# Using surrogate models to speed up the creation of aerodynamic databases in CEASIOMpy

Aidan Jungo, Vivien Riolo, and Jan B. Vos

CFS Engineering, Lausanne, Switzerland

October 2, 2020

**Abstract**

Engineers always look for new methods to speed-up the aircraft design process. Aerodynamic analysis are often the most costly in terms of computational time. Machine learning techniques as surrogate modeling are more and more used in this domain, as well as for optimisation workflows.

CEASIOMpy is an open source conceptual aircraft design software written in Python and using the CPACS standard, an XML data definition for aircraft. CEASIOMpy includes modules which cover several of the main aircraft design disciplines. These modules can be connected and executed in an order defined by the user depending on his needs. CEASIOMpy includes aircraft geometry CAD, Weight & Balance estimation, aerodynamics (Vortex Lattice Methods and SU2), and stability analysis modules. The aerodynamic modules of CEASIOMpy are being used and further developed in the framework of the H2020 project AGILE4.0 in collaboration with other European partners in order to run Multidisciplinary Design Analysis and Optimization (MDAO) on aircraft design cases.

Surrogate modeling was implemented in CEASIOMpy using the SMT libraries. First, a few high fidelity Euler calculation are performed for different flight state parameters (angle of attack, Mach number and altitudes), then these results are used to train a surrogate model that can be used to generate a more complete aerodynamics database or replace costly aerodynamic calculations in an optimisation workflow.

In this paper we will describe the different parameters that are used to create and employ surrogate models efficiently in CEASIOMpy. Accuracy testing will be performed on different test cases. We will also evaluate the possibility to add geometry parameters (such as wing span, fuselage length, etc.) in a surrogate model to make it suitable for a real optimisation workflow.

***Keywords:*** aircraft design, aerodynamics, surrogate model

# 1 Introduction

Aircraft design is a long process that requires a lot of knowledge and computational tools in many different disciplines. Machine learning techniques are increasingly used in the field of aircraft design to speed up the design process. Surrogate modeling is one of the most commonly used methods, being particularly useful to interpolate data that has not been calculated before. In aircraft design, the generation of aerodynamic databases is generally the most computationally expensive discipline and various techniques including surrogate modeling are being developed to reduce these costs [1].

## 1.1 CEASIOMpy environment

### 1.1.1 CPACS format

CPACS [2] is an XML data definition that permits to describe in a structured, hierarchical manner the characteristics of the aircraft as its geometry, engines, performances and many other discipline specific data. The CPACS format has been developed for more than 10 years at DLR in Germany.

Among other data the CPACS format can store one or several aerodynamic databases, called aeroMaps. These aeroMaps are useful to save all the different flight state parameters (Mach number, altitude, angle of attack, angle of side slip) and the corresponding aerodynamic coefficients (3 for the forces and 3 for the moments). The atmospheric model used - the International Standard Atmosphere (ISA) in our case - is also saved.

```xml
<aeroMap uID="Train9pts">
  <name>Train9pts</name>
  <description>Auto-generated Aeromap</description>
  <boundaryConditions>
    <atmosphericModel>ISA</atmosphericModel>
  </boundaryConditions>
  <aeroPerformanceMap>
    <altitude mapType="vector">0;0;0;0;0;0;0;0;0</altitude>
    <machNumber mapType="vector">0.3;0.3;0.3;0.3;0.3;0.3;0.3;0.3;0.3</machNumber>
    <angleOfAttack mapType="vector">-8;-6;-4;-2;0;2;4;6;8</angleOfAttack>
    <angleOfSideslip mapType="vector">0;0;0;0;0;0;0;0;0</angleOfSideslip>
    <cl mapType="vector">-0.423309;-0.213923;-0.00447686;0.204856;0.413917;0.622573;0.830741;1.03842;1.24558</cl>
    <cd mapType="vector">0.00767418;0.00430734;0.00252977;0.0023323;0.00370036;0.00661524;0.0110576;0.0170133;0.0244752</cd>
    <cs mapType="vector">3.72226e-06;2.94362e-06;2.16869e-06;1.43416e-06;6.9443e-07;-3.61777e-08;-7.55585e-07;-1.46418e-06;-2.15606e-06</cs>
    <cml mapType="vector">8.27135e-06;5.39351e-06;3.131e-06;1.56766e-06;5.32109e-07;1.67902e-08;-2.60453e-08;3.5041e-07;1.11936e-06</cml>
    <cmd mapType="vector">-4.47159e-05;-3.44746e-05;-2.45089e-05;-1.50289e-05;-6.098e-06;2.1763e-06;9.7602e-06;1.66763e-05;2.2986e-05</cmd>
    <cms mapType="vector">0.147681;0.113345;0.0791058;0.0450909;0.0113747;-0.0220798;-0.055471;-0.0892314;-0.12375</cms>
  </aeroPerformanceMap>
</aeroMap>
```

Figure 1: Example of an aeroMap in a CPACS file

### 1.1.2 CEASIOMpy framework

CEASIOMpy [3] is an open source conceptual aircraft design environment. CEASIOMpy can be used to set up complex design and optimization workflows, both for conventional and unconventional aircraft configurations. It provides tools for various disciplines in aircraft design, like: weight and balance, aerodynamics, structures, mission analysis and stability analysis.

CEASIOMpy is a collection of domain specific modules which can be connected together in different order depending on the application. CEASIOMpy uses the CPACS format to store and exchange information between the modules.

One of the main use of CEASIOMpy, for now, is to generate aerodynamic databases for preliminary aircraft design. It has been extensively use in the framework of the European project AGILE [4, 5, 6] and today in the follow-up project AGILE4.0 [7].

## 1.2 Surrogate model

With the ever-increasing amount of data at disposal and the higher complexity of today's models for various systems, machine learning that exploits those data to make predictions for a given problem without directly solving a large system appears as an interesting candidate to be used in industry.

Many machine learning methods are already being used in various sub-domains of aircraft design, and one of the most commonly used methods in the domain of aircraft design appears to be surrogate modeling [8].

By providing a training data-set containing inputs and their corresponding outputs, a surrogate model can be trained and used to predict an output value for new input points. A surrogate model can be considered as a black box as the user can specify which parameters shall be taken as inputs and outputs. Thus one can create a model based on a workflow by including all the desired modules.

One approach that can be adopted is to make a first search using a surrogate model, which can be used to extrapolate a result based on a set of predefined data for which the output is already known. In the domain of CFD, such model could become particularly handy in order to guess an initial solution to a problem based on previous simulations results and so converging to a solution faster than without an initial guess.

The implementation of surrogate models in CEASIOMpy has been made using the Surrogate Modelling Toolbox (SMT) [9] which is an open-source Python package consisting of libraries of surrogate modeling methods. The methods that were implemented in CEASIOMpy are different forms of the Kriging model and a least-square approximation. These methods and others are described in detail on the SMT website [10].

## 1.3 OPTIMALE aircraft

The unmanned aerial vehicle OPTIMALE is a Medium Altitude Long Endurance (MALE) conventional low wing configuration with a T-tail. Aircraft and mission specifications used for this study are given in Table 1.

| | |
|---|---|
| Semi-span | 16.2 $m$ |
| Reference area | 55 $m^2$ |
| Cruise speed | 150 $m/s$ |
| Dive speed | 180 $m/s$ |
| Cruise altitude | $15'500\,m$ |
| Ceiling altitude | $18'000\,m$ |

Table 1: OPTIMALE specifications

The OPTIMALE configuration has been developed during the German AeroStruct research project [11]. This aircraft has been used in several European project as for example AGILE [5, 12, 6]. A visualization of the model (without engines and tanks) using the CPACS visualization and modification tool CPACSCreator [13] is shown in Figure 2.

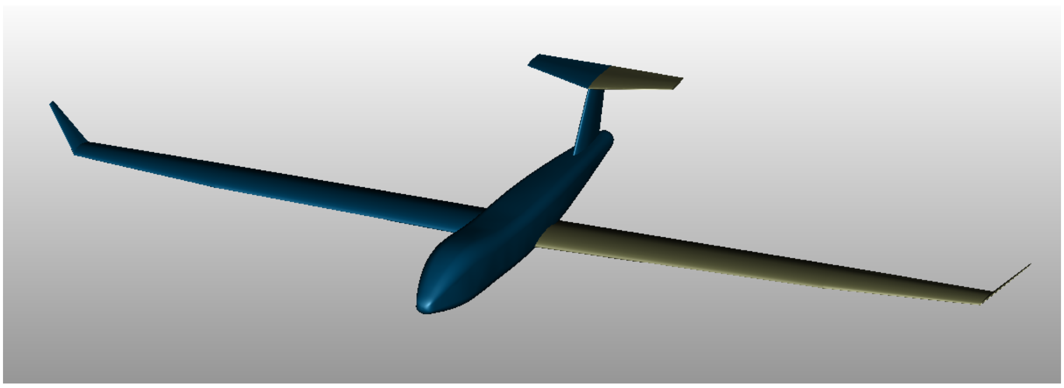

Figure 2: OPTIMALE CPACS configuration in CPACSCreator

## 2 Implementation

### 2.1 Data generation

To create a surrogate model one needs to first generate initial data (aerodynamic data in our case). In CEASIOMpy different possibilities can be used to generate aerodynamic data.

#### 2.1.1 PyTornado (VLM)

PyTornado [14] is an implementation of the vortex lattice method (VLM). The VLM, based on potential flow theory, is the simplest general method for 3D aerodynamic analyses of aircraft. The method requires only a coarse definition of the aircraft geometry and the flight state. The mesh is created from the CPACS definition during the initialisation of PyTornado.

Owing to the low number of input parameters, analyses can be set up with little effort and they are computationally inexpensive. PyTornado is an ideal tool for conceptual aircraft design. Short computation times makes it possible to easily obtain estimates of aerodynamic loads and to benchmark different concepts.

VLM methods also contain some drawbacks, among them: the effect of the fuselage is neglected, and the thickness of airfoils is not taken into account. Generally the lift is well predicted by these methods, especially for low angle of attack, however, the drag is underestimated.

In Figure 3, we can see how PyTornado can be used in a CEASIOMpy workflow. A CPACS file is provided as input, it contains the aircraft geometry and the flight state to calculate. Then, PyTornado calculates the aerodynamic coefficients for all the flight states and stores them in an aeroMap. Finally the updated CPACS file can be sent to another module, in this case 'PlotAeroCoef', which generates automatically several plots of the aerodynamic coefficients.

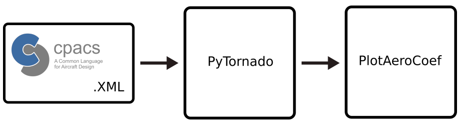


Figure 3: CEASIOMpy workflow to generate an aerodynamic database with SU2

#### 2.1.2 SU2 (Euler)

From a CPACS geometry, an unstructured tetrahedral mesh can be generated automatically with the open-source mesh generator SUMO [15]. A CEASIOMpy module called 'CPACS2SUMO' allows to convert a CPACS XML geometry format into a SUMO XML geometry format. SUMO is able to mesh automatically the aircraft by first creating a surface triangular mesh and then a volume tetrahedral mesh by using Tetgen [16].

Configuration files are also generated automatically from the aircraft parameters and flight conditions stored in the CPACS file. These configuration files are run automatically with the CFD solver SU2 [17]. The SU2 software suite from Stanford University is an open-source, integrated analysis and design tool for complex, multi-disciplinary problems on unstructured computational grids. In SU2, we use the Euler equations solver, as the mesh generated by SUMO does not contain enough cells in the boundary layer and is not suited for Navier-Stokes equations. The drawback of this method is that the friction drag is not taken into account and must be added later (see section 2.1.3), however it is faster than using the Navier-Stokes equations solver both to generate the mesh and to solve the equations.

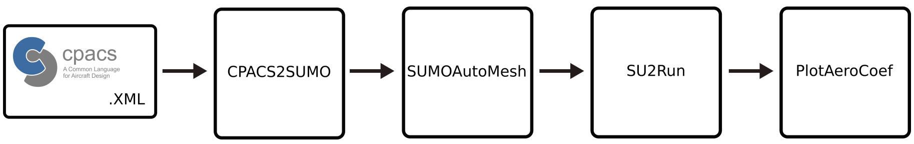


Figure 4: CEASIOMpy workflow to generate an aerodynamic database with SU2

The mesh refinement can be easily changed with the module 'SUMOAutoMesh', with a parameter called Refinement Level (RL). Meshes with different refinement have been created and used with SU2 to check the mesh convergence. The flight condition was AoA=0, M=0.3 @ sea level. The refinement level used for the rest of the paper has been chosen after a convergence analysis with a trade off between accuracy and the available computation time to performe the calculations. In Figure 5, we can see that a mesh with approximately 6-8 million cells gives a satisfactorily result and is used in the rest of this study.

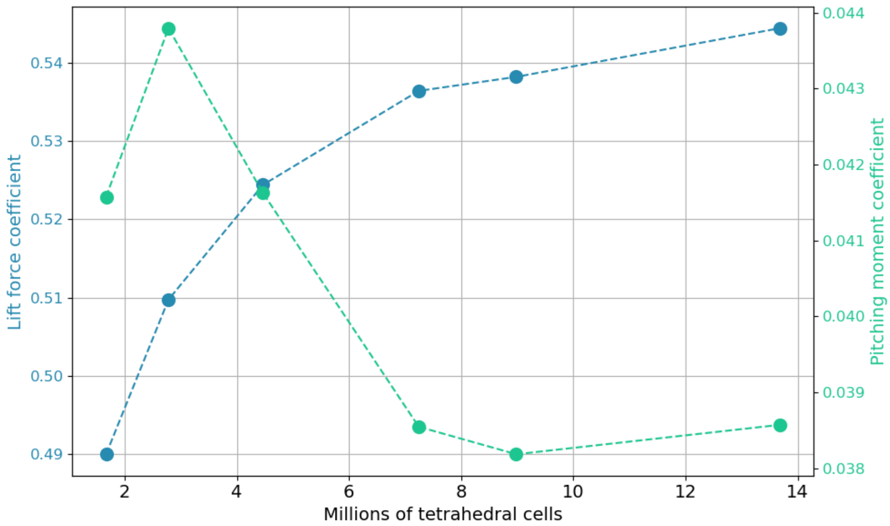


Figure 5: Mesh convergence with CL and Cm coefficients

### 2.1.3 Skin friction

In both methods mentioned above the effects of the skin friction (friction drag) are neglected. In order to get a better approximation of the total drag, in CEASIOMpy, there is the possibility to use the 'SkinFriction' module. It allows to estimate using empirical methods [18] the drag due to skin friction. The only input parameters required are the wing span, the wetted area and the Reynolds number. This module can be used either after the creation of the aerodynamic database with PyTornado or SU2 or after a surrogate model prediction if the surrogate have been trained without skin frictions.

## 2.2 Surrogate model

To use a surrogate model in CEASIOMpy, two new modules have been created, one to train the surrogate model and one to predict values from an existing surrogate model, see section 2.2.2. Both modules can be used independently from each other.

### 2.2.1 Training

As a start, the 'SMTrain' module in CEASIOMpy requires a CSV file with a list of inputs and outputs, that will serve as the model data set for training and optional validation. In the case of parameters which can all be found in an CPACS aeroMap the CSV file does not need to be specified. The module generates a model which is saved as a binary file, along with the information of the inputs and outputs that the model takes.

SMT contains different surrogate modeling methods and some of them have been implemented and are available in 'SMTrain':

- Kriging
- Least-squares
- KPLS
- KPLSK

### 2.2.2 Predicting

After a surrogate model has been generated, it can be used with the 'SMUse' module and run within any workflow. The particularity of this module comes from the entries that the surrogate will take, which will change depending on the model that is used. It is necessary to make a distinction here between the inputs and outputs of the 'SMUse' module and the inputs and outputs of the surrogate model, which is used by the module.

Once the surrogate has been trained, as explained in section 2.2.1, aerodynamic coefficients can be predicted for different values of the AoA, Mach number or geometry parameter. In Section 3, we will compare predicted values with some calculated ones and try to find which surrogate method and parameter distribution is best suitable to predict new data with the minimum amount of computation time to train the surrogate model.

# 3 Applications

## 3.1 Aerodynamic databases

### 3.1.1 Angle of Attack

The first test we performed with the surrogate model was to the predict aerodynamic coefficients for not calculated Angles of Attack (AoA). To evaluate the performance, different types of surrogate models and different number of training points were used. This first test was performed with aerodynamic data from PyTornado using the following workflow:

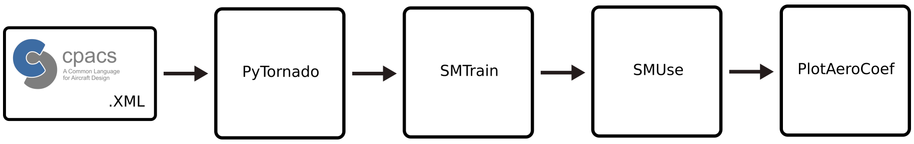


Figure 6: CEASIOMpy workflow to train and use the surrogate model for AoA prediction

To test the minimum number of points required to get a satisfactory surrogate model, we trained a simple Kriging model with different training sets (from 17 to only 3 different AoA). Then, we compared the predicted value of the surrogate model at two other AoA (1.5 and 6.5 degrees) with values calculated using PyTornado.

Figure 7 shows the error between the values predicted by trained surrogate model using different number of points, and the ”real” value calculated by PyTornado. Note that the validation point used for this test was not in the training set of points. For this simple application we can see that a surrogate model trained with only 5 points gives an error in the prediction of the aerodynamic coefficients of less than 0.1%, which is totally acceptable.

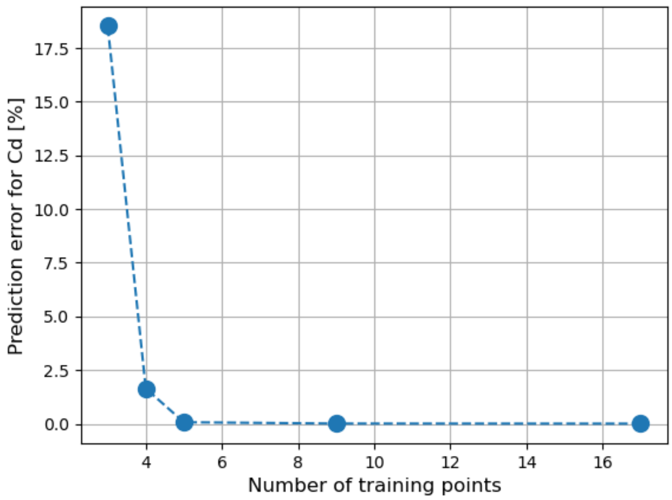


Figure 7: Error in prediction of Cd @$AoA = 6.5°$

### 3.1.2 Angle of Attack and Mach number

This case is closer to our intended use of surrogate models for real life applications. That is to say, given a few points at different AoA, Mach number and altitude, predict as accurately as possible all the points within the domain of these input values. This could, for example, be useful to generate aerodynamic databases for stability or mission analysis. A small number of points is used to generate the surrogate model, and then aerodynamic coefficients can be queried across the whole domain during the analysis without loosing time to wait for a new aerodynamic results.

For this test 25 training points have been calculated with SU2. The distribution of all the training points can be seen on the Figure 8 and the aerodynamic coefficient obtained are shown in Figure 9. The surrogate model has been trained using the Kriging method, the three other implemented methods did not give as good results.

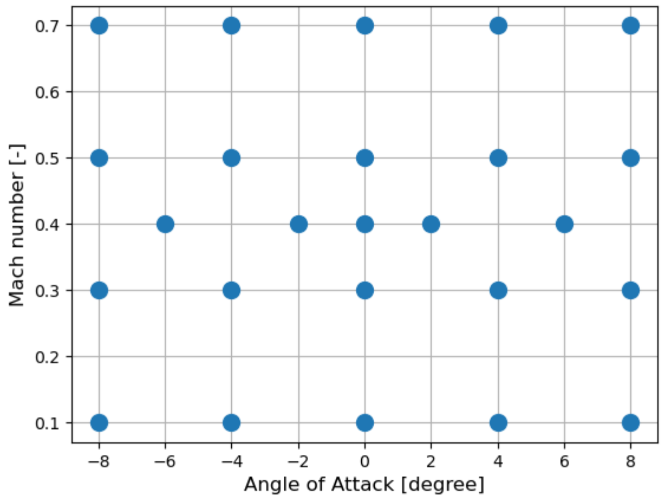


Figure 8: Distribution of AoA and Mach number values for the training set

Once the surrogate model is trained, it becomes easy and very fast to produce aerodynamic coefficients across the whole the domain. In Figure 10, we can see an example of aerodynamic coefficients plotted at different Mach numbers, which were not included in the training data set. Drag coefficients show the expected behaviour, however we can see that lift and pitch coefficients are not as linear as calculated data. This is due to the intrinsic definition of Kriging model and the few number of training points used in this case.

In Table 2 we can see a numerical comparison between the coefficients CL, CD, Cm predicted by the surrogate model and the actual data computed by SU2. The differences are generally lower than 5% but tend to increase when point are further away from the train dataset point. This problem could be overcome by using more points (which is longer) or using the same number of point but distributed with a different method like for example the Latin hypercube sampling method.

| Mach | AoA [°] | source | CL | CD | Cm |
|---|---|---|---|---|---|
| 0.35 | 3 | SU2 | 0.779 | 0.0360 | 0.0190 |
| | | Predicted | 0.778 | 0.0372 | 0.0185 |
| | | Error [%] | 0.1 | -3.3 | 2.8 |
| 0.45 | -3 | SU2 | 0.0711 | 0.0329 | 0.0806 |
| | | Predicted | 0.0728 | 0.0315 | 0.0794 |
| | | Error [%] | -2.4 | 4.1 | 1.4 |
| 0.55 | -3 | SU2 | 0.0851 | 0.0343 | 0.0804 |
| | | Predicted | 0.0940 | 0.0345 | 0.0770 |
| | | Error [%] | -10.4 | -0.5 | 4.1 |

Table 2: Comparison between aerodynamic coefficients predicted by the surrogate model and calculated using SU2

## 3.2 Aerodynamic database and geometry parameter

Another example of application of surrogate models in aircraft design is a case which implies the variation of one or more geometrical parameters. The main advantage of this technique is to pre-calculate a surrogate model from a Design of Experiment (DoE) and then to query the surrogate prediction (which is almost instantaneously) during an optimisation process to avoid time latency to get aerodynamic results.

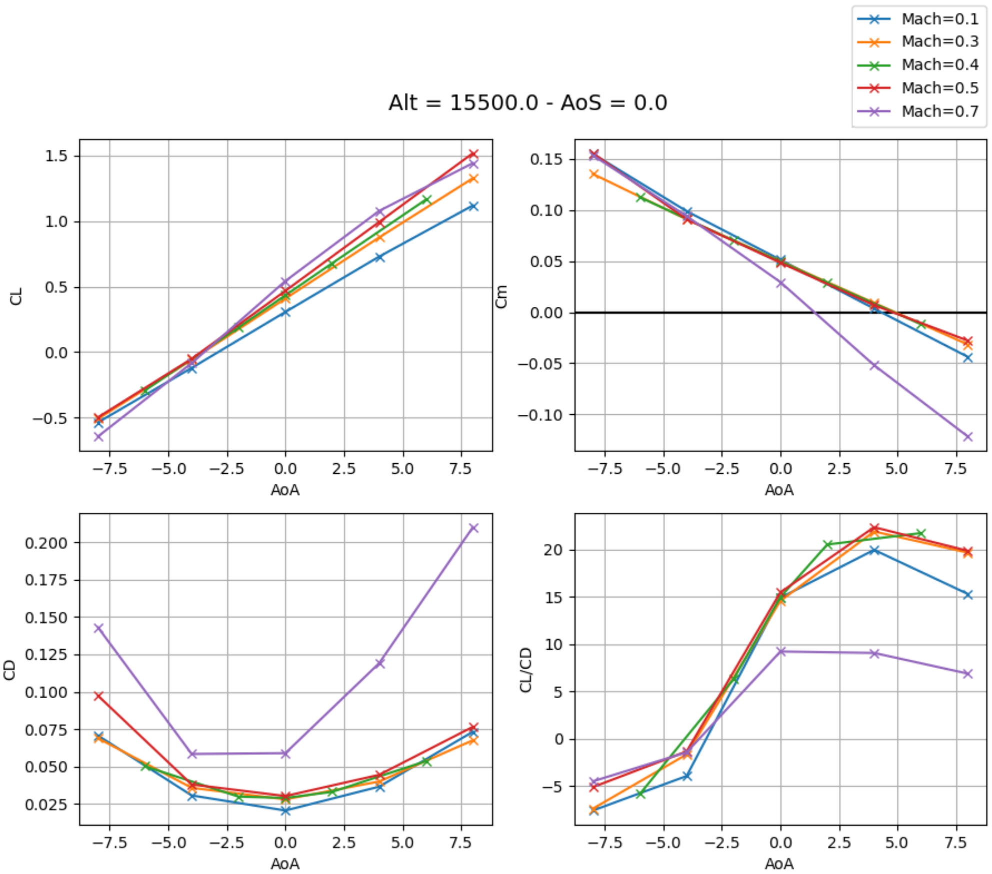


Figure 9: Aerodynamic coefficients calculated by SU2 and used to train the surrogate model

In this example we used PyTornado to generate the aerodynamics database, in real life applications the use of a surrogate model would be much more interesting when SU2 is used to generate the aerodynamic database because calculation times with SU2 could be very long, thus the time savings will be significant.

In this case, the design variable is the angle of attack, the angle of side slip and the span of the aircraft. A full factorial matrix with 5 values for each parameter has been used (125 in total), with the following lower and upper bounds:

- Angle of attack: from -4 to 4 $^{\circ}$
- Angle of side slip: from -5 to 5 $^{\circ}$
- Wing span: from 10 to 20 $m$

Then, the surrogate model has been used to predict the aerodynamic coefficients for a wing span of 14m and an angle of side slip of 1 degree and the results are compared with the results of a calculation using PyTornado. The results are shown in Figure 11, and one can observe an almost perfect prediction of the surrogate model in this case. The surrogate model works well for this case because it has a large number of training points.

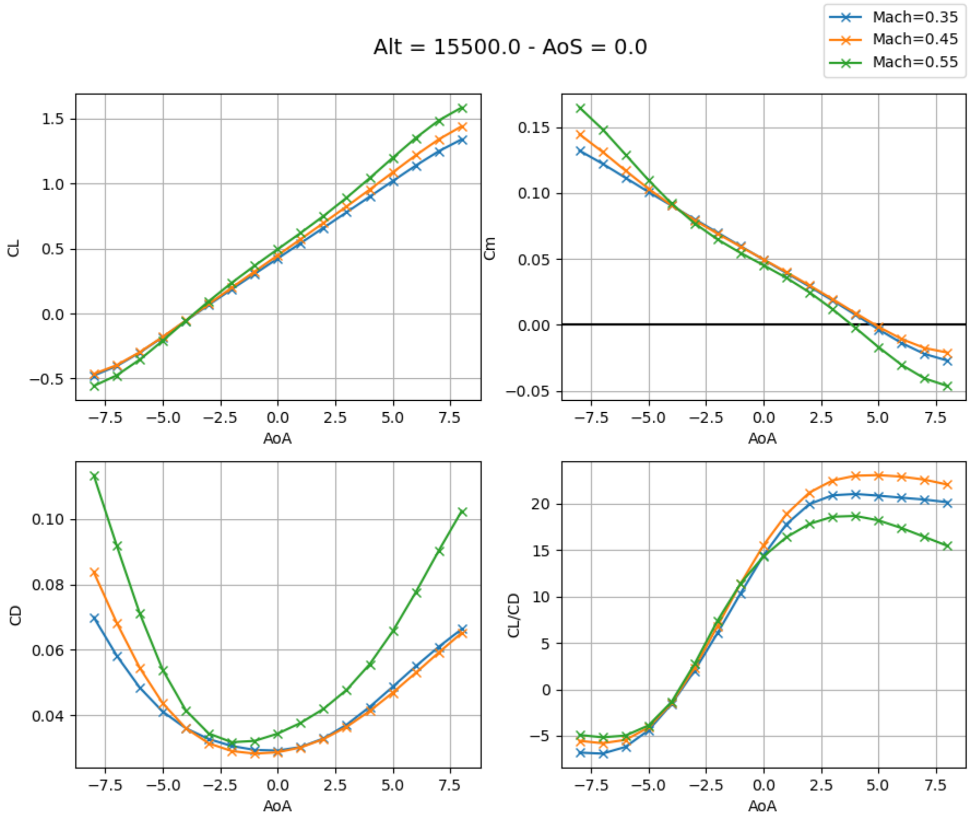


Figure 10: Example of predicted aerodynamic coefficients

# 4 Conclusions

In order to speed up the aerodynamic database creation during the conceptual aircraft design, we implemented a surrogate modeling tool in CEASIOMpy. Two modules have been added to CEASIOMpy, a surrogate modeling tool creation 'SMTrain' and a surrogate modeling prediction tool 'SMUse'. Both can be used with data provided by the already available aerodynamic tools PyTornado and SU2.

In this paper we showed how these tools can be used and which parameters should be chosen. Three different applications of the surrogate model in CEASIOMpy were shown. The first one was a simple case in which the only design variable was the angle of attack. We showed that with only five input points for the surrogate model (SMTrain) it was possible to get very accurate results.

In the second case, two design variables were used, the Mach number and the angle of attack. The results obtained with the surrogate model was compared to the "true" value calculated by SU2, and showed a reasonable agreement. But it, could still be improved by using a better sampling distribution.

The last case was a demonstration of the use of the surrogate model to predict aerodynamic coefficients with a modification of the angle of attack, angle of side slip and aircraft wing span. It showed that geometrical parameters can also be used to create a surrogate model, which makes this method very efficient when it is used in an optimisation process. We also observed that a larger number of training points leads to better predictions for complex cases.

Finally, more tests should be carried to find the best trade off between the number of training points and the accuracy of the surrogate model for more complex cases. With more experience on the use of these tools in CEASIOMpy, it has the potential to become a powerful tool that can be used in many different aircraft design optimisation studies.

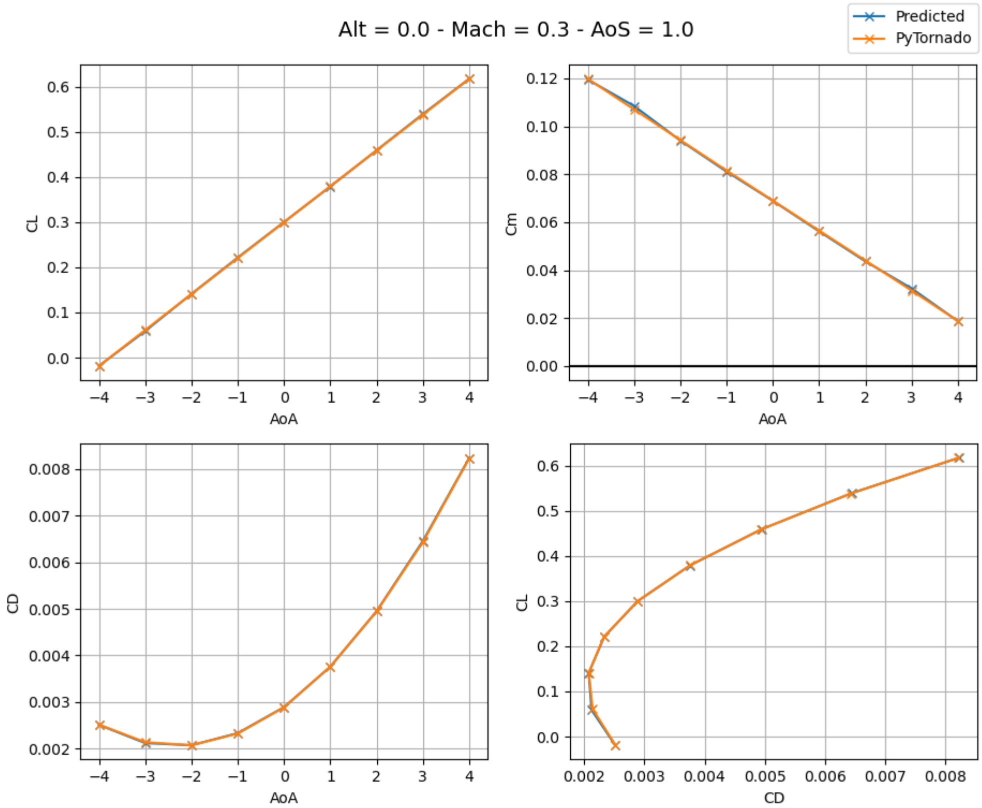


Figure 11: Comparison between predicted and calculated by PyTornado aerodynamic coefficients


# Acknowledgement

The work presented in this report has been performed in the framework of the AGILE 4.0 project (Towards Cyber-physical Collaborative Aircraft Development) and has received funding from the European Union Horizon 2020 Programme under grant agreement nr°815122.


# References


[1] M. Zhang, *Contribution to Variable Fidelity MDO Framework for Collaborative and Integrated Aircraft Design*. PhD thesis, KTH Royal Institue of Technology, 2015.

[2] DLR, "CPACS, common parametric aircraft configuration schema." "https://cpacs.de/". Accessed: 24.06.2020.

[3] CFSEngineering, "CEASIOMpy documentation." "https://ceasiompy.readthedocs.io/en/latest/". Accessed: 21.07.2020.

[4] M. Zhang, N. Bartoli, A. Jungo, W. Lammen, E. Baalbergen, and M. Voskuijl, "Enhancing the handling qualities analysis by collaborative aerodynamics surrogate modelling and aero-data fusion," vol. 119, p. 100647.

[5] J.-N. Walther, A.-A. Gastaldi, R. Maierl, A. Jungo, and M. Zhang, "Integration aspects of the collaborative aero-structural design of an unmanned aerial vehicle," vol. 11, no. 1, pp. 217–227.

[6] P. D. Ciampa, P. S. Prakasha, F. Torrigiani, J.-N. Walther, T. Lefebvre, N. Bartoli, H. Timmermans, P. D. Vecchia, L. Stingo, D. Rajpal, I. v. Gent, G. L. Rocca, M. Fioriti, G. Cerino, R. Maierl, D. Charbonnier, A. Jungo, B. Aigner, K. Anisimov, A. Mirzoyan, and M. Voskuijl, "Streamlining cross-organizational aircraft development: Results from the AGILE project," in *AIAA Aviation 2019 Forum*, American Institute of Aeronautics and Astronautics. _eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2019-3454.

[7] "AGILE 4.0 – towards cyber-physical collaborative aircraft development." https://www.agile4.eu/. Accessed: 2020-09-29.

[8] C. Jacob, J. Bieler, and A. Bardenhagen, "Introducting surrogate models to the structural preliminary aircraft design phase," in *Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V.*, 2018.

[9] M. A. Bouhlel, J. T. Hwang, N. Bartoli, R. Lafage, J. Morlier, and J. R. R. A. Martins, "A python surrogate modeling framework with derivatives," *Advances in Engineering Software*, p. 102662, 2019.

[10] "Smt: Surrogate modeling toolbox." https://smt.readthedocs.io/en/latest/index.html. Accessed: 2020-07-14.

[11] *AeroStruct: Enable and Learn How to Integrate Flexibility in Design.* Springer International Publishing, Heinrich, R. ed.

[12] R. Maierl, A. Gastaldi, J.-N. Walther, and A. Jungo, "Aero-structural optimization of a MALE configuration in the AGILE MDO framework," in *Flexible Engineering Toward Green Aircraft* (M. E. Biancolini and U. Cella, eds.), Lecture Notes in Applied and Computational Mechanics, pp. 169–187, Springer International Publishing.

[13] DLR, "CPACSCreator documentation." "https://dlr-sc.github.io/tigl/doc/cpacscreator-0.1/index.html". (accessed: 23.07.2020).

[14] Airinnova, "Pytornado documentation." "https://pytornado.readthedocs.io/en/latest/". (accessed: 23.07.2020).

[15] M. Tomac and D. Eller, "From geometry to cfd grids—an automated approach for conceptual design," *Progress in Aerospace Sciences*, vol. 47, no. 8, pp. 589 – 596, 2011. Special Issue - Modeling and Simulating Aircraft Stability and Control.

[16] S. Hang, "TetGen: A quality tetrahedral mesh generator and a 3d delaunay triangulator."

[17] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso, "Su2: An open-source suite for multiphysics simulation and design," *Aiaa Journal*, vol. 54, no. 3, pp. 828–846, 2016.

[18] G. W. H. van Es, "Rapid estimation of the zero-lift drag coefficient of transport aircraft," *Journal of Aircraft*, vol. 39, no. 4, pp. 597–599.

[19] P. Ciampa, B. Nagel, and G. L. Rocca, "A mbse approach to mdao systems for the development of complex products," vol. 2020-3150, (Virtual), AIAA Aviation, 2020.

[20] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor, "OpenMDAO: An Open-Source Framework for Multidisciplinary Design, Analysis, and Optimization," *Structural and Multidisciplinary Optimization*, vol. 59, pp. 1075–1104, 2019.