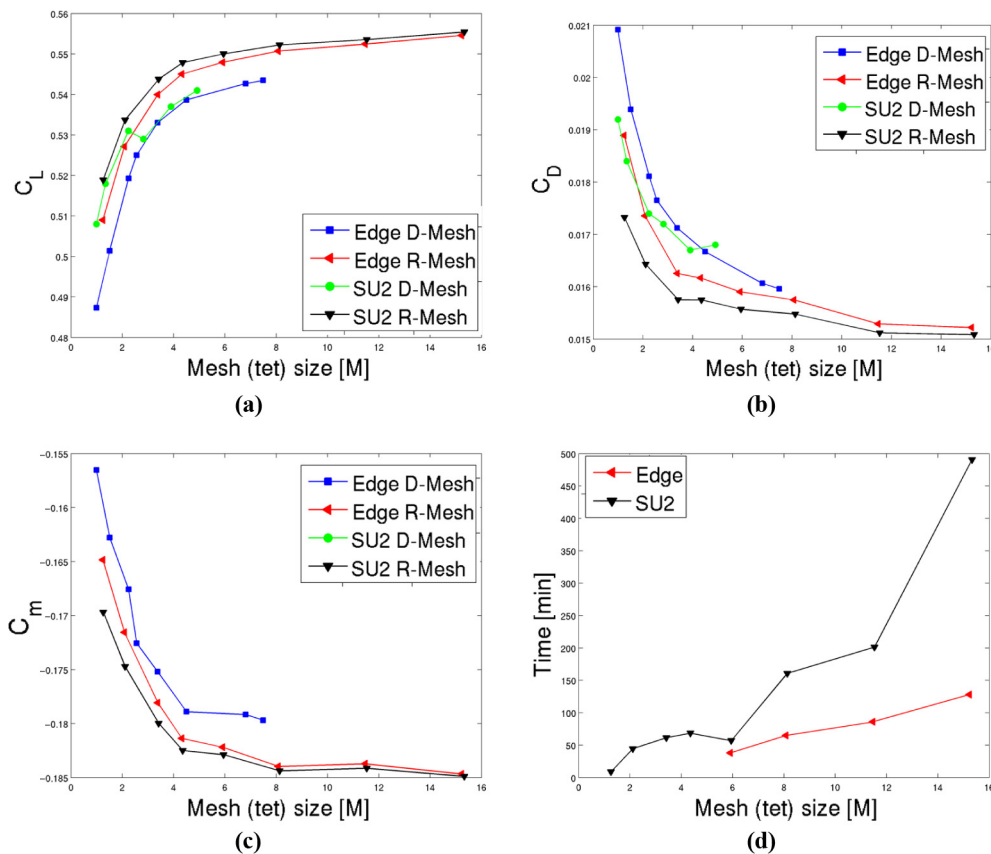
predicted by both Datcom and Tornado. The flight conditions used for this comparison are summarized in Table III.

Figure 17 shows that the lift coefficient is well predicted by the three different methods between angles of attack of  $-5$  and  $+5^\circ$ . Above this range, we can observe that stall occurs at an angle of approximately  $8^\circ$ . This is more clearly visible in the SU2 results; however, the stall angle cannot be determined precisely because Euler solver is unable to predict the boundary layer separation point. Some comparison should be performed with RANS solver and/or wind tunnel experiment.

Figure 18 shows that the minimum drag coefficient is obtained for an angle of attack of about  $-2.5^\circ$  which corresponds to the angle where the lift coefficient is minimum. This value is almost zero with Tornado and SU2 because they do not include skin friction in their physical model. With SU2, the drag coefficient seems to be overestimated for high angle of attack, and this is probably due to the stall and the unsteady flow condition.

Figure 19 shows that the aircraft is horizontally stable ( $\delta C_m / \delta \alpha < 0$ ) for angle of attack from  $-5$  to  $+5^\circ$ . The break in the slope that appears for Tornado and SU2 results after approximately  $+5^\circ$ , is probably due to the stall of the horizontal stabilizer which happens at a lower angle of attack than the stall of the main wing. As a reminder, this aircraft is only the first

Figure 16 Mesh sensitivity study using edge and SU2



Notes: (a) Lift coefficient  $C_L$ ; (b) drag coefficient  $C_D$ ; (c) pitching moment coefficient  $C_m$ ; (d) computational time comparison for the R-mesh

Table III Flight condition used for the comparison of the different methods

Parameter	Value	Unit
Mach number	0.6	–
Altitude	5000	m
Angle of attack	–5 to 12.5	deg
Sideslip angle	0	deg

Figure 17 Lift coefficient versus angle of attack

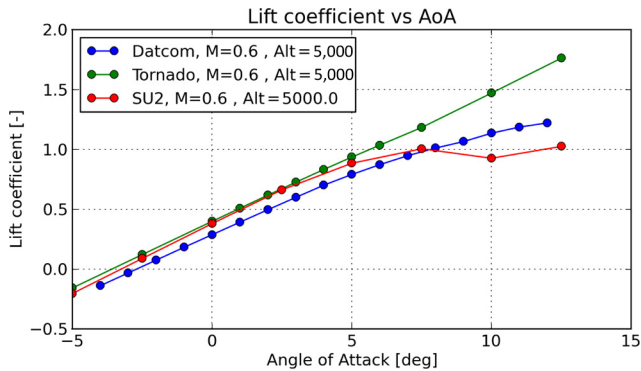


Figure 18 Drag coefficient versus angle of attack

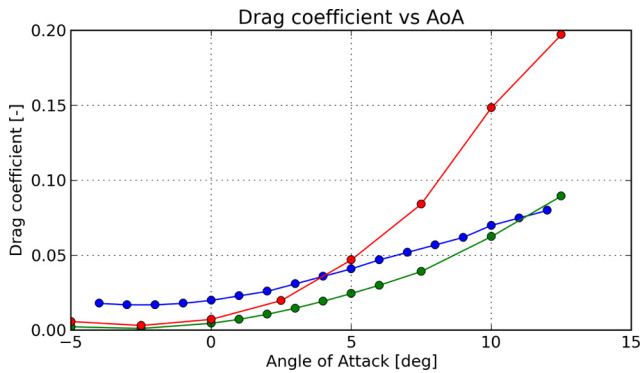
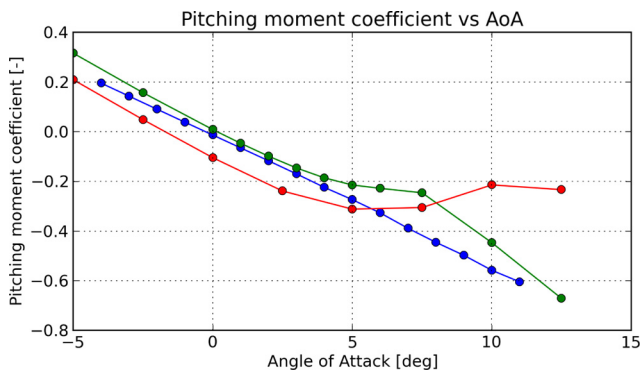


Figure 19 Moment coefficient versus angle of attack



phase of its design, and it has not been optimized in terms of stability.

The value calculated with SU2 that occurs after the stall ( $\alpha > 7.5^\circ$ ) are not completely accurate. Due to the separated flow regions that result from the stall, the flow is highly unsteady and the aerodynamic forces are not entirely converged. Nevertheless, these values are not so important as they are completely out of the normal range of flight condition.

The good agreement in results obtained by the different aerodynamic modules shows that it will be possible to create an automatic sampling approach that takes advantage of each method according to their fidelity levels and their limitations. For example, it would be useless to spend plenty of computation time with Euler calculation in a range where Datcom or Tornado can give trustful results. This computation time can be better used to perform some simulations at higher mach number or angle of attack, where the other methods are unusable.

### 3.3 Flight simulation

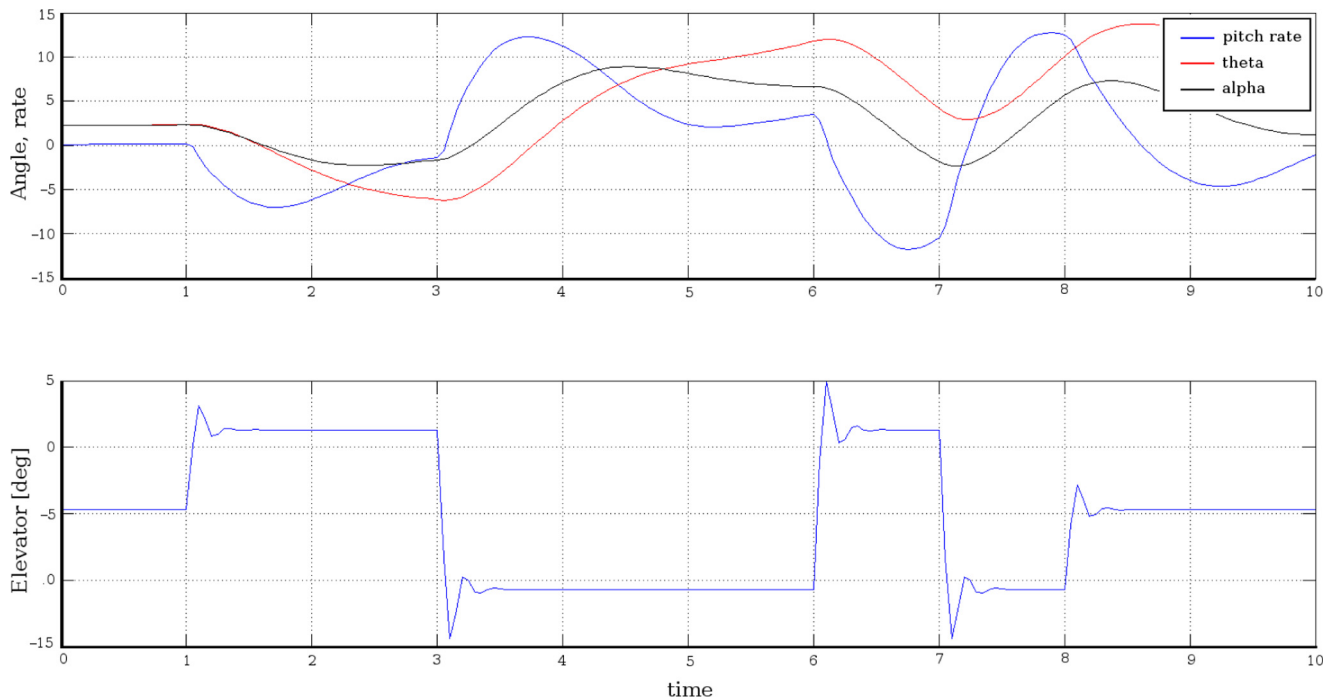
The classical modes of motion indicate the linear stability of the aircraft, i.e. the response that would be expected to small disturbances. Simulating a flight maneuver in a flight simulator allows the nonlinear stability of the aircraft to be assessed. The time history in Figure 20 shows how the  $\theta$ ,  $\alpha$  and the pitch rate oscillate under the step-function-type small disturbance of the elevator from PHALANX, which is a flight simulation tool from Delft University of Technology. The authors would like to thank Dr Mark Voskuil to provide the analysis for Figure 20. PHALANX has been used instead of SDSA (which was in the SimSAC version of CEASIOM) due to its compatibility with the CPACS format. The time domain simulation is in trimmed flight at sea level conditions and True Air Speed TAS = 130 m/s, after 1 s it performs a 2-3-1-1 maneuver in the pitch axis, namely, longitudinal stick 2 s nose down, 3 s nose up, 1 s nose down and 1 s nose up.

## 4. Conclusion

The new version of CEASIOM (CEASIOMpy) provides an enhanced MDO framework for aircraft preliminary design with the input as a CPACS format file, particularly with its capability to perform the automatic aerodynamic analysis. This “automation” ability is especially valuable, for instance, for the Euler calculation with SU2 which needs an appropriate mesh. To perform this operation, a converter has been developed, which can read a CPACS XML file and output a meshable SUMO file. This meshable SUMO file allows to automatically create unstructured isotropic meshes for Euler analysis as well as the hybrid meshes for RANS computations.

The possibility to create a multi-fidelity aerodynamic database in an automatic way has been demonstrated to construct the entire aero-database for Stability & Control analysis. As a test case, results obtained by Tornado (saved in CPACS format) are analyzed in flight simulation tool PHALANX, and the simulated flight indicated the analyzed aircraft’s maneuverability.

The new CEASIOM version is still under development. One of the next steps will be to convert the last Matlab module (Tornado) into Python to make it easier to work together in the

**Figure 20** Flight time domain simulation from PHALANX, trimmed flight with true air speed TAS = 130 m/s at sea level

CPACSpy framework. This will also facilitate the implementation of the automatic data sampling or data fusion methods. A Weight & Balance module should be written in Python with some enhancements, which should permit to calculate masses and moment of inertia not only for conventional aircraft but also for unconventional aircraft geometry such as Blended Wing Body or Strut Braced Wing.

### Acknowledgements

The research presented in this paper has been performed in the framework of the AGILE project (Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts) and has received funding from the European Union H2020 program (H2020-MG-2014-2015) under Grant Agreement No. 636202. The Swiss participation in the AGILE project was supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 15.0162. The authors are grateful to the partners of the AGILE consortium for their contributions and feedback.

### Notes

- 1 This category of level (L) definition is generally suitable for all the design disciplines. In the automated MAA, it refers to aerodynamic analysis.
- 2 EU H2020 Project: Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts, [www.agile-project.eu](http://www.agile-project.eu)

- 3 The position in vertical  $z$  direction of the main wing was change from  $0.9m$  to  $0.85m$  and the position in  $z$  of the horizontal stabilizer from  $-0.9$  to  $0.88 m$ . The definition of the CPACS coordinate system can be found in [CPACS Documentation \(2015\)](#).

### References

- Anderson, J.D. (2004), *Modern Compressible Flow with Historical Perspective*, 3rd ed., McGraw-Hill Education.
- Baseline Definition Reference Aircraft (2016), "Baseline definition reference aircraft", Tech. rep., AGILE Project Deliverable Report D2.5, Consortium Only.
- CPACS – A Common Language for Aircraft Design (2016), "CPACS – a common language for aircraft design", available at: <https://software.dlr.de/p/cpacs/home/> (accessed 26 February 2016).
- CPACS Documentation (2015), "CPACS documentation", Tech. rep.
- Eliasson, P. (2002), "Edge, a Navier-Stokes solver for unstructured grids", June 2002, pp. 527-534.
- Goetzendorf-Grabowski, T., Mieszalski, D. and Marcinkiewicz, E. (2011), "Stability analysis using SDSA tool", *Progress in Aerospace Sciences*, Vol. 47 No. 11, pp. 636-646, doi: [10.1016/j.paerosci.2011.08.007](https://doi.org/10.1016/j.paerosci.2011.08.007).
- Griva, I., Nash, S.G. and Sofer, A. (2009), *Linear and Nonlinear Optimization*, 2nd ed., Society for Industrial Applied Mathematics, Philadelphia.
- Isikveren, A. (2002), "Quasi-analytical modeling and optimization techniques for transport aircraft design",

- PhD thesis, Royal Institute of Technology, KTH, Stockholm.
- Katz, J. and Plotkin, A. (1991), *Low-Speed Aerodynamics: From Wing Theory to Panel Methods*, McGraw-Hill.
- Melin, T. (2003), "Using internet interactions in developing vortex lattice software for conceptual design", Software available at: <http://redhammer.se/tornado/>, July 2009.
- Palacios, F., Economon, T.D., Wendorff, A.D. and Alonso, J. (2015), "Large-scale aircraft design using SU2", *53rd AIAA Aerospace Sciences Meeting, Kissimmee, FL*, AIAA 2015-1946, 5-9 January 2015.
- Palacios, F., Colonno, M.R., Aranake, A.C., Campos, A., Copeland, S.R., Economon, T.D., Lonkar, A.K., Lukaczyk, T.W., Taylor, T.W.R. and Alonso, J. (2013), "Stanford university unstructured (SU2): an open-source integrated computational environment for multi-physics simulation and design", *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, AIAA 2013-0287, Grapevine, Texas*, 7-10 January 2013.
- (PDAS), P.D.A.S. (2013), "Digital datcom", available at: [www.pdas.com/datcomDescription.html](http://www.pdas.com/datcomDescription.html), (accessed 17 November 2016).

- Si, H. (2013), "TetGen: a quality tetrahedral mesh generator and 3D delaunay triangulator", Tech. rep., User's Manual, WIAS technical Report No. 13.
- Tomac, M. (2014), "Towards automated CFD for engineering methods in aircraft design", PhD thesis, Royal Institute of Technology, KTH, Stockholm.
- Tomac, M. and Eller, D. (2011), "From geometry to CFD grids. an automated approach for conceptual design", *Progress in Aerospace Sciences*, Vol. 47 No. 8, pp. 589-596, doi: [10.1016/j.paerosci.2011.08.005](https://doi.org/10.1016/j.paerosci.2011.08.005).
- Zhang, M. (2013), "Computational design framework for a fully parametric virtual aircraft with morphing surfaces", Tech. rep., NOVEMOR Project Deliverable Report D3.2.
- Zhang, M. (2015), "Contributions to variable fidelity MDO framework for collaborative and integrated aircraft design", PhD thesis, Royal institute of Technology KTH, Stockholm.

**Corresponding author**

**Aidan Jungo** can be contacted at: [aidan.jungo@cfse.ch](mailto:aidan.jungo@cfse.ch)