Multifidelity Aeroelastic Optimization

Development of a low fidelity structural model in a multifidelity efficient global optimization framework.

Giovanni QUAGLIA

March 2020

Abstract

The reduction of computational time during the aircraft design process is a major issue with direct industrial applications [18]. In the last two decades a strategy named multifidelity has offered significant improvements to the already existing surrogate model-based optimization technique. The innovative approach is based on the combination of sources with different accuracy in order to accelerate the analysis. The research project is part of a joint effort between the mechanical department of university ISAE Supaero and the ONERA, The French Aerospace Lab [2]. The scope is the development of a more efficient framework able to design an optimized blended wing body aircraft. This paper presents the theoretical arguments related to the research and focuses on the improvements obtained during the project. In particular, the latter consist in the design and validation of a low fidelity structural model. The main topics treated are multifidelity, surrogate models, in particular Kriging and Co-Kriging interpolation, and more general optimization schemes such as Efficient Global Optimization and its enhancements.

I Introduction

In the last century, complex mathematical models have been developed in order to simulate real systems in many areas of scientific research. The main obstacle to an intensive exploitation of such models, in the process of visualization and analysis, is the computational cost which can easily become prohibitive even for simple cases [12]. This problem is more crucial in the field of aerostructural design and optimization since many simulations are required and one single Navier-Stokes CFD run can demand hours of CPU time. The challenge has been tackled by analyzing surrogate models, sometimes also referred to as meta-models, which are an approximation of the actual system based on a number of collected samples [10]. These techniques have their origin in many scientific fields, from geology and statistics to engineering. Some examples are Kriging, Radial Basis Function (RBF) and Neural Network. Kriging, in particular, is an interpolation method that uses a Gaussian process as reference [7]. Different variants have been efficiently applied to aeroelastic optimization. Nevertheless, the actual cost of sampling would still be considerable, unless a trade-off between time and accuracy is considered. In recent years, multifidelity (MF) has been proposed as a valuable strategy to reduce computational time. Under such approach, the results employed to train and enrich the surrogate model derive from sources of different precision and are balanced in order to reduce the overall cost. These sources can be seen as black-box functions [11] and are connected in an optimization environment.

In the last decade ONERA, The French Aerospace Lab, in a joint effort with the university ISAE-Supaero, has developed the SEGOMOE (Super Efficient Global Optimization with the Mixture Of Experts) framework [2]. It is a single-fidelity, surrogate-model based, Bayesian optimization tool, able to deal with high-dimensional problems with a large number of constraints. As a further progress, Bartoli et al. (2019) [3] exploited this technique as a benchmark to demonstrate the improvements offered by the use of a multi-fidelity approach in terms of computational time. In the research, a subsonic airfoil with 15 variables has been designed using MFEGO (Multi-Fidelity Efficient Global Optimization). Finally, Mas-Colomer (2018) [13] produced an aeroelastic black-box function implemented in the OpenMDAO framework [16], which optimizes the NASA CRM (Common Research Model) wing using a computationally cheap aerodynamic and structural solvers.

The next step in this field of research aims to successfully optimize the CRM wing inside the SEGOMOE framework enhanced by the use of multifidelity. On one side, concerning the aerodynamics, the two levels of accuracy are currently represented by a CFD and a panels codes. On the other, the subject of this article is the development of a low fidelity version of the structural model. It features a reduced mesh resolution, a shorter analysis time but a substantial correlation with respect to the original high fidelity model. In particular, a MATLAB tool has been initially designed in order to produce a FEM mesh with a level of complexity fixed by the user. Later the model properties have been assessed through different tests.

In section II the different theoretical arguments are discussed, along with their mathematical formulation, to provide the general framework. Section III presents the simplifications made on the original CRM model and how the low fidelity structural mesh is written. In IV the static and dynamic behaviours are validated and the model is tested on a simple multifidelity optimization. Lastly, section V reports the conclusions and lists the future objectives.

II Theoretical framework

1 The aerostructural problem

During the design phase of an aircraft the focus is the enhancement of multiple objective functions, namely the QoI (Quantities of Interest), such as drag or fuel consumption under mass and maximum stress constraints. In the case of steady cruise flight, the QoI are computed as result of a static aeroelastic problem. In literature, two approaches to solve it can be found, the monolithic and the partitioned. The latter is the most applied nowadays and it consists in the coupling of two different solvers for the structural and fluid problems, such as a FEM and a CFD software. The theory is reported with the notation used in Scholcz (2015) [18].

$$\mathbf{y} = \mathcal{F}(\mathbf{x}), \qquad \mathbf{x} = \mathscr{S}(\mathbf{y}) \tag{1}$$

Here, $\mathbf{x}$ and $\mathbf{y}$ are the vectors containing the structural and aerodynamic variables, namely the nodes displacements and the fluid pressure along the airfoil. $\mathscr{S}$ and $\mathcal{F}$ are the structural and fluid problems. The coupling can be expressed as:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in I\!R^{N_s}} \mid\mid \mathcal{R}(\mathbf{x}) \mid\mid \tag{2}$$

$$with \quad \mathcal{R}(\mathbf{x}) = \mathscr{S} \circ \mathcal{F}(\mathbf{x}) - \mathbf{x} \tag{3}$$

Once the two software required by the disciplines are chosen, they are wrapped in a framework able to process the outputs in order to be provided as inputs one to the other. Given a set of design variable, the solver iterates until a predefined convergence criterion is reached, and finally the QoI and the constraints values are extracted. Hence, the ensemble can be seen as a black-box function as shown in figure 1.

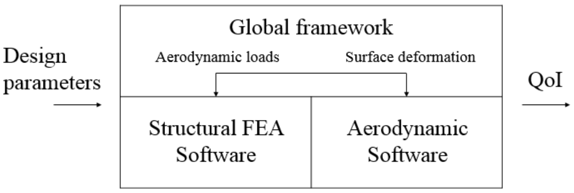


Figure 1: Black-box aeroelastic function diagram.

2 Surrogate models

In aeronautical engineering, relying on computationally expensive codes for the variable design space exploration is extremely time-consuming. The use of surrogate models is an attempt to circumvent this limit. Although one comprehensive definition is not possible, the main features are the dependency on some initial values of the expensive function, the ability to mimic its behaviour and being cheap to evaluate. A thorough description of the techniques available nowadays is found in Yondo et al. (2018) [20]. One main separation is made between data fit surrogate models and ROM (Reduced Order Model) methods. In the first case, the original system is approximated by a function interpolating some known values of the QoI, and multiple choices can be made regarding the type of fitting. Some examples are Polynomial Regressions, Kriging and Radial Basis Functions. On the other side, ROM's methods are based on the projection of the governing equations onto a low-dimensional subspace. The aim is to maintain the main characteristics (physical and dynamical) neglecting, in first approximation, irrelevant features. Since the SEGOMOE research implements data fit surrogate models, the focus here is given to the first group. In the following part it is first explained how to create a surrogate model through Kriging interpolation according to Jones (2001) [10], then the Co-Kriging is introduced.

2.1 Kriging

The method was invented by D.G. Krige in the field of mineral extraction and later developed by G. Matheron. Given $\mathbf{y}_i(\mathbf{x}_i)$, a set of n initial design of experiments, it defines the predictor that minimizes the expected squared prediction error subjected to: being unbiased and, being a linear function of the observed $\mathbf{y}_i$'s. The process is modeled as a mean base term μ plus random variable $\mathbf{Y}(\mathbf{x})$ with zero mean, variance σ^2 and correlation function of two realizations approximated as:

$$Corr[\mathbf{Y}(\mathbf{x}_i), \mathbf{Y}(\mathbf{x}_j)] = exp(-\sum_{l=1}^{d} \theta_l \mid \mathbf{x}_{il} - \mathbf{x}_{jl} \mid^{p_l}) \tag{4}$$

Figure 2 shows two different behaviours of the correlation function, which can be seen as a definition

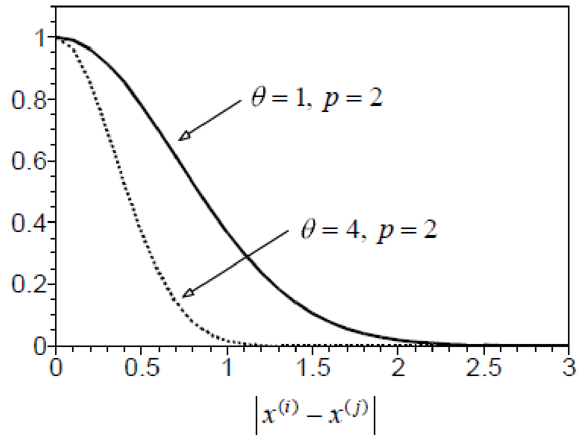


Figure 2: Examples of Kriging covariance function. Jones et al. (1998) [11].

of the distance between the two sets of variables $\mathbf{x}_i$ and $\mathbf{x}_j$. In fact when the two vectors are close, their covariance is large while it decreases if the experiments are distant. The θ_l and p_l values have a key role in defining how sharply and smoothly the contribution to the correlation drops for each variable $\mathbf{x}_l$. If even a small difference in the variable produces a large change in the function value it is then considered active and the resulting θ_l will be greater, causing a faster lowering of the covariance. p_l defines in the same manner the smoothness in the $\mathbf{x}_l$ direction.

The n initial function evaluations can be seen as realizations of the random variable and stored in a vector:

$$\mathbf{Y} = \begin{bmatrix} Y(x_1) \\ Y(x_2) \\ \vdots \\ Y(x_n) \end{bmatrix} \tag{5}$$

The covariance matrix is therefore defined as in (6), where $\mathbf{R}$ is an $n \times n$ matrix with (i, j) elements given by (4).

$$Cov(\mathbf{Y}(\mathbf{x})) = \sigma^2 \mathbf{R} \tag{6}$$

The parameters μ, σ^2, θ_l and p_l are tuned using the samples and maximizing the log-likelihood function (MLE), this is usually accomplished with a Genetic Algorithm (GA) but the size of this problem can easily become demanding. As a consequence two possible choices are avoiding the update of the parameters when a new evaluation is used to train the model or fixing $p = 2$ to reduce the problem size [7]. Finally the Kriging predictor (7) can be recovered by adding an unknown additional sample and maximizing again the augmented likelihood function which is only dependent from $\hat{\mathbf{y}}$.

$$\hat{\mathbf{y}}(\mathbf{x}) = \hat{\mu} + \mathbf{r}^{'}\mathbf{R}^{-1}(\mathbf{y} - 1\hat{\mu}) \tag{7}$$

$\mathbf{r}$ and $\mathbf{R}$ contain the covariances of the samples and are functions of the parameters.

This method is extremely useful because it provides a prediction of the function in a new point in the design space along with its uncertainty based on the Gaussian process used. The possible deviation is zero in the already known points and increases moving away from them as it can be seen in figure 3.

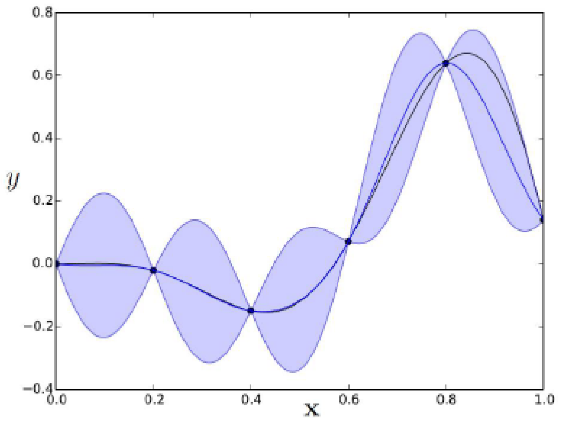


Figure 3: Example of Kriging interpolation. Bartoli et al. (2016) [2].

2.2 Co-Kriging

Co-Kriging [7], is an extension of this method in the case when values come from two sources, the firsts $\mathbf{y}_e$, where the **e** stands for expensive, more precise, called HF (high fidelity), and the other less accurate $\mathbf{y}_c$, **c** for cheap, named LF (low fidelity). The first assumption made is that high fidelity computed values are exacted, hence in those points the error must be zero. This is reflected in property (8), meaning that no more can be learnt from the cheap code when the value is known in $\mathbf{x}_i$.

$$Cov\{\mathbf{Y_e}(\mathbf{x}_i), \mathbf{Y_c}(\mathbf{x})|\mathbf{Y_c}(\mathbf{x}_i\} = 0, \quad \forall \mathbf{x} \neq \mathbf{x}_i \tag{8}$$

The model builds the predictor in the same fashion as before, full derivation of equations is given in Jones (2001) [10]. Both the expensive and cheap functions are interpolated with a Gaussian process but the first one is redefined with an auto-regressive method as the cheap model, scaled by ρ, plus the Gaussian process $\mathbf{D}$, which represents the difference between the two.

$$\mathbf{Y}_e = \rho \mathbf{Y}_c + \mathbf{D} \tag{9}$$

The procedure consists in stacking the expensive and cheap values into an unique array as before (5). The definition of the correlation matrix becomes more dense with respect to Kriging, containing also the cross correlation between fidelity.

After substituting $\mathbf{Y}_e$ with its redefinition, the parameters to be computed through MLE are twice as before (μ, σ^2, θ_l and p_l both for the cheap and difference processes) plus ρ. Since the cheap data are considered independent from the expensive ones, it is possible to maximize the log-likelihood of the covariance matrix using a GA, obtaining the expression for μ_c, σ_c^2,θ_{cl} and p_{cl}. Finally one can write $\mathbf{d}$ and maximize also its likelihood function.

$$\mathbf{d} = \mathbf{y}_e - \rho \mathbf{y}_c(\mathbf{x}_e) \tag{10}$$

If the values $\mathbf{y}_c(\mathbf{x}_e)$ are unknown, they can be computed using the cheap model already estimated without excessively increasing the overall cost. Finally, the predicted value for $\hat{\mathbf{y}}(\mathbf{x})$ can be expressed as in (7) where $\mathbf{r}$ and $\mathbf{R}$ now contain the expanded covariance. Figure 4 compares the two interpolation method presented, showing the improvements offered by the use of Co-Kriging and multifidelity data with respect to normal Kriging.

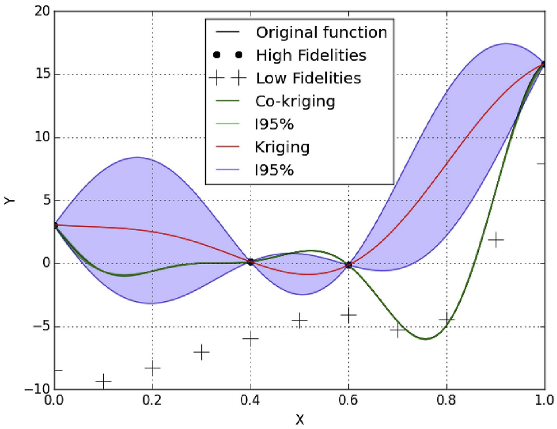


Figure 4: Example of Co-Kriging interpolation. Yondo et al. (2018) [20].

This meta-model is crucial in allowing the implementation of multifidelity in the optimization process, as it is shown in the next section. Literature provides many improvements to tackle problems such as: noisy data [7], parameters tuning time and ill conditioned functions [11], offering a variety of meta models to be used in optimization environments [19] [1]. Finally, Kennedy and O'Hagan (2000) [12] propose a more general Bayesian derivation, valid for an increased number of fidelity, of which Co-Kriging is just a reduction.

3 Multifidelity

The concept of multifidelity has already been introduced in the previous sections. When it is not possible to have more than few high-fidelity samples, the use of a cheaper solver allows to explore the variable design space. The large number of LF simulations permits to enrich the surrogate model and to address the optimum search. Moreover, the space of possible sources is huge. Samples used to train the model could come from experiments, software of different levels of modelling, such as CFD or panels codes in aerodynamics, a coarse and a fine mesh on the same problem or even a change in the convergence criteria for an iterative algorithm [18]. Multifidelity is efficient if the accuracy of the optimum found is maintained while saving computational time performing less HF simulations. The main property required is the correlation between the results. The LF model must evolve in accordance with the HF one to correctly predict the behaviour of the objective function. Furthermore, the cheap code should be one or more orders of magnitude faster, allowing consistent scaling and time saving.

The theoretical background regarding MF methods is still ambiguous. While the results in terms of computational time are generally positive, it has been proven that the use of LF data can worsen the surrogate precision [8]. Fernandez et al. (2019) [6] delve into over 100 papers which make use of a MF approach. The research compared the studies in terms of levels of fidelity, disciplines, types of solvers, and data of correlation and time savings when made available. The conclusion is the absence of a clear pattern, able to determine whether or not it is convenient an MF implementation, and the lack of research in this direction.

4 Surrogate-based optimization

4.1 Design of Experiments techniques

The optimization process requires an initial metamodel in order to begin the minimum research. The computation of the first samples is called DoEs (Design of Experiments) and the theory dates back to the beginning of the 20th century when it was developed to plan and conduct physical experiments. The difficulties of noise and bias are no longer present in modern numerical simulations and the objective of the research has shifted to maximize the amount of information regarding the design space minimizing the number of samples. DoEs techniques can be grouped in Classical (CDoE) or Modern (MDoE) [20]. Two issues common to both groups are the choice of the number of samples and whether to perform the analysis offline or online. In the first case the experimental campaign is predefined while in the second case the additional sample is determined by the previous. In this sense a trade-off between accuracy and computational cost is always needed.

CDoE designs were studied first. They assume a general non-repeatability of measurements and search for space-filling by mean of a uniform distribution of the training points. In spite of being more appropriate for physical experiments, they are here cited since MF techniques can in principle include such data. Some examples are grid designs, central composite designs and box-behnken designs.

MDoE were developed in parallel to the evolution of computer science and are characterized by deterministic samples excluding any source of inaccuracy due to randomness. These techniques generally place more points in the interior of the design space, with the aim of producing the surrogate model minimizing the bias error. As a result this class is considered less space filling with respect to CDoE but is usually less time-consuming and provides a better knowledge of the design space. A wide broad of methods exists, and they all share the attempt to perform a better exploration. A first classification includes Random designs such as: Monte Carlo methods, Projection-based designs, Statistical Measure-based designs and Miscellaneus designs. Latin Hypercubes (LHS) belongs to the last group and it is the most common choice in aerospace engineering. The model is the multidimension generalization of the Latin Square, a grid having only one sample on each row and column. Thus in an k variables space, each of them divided in N intervals, the Latin Hypercube Design (LHD) is a $(N \times k)$ matrix in which each column is a permutation of $\{1, 2, ..., N\}$. A 2D example is shown in figure 5, with $k = 2$ and $N = 20$. The SEGOMOE framework implements one of the many variants to this technique, the Enhanced Stochastic Evolutionary (ESE) algorithm, which chooses the points optimizing a distance criterion.

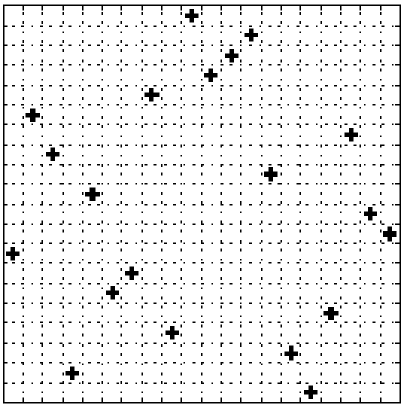
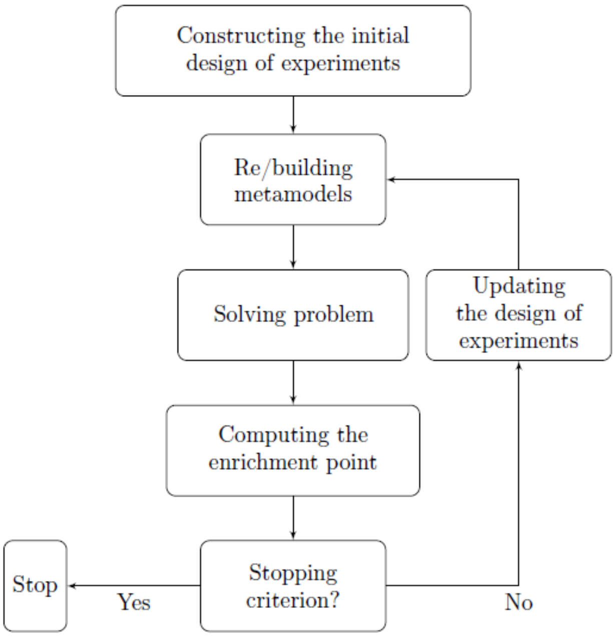


Figure 5: Example of Latin Hypercube Sampling. Forrester et al. (2007) [7].

4.2 Efficient Global Optimization (EGO)

Once obtained the initial surrogate model, it can be optimized in place of the original numerical system. The algorithm flowchart is well described in figure 6: the surface is updated each time a new sample is computed until a convergence criterion is met. This kind of process is characterized by the choice made on the next sample. A simple criterion could be investigating the lowest point of the interpolation, but obviously it will lead only to a more accurate estimate of the local minimum. Thus the approach would put too much emphasis on the exploitation of the predictor neglecting the exploration of the remaining design space. In order to balance the two aspects, the most common solution to this issue is adopting the Maximum Expected-Improvement (EI) criterion. This method takes in consideration the already found minimum and the uncertainty in all the interpolation surface, providing a trade-off between local and global search.



Constructing the initial design of experiments
Re/building metamodels
Solving problem
Updating the design of experiments
Computing the enrichment point
Stopping criterion?
Stop
Yes
No

Figure 6: Optimization Algorithm Flowchart. Bartoli et al. (2016) [2].

$$E[I(\mathbf{x})] = E[max(f_{min}, 0)] \qquad (11)$$

The Expected-Improvement (11) is a function that can be defined for statistical processes such as Kriging and Co-Kriging, and the optimizing problem shifts to find its maximum. The most common choices are derivative-free optmizers and a useful property is that in some cases [11] the EI has been proven monotone with respect to $\mathbf{y}$ and σ. Moreover, the advantages of an EI approach are two-fold: firstly, it avoids the definition of a target function as improvement, secondly, it provides an automatic stopping criterion based on a fixed lower threshold of its value. The algorithm under such formulation is usually referred in literature as Efficient Global Optimization (EGO). Figure 7 presents one step of the algorithm. As indicated, one local minimum is found in $x = 3$. Nevertheless, the high uncertainty in the region $x = [6 - 11]$, as visualized in the probability function, results in a peak in the EI at $x = 8$. In conclusion, this kind of algorithm has been proven to converge to a global maximum but with one possible disadvantage of taking a significant number of iterations to start searching globally when the initial interpolation is highly deceptive [10].

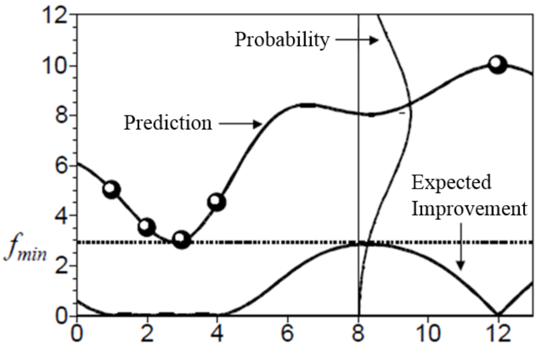


Figure 7: Expected-Improvement Optimization. Jones et al. (1998) [11]

4.3 Multifidelity EGO (MFEGO)

When multifidelity is applied to Efficient Global Optimization, the method is then called MFEGO and many examples can be found in literature. Both in Bartoli et al. (2019) [3] and Ariyarit and Kanazaki (2017) [1], the algorithm was successfully tested on the optimization of an airfoil shape subjected to multiple constraints.
The augmented framework requires two aeroelastic black-box functions, depicted before in section 1. Both must be able to compute the same quantities of interest but one, the LF, should be significantly faster to execute than the HF at the cost of a reduction in accuracy. Not only the two functions should have high correlation in their behaviour, it is also mandatory that the design variables have a correct representation in the models. For instance the LF, even if it is a simplified version of the more detailed original, it should preserve all the parameters that are optimized in the algorithm.
The addition of multiple fidelity impose two more criteria. Firstly, during the DoEs, a choice is made on whether a point is computed through HF or LF. The issue is strongly problem dependent, Forrester et al. (2007) [7] suggests a rule of thumbs, setting the number of cheap and expensive points at $n_c > 10k$ and $n_e < 10k$, k being the design variable vector dimension. On large problems, one common procedure is computing a dense LF surrogate adding few HF data in the points where the uncertainty is maximum, until a certain threshold. Secondly, the most interesting improvement regards the choice of t, the next fidelity level during the optimization loop. The standard strategy deals with this problem in a separate step, after selecting the next enrichment point x^*.

$$t = \arg \max_{k \in (0,\dots,l)} \frac{\sigma^2_{red}(k, x^*)}{cost_{total}(k)^2} \tag{12}$$

Being:

$$\sigma^2_{red}(k, x^*) = \sum_{i=0}^{k} \sigma^2_{d,i}(x^*) \prod_{j=1}^{l-1} \rho^2_j \tag{13}$$

$$cost_{total}(k) = \sum_{i=0}^{k} c_i \tag{14}$$

It is reported here the criterion (12) adopted in Bartoli et al. (2019) [3]. The formula is written for a system with a number of l fidelity and based on the ratio between $\sigma^2_{red}(k, x^*)$, the reduced variance in the next query point, and the square of the total computational cost of the solvers. One difficulty is the correct prediction of the cost value and scaling it to a unit of measurements comparable to the variance. Finally, one more feature in an MFEGO algorithm is employing only HF f_{min} in the EI criterion.

4.4 The SEGOMOE Framework

Super Efficient Global Optimization with Mixture Of Experts (SEGOMOE) is the name of the research currently developed in a joint effort between ONERA, The French Aerospace Lab and the mechanical department at university ISAE-Supaero [2]. The single fidelity algorithm that has been developed differs from the theory already presented in two aspects. The first straightforward improvement is the substitution of the EI criterion by the WB2 (15).

$$\text{WB2}(\mathbf{x}) = -\hat{y}(\mathbf{x}) + EI(\mathbf{x}) \tag{15}$$

The WB2 function is more local and smoother than the EI, so that it is easier to optimize and accelerates the convergence of the algorithm even to a local solution. This is very important for high dimensional problems with great uncertainties.
The application of the MOE technique is on the other side the key enhancement to the method. A surrogate model may be the overall best choice for one problem, but at the same time it could provide a bad fitting in some regions of the design space. The MOE algorithm clusters the DoEs depending on the function behaviour in each area and applies the best suited model for such group. The available options include Least Square Model (LS), Polynomial Regression Models (PA2 or PA3), Ordinary Kriging, RBF, Multifidelity Kriging (MFK) and Kriging Partial Least Square (KPLS). The last one is an approximation of Kriging in which the computation of the hyper-parameters is speed up by the use of Partial Least Square, thus projecting the unknowns on a smaller subspace in which more weight is given to the most active variables.
Finally, a global model is obtained from the clusters through a Gaussian Mixture Model (GMM) tuned with Expectation-Maximization. Two algorithms have been developed to choose the number and type of clusters but also to evaluate the posterior probability of each of them, which is then used in the GMM. The resulting model can both be smooth or a discontinuous approximation, depending on the formula introduced for the last quantity.

III Design and analysis of a low fidelity structural model

1 General framework

The focus in the current research is optimizing the NASA CRM wing in the SEGO framework enhanced by the use of multifidelity. First of all, the programming language of the SEGO optimization algorithm is python and, in this sense, the adoption of the OpenMDAO library results extremely useful. OpenMDAO is an open-source computing platform for multidisciplinary analysis and optimization. Its structure makes it well suited for wrapping the two disciplines solvers, creating an aeroelastic black-box function. In general OpenMDAO codes are written connecting components, which can be explicit or implicit, each of them representing a part of the problem. Every component has inputs and outputs which must all be connected in the main function. Once run, the solver iterates until also the implicit parts reach convergence.
One key part of this next step in the research is the OpenMDAO package *aerostructures* [13]. The framework is used for solving static and dynamic aeroelastic analysis and optimizations. The XDSM diagram 8 presents its functioning. Each block represents a component, one can find the two main disciplines which run at their interior Panair [4], an open-source aerodynamic panels code and Nastran95 [14], used for solving the structural problem. All the other boxes have linking scopes being the interpolation and the transfer of loads and displacements between the fluid and structural mesh.

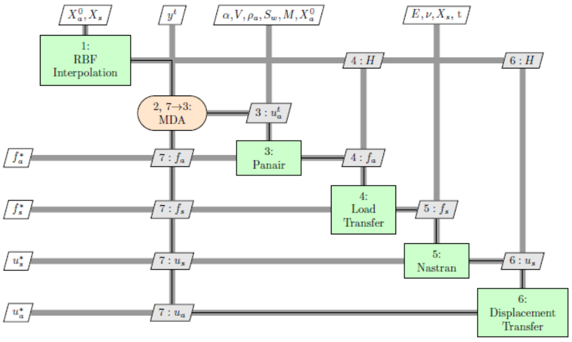


Figure 8: XDSM diagram of the static aeroelastic OpenMDAO LF solver. Mas-Colomer (2018) [13].

Taking advantage of this package, setting up the new optimization framework has been straightforward. The SEGO environment can easily encapsulate *aerostructures* and creates two versions of the main function. The first type is directly the same as before, while in the other case Panair has been substituted by a component calling the execution of the CFD software ADFlow. In this way two levels of fidelity are available.

While this first version is currently being tested, one more step was needed to improve the multifidelity approach. In fact, both functions had as an input the original CRM FEM model, which is given with a high resolution mesh, weakening the possible advantages of MF. Finally, the scope of this work has been twofold. Firstly, the accuracy of the structural model has been reduced to allow faster simulations, secondly, the preserved correlation with the initial system has been analyzed. Figure 9 illustrates how the framework should resemble once this tasks have been achieved.

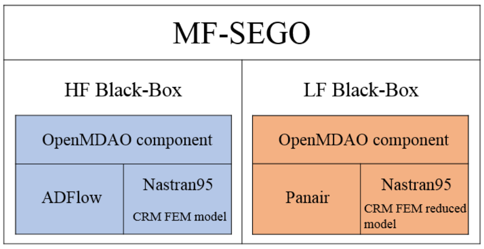


Figure 9: Diagram representing the current research framework.

2 The NASA CRM model

The Common Research Model is an attempt made by NASA to provide an experimental database for the validation of application of Computational Fluid Dynamics [15]. The related website is a repository of results from numerical and experimental analysis. In particular, some versions of different FEM models for the aircraft wing are released in Nastran language, one is drawn in figure 10. The one considered here has the following characteristics. The material is an aluminum alloy with $\rho = 2795.67$ and E = 68.9 GPa. The model consists of over 10500 nodes connected in 26600 elements. The structure is modelled as wing boxes delimited by 58 ribs closed by the upper and lower skins and with two spars at the leading and trailing edge. All this parts are explicitly represented by CQUAD4 elements depending on 12 PSHELL cards of different thickness. Moreover, ribs and skins are reinforced by stringers created with the CBAR class and with 6 different sections and principals of inertia. Some figures showing in details the various component are reported in appendices A.

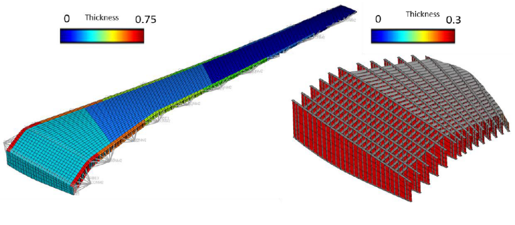


Figure 10: Illustration of the NASA CRM wing FEM model. NASA (2017) [15].

3 Design phase

The first task of the project consisted in coding a routine able to receive as input the nastran *.bdf* file related to the original model and producing as outputs the reduced list of nodes and the new mesh description. As an additional objective it was planned to parametrise the code in order to allow choosing the desired complexity of the model. The environment used for coding the tool has been MATLAB. An early difficulty has been faced when dealing with the NASA input file. The list containing the nodes was given with no apparent order in the IDs so that it was impossible to find a mathematical law able to define their succession. As a consequence the problem as been split in writing two separate functions to order the nodes and to write the new mesh.

MainCreatePlanes.m is the first routine. One initial assumption has been made neglecting the intermediate nodes between two ribs with the outcome of having the elements representing skins and spars as long as each section. The removal of the additional rows of nodes in the cord direction diminishes the overall model flexibility with respect to the rotation around the body axis. As a result of such hypothesis the function operates with the following procedure. First, it reads a sub-list of the stringers elements inside the ribs and takes the two extreme points defining the first one. After that, it searches for a third node as the closest among the remaining and computes the 4 geometrical parameters of a plane passing through them. Finally, the node list is scanned, selecting and removing the entries whose distance from the plane is less than a fixed tolerance. The picked nodes are saved in a structure referring to a particular rib, and the process is iterated on the reduced file. As a final step, the 58 sections are sorted root to tip and the nodes belonging to each of them are locally enumerated. The result is presented in figure 11 where the initial chaotic cloud of points has been ordered.

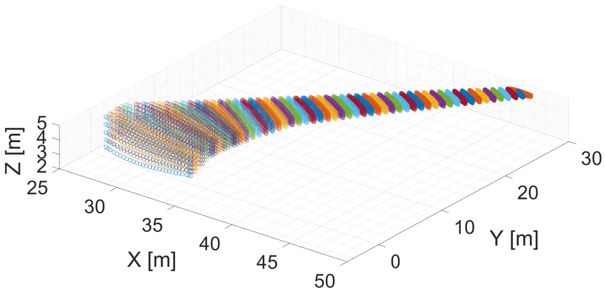


Figure 11: Matlab representation of the ordered nodes colored by rib section.

The second routine MainCreateMesh.m loads the structure containing the sorted nodes and writes a new coarser model. At the beginning of the function two main parameters, L and V, are defined, as they set the number of points in the reduced mesh on the vertical and cord directions. One first subroutine creates the list with the new nodes of the model. In order to accomplish this goal for each section the coordinates are transformed in a temporary reference frame so that the first two axes are in the rib plane and the third one is orthogonal to them. Then a spline interpolation is created for each row of nodes in the section, and it is evaluated in the number of points requested by L and V. The 57th rib has less nodes than the others and it is joint with the last one. Since the precise location of the intersection cannot be respected considering a coarser mesh, the function shifts all the points in the rib in order to place it as close as possible to the original.

Once created, the list with the new set of nodes is taken as database by a second sub-routine which has the scope of redefining all the elements as in the starting model. The last hypothesis regards the definition of the spars. In the original CRM model they are modelled explicitly by CQUAD4 elements in a I-beam. In the spirit of reducing the complexity of the system, the flanges were substituted with simple CROD elements as they are loaded only axially in first approximation. All the elements preserve the original properties, and in particular the V vectors are reassigned to the CBAR entries looking for the closest and identically oriented stringer. One realization of the LF structural model is presented in figure 12 for the choices of L = 4 and V = 3.

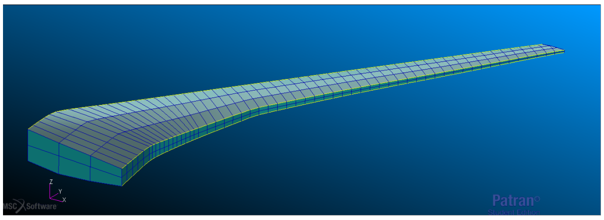


Figure 12: Low Fidelity structural model visualized in Patran for the choices of L = 4 and V = 3.

The last task during the design phase consisted in adapting the python *aerostructures* package to receive the same model with different mesh definitions. In fact the structure created by the MATLAB tool would have a decreased stiffness, due to the lower number of stringers, if the geometrical and inertial properties are kept as in the HF model. The problem was overcome providing to the main function the parameters L and V, the summation on all the stringer areas and principals of inertia as well as the size of the spar flanges. A new component is then added to the OpenMDAO group and it reassigns the properties according to the actual number of bars.

IV Model validation

1 Static analysis

In a preliminary phase of the validation process, all the simulations performed were static aeroelastic problems representing the aircraft in cruise condition. After computing the converged solution in the OpenMDAO framework, Gmsh has been used to visualize the deformation of the wing subject to the aerodynamic loads along with the Von Mises stress distribution. As an example, figure 13 compares the solution of the original model with the one obtained with the L = 4 low fidelity structure. The standard geometrical values for wing surface, wing span and sweep angle were taken from the original NASA model [15], while the case parameters are reported in table 1. The QoI selected to be compared in the analysis have been the lift coefficient (Cl) and the induced drag (CDi) computed by Panair, and the tip vertical displacement (δ_{tip}) and the maximum Von Mises stress (VM$_{max}$) that come from the postprocessing of the Nastran95 output file.

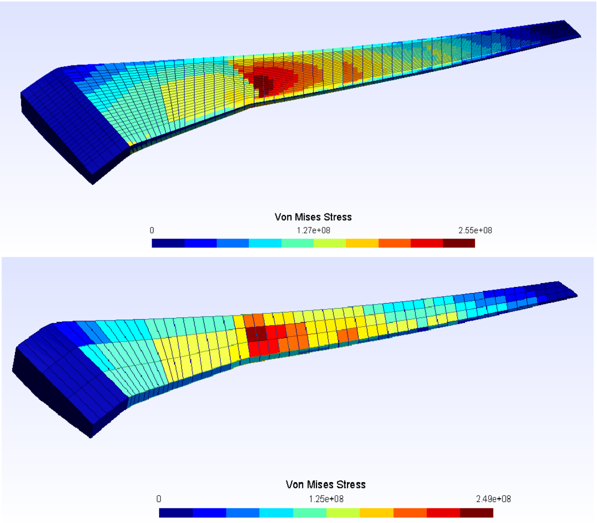


Figure 13: Gmsh visualizations of the converged solutions of the CRM wing for the HF case (up) and the L = 4 LF case (down).

Table 1: Aircraft cruise parameters

Name	Value
Flight speed	252.16168 [m/s]
Mach number	0.85
Air density	0.380580 [kg/m^3]
Wing incidence angle	1.34 [°]

1.1 Mesh resolution influence

As a first step, the influence of the mesh resolution has been studied. The MATLAB tool designed was used to produce different FEM models for the parameter L varying in the interval [4-28]. The lower bound allows to have at least 2 stringers running along the wing span, while the upper bound reaches the same value of nodes in the cord direction as for the original model.

The computational cost is the major interest of MF, to this purpose, two timers, measuring the setup and run times of the OpenMDAO routine, were added to the main function . Figure 14 shows the results normalized by the values obtained running the full model. The run time includes all the executions of Nastran95, hence it presents the greatest reduction. As expected, the line grows with a power trend due to the increasing number of elements for each node added. The key outcome is the ratio between the reference and the fastest simulation time (16), computed with L = 4. This is the maximum gain obtained and it can be called Fidelity Cost Ratio (FCR) and employed as an indicator, for example in the criterion for the next point fidelity.

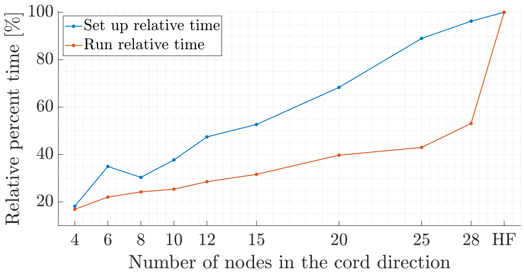


Figure 14: Setup and execution normalized timers as a function of the L parameter.

$$FCR = \frac{T_{run}^{L4}}{T_{run}^{HF}} = 16.9\% \qquad (16)$$

Figure 15 shows the computed values of Cl depending on the L parameter along with the reference produced by the original model. The graphs for all the QoI are reported in appendices B. The LF results have a general convergent trend, as the structure representation becomes more and more refined with the mesh, proving the coherent design of the model. The offset between the final values at L = 28 and the HF solution is the trade-off of a MF approach. It is caused by the simplifications assumed during the design phase, namely the removal of the rib intermediate nodes, the redefinition of spars and the shifting of the next to last rib.

1.2 Multifidelity correlation analysis

Once the MATLAB tool has been validated and the FCR computed, it was possible to focus on the correlation analysis between the LF model with L = 4 and the HF original one. It is hopeful that the LF maintains the same offset found in the previous

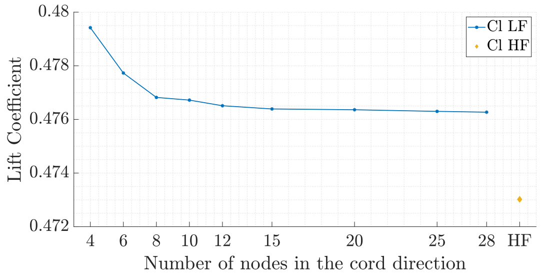


Figure 15: Lift coefficient as a function of the L parameter with reference HF value.

section even for other design variables inputs. For this reason maximum and minimum relative variances must be always checked during the design space exploration. Moreover, the most important indicator is the correlation between the two fidelity, since it states if the LF can provide true information on the HF behaviour. This property has been tested in two cases, one with a random approach and one with predefined parameter steps. Appendices C contains all the modified design variables and results.

In a first experiment the influence of a physical property of the structure has been analyzed. The vector containing the thickness of the 12 different PSHELL cards has been randomly modified changing its components independently inside the ±40% range. 10 different realizations of the properties have been considered, the 10th being the original one. The two fidelity have been tested on each configuration. The first outcome has been the representation of the four QoI values as a function of the scalar product between the modified and original thickness vectors normalized by themselves. This index is equal to 1 when the parameters are unchanged and it decreases expressing an overall distance in the variable design space. Figure 16 presents the two fidelity results for δ_{tip} and as well as the other graphs, it shows a perfect accordance in the results.

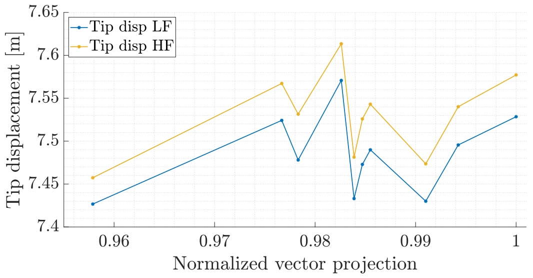


Figure 16: Tip vertical displacement as a function of the normalized projection of the shell elements thickness vectors onto the standard CRM property.

Moreover, the computation of the actual Pearson's correlation coefficient (PCC) between the HF and LF QoI validates the trend. Table 2 reports its values along with the relative percent variances with respect to the HF results.

Table 2: Comparative indices between HF and LF for different QoI, panels thickness case.

Indices:	**Cl**	**CDi**	δ_{tip}	**VM**$_{max}$
PCC	0.9995	0.9987	0.9903	0.9849
Δ_{max}	1.365%	1.693%	0.704%	3.477%
Δ_{min}	1.126%	0.790%	0.404%	0.219%

The second experiment has focused on the effect of two geometrical properties, the wing span (b) and the sweep angle (φ). Their standard values were b = 58.7629 [m] and $\varphi = 37.16°$, they were updated two times in both direction by a $\Delta b = 5$ [m] and $\Delta\varphi = 2.5°$ step, resulting in 25 configuration cases. The main function was modified by the use of the *aerostructures* components able to deform the mesh according to a variation in those parameters. Figure 17 shows three realizations of the wing in the software Gmsh, the original and the two extreme cases.

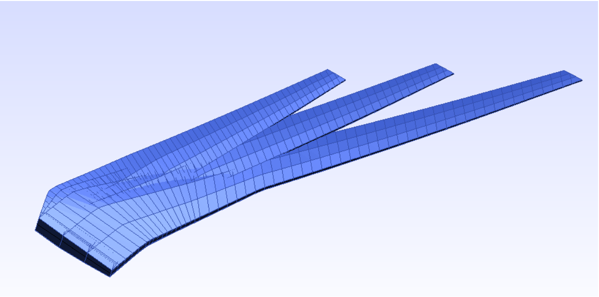


Figure 17: Gmsh visualization of the standard wing and of the two extreme configurations tested.

In this second test the QoI has been represented on the Z axis of a 3D graph with the design variables on the XY plane. Figure 18 shows the surfaces for the VM$_{max}$ quantity. Once more the LF is able to correctly predict the behaviour of the HF. All the figures present a partial crossing of the surfaces for high values of the b parameter. Such long-span configuration could present some nonlinearity badly predicted by the aeroelastic function and should be object of further analysis. Finally the comparative indices, obtained from the matrices containing the QoI values, are reported in table 3.

2 Dynamic analysis

The second phase of the validation has concerned the dynamic behaviour of the LF model. The objective has been the assessment of the correlation between the frequency response of the simplified

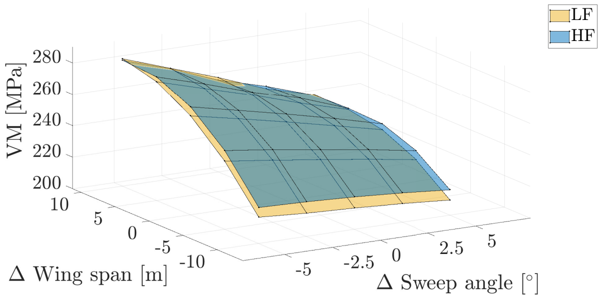


Figure 18: Maximum Von Mises stress as a function of the wing span and the sweep angle.

Table 3: Comparative indices between HF and LF for different QoI, 2 variables case.

Indices:	**Cl**	**CDi**	δ_{tip}	$\mathbf{VM}_{max}$
PCC	0.9998	0.9982	0.9997	0.9945
Δ_{max}	1.694%	1.547%	0.884%	2.841%
Δ_{min}	0.354%	0.119%	0.015%	0.118%

structure and the original one. Nastran95 provides a solution for normal mode extraction even if it does not offer the same advanced options as in commercial releases. For example the standard Lanczos method is not available and thus a basic inverse algorithm has been chosen for all the simulations. One common performance indicator for a dynamic comparison is the Modal Assurance Criterion (MAC) [17]. The Modal Assurance Criterion is a statistical indicator that is most sensitive to large differences and relatively insensitive to small differences in the mode shapes. When comparing two sets of eigenvectors in the same space, MAC indices are usually stacked in a matrix where a maximum value of 1 states full correlation, 0 being no correlation. Thus, the MAC matrix is calculated as the normalized scalar products of the two sets of vectors Φ_{HF} and Φ_{LF} as in (17).

$$MAC(i,j) = \frac{\left\|\Phi_{HF_i}^{\,T}\Phi_{LF_j}\right\|^2}{(\Phi_{HF_i}^{\,T}\Phi_{HFi})(\Phi_{LF_j}^{\,T}\Phi_{LFj})} \qquad (17)$$

Mode shapes must belong to the same space, therefore the HF ones are projected, through an interpolation, onto the LF subspace. Moreover, a good practice during the comparison is to apply a mode tracking algorithm able to find the most correlated modes regardless of their extraction order. Finally, this correlation criterion takes into account only eigenvectors while frequencies must be separately compared on a relative error basis.
The first result presented in table 4 is the FCR between a 10 modes extraction for the two fidelity.

HF extraction	**LF extraction**	**FCR**
840.53 s	21.44 s	2.55 %

Table 4: Computational cost of modal analysis.

The LF model gives a sharp decrease in the computational time while the FCR value can be use as a cost index in the multifidelity optimization.
Following the static validation, the fidelity correlation has been tested on the b and φ design variables with the same original values and variations. Since mode shapes might shift in order from one model to the other, the code extracts 10 eigenvectors from the HF and 30 from the LF, using mode tracking to find the best matches. From each analysis it is possible to compute the MAC matrix and a table of relative errors on the first 10 modes. Figure 19 and table 5 show the results for the CRM original values of the variables while more cases are reported in D. Models are proven to have great dynamic correlation, in particular no mode tracking is needed since LF frequencies are ordered and the relative error is almost zero for the first three modes.

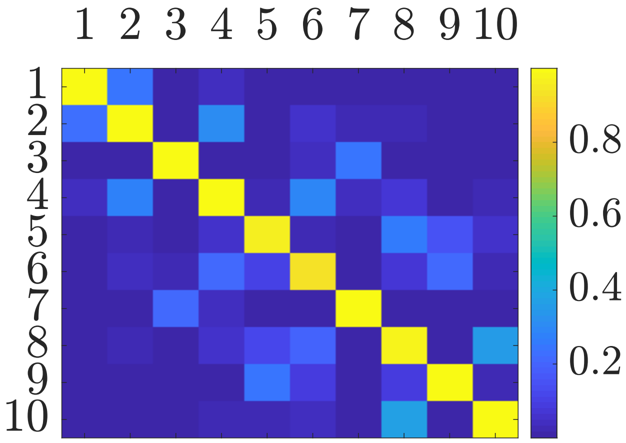


Figure 19: MAC matrix for the original values of b and φ.

HF	**LF**	**Relative error**
1.9320	1.9398	-0.400
5.7476	5.7409	0.117
7.9694	7.9275	0.525
11.650	11.402	2.126
17.672	16.227	8.172
19.002	18.413	3.102
22.289	22.067	0.992
26.817	26.038	2.906
31.016	28.620	7.725
35.742	34.826	2.561

Table 5: Frequencies [Hz] and relative error.

3 Multifidelity test case application

The final part of the validation process has consisted in a simple unconstrained optimization using the MFEGO algorithm. Both the static and dynamic HF behaviours are well represented by the LF model but the latter has been chosen as analysis type. The objective function to be minimize is the inverse of the distance between the first two frequencies, b and φ being taken as design variables in accordance with the tests made. Thus the problem is defined as in (18).

$$\begin{array}{rl} \min\limits_{b,\varphi} & \dfrac{1}{f_2 - f_1} \\ \text{for} & b \in [40m, 70m] \\ & \varphi \in [0°, 30°] \end{array} \tag{18}$$

The MFEGO algorithm requires two black box functions, namely the LF and HF codes for normal modes extraction, the number of points used to initially train the Co-Kriging surrogate, 7 for each fidelity, and a cost evaluation. Based on the previous section the last vector is set to [0.01,1] and a budget of 15 is chosen. Each call of a function evaluation counts for a cost against the budget.
Figure 20 shows the initial LHS sampling of the design space. The same number of LF and HF points is chosen to train the first surrogate model.

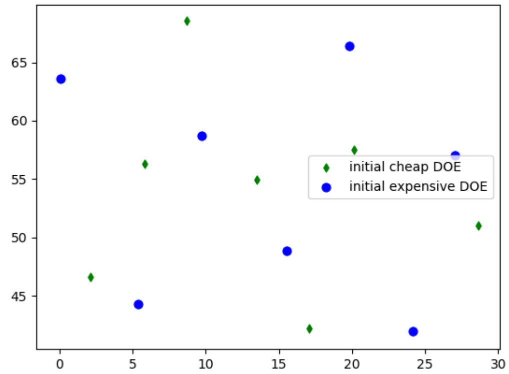


Figure 20: Initial LHS sampling in the design space.

Once the initial meta-model is obtained, the EGO algorithm successfully balances the exploration and exploitation of the design space choosing for each enrichment point the level of fidelity to be used. The final landscape is presented in figure 21 along with the location of the last point considered as optimum. Table 6 summarises the analysis results.

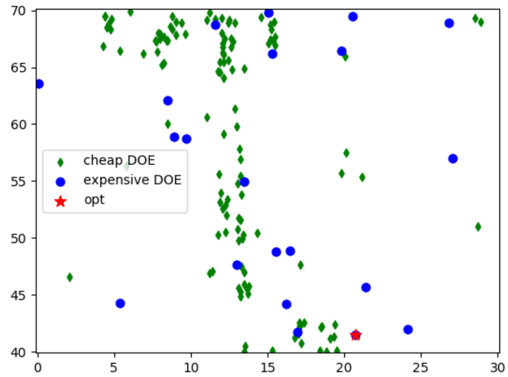


Figure 21: Design space after the optimization.

LF calls	127
HF call	21
Total iterations	134
b solution	41.53 m
φ solution	20.69°
objective function	3.393786858646402e-05

Table 6: MFEGO test results.

The simple test performed verifies the possibility to use the new LF in a multifidelity environment. The imposed cost ratio leads the algorithm to exploit three regions using many calls of the cheap code while the HF points are fewer and better spread. Nevertheless, the number of calls to the HF model is still consistent and a further reduction of the computational cost ratio would be an interesting development.

V Conclusions

The objective of the research project has been in large part achieved. The MATLAB tool designed is able to produce a low fidelity version of the original CRM model. Increasing the mesh resolution by use of the L and V parameters does not lead to a closer representation of the high fidelity. Nevertheless it allows to verify the tool accuracy and the scaling of the computational cost. One first possible improvement is the exploration of even more simplified LF structures obtained, for example, considering less ribs than the standard number.
Moving on to the validation phase, the first static and dynamic results have been great in terms of precision and correlation with respect to the HF model. More variable configurations could be simulated to increase the confidence in the correlated behaviour.
The test case has shown the interest in even more extremely simplified fidelity levels. During the last months, various attempts have been made in order to define the procedure for the reduction to a beam stick model of the CRM wing. Among the methods

tested, both applying unitary loads to each wing box for stiffness matrix extraction, as in [5],[9], and the analytical computation through wing structure theory, have failed. The resultant beam models are too stiff or flexible. Thus a it is still not possible to introduce a third level of fidelity.

Finally, the next step in this research would be implementing the low fidelity structural models in the optimization framework. As a first try it is still possible to adopt Panair as the aerodynamic software for both functions to create an initial fast comparison. Later the reduced model can be inserted in the MF SEGO analysis, as depicted in diagram 9, in order to study the real trade-off between the reduction in the computational cost and the lost in accuracy. These tests will produce important data regarding the actual benefits of the MF approach and how far the simplification of the LF model could be pushed.

References

[1] Ariyarit, A. and Kanazaki, M. (2017). Multi-fidelity multi-objective efficient global optimization applied to airfoil design problems. *Applied Sciences*, 17.

[2] Bartoli, N., Kurek, I., Lafage, R., Lefebvre, T., Priem, R., Bouhlel, M. A., Morlier, J., Stilz, V., and Regis, R. (2016). Improvement of efficient global optimization with mixture of experts: methodology developments and preliminary results in aircraft wing design.

[3] Bartoli, N., Meliani, M., Morlier, J., Lefebvre, T., Bouhlel, M.-A., and Martins, J. (2019). *Multi-fidelity efficient global optimization: Methodology and application to airfoil shape design.*

[4] Boeing (1992). *A502I User's Guide-PAN AIR Technology Program for Solving Potential Flow about Arbitrary Configurations.*

[5] Cirillo, R. (2011). Detailed and condensed finite element models for dynamic analysis of a business jet aircraft.

[6] Fernandez, G., Park, C., Kim, N., and Haftka, R. (2019). Issues in deciding whether to use multifidelity surrogates. *AIAA Journal*, pages 1–16.

[7] Forrester, A. I., Sóbester, A., and Keane, A. J. (2007). Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2088):3251–3269.

[8] Guo, Z., Song, L., Park, C., Li, J., and Haftka, R. T. (2018). Analysis of dataset selection for multi-fidelity surrogates for a turbine problem. *Structural and Multidisciplinary Optimization*, 57(6):2127–2142.

[9] Iacono, A. (2018). Development of a code for aeroelastic optimization of wings with stress constraints aggregation.

[10] Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383.

[11] Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492.

[12] Kennedy, M. C. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13.

[13] Mas-Colomer, J. (2018). *Aeroelastic Similarity of a Flight Demonstrator via Multidisciplinary Optimization.* PhD thesis.

[14] NASA (1980). *The Nastran User's Manual.*

[15] NASA (2017). Nasa common research model.

[16] openMDAO (2016). Openmdao source docs.

[17] Pástor, M., Binda, M., and Harčarik, T. (2012). Modal assurance criterion. *Procedia Engineering*, 48:543–548.

[18] Scholcz, T. (2015). Multi-fidelity methods for fluid-structure interaction and uncertainty quantification.

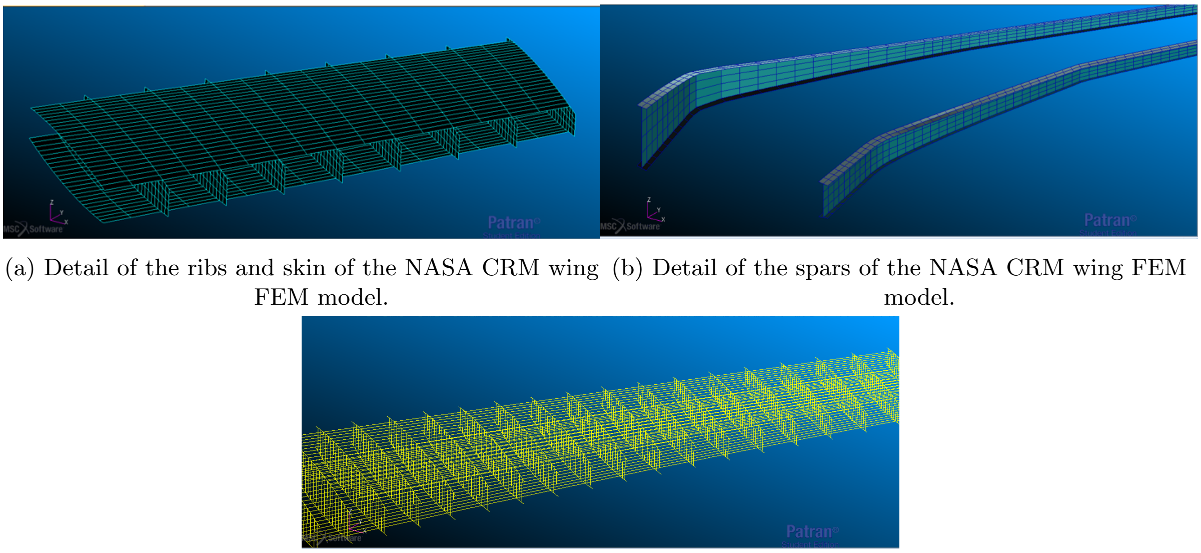
[19] Xiao, M., Zhang, G., Breitkopf, P., Villon, P., and Zhang, W. (2018). Extended co-kriging interpolation method based on multi-fidelity data. *Applied Mathematics and Computation*, 323:120–131.

[20] Yondo, R., Andrés, E., and Valero, E. (2018). A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses. *Progress in Aerospace Sciences*, 96:23 – 61.

Appendices

A

Pictures taken in Patran of the NASA CRM wing model components.



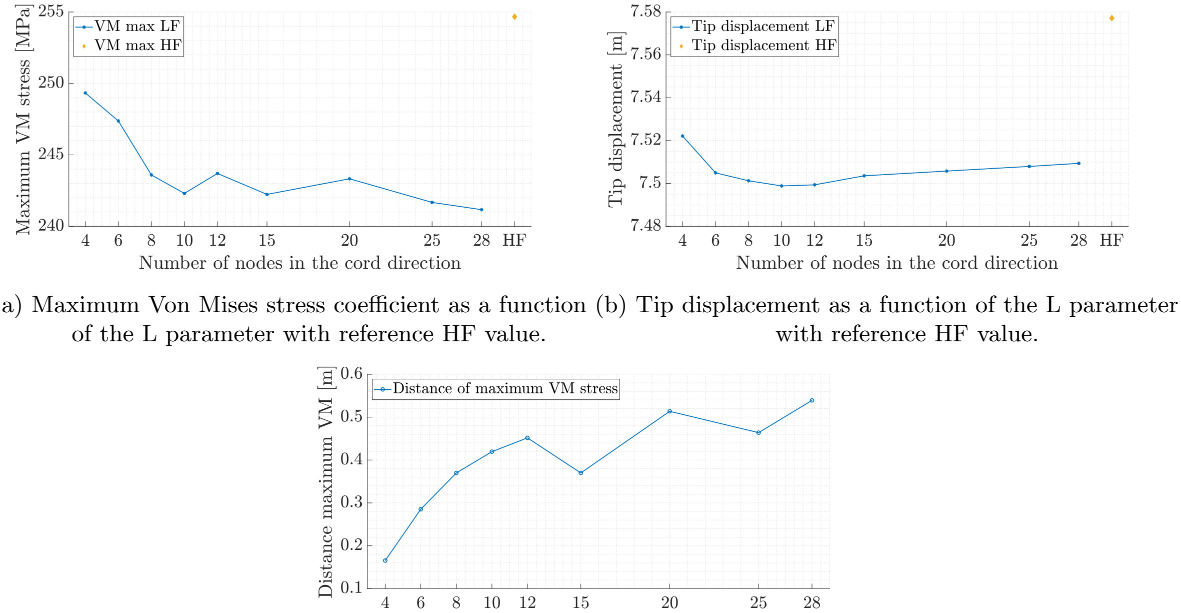
(a) Detail of the ribs and skin of the NASA CRM wing FEM model.

(b) Detail of the spars of the NASA CRM wing FEM model.

(c) Detail of the stringers of the NASA CRM wing FEM model.

B

Additional figures relative to the L parameter influence analysis.



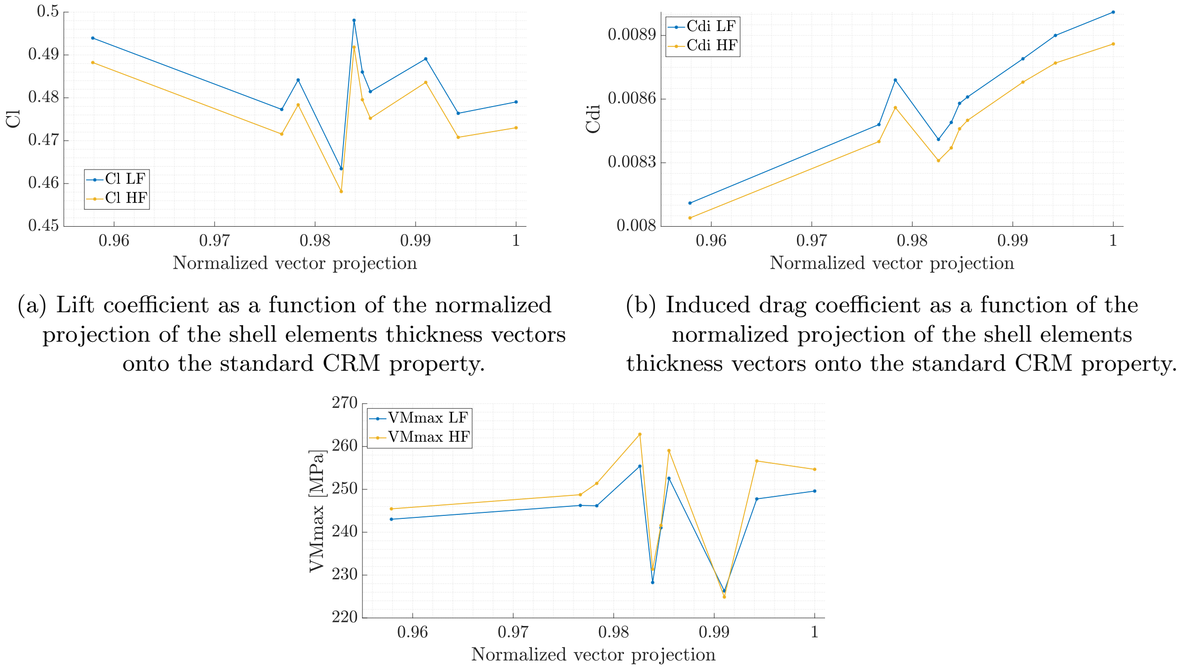
(a) Maximum Von Mises stress coefficient as a function of the L parameter with reference HF value.

(b) Tip displacement as a function of the L parameter with reference HF value.

(c) Distance of the location of the maximum Von Mises stress with respect to the HF solution, as a function of the L parameter.

C

Additional figures relative to the panels thickness influence study.

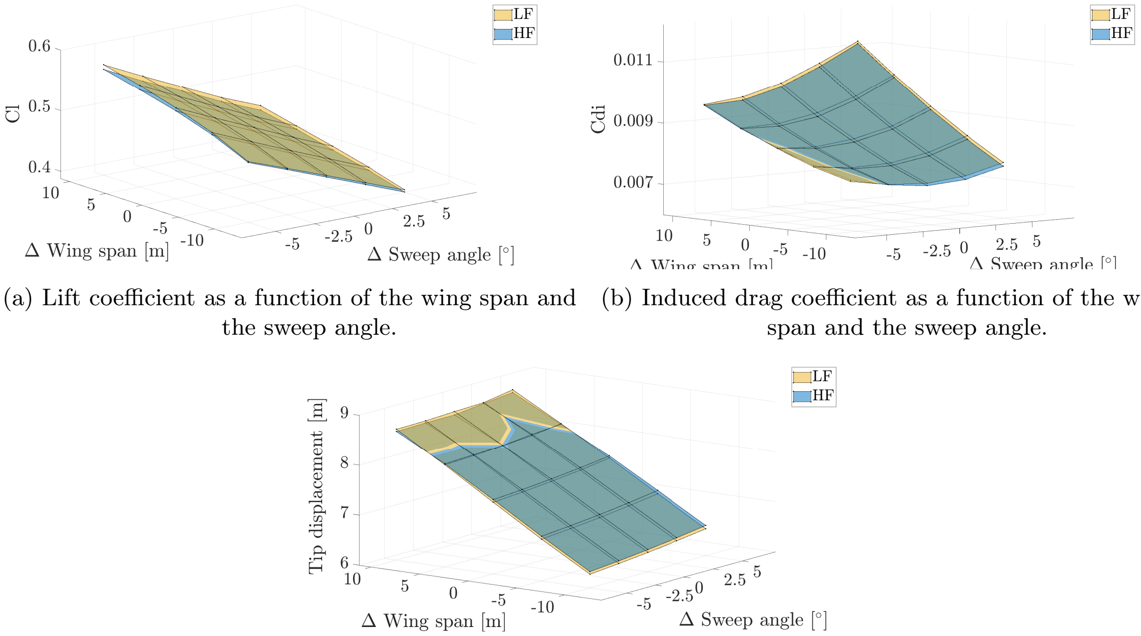


(a) Lift coefficient as a function of the normalized projection of the shell elements thickness vectors onto the standard CRM property.

(b) Induced drag coefficient as a function of the normalized projection of the shell elements thickness vectors onto the standard CRM property.

(c) Maximum Von Mises stress as a function of the normalized projection of the shell elements thickness vectors onto the standard CRM property.

Additional figures relative to the wing span - sweep angle influence study.



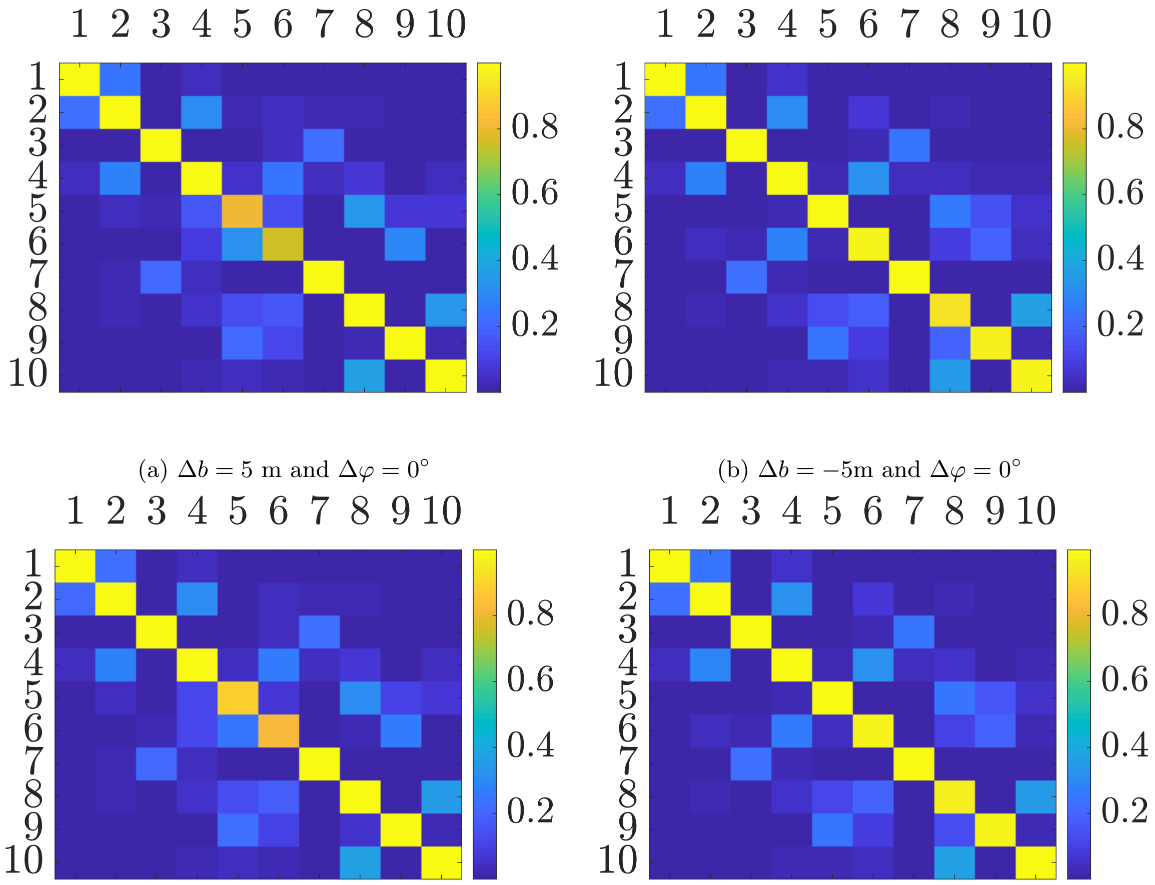
(a) Lift coefficient as a function of the wing span and the sweep angle.

(b) Induced drag coefficient as a function of the wing span and the sweep angle.

(c) Tip displacement as a function of the wing span and the sweep angle.

D

MAC matrices for different values of the design variables.



(a) $\Delta b = 5$ m and $\Delta\varphi = 0°$

(b) $\Delta b = -5$m and $\Delta\varphi = 0°$

(c) $\Delta b = 0$m and $\Delta\varphi = 2.5°$

(d) $\Delta b = 0$m and $\Delta\varphi = -2.5°$