

Multifidelity aeroelastic optimization with application to a BWB

Gilberto Ruiz Jiménez* and Joan Mas Colomer†
ISAE-SUPAERO, Université de Toulouse, Toulouse, France

Joseph Morlier‡
ICA, Université de Toulouse, ISAE-SUPAERO, MINES ALBI, UPS, INSA, CNRS, Toulouse, France

A multifidelity approach for the solution of aeroelastic optimization problems is developed using open-source tools. The Multidisciplinary Optimization problem is managed via OpenM-DAO and a dedicated python library. Aerodynamics are solved with PANAIR, with a detailed mesh for the High-Fidelity case and a coarser mesh for the Low-Fidelity one. Structural displacements are computed via NASTRAN and transferred to the aerodynamic mesh with radial basis functions (RBF). The program starts solving the coupled problem at a low fidelity level, and once some predefined convergence tolerance is reached, the high fidelity solver is called. The Multidisciplinary Analysis (MDA) solution is then treated by the outer-loop optimizer, which modifies the geometry in order to comply with design constraints and minimize fuel consumption. The method was implemented on the NASA CRM wing section, on a Blended Wing Boyd (BWB) concept, and using analytical test functions. It was found that a simple transfer of the displacement field between fidelities helps to slightly reduce optimization iterations while improving constraint satisfaction.

Keywords— Multifidelity, aeroelasticity, BWB, MDO

I. Nomenclature

<i>MDO</i>	=	MultiDisciplinary Optimization
<i>MDA</i>	=	MultiDisciplinary Analysis
<i>MDF</i>	=	MultiDisciplinary Feasible
<i>BWB</i>	=	Blended Wing Body
<i>TRMM</i>	=	Trust Region Model Management
<i>MLE</i>	=	Maximum Likelihood Estimate
<i>RANS</i>	=	Reynolds-Averaged Navier–Stokes equations
<i>IDF</i>	=	Individual Discipline Feasible
<i>COBYLA</i>	=	Constrained Optimization By Linear Approximation
<i>RBF</i>	=	Radial Basis Functions

II. Introduction

MODERN engineering design problems rely increasingly on the interdisciplinary interactions between subsystems. As a consequence, multidisciplinary design strategies have been developed, these allow to manipulate design variables from various disciplines simultaneously. Moreover, the incorporation of optimization tools within these methodologies leads to what is referred collectively as Multidisciplinary Design Optimization (MDO). The present work deals with the interaction of aerodynamics and structures (so called Fluid-Structure Interaction, FSI), which is a key feature in aircraft design. Coupled aeroelastic analyses allow a better prediction of the aerodynamic forces and structural displacements that the aircraft experiences in real flight, hence, an MDO platform is implemented in order to study these interactions.

Each discipline inside the MDO problem requires a solver that computes its effects on the global problem, with the aerodynamic solution being the main source of computational cost. High fidelity CFD analyses offer accurate results, but at the expense of high

*M.Sc. Graduate, ISAE-SUPAERO, 10 Av. Edouard Belin, 31400, Toulouse, France, gilberto.ruiz-jimenez@isae-alumni.net

†Postdoctoral Researcher, Institut Clément Ader, ISAE-SUPAERO, 10 Av. Edouard Belin, 31400, Toulouse, France

‡Professor, ICA, Université de Toulouse, ISAE-SUPAERO, MINES ALBI, UPS, INSA, CNRS, Toulouse, France, AIAA Member

computational time. On the other hand, potential flow theory, and its computational implementation, the panel codes, offer reasonable approximations with low computational demands. The main goal of this project is to efficiently use low-fidelity analyses, combined to high-fidelity ones, to conduct aeroelastic optimizations, with an application to a Blended Wing Body (BWB) configuration.

The starting point for the bibliography research is a Survey of Multifidelity Methods by Peherstorfer [1], where a variety of alternatives are listed and classified. Surrogate modelling is a global optimization strategy. It uses co-kriging as a regression technique in order to link High-Fidelity sources with one or many Low-Fidelity functions. This correlated model can be used to find optimal solutions faster [2]. On the other hand, simpler approaches such as a linear regression are found to offer a good balance between accuracy, cost and simplicity [3]. Other advantages include: ease to combine many low-fidelity sources and robustness for High-Fidelity samples with the presence of noise [3].

Newton methods and their variations have been explored as well. Jovanov and Breuker propose to solve the High-Fidelity problem by adding an aerodynamic correction load to the solution of the Low-Fidelity equation. The defect-correction method accelerates the convergence compared to the Quasi-Newton method [4]. This last case does not consider the methods as black-boxes, but rather exploits the fact that the algorithms of both fidelity levels are known and can be modified at any stage to accommodate the necessary corrections between them. Scholz presents an Aggressive Space Mapping methodology, it solves for the low fidelity fluid-structure interaction solution and then feeds that information to a Quasi-Newton algorithm. The final results are obtained from a mapping function between both fidelity levels [5].

When it comes to the application of multifidelity models to aerospace design, there are several publications on the subject. A Bayesian-enhanced Low-Fidelity correction proved the ability to maintain high-fidelity accuracy while reducing computational cost associated with the optimization of an airfoil shape [6]. An aeroelastic optimization of a BWB was carried out by Bryson [7] using a new TRMM (Trust Region Model Management) approach. Its main difference is that it adds hybrid additive-multiplicative corrections (or bridge functions) to the low-fidelity analysis [8]. Another approach to the aeroelastic optimization of a BWB is presented by Marques [9]. The flutter boundary problem is solved with a contour location approach (i.e. the zero contour of the aeroelastic damping coefficient). It also incorporates an active learning strategy, where the model evaluations are selected iteratively based on how much they are estimated to improve the predictions. All of the aforementioned techniques require statistical processing and complex combinations of stochastic regression and surrogate functions. This work explored a simpler approach: transferring of information within the routines of the MDAO, taking advantage of the open-source tools at every step of the process.

Section III describes in detail the proposed method, as well as the tools that were integrated in the MDAO and how they were programmed. Section IV presents three different test approaches for the current formulation: optimizing the NASA Common Research Model (CRM) wing, optimizing a BWB configuration, and solving analytical optimization problems used to benchmark MDAO architectures. The results and perspectives for future work are discussed in section V.

III. Methodology

A series of tools will be used together in order to create a complete multifidelity optimization solver. This section presents each of them as well as their coordinated implementation.

A. OpenMDAO

First of all, the Multidisciplinary Design Analysis has to be established. The present work uses the OpenMDAO platform [10], an open source python library that allows the user to create, solve and optimize a wide variety of coupled problems. As a complement to this tool, the aerostructures package [11] assists the user in the creation of aerospace-related problems within the platform.

The selected MDAO architecture is MultiDisciplinary Feasible or MDF, this means that the coupled problem is solved completely before the output is sent to the outer-loop optimizer. One of the advantages being that the optimization process can be interrupted before converging and the result at any point will satisfy consistency constraints (but not necessarily optimization constraints) [12]. There are many different MDAO architectures, in this case, the MDF was chosen because of the relatively simple problem (only 2 disciplines involved) and the ease of formulation. Another widely used architecture is Individual Discipline Feasible (IDF). It decouples the MDA, adding consistency constraints, and giving the optimizer control of the coupling variables. This strategy generates a problem that is potentially much larger than MDF, depending on the number of coupling variables [12]. A wide study of different MDAO architectures is out of the scope of the current problem.

Figure 1 shows a diagram of the MDAO formulation, the main components are the structural solver, the aerodynamic solver and a custom discipline called "Filter" that coordinates both fidelity levels.

B. Structural Solver

NASTRAN simulations are treated using input files filled with instructions. These inputs are generated automatically at each run. Two files are needed to run the program, the first one, 'nastran_static_template.inp' looks mostly like a normal NASTRAN file but instead of the classical Nastran FORCE cards, there are modified cards that include a dictionary key for each force component [11], for example:

```
FORCE,201,33566,0,1.0,{Fx33566}, \\\
```

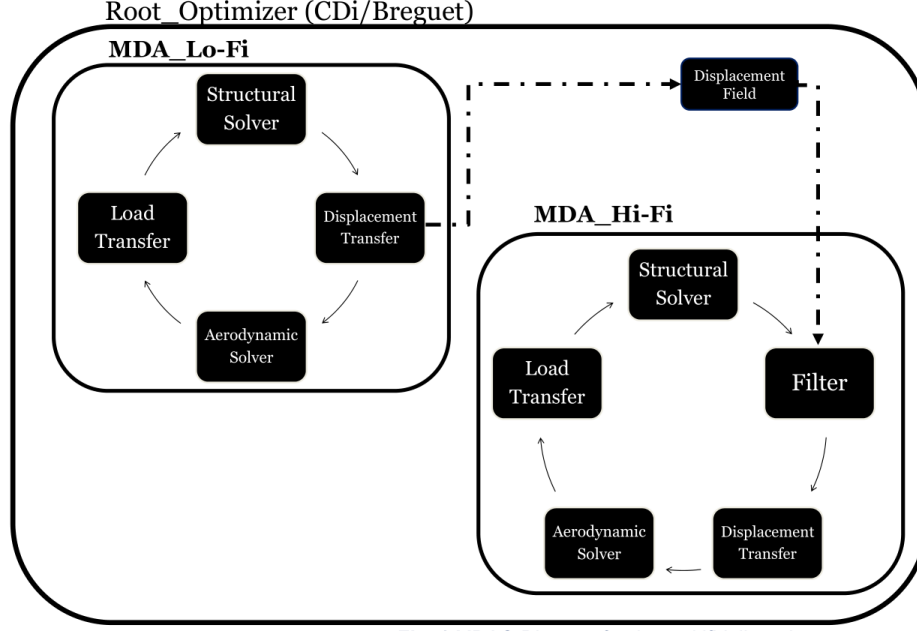


Fig. 1 MDAO diagram of the coupled multifidelity problem

{Fy33566},{Fz33566}

Another difference with respect to a normal NASTRAN file is that the GRID cards, which contain the node coordinates, are also modified to include a dictionary key for each component of the nodal coordinates [11], for example:

```
GRID,33566,,{x33566},{y33566},{z33566}
```

The dictionary keys are used to create input files for NASTRAN in an automated way. As the nodal forces vary during the MDA, the input force values are updated by substituting the dictionary keys in the template by the contents of the dictionary. The same operation is performed for the GRID cards. Even if in the case of an MDA the nodal coordinates remain constant, this feature is useful for shape optimization, in which the nodal coordinates will change with each optimization iteration [11].

The last difference with respect to an ordinary NASTRAN file is that, at the end of the file, there is a list of node IDs. The nodes in this list represent the nodes used for displacement interpolation. Typically, this list is a subset of the total nodes in the structural model, since not all nodes are required to perform the displacement interpolation. Each node ID is preceded by a '\$' sign, so they appear as comments to NASTRAN [11]. An example:

```
$List of nodes belonging to the outer skin
$33566
```

The file 'nastran_input_geometry.inp' is a typical NASTRAN input file containing all the GRID points of the jig shape of the structural model (i.e., the shape of the wing structure when no forces are applied onto it) [11]. For example, for one GRID point, this looks like:

```
GRID,33566,,33.5157,0.0010402,2.554365
```

The structural displacements are treated by a special script and interpolated using RBF to the aerodynamic mesh in order to update its shape. This module is treated as if it was a discipline in the OpenMDAO architecture.

C. Filter

The filter component requires two inputs and gives one output. The first input is the displacement field calculated by the previously converged low fidelity MDA group. Convergence depends on the relative tolerance setup by the user at startup. The second input is the displacement field from the high fidelity MDA group, which is equal to zero for the first iteration of the program. This characteristic property of the displacement field initialization is used by the filter to ensure that once the information from the previous low fidelity solution has been imported, the next cycle will use the values from the high fidelity MDA until the iterative solution converges. Since the structural meshes of both fidelity levels are identical, there is no need to interpolate or adjust the displacement field values in any way. It is also important to note that no statistical treatment or surrogate function is used at any point when the displacement field is imported, as the key idea to be explored in this research paper is the direct connection between fidelities.

D. Fluid solvers

There are two ways in which the multifidelity of the aerodynamic solver was addressed. The first one is to use the same solver at both fidelities and vary the mesh refinement, this approach has the advantage of a faster implementation and is better suited to try different strategies. For the second one, high and low fidelity solvers were established to obtain the aerodynamic loads from the wing shape and the fluid characteristics. ADFlow is an open-source CFD code developed by the MDOLab at the University of Michigan. It solves the compressible Euler, laminar Navier-Stokes and Reynolds-Averaged Navier-Stokes equations [13]. On the other hand, PANAIR [14] was chosen as the low fidelity model. It is an open-source implementation of the potential flow theory via panel methods.

The exchange between fidelities will take place at this stage: the idea is that for every call of the MDA, PANAIR will initially compute the solutions that are fed to the Gauss-Seidel algorithm. Next, when the norm of the unknowns reaches a given tolerance, ADFlow will start from the previously computed solution. As a consequence, the MDA computation will be more accurate than the one calculated in low fidelity and faster than the one calculated in pure high fidelity.

In a similar way to the structural solver, the aerodynamic loads obtained from either one of the solvers are interpolated using the load transfer script of the aerostructures package. The result is sent to the structural discipline to restart the cycle until convergence is reached.

E. Optimizer & post-processing

The solution of the MDA is then sent to the outer-loop optimizer, a COBYLA algorithm readily available in the OpenMDAO framework. The main characteristic of COBYLA is that it is a numerical optimization method for constrained problems where the derivative of the objective function is not known [15]. Finally, post-processing of the data is achieved through python scripts that transform outputs from NASTRAN and PANAIR to mesh files (.msh) that can be opened in the open source software Gmsh [16]. In the case of the analytical functions, it was possible to use a SLSQP algorithm [17]. This is because the analytical functions could be differentiated to evaluate the gradient whenever it was needed. The main advantage of this optimizer strategy is that the optimum solution is found faster than when using COBYLA.

IV. Demonstration problems

Three different demonstration problems were solved with the proposed method. We first addressed classic analytical functions with different behaviours. This allowed to get a more general understanding of the performance of the multifidelity strategy. Then, the strategy was implemented to optimize the NASA CRM wing in cruise conditions. Lastly, the same analysis was carried out for a Blended Wing Body concept. Each demonstration problem is described in detail in the following subsections.

A. Analytical functions

With the aim of validating the performance of the simple transfer strategy when the low and high fidelity functions have different shapes, 5 different analytical formulations of a high and low fidelity functions [18] were combined with the classic Sellar problem [19] and programmed in OpenMDAO along with the filter component to transfer data from the low fidelity MDA to the high fidelity MDA. The problems include an expensive function and a close cheap approximation that lacks local details. As described in [18]:

Two high fidelity functions (Rosenbrock and Brown) have single minima, while the others are multimodal. The test suite encompasses a variety of mathematical forms, including polynomials, sinusoids, and exponentials.

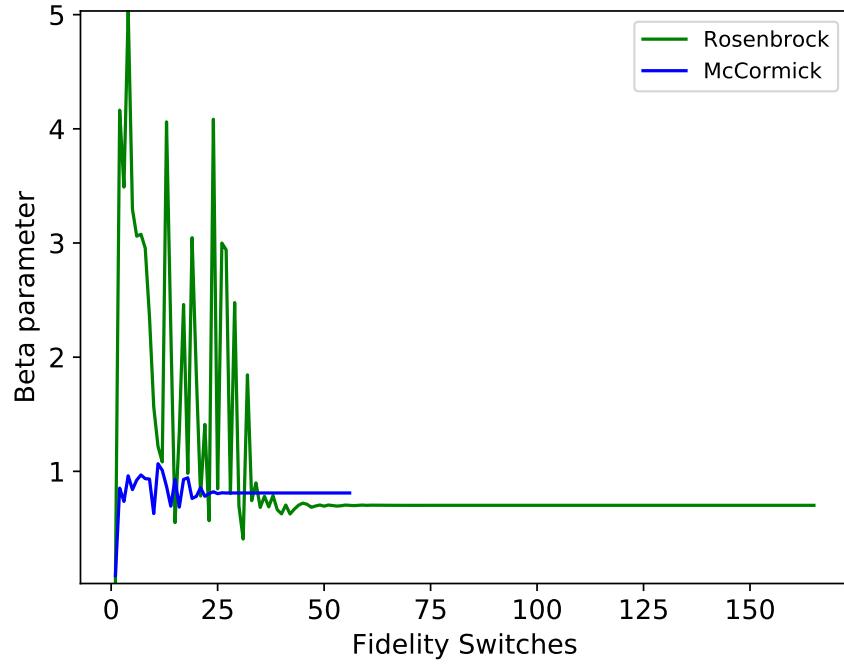
All of the formulations are listed in the Appendix A, alongside with the corresponding plots of the results. The sample problems were solved using two different optimization algorithms: COBYLA and SLSQP. Tab. 1 summarizes the results of the tests.

As usual, the efficiency of the optimization algorithm depends heavily on the local characteristics of the objective functions. The present strategy works best with the Styblinski-Tang formulation, and its worst performance occurs with the Rosenbrock function. This tendency was maintained for both optimizer strategies, which confirms that it is the shape of the functions and not the optimizer what makes a problem more suitable for acceleration with the current strategy. A marked difference in the amount of optimization between SLSQP and COBYLA is observed, this is expected because of the advantage of a defined gradient function in the case of SLSQP and has nothing to do with the multifidelity strategy.

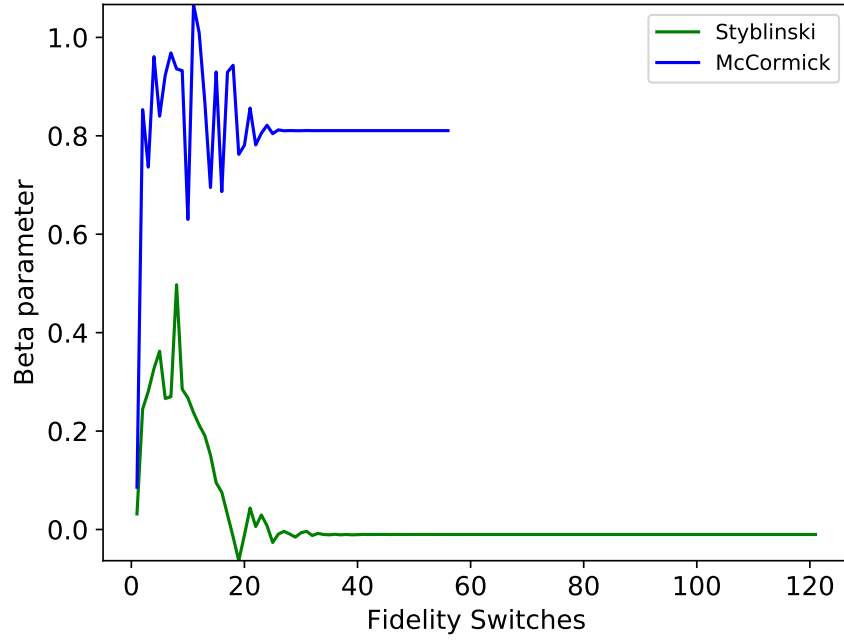
In order to characterize the compatibility of a problem with the current strategy, a simple proportional parameter was proposed and calculated at each fidelity switch in the previous formulations. Mathematically, it is defined as:

$$\beta = \frac{y_2(y_{1h})}{y_2(y_{1l})} \quad (1)$$

With this computation, the best, intermediate and worst functions were compared. Fig. 2 shows the two corresponding plots of β as the solution of the MDAO advances with each fidelity switch. One can see that the coefficient in the case of the Rosenbrock function is very high, reaching values much larger than 1. On the other hand, the Styblinski-Tang beta coefficient is at all times smaller than 1, which would be expected when the predicting capability of the low fidelity function is very good. The McCormick function has values that at most get to 1, which corresponds to the similarity between both strategies (single fidelity and multifidelity).



(a) Rosenbrock (worst) -McCormick (intermediate)



(b) Styblinski (best) - McCormick (intermediate)

Fig. 2 Comparison of the beta parameter for the analytical functions with respect to the fidelity switches

Table 1 Optimizer iterations needed to converge to a minimum, various formulations.

Formulation	COBYLA Iterations		SLSQP Iterations	
	Single	Multifidelity	Single	Multifidelity
McCormick	55	55	11	19
Brown	63	124	13	27
Styblinski-Tang	122	120	17	15
Rosenbrock	61	164	20	26
Trigonometric	83	172	12	14

B. NASA CRM wing

In this case, the optimization of the aeroelastic structure was carried out with respect to the induced drag. The design was constrained to generate enough lift for cruise conditions and not to exceed the maximum equivalent stress on the structure including the regulatory safety factor. Detailed problem setup up is available in the GitHub repository.

Fig. 3 shows the evolution of the main solution parameters for this case in both single and multifidelity modes. When it comes to the objective function, it is clear that the amount of iterations to converge is only slightly reduced. On the other hand, both constraints are closer to zero for the multifidelity case. A clear gap between the compared strategies means that constraint satisfaction is never as good in single fidelity mode. Which means that for the same solution, one could reduce the optimizer iterations and get to a similar overall configuration that satisfies the imposed constraints just as well. An implementation using ADFlow as the high fidelity fluid solver was explored as well. However, despite the numerous attempts, the solution of the problem never converged. This was due to the fact that the partial solutions from the low fidelity MDA contained regions with geometries that caused the CFD solution to diverge. Since the CFD mesh could not be manually corrected at each step, ADFlow crashed at a certain point and thus the optimizer as well.

C. Blended Wing Body

In this case, the optimization variable to minimize is the fuel consumption (calculated using the Breguet equation). The parameters and baseline geometry are available in the tutorial section of the GitHub repository. Fig. 4 shows the structural and aerodynamic solution at the end of the optimization loop for the single fidelity case. After successfully solving the problem in single fidelity, the aerodynamic mesh was refined using a python library developed at ONERA: CASIOPÉE [20].

A series of issues arose after the mesh refinement for the high fidelity case, mainly that the enriched geometry had some non-convex panels that were not compatible with the high order panel code. As a consequence, the refinement was repeated, this time only doubling the mesh nodes instead of tripling them. In this case, Panair did not crash, but the resulting C_p distribution had some panels with undefined values. This, in turn, causes the optimizer to crash because it cannot parse the current vector into the inner algorithm. Despite numerous attempts to create a viable high-fidelity mesh for Panair, the very well known issue of convexity between panels prevented a smooth run of the current method.

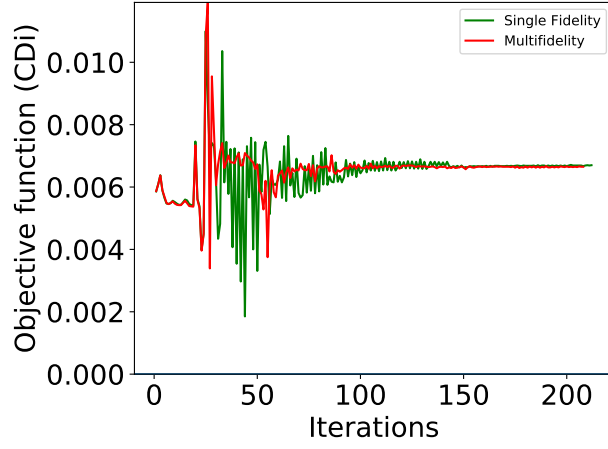
Just like in the previous test case, an implementation using ADFlow as the high fidelity fluid solver was explored. The geometry is in now even more complex, hence, it was not surprising to get the same crash of the program after some optimizer runs, this time due to the large deformations of the structure produced by the optimizer, which in turn generate spikes in the aerodynamic mesh that cause the CFD code to fail.

V. Conclusions and perspectives

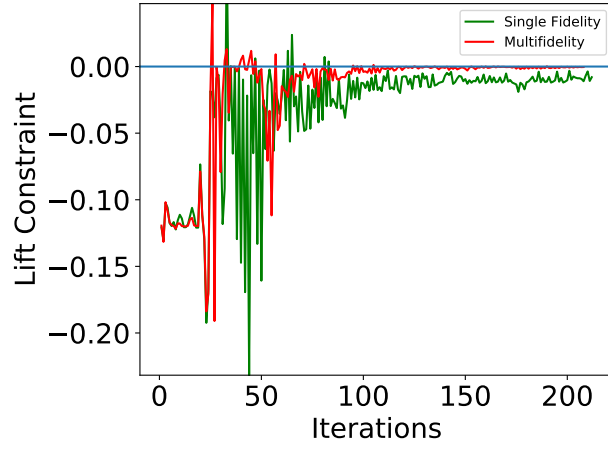
After a thorough review of different implementations of the presented method, it was shown that the constraint satisfaction is the main part of the MDAO problem that benefits from the exchange of information between fidelities. The overall number of optimizer calls is only slightly reduced, however, one must remember that the simplicity of implementation is unlike any other existing strategies. This is all contingent of course on the nature of the low and high fidelity functions, as demonstrated in the analytical test section of this work.

Another important issue is the morphing of the structure in a way that does not disrupt the aerodynamic mesh to the point that singularities start to show. This is specifically problematic when the high fidelity method is a CFD solver. Constraining the design variables of the structural solver is one of the solutions, however, this must be done carefully so as not to over-constrain the structural optimization in such a way that the mass reduction from the optimization is affected.

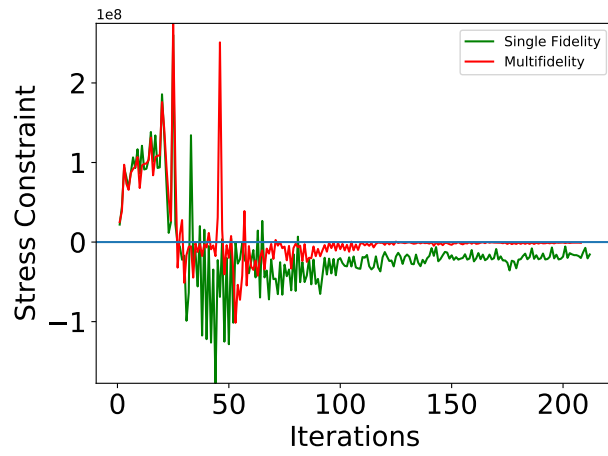
Some other improvements could be possible with the inclusion of a scaling strategy that used the computed β factor at each step to reduce or increase the weight of the low fidelity function with respect to the high fidelity one. Most of the strategies that follow this path use stochastic predictors or surrogates, but if the objective was to try and keep the model as simple as possible, the case can be made for a linear regression at the displacement field transfer. This could potentially allow to keep the constraint convergence acceleration while also reducing the optimization runs.



(a) Objective function



(b) Lift Constraint



(c) Stress Constraint

Fig. 3 Evolution of the solution parameters for both fidelity levels, NASA CRM wing, Panair–Panair

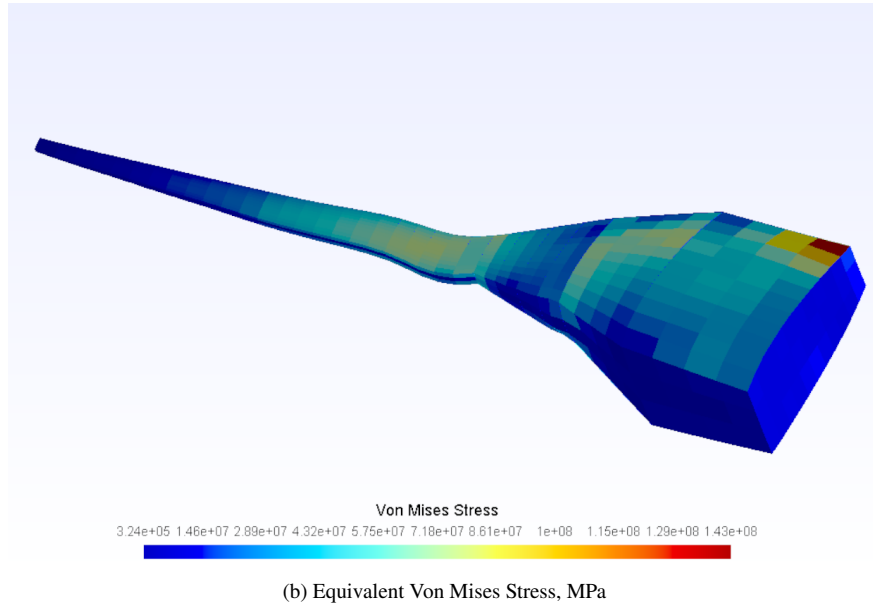
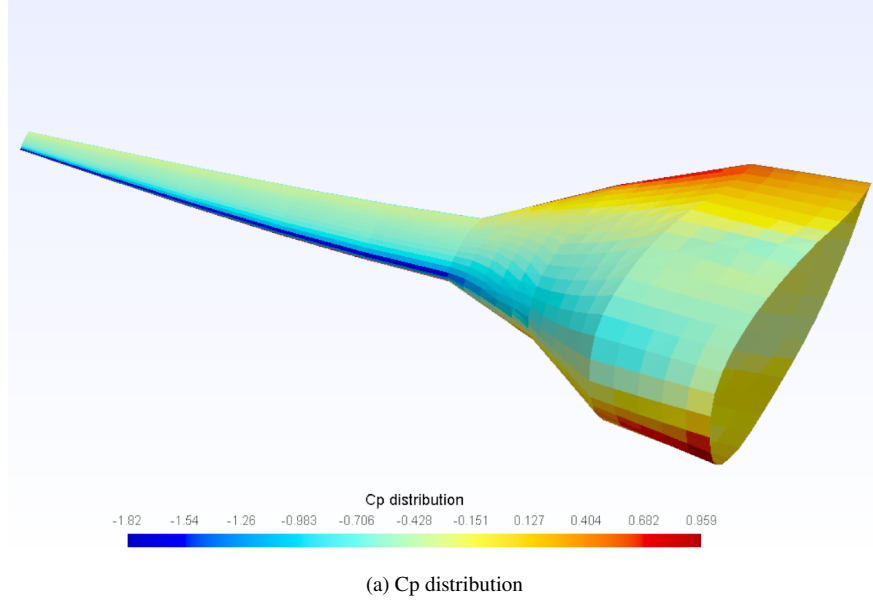


Fig. 4 Optimization result for the BWB case, single fidelity.

Appendix

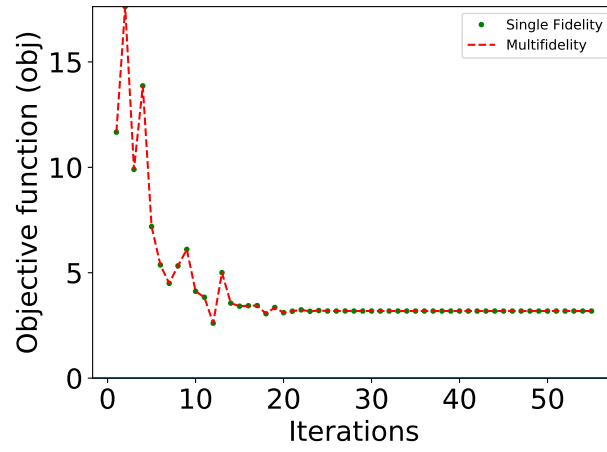
A. Formulation and plots of the analytical function tests

1. McCormick-Sellar problem

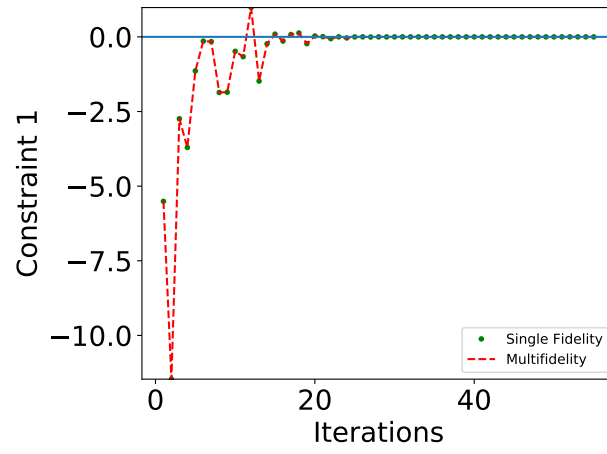
$$\begin{aligned}
 & \text{Minimize } x_1^2 + z_2 + y_1 + e^{-y_2} \\
 & \text{with respect to } z_1, z_2, x_1 \\
 & \text{subject to} \\
 & \quad \frac{y_1}{3.16} - 1 \geq 0 \\
 & \quad 1 - \frac{y_2}{24} \geq 0 \\
 & \quad -10 \leq z_1 \leq 10 \\
 & \quad 0 \leq z_2 \leq 10 \\
 & \quad 0 \leq x_1 \leq 10 \\
 & \text{Disc. 1: } \begin{cases} y_{1h} = \sin(z_1 + z_2) + (z_1 - z_2)^2 \\ \quad -\frac{3}{2}z_1 + \frac{5}{2}z_2 - 0.2y_2 + x_1 + 1 \\ y_{1l} = (z_1 - z_2)^2 + 2z_1 + 2z_2 \\ \quad -0.2y_2 + x_1 \end{cases} \\
 & \text{Disc. 2: } y_2 = \sqrt{y_1} + z_1 + z_2
 \end{aligned} \tag{2}$$

2. Brown-Sellar problem

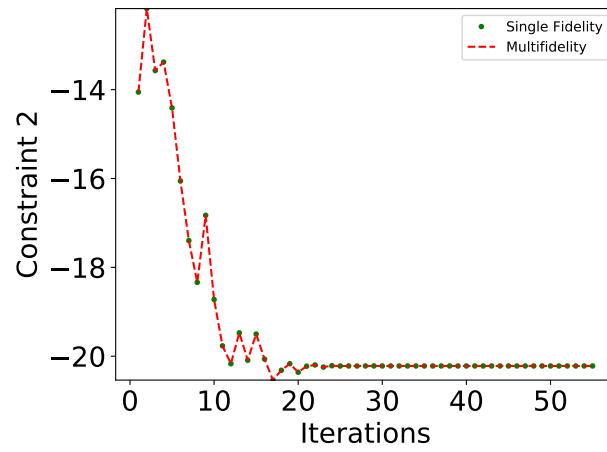
$$\begin{aligned}
 & \text{Minimize } x_1^2 + z_2 + y_1 + e^{-y_2} \\
 & \text{with respect to } z_1, z_2, x_1 \\
 & \text{subject to} \\
 & \quad \frac{y_1}{3.16} - 1 \geq 0 \\
 & \quad 1 - \frac{y_2}{24} \geq 0 \\
 & \quad -1 \leq z_1 \leq 2 \\
 & \quad -1 \leq z_2 \leq 2 \\
 & \quad 0 \leq x_1 \leq 10 \\
 & \text{Disc. 1: } \begin{cases} y_{1h} = (z_1^2)^{z_2^2+1} + (z_2^2)^{z_1^2+1} + x_1 - 0.2y_2 \\ y_{1l} = 10^{-3} + \exp(z_1^2 + z_2^2) + x_1 - 0.2y_2 \end{cases} \\
 & \text{Disc. 2: } y_2 = \sqrt{y_1} + z_1 + z_2
 \end{aligned} \tag{3}$$



(a) Objective function

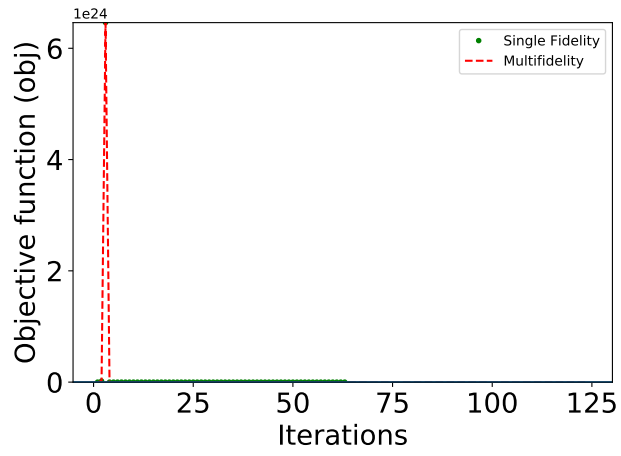


(b) Constraint 1

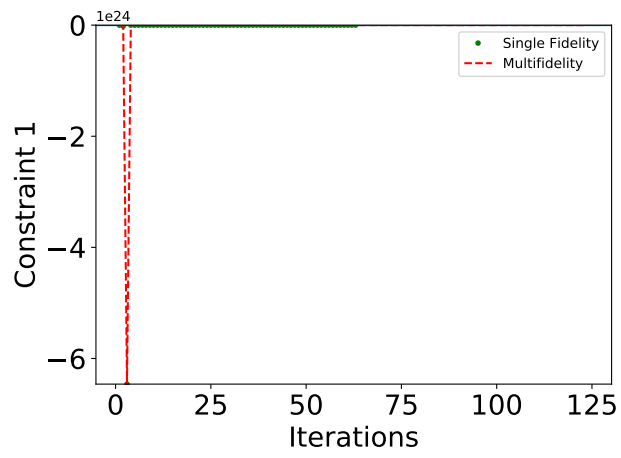


(c) Constraint 2

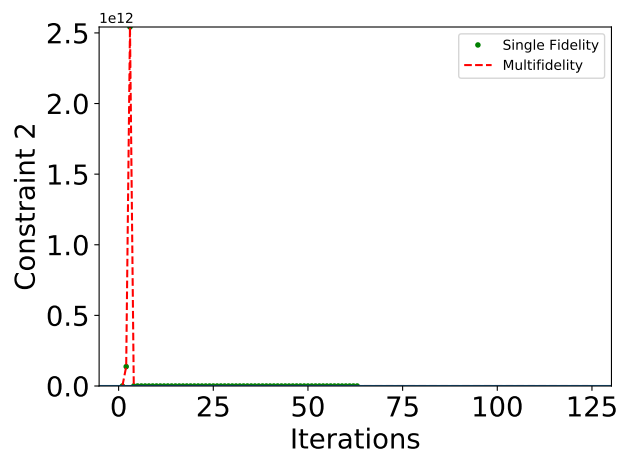
Fig. 5 Comparison of evolution of solution parameters for the McCormick function



(a) Objective function



(b) Constraint 1



(c) Constraint 2

Fig. 6 Comparison of evolution of solution parameters for the Brown function

3. Styblinski-Tang-Sellar problem

$$\begin{aligned}
& \text{Minimize } x_1^2 + z_2 + y_1 + e^{-y_2} \\
& \text{with respect to } z_1, z_2, x_1 \\
& \text{subject to} \\
& \frac{y_1}{3.16} - 1 \geq 0 \\
& 1 - \frac{y_2}{24} \geq 0 \\
& -5 \leq z_1 \leq 5 \\
& -5 \leq z_2 \leq 5 \\
& 0 \leq x_1 \leq 10
\end{aligned} \tag{4}$$

$$\text{Disc. 1: } \begin{cases} y_{1h} = \frac{1}{2} \left[\left(z_1^4 - 16z_1^2 + 5z_1 \right) \right. \\ \quad \left. + \left(z_2^4 - 16z_2^2 + 5z_2 \right) \right] + x_1 - 0.2y_2 \\ y_{1l} = z_1^4 \cos\left(\frac{2\pi z_1}{5}\right) + z_2^4 \cos\left(\frac{2\pi z_2}{5}\right) \\ \quad + x_1 - 0.2y_2 \end{cases}$$

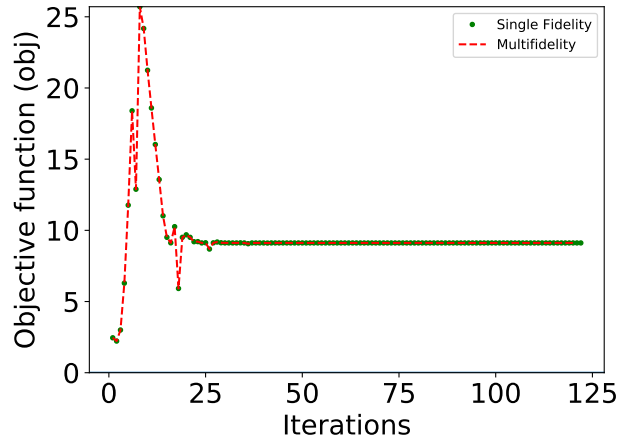
$$\text{Disc. 2: } y_2 = \sqrt{y_1} + z_1 + z_2$$

4. Rosenbrock-Sellar problem

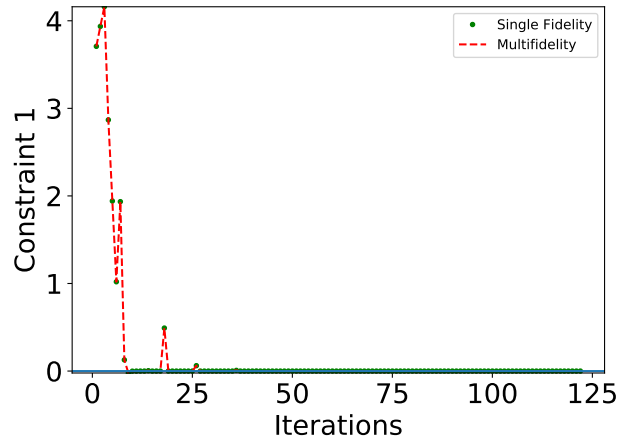
$$\begin{aligned}
& \text{Minimize } x_1^2 + z_2 + y_1 + e^{-y_2} \\
& \text{with respect to } z_1, z_2, x_1 \\
& \text{subject to} \\
& \frac{y_1}{3.16} - 1 \geq 0 \\
& 1 - \frac{y_2}{24} \geq 0 \\
& -2 \leq z_1 \leq 2 \\
& -2 \leq z_2 \leq 2 \\
& 0 \leq x_1 \leq 10
\end{aligned} \tag{5}$$

$$\text{Disc. 1: } \begin{cases} y_{1h} = (1 - z_1)^2 + 100 \left(z_2 - z_1^2 \right)^2 \\ \quad + x_1 - 0.2y_2 \\ y_{1l} = 1 + 50z_1^2 \left[(z_2 - 2)^2 + 2 \right] \\ \quad + x_1 - 0.2y_2 \end{cases}$$

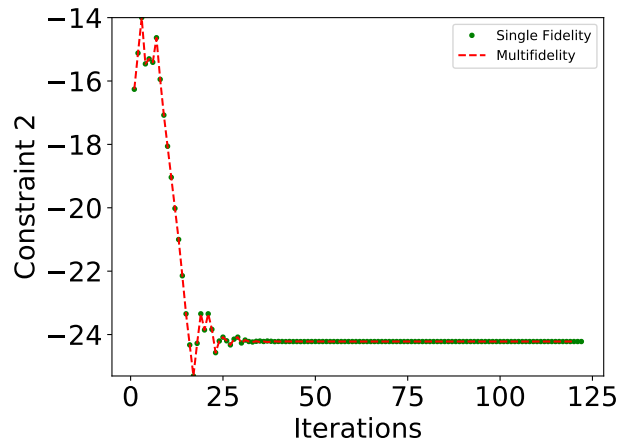
$$\text{Disc. 2: } y_2 = \sqrt{y_1} + z_1 + z_2$$



(a) Objective function

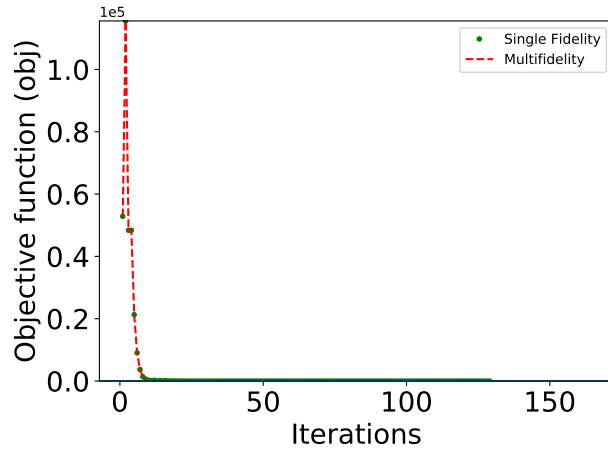


(b) Constraint 1

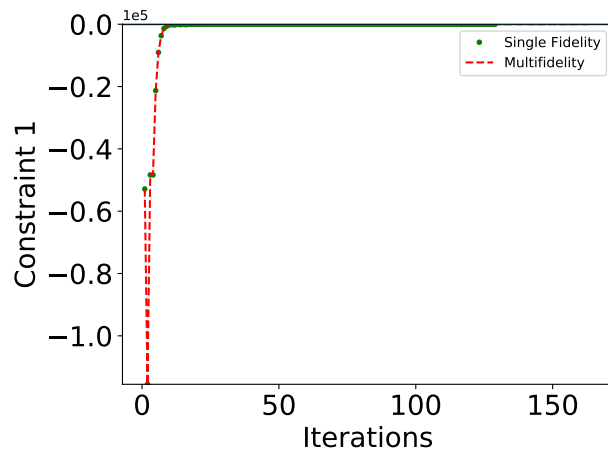


(c) Constraint 2

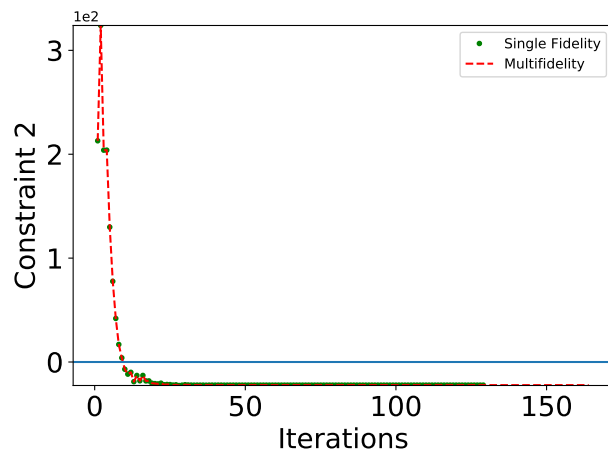
Fig. 7 Comparison of evolution of solution parameters for the Styblinski-Tang function



(a) Objective function



(b) Constraint 1



(c) Constraint 2

Fig. 8 Comparison of evolution of solution parameters for the Rosenbrock function

5. Trigonometric-Sellar problem

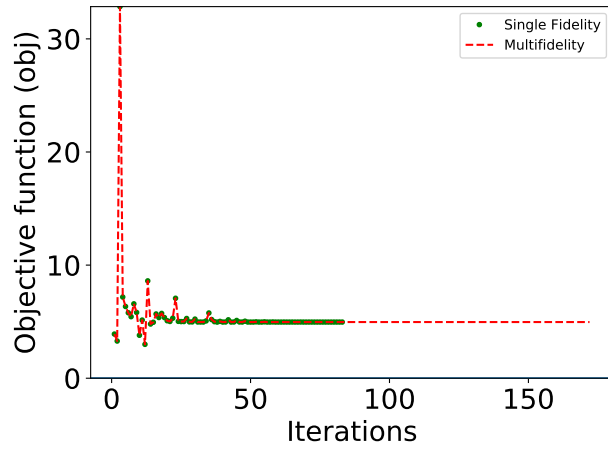
$$\begin{aligned}
& \text{Minimize } x_1^2 + z_2 + y_1 + e^{-y_2} \\
& \text{with respect to } z_1, z_2, x_1 \\
& \text{subject to} \\
& \frac{y_1}{3.16} - 1 \geq 0 \\
& 1 - \frac{y_2}{24} \geq 0 \\
& -1.5 \leq z_1 \leq 3 \\
& -1.5 \leq z_2 \leq 3 \\
& 0 \leq x_1 \leq 10 \\
& \text{Disc. 1: } \begin{cases} y_{1h} = \sum_{i=1}^2 \left[2 - \sum_{j=1}^2 \cos(x_j) \right. \\ \quad \left. + i(1 - \cos(x_i) - \sin(x_i)) \right]^2 \\ y_{1l} = 10^{-3} + \sum_{i=1}^2 5 \left(x_i - \frac{1}{4} \right) \\ \quad - 0.2y_2 + x_1 \end{cases} \\
& \text{Disc. 2: } y_2 = \sqrt{y_1} + z_1 + z_2
\end{aligned} \tag{6}$$

Acknowledgments

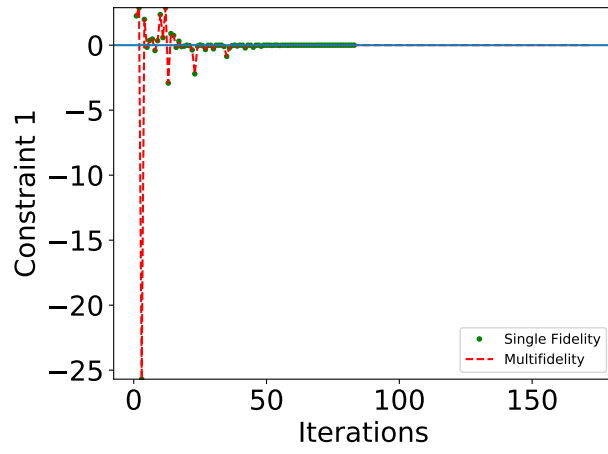
This work was supported by CONACYT (The Mexican National Council for Science and Technology), grant number 2018-000041-01EXTF-00038.

References

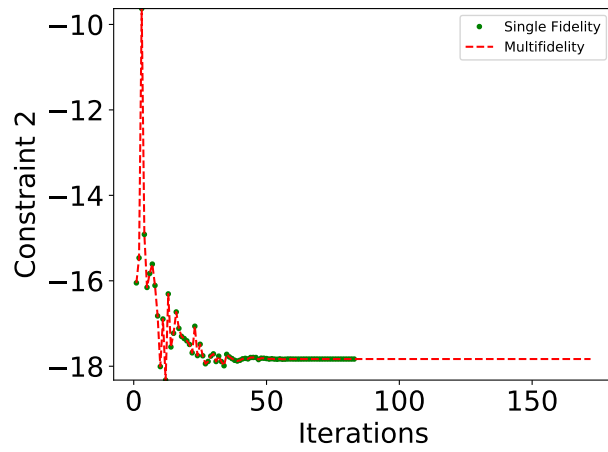
- [1] Peherstorfer, B., Willcox, K., and Gunzburger, M., “Survey of multifidelity methods in uncertainty propagation, inference, and optimization,” *SIAM Review*, Vol. 60, No. 3, 2018, pp. 550–591.
- [2] Forrester, A. I., Sóbester, A., and Keane, A. J., “Multi-fidelity optimization via surrogate modelling,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 463, No. 2088, 2007, pp. 3251–3269. <https://doi.org/10.1098/rspa.2007.1900>.
- [3] Zhang, Y., Kim, N. H., Park, C., and Haftka, R. T., “Multifidelity Surrogate Based on Single Linear Regression,” *AIAA Journal*, Vol. 56, No. 12, 2018, pp. 4944–4952. <https://doi.org/10.2514/1.j057299>, URL <https://doi.org/10.2514/1.J057299>.
- [4] Jovanov, K., and Breuker, R. D., “Accelerated convergence of static aeroelasticity using low-fidelity aerodynamics,” *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, 2015. <https://doi.org/10.2514/6.2015-0175>.
- [5] Scholcz, T., van Zuijlen, A., and Bijl, H., “Space-mapping in fluid–structure interaction problems,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 281, 2014, pp. 162 – 183. <https://doi.org/https://doi.org/10.1016/j.cma.2014.07.028>, URL <http://www.sciencedirect.com/science/article/pii/S0045782514002606>.
- [6] Fischer, C. C., Grandhi, R. V., and Beran, P. S., “Bayesian-Enhanced Low-Fidelity Correction Approach to Multifidelity Aerospace Design,” *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Vol. 56, American Institute of Aeronautics and Astronautics, 2017, pp. 3295–3306. <https://doi.org/10.2514/1.j056529>, URL <https://doi.org/10.2514/1.J056529>.
- [7] Bryson, D., Rumpfkeil, M., and Durscher, R., “Framework for Multifidelity Aeroelastic Vehicle Design Optimization,” *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, No. 18 in AIAA AVIATION Forum, American Institute of Aeronautics and Astronautics, 2017. <https://doi.org/10.2514/6.2017-4322>, URL <https://doi.org/10.2514/6.2017-4322>.
- [8] Bryson, D., and Rumpfkeil, M. P., “Comparison of Unified and Sequential-Approximate Approaches to Multifidelity Optimization,” *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, 2017. <https://doi.org/10.2514/6.2017-0131>, URL <https://doi.org/10.2514/6.2017-0131>.



(a) Objective function



(b) Constraint 1



(c) Constraint 2

Fig. 9 Comparison of evolution of solution parameters for the Trigonometric function

- [9] Marques, A. N., Lam, R., Chaudhuri, A., Opgenoord, M. M., and Willcox, K. E., “A multifidelity method for locating aeroelastic flutter boundaries,” *AIAA Scitech 2019 Forum*, No. 0 in AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2019. <https://doi.org/10.2514/6.2019-0438>, URL <https://doi.org/10.2514/6.2019-0438>.
- [10] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., “OpenMDAO: An Open-Source Framework for Multidisciplinary Design, Analysis, and Optimization,” *Structural and Multidisciplinary Optimization*, 2019. In Press.
- [11] Mas-Colomer, J., “Aeroelastic Similarity of a Flight Demonstrator via Multidisciplinary Optimization,” Theses, Université de Toulouse, Dec. 2018. URL <https://hal.archives-ouvertes.fr/tel-02023612>.
- [12] Gray, J., Moore, K. T., Hearn, T. A., and Naylor, B. A., “Standard platform for benchmarking multidisciplinary design analysis and optimization architectures,” *AIAA journal*, Vol. 51, No. 10, 2013, pp. 2380–2394.
- [13] Lyu, Z., Kenway, G. K., Paige, C., and Martins, J., “Automatic differentiation adjoint of the Reynolds-averaged Navier-Stokes equations with a turbulence model,” *21st AIAA Computational Fluid Dynamics Conference*, 2013, p. 2581.
- [14] Carmichael, R., and Erickson, L., “PAN AIR-A higher order panel method for predicting subsonic or supersonic linear potential flows about arbitrary configurations,” *14th Fluid and Plasma Dynamics Conference*, 1981, p. 1255.
- [15] Powell, M. J., “A direct search optimization method that models the objective and constraint functions by linear interpolation,” *Advances in optimization and numerical analysis*, Springer, 1994, pp. 51–67.
- [16] Geuzaine, C., and Remacle, J.-F., “Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities,” *International journal for numerical methods in engineering*, Vol. 79, No. 11, 2009, pp. 1309–1331.
- [17] Kraft, D., and Schnepfer, K., “SLSQP—a nonlinear programming method with quadratic programming subproblems,” *DLR, Oberpfaffenhofen*, 1989.
- [18] Bryson, D. E., and Rumpfkeil, M. P., “Multifidelity Quasi-Newton Method for Design Optimization,” *AIAA Journal*, Vol. 56, No. 10, 2018, pp. 4074–4086. <https://doi.org/10.2514/1.j056840>.
- [19] Sellar, R., Batill, S., and Renaud, J., “Response surface based, concurrent subspace optimization for multidisciplinary system design,” *34th Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, 1996. <https://doi.org/10.2514/6.1996-714>.
- [20] Benoit, C., Péron, S., and Landier, S., “Cassiopee: A CFD pre- and post-processing tool,” *Aerospace Science and Technology*, Vol. 45, 2015, pp. 272–283. <https://doi.org/10.1016/j.ast.2015.05.023>.