Multi-fidelity efficient global optimization: Methodology and application to airfoil shape design

Mostafa Meliani, Nathalie Bartoli, Thierry Lefebvre, Mohamed-Amine Bouhlel, Joaquim Martins, Joseph Morlier

Multi-fidelity efficient global optimization: Methodology and application to airfoil shape design

Mostafa Meliani*
Université de Toulouse, ISAE-SUPAERO, Toulouse, France

Nathalie Bartoli†, Thierry Lefebvre‡
ONERA, DTIS, Université de Toulouse, Toulouse, France

Mohamed-Amine Bouhlel§, Joaquim R. R. A. Martins¶
University of Michigan, Ann Arbor, Michigan, 48109, United States

Joseph Morlier‖
Université de Toulouse, Institut Clément Ader, CNRS, ISAE-SUPAERO, Toulouse, France

Predictions and design engineering decisions can be made using a variety of information sources that range from experimental data to computer models. These information sources could consist of different mathematical formulations, different grid resolutions, different physics, or different modeling assumptions that simplify the problem. This leads to information sources with varying degrees of fidelity, each with an associated accuracy and querying cost. In this paper, we propose a novel and flexible way to use multi-fidelity information sources optimally in the context of airfoil shape optimization using both Xfoil and ADflow. The new developments are based on Bayesian optimization and kriging metamodeling and allow the aerodynamic optimization to be sped up. In a constrained optimization example with 15-design variables problem, the proposed approach reduces the total cost by a factor of two compared to a single Bayesian based fidelity optimization and by a factor of 1.5 compared to sequential quadratic programming.

Nomenclature

DOE	=	Design of Experiments
MOE	=	Mixture of Experts
EGO	=	Efficient Global Optimization (for unconstrained optimization)
SEGO	=	Super Efficient Global Optimization (for constrained optimization)
SEGOMOE	=	SEGO with Mixture of Experts
MFEGO	=	Multi-Fidelity EGO
MFSEGO	=	Multi-Fidelity SEGO

I. Introduction

The use of multiple fidelities is particularly interesting for aircraft design decisions where the high-dimensionality of the design space and the cost of the high fidelity analyses make global optimization near impossible. Many efforts have used multiple information sources for surrogate modeling (different accuracies for the same quantity of interest) [1] or optimization [2, 3]. There have also been efforts in modeling the differences in fidelity in an additive way

*MSc Student, ISAE-SUPAERO, melimostafa@gmail.com
†Senior Researcher, Information Processing and Systems Department, nathalie.bartoli@onera.fr, AIAA Member
‡Research engineer, Information Processing and Systems Department, thierry.lefebvre@onera.fr
§Postdoctoral Fellow, Department of Aerospace Engineering, mbouhlel@umich.edu
¶Professor, Department of Aerospace Engineering, jrram@umich.edu, AIAA Associate Fellow
‖Professor, Structural Mechanics, Joseph.Morlier@isae.fr, AIAA member

[4, 5]. However these approaches fail to work in the general case where different assumptions about the physics of the problem exist between fidelity levels. In this context, the work of Kennedy and O'Hagan [6] provided powerful tools for multi-fidelity surrogate modeling which Forrester et al. [7] used for a multi-fidelity extension of the Efficient Global Optimization algorithm (EGO) [8] for unconstrained problems. In 2013, Le Gratiet [9] proposed an alternative formulation of multi-fidelity co-kriging with explicit contributions of each fidelity level to the overall model. In 2002, EGO was extended to constrained problems with the development of SEGO [10]). More recently SEGO was coupled to MOE (mixture of experts) methodology in order to solve high dimensional variables and constrained optimization problems such as wing shape aerodynamic optimization in the SEGOMOE framework [11–13]. SEGOMOE will be used in this work as a reference for surrogate based constrained optimizer with a single high fidelity (HF). Of course, we will also compare to the standard gradient-based optimizer SNOPT [14]. In practice, the SEGOMOE framework is able to treat either unconstrained and constrained problem with or without MOE for black box optimization problems.

In this paper these works are used to improve the efficiency of global optimization algorithms which suffer from the curse of dimensionality. We demonstrate that the use of multiple fidelities helps alleviate this curse and show that we can perform high-dimensional Bayesian optimization. We begin by reviewing kriging and multi-fidelity co-kriging formulations; we then present the multi-fidelity extension of EGO called MFEGO. We provide an analysis of the properties of the proposed algorithm including its global convergence. Finally, we present two kinds of results on both unconstrained and constrained optimizations of a subsonic airfoil with 9 and 15 design variables and compare the performance of MFEGO to standard methodologies such as EGO and SNOPT. The constrained version of the algorithms will be later referred to as SEGO and MFSEGO.

II. Kriging and multi-fidelity co-kriging description

There are multiple ways to build an approximation of a function, such as artificial neural networks (ANN), radial basis functions (RBF) or support vector regressions (SVR) [7]. In all these metamodels (or surrogate models), a fundamental assumption is that the quantity of interest $y(x)$ can be written $y(x) = \hat{y}(x) + \epsilon$, where ϵ, the residuals, are independent identically distributed normal random variables, so that fitting the model $\hat{y}(x)$ is performed by minimizing a measure over ϵ.

Following the same assumptions, the specificity of kriging is that the 'errors'—or more accurately, the deviations from the base model—in the predicted values $\hat{y}$, are not independent. Rather, we take the view that the errors are a systematic function of the locations of the samples. The kriging surrogate model $\hat{y}(x) = m(x) + Z(x)$, is comprised of two parts: 'regression' term $m(x)$ and a functional departure from that regression: $Z(x)$ [10]. We can write the regression term for the *universal kriging* as the following:

$$m(x) = \sum_{j=1}^{k} \beta_j f_j, \tag{1}$$

where $\{f_1, \ldots, f_k\}$ are basis functions of $m(x)$ and β_j are coefficients weighting these basis functions. *Ordinary kriging* denotes the special case where $k = 1$ and $f_1 = 1$, meaning the regression function takes the form of a constant, leaving most of the prediction work to $Z(x)$. $Z(x)$ on the other hand is a Gaussian process with a covariance function:

$$\mathrm{cov}(Z(w), Z(x)) = \sigma^2 R(w, x), \tag{2}$$

where σ^2 is a scaling factor known as the process variance, w and x are two design points in $\mathbb{R}^d$ and $R(w, x)$ is the spatial correlation function of the process also known as the kernel of the process. The choice of the kernel influences greatly the way data is fitted. The kernel quantifies how quickly and smoothly the function changes from point x to point w.

One of the most commonly used kernels in kriging is the squared exponential correlation kernel:

$$R(z, x) = \exp\left(-\sum_{k=1}^{d} \theta_{(k)}(z_{(k)} - x_{(k)})^2\right) \tag{3}$$

where $x \in \mathbb{R}^d$, $z \in \mathbb{R}^d$ and $\theta \in \mathbb{R}^d$ is a vector of hyperparameters of the kriging model, denoting the correlation along the different axes of space. This Kernel has the particularity that it only depends on the weighted distance between points x and z and not their actual positions. It is called a stationary Kernel.

Once we have defined the Kernel, and "fitted" the vector of hyperparameters θ using the vector $X_T = \{x_1, \ldots, x_n\}$ (with $x_i \in \mathbb{R}^d$) yielding the responses $Y_T = \{y_1, \ldots, y_n\}$ (with $y_i \in \mathbb{R}$), we can express the mean and covariance of the

Gaussian Process $Z(x)$:

$$\mu_Z(x) = r(x, X_T)' R^{-1}(X_T, X_T) Y_T \tag{4}$$

$$\sigma_Z^2(x) = \sigma^2(1 - r(x, X_T)' R^{-1}(X_T, X_T) r(x, X_T)) \tag{5}$$

where $x \in \mathbb{R}^d$ is the prediction point, and X_T is the locations of training set. $R(X_T, X_T)$ is the matrix of correlations among the training points. $r(x, X_T)$, on the other hand, denotes the correlation between the prediction point and the training points. Note that Eq. (4) ensures the interpolation of the training points. In fact, if we make a prediction at training point x_i, the vector $r(x_i, X_T)$ will correspond to the i^{th} line of $R(X_T, X_T)$, so that $(r(x_i, X_T)' R^{-1}(X_T, X_T))'$ will give the i^{th} unit vector and so $\mu_Z(x_i) = y_i$.

If we take into account the 'regression' term $m(x)$ the prediction becomes:

$$\mu(x) = m(x) + r(x)' R^{-1}(Y_T - m(x))$$

which we can also express using the basis introduced in Eq. (1):

$$\mu(x) = f(x)'\beta + r(x)' R^{-1}(Y_T - F\beta) \tag{6}$$

$$\begin{aligned} \sigma^2(x) = \sigma^2 \Big[& 1 - r(x)' R^{-1} r(x) + \\ & \Big(f(x)' - r(x)' R^{-1} F\Big) \Big(F' R^{-1} F\Big) \Big(f(x)' - r(x)' R^{-1} F\Big)\Big] \end{aligned} \tag{7}$$

where β is the vector of coefficients β_j introduced in Eq. (1). F is the matrix of the values of the regression basis function at the positions of the training points, whereas $f(x)$ is the vector of values of these functions at the prediction point.

In the case of multi-fidelity surrogate models, we can make assumptions to simplify the problem and inform our model (which results in a decrease of the data needed to learn the model). Lewis and Nash [4] proposed an additive bridge to link the high and low fidelity of a multigrid approach:

$$f_{HF}(x) = f_{LF}(x) + \gamma(x). \tag{8}$$

When using different grids for a partial differential equation, it is expected that the low and high fidelity would have the same scale and a very high correlation. The coarse and finer meshes are expected to capture the same low frequencies, the coarser mesh, however, would fail to capture high-frequency phenomena. $\gamma(x) \in \mathbb{R}$ can then be used to capture the difference. However, in general when modeling multiple fidelities that involve different assumptions on the physics of the problem (e.g., viscosity, compressibility, or turbulence in aerodynamics models), chances are that the low and high fidelities will have different scales and sometimes (in rare cases) poor correlations. Kennedy and O'Hagan [6] proposed a new formulation that takes the correlation and scaling into account by introducing a factor $\rho \in \mathbb{R}$ in the formulation above:

$$\begin{cases} f_{HF}(x) = \rho f_{LF}(x) + \delta(x) \\ \text{with} \qquad f_{LF}(\cdot) \perp \delta(\cdot) \end{cases} \tag{9}$$

where $\delta(\cdot)$ is the discrepancy function tasked with capturing the differences between the low- and high-fidelity functions beyond scaling. The addition of the term ρ increases the robustness of the model. Le Gratiet [9] proposed an implementation using the regression term expression of *universal kriging* introduced in Eq. (1) and extends it to take the lower fidelity model as a basis function such that the regression term becomes:

$$m(x) = \sum_{j=1}^{k} \beta_j f_j + \beta_\rho f_{LF},$$

where β_ρ is an estimation of ρ performed by a classic parameter estimation such as the likelihood maximization [15, 16].

Assuming the independence of the high and low-fidelity models (two levels here), the mean and variance of the high-fidelity model are expressed:

$$\mu_{HF} = \rho\ \mu_{LF} + \mu_\delta \tag{10}$$

$$\sigma_{HF}^2 = \rho^2\ \sigma_{LF}^2 + \sigma_\delta^2 \tag{11}$$

The approach can be extended to l levels of fidelity. To make this explicit, let us denote $f_0, \ldots, f_l$ the hierarchically ranked fidelity codes (from lowest $f_0 = f_{LF}$ to highest $f_l = f_{HF}$). Using the recursive formulation, we write:

$$\mu_k = \rho_{k-1}\ \mu_{k-1} + \mu_{\delta_k} \tag{12}$$

$$\sigma_k^2 = \rho_{k-1}^2\ \sigma_{k-1}^2 + \sigma_{\delta_k}^2. \tag{13}$$

The formulation by Le Gratiet [9], if satisfying the nested DOE requirement, offers explicit expressions of the contribution of fidelity levels to the uncertainty of the model. The nested DOE requirement states that $X_l \subseteq X_{l-1} \ldots \subseteq X_0$ where X_i is the vector of training points of the fidelity i and l is the highest fidelity. With this assumption, a point computed at the highest fidelity has to be also computed at the lowest fidelities.

We introduce the following notation in Eq. (13):

$$\begin{aligned} \sigma_{\delta,k}^2 &= \sigma_{\delta_k}^2 \qquad \text{for } k \in \{1, \ldots, l\} \\ \sigma_{\delta,0}^2 &= \sigma_0^2 \qquad \text{for } k = 0. \end{aligned} \tag{14}$$

We express the uncertainty contribution of the fidelity level k at design point x (corrected from page 163, [9]) as:

$$\sigma_{\text{cont}}^2(k, x) = \sigma_{\delta,k}^2(x) \prod_{j=k}^{l-1} \rho_j^2, \tag{15}$$

which means that the variance contribution of the fidelity level k: $(\sigma_{\delta,k}^2)$ is scaled using the recursive values of ρ_j until we get to the highest fidelity l. These contributions are essential to build Sequential Design or Optimization strategies.

Recent works on linear regression multi-fidelity surrogates have been proposed in [17] where the accuracy is proved to be better than co-kriging [9] but in the context of this research where we extend design optimization to multi-fidelity information sources, the access to an estimation of the variance is essential as it gives us an estimation of the error at each point of the design space. It thus helps us balance exploration/exploitation in the context of global optimization as explained in the following section.

A Python implementation of multi-fidelity co-kriging based on Le Gratiet's work can be found in the open source Surrogate Modeling Toolbox [18] (SMT).*

III. MFEGO methodology

Bayesian optimization is defined by Močkus [19] as an optimization technique based upon the minimization of the expected deviation from the exteremum of the studied function. The objective function is treated as a black-box function. A Bayesian strategy sees the objective as a random function and places a prior over it. The prior captures our beliefs about the behavior of the function. After gathering the function evaluations, which are treated as data, the prior is updated to form the posterior distribution over the objective function. The posterior distribution, in turn, is used to construct an acquisition function (often also referred to as *infill sampling criterion*) that determines what the next query point should be.

A. Efficient Global Optimization: EGO

We describe here the expected improvement as well as the EGO algorithm based on Jones et al. [8]. Let F be an expensive black-box function to be minimized. We sample F at the different locations $X = \{x_1, x_2, \ldots, x_n\}$ yielding the responses $Y = \{y_1, y_2, \ldots, y_n\}$. We build a kriging model (also called Gaussian process) with a mean function μ and a variance function σ^2 as presented in Section II. The next step is to compute the expected improvement (EI) criterion. To do this, let us denote:

$$f_{min} = \min\{y_1, y_2, \ldots, y_n\},$$

the expected improvement function (EI) can be expressed:

$$E[I(x)] = \mathbb{E}[\max(f_{min} - Y, 0)] \tag{16}$$

*https://www.github.com/SMTorg/smt

where Y is the random variable following the distribution $\mathcal{N}(\mu(x), \sigma^2(x))$. By expressing the right-hand side of Eq. (16) as an integral, and applying some tedious integration by parts, one can express the expected improvement in closed form:

$$E[I(x)] = \begin{cases} (f_{min} - \mu(x))\Phi\left(\frac{f_{min}-\mu(x)}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{f_{min}-\mu(x)}{\sigma(x)}\right), & \text{if } \sigma > 0 \\ 0, & \text{if } \sigma = 0 \end{cases}, \tag{17}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are respectively the cumulative and probability density functions of $\mathcal{N}(0, 1)$. Next, we determine our next sampling point as:

$$x_{n+1} = \underset{x}{\operatorname{argmax}} \left(E[I(x)]\right). \tag{18}$$

We then test the response y_{n+1} of our black-box function F at x_{n+1}, rebuild the model taking into account the new information gained, and research the point of maximum expected improvement again. We summarize here the EGO strategy in Algorithm 1.

Algorithm 1 EGO Algorithm

```
procedure RUN(F, n_iter)                    ▷ Find the best minimum of F in n_iter iterations
    while i ≤ n_iter do
        mod ← model(X, Y)                   ▷ Surrogate model based on sample vectors X and Y
        f_min ← min Y
        x_{i+1} ← argmax EI(mod, f_min)     ▷ Choose x that maximizes EI
        y_{i+1} ← F(x_{i+1})                ▷ Probe the function at most promising point x_{i+1}
        X ← X ∪ {x_{i+1}}
        Y ← Y ∪ {y_{i+1}}                   ▷ Update the samples vectors
        i ← i + 1
    f_min ← min Y
    return f_min                            ▷ This is the best known solution after n_iter iterations
```

B. EGO extension to multi-fidelity: MFEGO

We extend the EGO algorithm to work with multiple fidelities. In practice as the SEGOMOE framework proposed by Bartoli et al. [11, 12, 13] is capable of handling both unconstrained and constrained problems and this with or without MOE, we choose to note in this paper the unconstrained version of SEGOMOE without MOE as EGO. The multi-fidelity is defined as MFEGO. Finally for constrained problems, we will use the notation SEGO for SEGOMOE without MOE and MFSEGO for its multi-fidelity adaptation.

The main idea of the proposed algorithm is that the search for the most promising sample and the choice of level of enrichment can be seen as problems to be tackled sequentially. Indeed, we can consider that given a Gaussian process (GP) or kriging model ($m_k \backsim GP(\mu, \sigma^2)$) and a current best ($f_{min}$), EI (or another infill sampling criterion) can be trusted to find the next *most promising point*. The choice of the *fidelity level of enrichment* is a different question that can be formulated thus: given the uncertainty at a chosen point, is it more interesting to query it at the highest-fidelity level or at lower ones? This choice of a two-stage decision process (fix *most promising point* then *fidelity level of enrichment*) offers the advantage of greatly reducing the time of *infill sampling criterion* optimization without being restrictive in any way. The choice of the *fidelity level of enrichment* translates an idea stating that one should favor the use of low-fidelity samples for exploration, and high-fidelity ones for exploitation, while making sure that there is no resampling at the same location.

Let $f_0, \ldots, f_l$ be the lowest- to highest-fidelity of a quantity of interest, with querying costs $c_0, \ldots, c_l$. Using the recursive formulation [9] with a constant ρ, we know that:

$$f_k = \rho_{k-1}\, f_{k-1} + \delta_k \qquad \text{for } k \in \{1, \ldots, l\} \tag{19}$$

$$\rho_{k-1} = \operatorname{corr}(f_k, f_{k-1}) \frac{\operatorname{std}(f_k)}{\operatorname{std}(f_{k-1})} \tag{20}$$

$$\sigma_k^2 = \rho_{k-1}^2\, \sigma_{k-1}^2 + \sigma_{\delta_k}^2 \tag{21}$$

where the notations are std$(\cdot)$ for the standard deviation and corr$(\cdot,\cdot)$ for the correlation. Using the notations introduced in Eq. (14), the variance contribution of the fidelity level k at design point x^* defined by Eq. (15) is recalled here:

$$\sigma^2_{\text{cont}}(k, x^*) = \sigma^2_{\delta,k}(x^*) \prod_{j=k}^{l-1} \rho_j^2$$

Due to the necessity of nested DOEs, all lower fidelities are enriched at the same time, the uncertainty reduction becomes:

$$\sigma^2_{\text{red}}(k, x^*) = \sum_{i=0}^{k} \sigma^2_{\delta,i}(x^*) \prod_{j=i}^{l-1} \rho_j^2$$

The corresponding cost to the enrichment of level of fidelities 0 through k is:

$$\text{cost}_{\text{total}}(k) = \sum_{i=0}^{k} c_i$$

We propose the level of enrichment criterion as follows:

$$t = \underset{k \in (0,\ldots,l)}{\text{argmax}} \quad \frac{\sigma^2_{\text{red}}(k, x^*)}{\text{cost}_{\text{total}}(k)^2}$$

where t is the highest fidelity level to be added (the nested DOE imposed to enrich all lower fidelities). It is questionable whether it is possible to always find a common unit of measurement for the cost of an observation and the variance reduction, but as the variance scales with the square of the correlation (Eqs. (20) and (21)), it is reasonable to penalize the uncertainty reduction by the square of the cost. The correlation translates the quantity of information shared between multiple functions. Note that Le Gratiet [9] introduces a sequential design strategy where the uncertainty is simply penalized by the cost. Tests were realized with both approaches (penalization with simply the cost, or the square of the cost), the penalization with the square of the cost constantly gave better results.

C. Algorithm

The choice of level heuristic having been presented, we summarize the proposed strategy in Algorithm 2. We note $\{X_0, \ldots, X_l\}$ the DOEs of the fidelity levels 0 through l. $\{Y_0, \ldots, Y_l\}$ are the corresponding responses.

Algorithm 2 MFEGO Algorithm

1: **procedure** ENRICH_LEVEL($model, x^*, costs$) ▷ Which fidelities to query
2: compute $\sigma^2_{red,0}(x^*, model)$ ▷ Uncertainty reduction by querying at x^* at level 0
3: $crit_0 \leftarrow \sigma^2_{red,0}(x^*, model)/costs[0]^2$
4: enrich Fidelity 0 ▷ LF has to be computed because of nested DOEs
5: update datasets X_0 and Y_0
6: **for** $k \in \{1, \ldots, l\}$ **do**
7: compute $\sigma^2_{red,k}(x^*, model)$
8: $crit_k \leftarrow \sigma^2_{red,k}(x^*, model)/(\sum_{i=0}^{k} costs[i])^2$
9: **if** $crit_k \geq crit_{k-1}$ **or** $\sigma^2_{red,k-1} \leq \epsilon$ **then** ▷ ϵ: machine resolution
10: enrich Fidelity k
11: update datasets X_k and Y_k
12: **else**
13: break;
14: **return** updated datasets
15:
16: **procedure** RUN(F, $n_{iter}, costs$) ▷ Find the best minimum of F in n_{iter} iterations
17: **while** $i \leq n_{iter}$ **do**
18: $mod \leftarrow \text{model}(\{X_0, \ldots, X_l\}, \{Y_0, \ldots, Y_l\})$ ▷ Multi-fidelity surrogate model
19: $f_{min} \leftarrow \min Y_l$
20: $x_{i+1} \leftarrow \text{argmax EI}(mod, f_{min})$ ▷ Choose x that maximizes EI
21: ENRICH_LEVEL($mod, x_{i+1}, costs$)
22: $i \leftarrow i + 1$
23: $f_{min} \leftarrow \min Y_l$
24: **return** f_{min} ▷ This is the best known HF solution after n_{iter} iterations

Note that the algorithm above does not take into account the constant cost of an iteration (surrogate model creation and maximization of EI). This can cause the optimization to last longer. A quick fix to take into account the constant cost of an iteration is to measure and to add that cost as an offset. This translates the idea that when we perform one iteration of the algorithm, the cost of an update is not only the querying time of the high- or low-fidelity code, but also the time needed to build the surrogate model and maximize the *infill sampling criterion*.

As can be seen in line 19 of the MFEGO algorithm, the value of f_{min} is only updated by the best HF value. That is because other fidelity codes and datasets are only considered to help MFEGO and are not the objective of optimization. By only updating the best solution when a high fidelity sample is requested, the optimization is made more robust. Indeed, low fidelity can only be used to reduce some amount of the expected improvement, which is an amount due to uncertainty (exploration), rather than knowledge (exploitation). This property ensures that MFEGO converges to the global optimum of the high-fidelity function (in the same sense that EGO converges to the global optimum of a function). Note that the criterion integrates the correlation of the fidelity levels and thus makes the algorithm more robust to cases where there is a poor correlation between fidelity levels. Indeed, when $\rho_k \to 0$, $\frac{\sigma^2_{\text{red}}(k,x^*)}{\text{cost}_{\text{total}}(k)^2} \to 0$, prompting the algorithm to move to the higher fidelities.

It is important that the algorithm will not resample high-fidelity points as the EI at these points is zero. It will also not resample low-fidelity points as the choice of level criterion will become zero for the low fidelity points, pushing the algorithm to query the high fidelity ones, thus enhancing the model and solution.

D. Illustration on 1-D analytic problem

The proposed strategy is illustrated on the following 1-D analytic problem [7]:

$$f_{HF}(x) = (6x - 2)^2 \sin(2(6x - 2))$$
$$f_{LF}(x) = 0.5 f_{HF} + 10(x - 0.5) - 5$$

In Figure 1 we represent graphically these two fidelity levels. f_{HF} is the expensive function and f_{LF} is the cheap one.

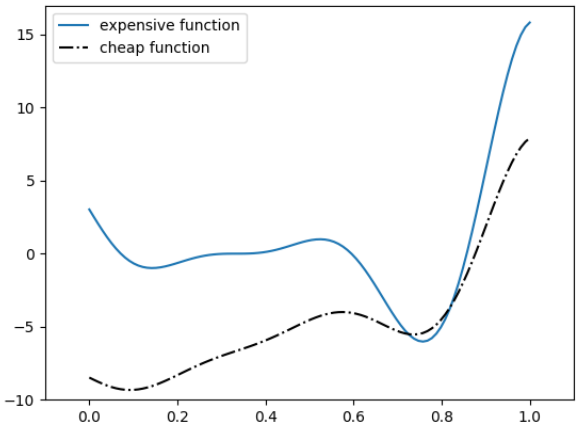


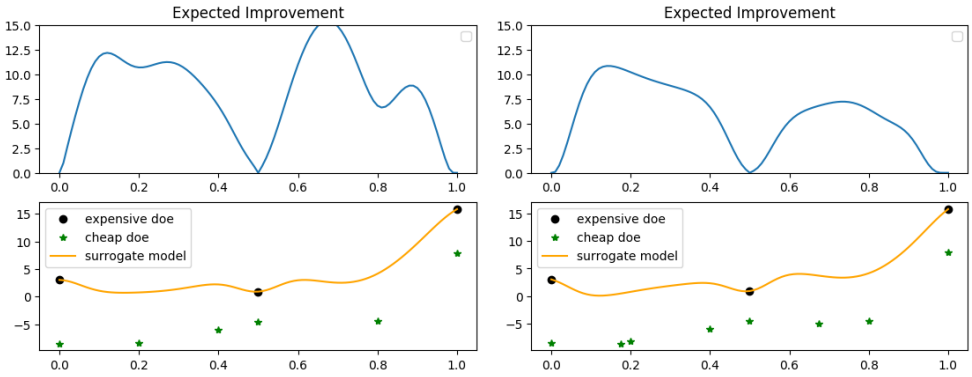
Fig. 1 Multi-fidelity of the 1-D analytic problem [7].

We make the assumption that the cost ratio between the fidelity levels is 1/1000. We start the optimization with 3 HF samples and 6 LF ones. We define the low- and high-fidelity datasets as:

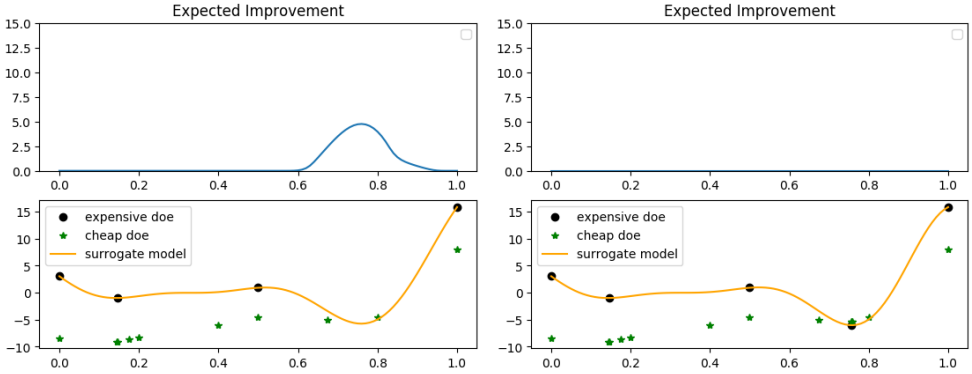
$$X_0 = \{x_1^{LF}, x_2^{LF}, x_3^{LF}, x_4^{LF}, x_5^{LF}, x_6^{LF}\} \text{ and } X_1 = \{x_1^{HF}, x_2^{HF}, x_3^{HF}\}.$$

The corresponding responses are Y_0 and Y_1 where $Y_0 = f_{LF}(X_0)$ and $Y_1 = f_{HF}(X_1)$. MFEGO algorithm is used to sequentially search through the space to minimize the maximum of EI. The algorithm has at each step two choices:

1) query once the low fidelity only,
2) query once the low fidelity and once the high fidelity if the EI cannot be reasonably reduced by a low-fidelity query.



(a) Initial DOE (b) 2nd iteration



(c) 4th iteration (d) 7th iteration

Fig. 2 Evolution of EI throughout MFEGO iterations on the 1-D analytic problem [7].

Figure 2 shows the reduction of EI resulting from cheap exploration, high-fidelity exploitation and model enhancement. After adding 4 LF points to explore the space and reduce EI (some low-fidelity points can be indistinguishable due to their proximity in the images above), the image (c) shows the local exploitation and enhancement of the model by querying the high-fidelity code once. The next HF sample finds the global optimum of the function. To summarize, we find the optimum after 7 iterations: 7 LF samples and 2 HF samples have been added, whereas the initial DOE was 6 LF points and 3 HF points (summary in Table 1). The classical mono-fidelity EGO approach with an initial DOE of 4 HF points required 11 additional HF points to find the global optimum, with a total cost three times higher than the MFEGO cost (15 compared to 5.013).

	HF DOE	LF DOE	HF Opt	LF Opt	Cost
MFEGO	3	6	2	7	5.013
EGO	4	-	11	-	15

Table 1 Analytic problem optimization summary.

To finish, we highlight that the *choice of level criterion* can be used with any exploration-versus-exploitation algorithm as the same idea can be applied. We use the low fidelity to reduce the uncertainty of the model and thus reduce the Exploration contribution to the *infill sampling criterion*. The highest fidelities are then used for the exploitation and effectively minimize the objective.

IV. Airfoil shape optimization

To validate MFEGO (resp. MFSEGO) and compare it to EGO (resp. SEGO) and later gradient-based approaches, three optimization problems have been considered:

1) 9-D unconstrained airfoil shape optimization
2) 15-D unconstrained airfoil shape optimization
3) 15-D airfoil shape optimization with 2 equality constraints

A supervised (analytic) test case is also provided in Appendix (2-D Rosenbrock function).

A. Test case description

For the airfoil shape optimization test case, we used a parametrization based on a mode decomposition proposed by Li et al. [20]. This decomposition is a Singular Value Decomposition (SVD) on the camber and thickness of an airfoil database. We use this parametrization to define an airfoil geometry for which we can compute characteristics such as the Lift Coefficient C_l, Drag Coefficient C_d, and Pitching Moment C_m. The goal is to find the global optimal geometry of the airfoil as computed by the high-fidelity code ADflow.[†] ADflow has a Reynolds Averaged Navier–Stokes (RANS) multi-block flow solver developed in the MDO Lab (University of Michigan) that has been successfully applied to a variety of aerodynamic shape optimization problems [21–24].

To help MFEGO find the optimum at the lowest cost possible, we use a low-fidelity code Xfoil [25], which takes a fraction (1/200) of the HF code time to give an approximation of the result using hypotheses simplifying the problem (linearization, ignoring shocks, compression corrections...). Although Xfoil gives results that are different from ADflow, it has a strong correlation with the RANS solver. As they both represent the same physical problem, the same trends can be observed (Lift and Drag tend to increase when we increase the Angle of Attack for example). We use the correlation between the low and high fidelities to improve the accuracy of the model and perform faster optimizations.

ADflow compute derivatives efficiently thanks to a discrete adjoint method implementation [26]. This has allowed us to have a gradient-based optimization reference to compare the Bayesian optimization approaches with, especially considering that 2D airfoil optimizations are unimodal [23].

To solve such optimization problems, work has been done on the noise estimation and regression/re-interpolation of the multi-fidelity surrogate model based on the work of Forrester et al. [7]; when using multi-fidelity analyses, we can modify the interpolating co-kriging formulation in Eq. (4) such that each analysis can be regressed appropriately to filter any noise present in the data [7]. In practice, this is done by adding a noise term λ to the diagonal of the covariance matrix and estimating it by maximum of likelihood [15, 16].

B. 9-D unconstrained optimization

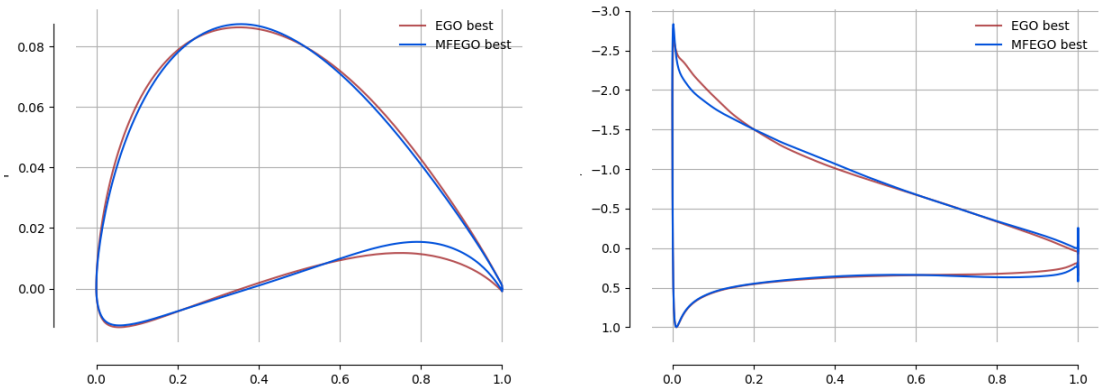
The main characteristics of the first airfoil shape optimization problem are summarized in Table 2.

	Function/variable	Description	Quantity	Range
maximize	L/D	Lift-to-Drag ratio	1	
with respect to	α	Angle of attack (deg)	1	$[0.0, 8.0]$
	θ	Thickness modes	4	$[0, 1]$
	δ	Camber modes	4	$[0, 1]$
		Total variables	**9**	

Table 2 Definition of the 9-D unconstrained optimization problem.

The modes design variables have been normalized to fit the segment $[0, 1]$. The SVD gives as output basis vectors that can be scaled so that the design variable associated to a particular mode has a range of $[0, 1]$. In practice the scaled basis vectors were chosen not only to allow the reconstruction of the airfoil database used by Li et al. [20], but also an additional margin was added to allow further exploration of new geometries. The Mach number (set to 0.25) and Reynolds number (set to 6×10^6) remain constant. With a cost ratio of 1/200, we obtain the results presented in Figure 3 and Table 3.

[†] `https://github.com/mdolab/adflow`



(a) Airfoil shape

(b) C_p distribution

Fig. 3 Comparison of EGO and MFEGO resulting airfoil shapes and C_p distributions for an unconstrained L/D maximization with 9 design variables.

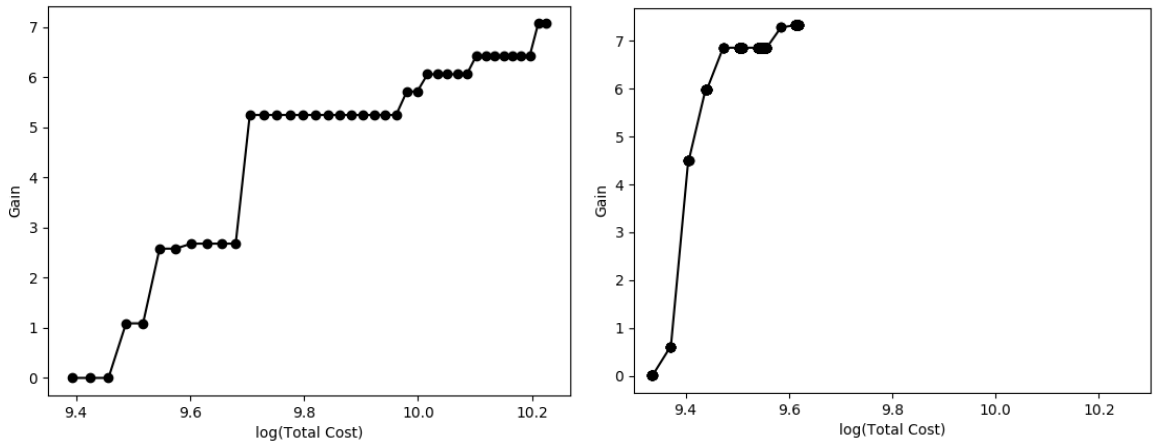
	HF DOE	LF DOE	HF Opt	LF Opt	Cost	Obj
EGO	30	-	40	-	70	108.44
MFEGO	28	58	8	214	37.41	109.94

Table 3 L/D maximization: Comparison of cost and objective for EGO and MFEGO unconstrained optimization with 9 design variables.

The columns HF DOE and LF DOE in Table 3 denote the number of samples of each fidelity used to build the initial design of experiment (DOE). The columns HF Opt and LF Opt are respectively the number of high-fidelity (HF) and low-fidelity (LF) calls that the optimization routine used after that. The cost is a cost normalized by the HF cost, so that HF calls contribute 1 to the total cost, and LF calls contribute $1/200$. It can be directly interpreted as the CPU time needed to reach the solution. We should note that EGO and MFEGO were stopped before 'convergence' (Bayesian optimization doesn't have a characterization of optimum, a budget of iterations is generally imposed as stopping criterion) and that at 'convergence', EGO and MFEGO should give the same optimum. But we can see from Table 3 that with lesser calls we are able to obtain a better *best known solution*. To understand the reason MFEGO performs better than EGO, let us define the series '*Gain*' of a Bayesian optimization at iteration i as:

$$\text{Gain}_i = |sol_i - sol_{DOE}|$$

where sol_i is the best known solution after the i^{th} iteration and sol_{doe} is the best known solution resulting from the random sampling of the initial DOE. We use the *Gain* to compare the efficiency of the EGO and MFEGO algorithms.



(a) EGO (b) MFEGO

Fig. 4 Comparison of '*Gain*' of EGO and MFEGO for unconstrained L/D maximization (9 design variables).

Figure 4 shows the *Gain* (improvement of the objective function compared to the start of the optimization) as a function of the cost. We see that EGO tends to have long plateaus where the objective does not improve, whereas MFEGO tends to improve at (almost) every HF call. This is due to the fact that EGO is an exploitation/exploration compromise. By using the lower-fidelity, MFEGO can explore the design space more cheaply, leaving HF calls for effective improvement of the objective. This is particularly important when increasing the dimension of the problem. As we increase the number of design variables, the space to explore becomes bigger. Using multiple fidelities allows MFEGO to scale better than EGO w.r.t the number of design variables.

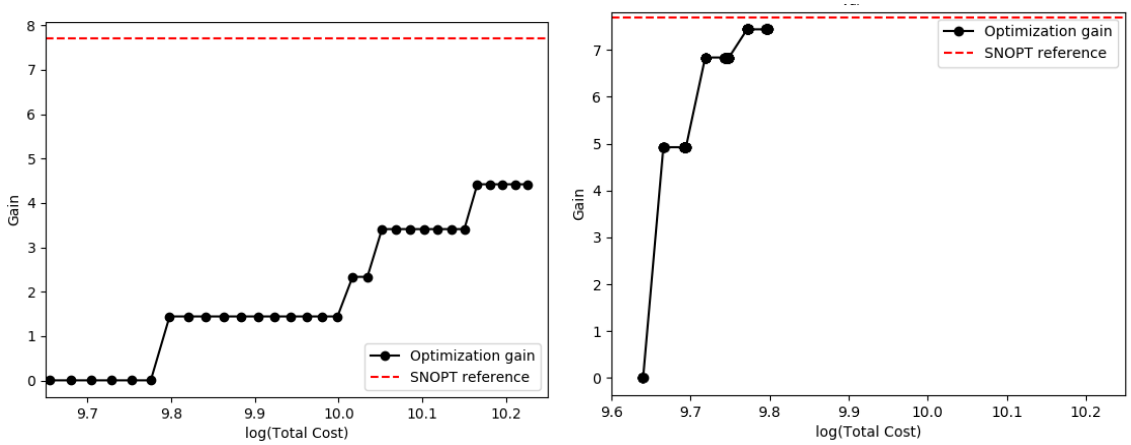
C. 15-D unconstrained optimization

We showcase the better scaling by solving the same shape optimization problem with more design variables are described in Table 4.

	Function/variable	Description	Quantity	Range
maximize	L/D	Lift-to-Drag ratio	1	
with respect to	α	Angle of attack	1	$[0.0, 8.0]$ (°)
	θ	Thickness modes	7	$[0, 1]$
	δ	Camber modes	7	$[0, 1]$
		Total variables	**15**	

Table 4 Definition of the 15-D unconstrained optimization problem.

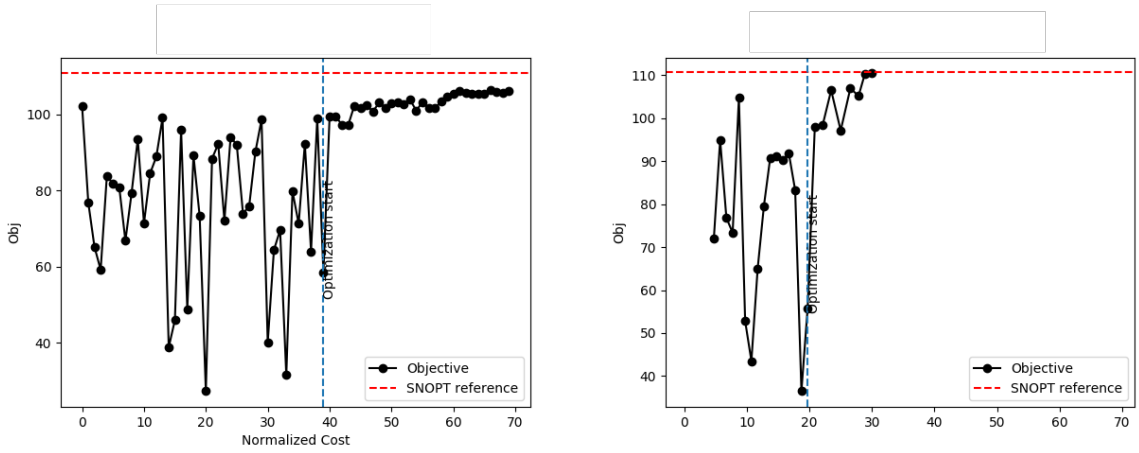
The Mach number (set to 0.25) and Reynolds number (set to 6×10^6) remain constant. As this particular problem is convex and unimodal, the local optimum found by SNOPT is also the global optimum of the function (red dotted line in Figure 5). We use this optimum as a reference to show the distance between the true optimum and the current solution of EGO and MFEGO. We can see from Figure 5 that the additional dimensions result in EGO spending a lot of time on the exploration of the space of the design variables. MFEGO is more immune to this trend as it can use the lower fidelity to explore the space before committing a HF computation. This observation is consistent across multiple runs.



(a) EGO (b) MFEGO

Fig. 5 Comparison of '*Gain*' of EGO and MFEGO for unconstrained L/D maximization (15 Design Variables). The red dashed line is the optimum found by SNOPT.

One way to make further use of the multi-fidelity kriging is to reduce the cost of the initial DOE. For example, instead of using 40 HF (for a total cost of 40), we use 16 HF and (approximately) 800 LF (for a total equivalent cost of (approximately) 20). We see in Figure 6 that by doing so, the space is better mapped, and that this improves greatly the speed of convergence at a much lower cost. For this problem we had access to the gradient-based optimum using SNOPT [14]. The dashed vertical blue line in Figure 6 separates the initial DOE building phase and the optimization driven by the Bayesian algorithm phase that comes after that.



(a) EGO (b) MFEGO

Fig. 6 Comparison of evolution of objective as function of iterations for EGO and MFEGO for unconstrained L/D maximization with 15 design variables.

Figure 6 shows that MFEGO gives better results than EGO even with a less expensive initial DOE. The low fidelity, much cheaper, contributes some information to the surrogate model. MFEGO searched through the design space using 437 LF points and probed 8 HF points to attain the SNOPT solution (110.7 for SNOPT against 110.5 for MFEGO). We summarize the results in Table 5.

	HF DOE	LF DOE	HF Opt	LF Opt	Cost	Obj
SNOPT (ref)	-	-	21	-	21	110.7
EGO	40	-	30	-	70	104.9
MFEGO	16	744	8	437	29.89	110.5

Table 5 L/D maximization: Comparison of cost and objective for EGO and MFEGO unconstrained optimization with 15 design variables. The reference value is obtained by SNOPT.

D. Constrained optimization

We have shown that for unconstrained optimizations, MFEGO gives superior results to EGO due to the availability of cheap exploration of the design space. We show in what follows the advantages of multi-fidelity surrogate modeling for constrained optimization in an approach called MFSEGO. Not only the lower fidelities give access to cheaper exploration, but the overall model is more accurate resulting in a smaller constraints violation. First, let us introduce the measure root mean squared constraint violation (RMSCV) for an equality constraint as:

$$\text{RMSCV} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\text{val}_j - \text{target}\right)^2}$$

where N is the number of iterations of the algorithm, val_j the value of the constraint function at iteration j and target is the target value of the constraint. A smaller RMSCV means that, on average, the algorithm respects the constraints more. We compare MFSEGO with SEGO and SNOPT through the following constrained optimization problem given in Table 6.

	Function/variable	Description	Quantity	Range
minimize	C_d	Drag Coefficient	1	
with respect to	α	Angle of attack	1	[0.0, 8.0] (°)
	θ	Thickness modes	7	[0, 1]
	δ	Camber modes	7	[0, 1]
		Total variables	**15**	
subject to	$C_L = 0.5$	Lift coefficient	1	
	$C_m = 0$	Pitching moment	1	
		Total constraints	**2**	

Table 6 Definition of the 15-D constrained optimization problem.

The Mach number (set to 0.25) and Reynolds number (set to 6×10^6) remain constant. To test the robustness of the algorithm we add a Pitching Moment (C_m) constraint. Note that the AIAA Aerodynamic Design Optimization Discussion Group [27] proposes a pitching moment inequality constraint for the RAE2822 airfoil test case. We transformed this constraint into an equality constraint to make sure it is active on our problem. For a design to be accepted, we require that the constraint violation must be below 10^{-3} for the absolute value of each of the two constraints.

In Table 7, the cost of SNOPT optimizations combines the number of direct problems solved (23), and the number of adjoint problems solved to compute derivatives ($23 \times 3 = 69$). The overall cost is then calculated as:

$$\text{cost}_{\text{SNOPT}} = \frac{\text{time}_{\text{Direct}} + \text{time}_{\text{Adjoint}}}{\text{time}_{\text{Direct}}} \times N_{iter}$$

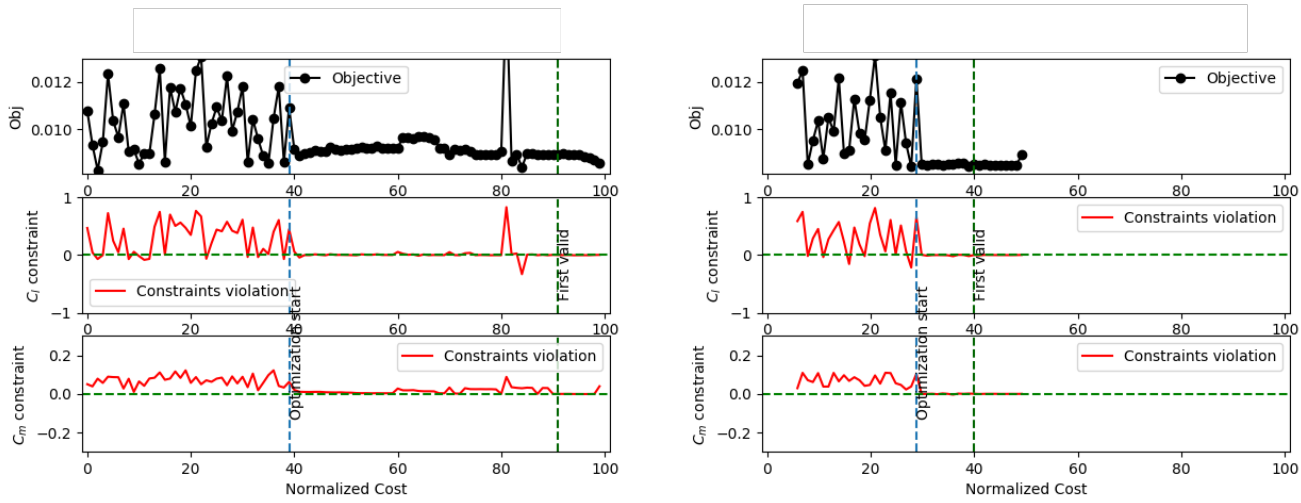
where $\text{time}_{\text{Direct}}$ is the time needed for solving the N_{iter} direct problems, and $\text{time}_{\text{Adjoint}}$ is the time needed to solve $N_{iter} \times N_{funcs}$ adjoint problems. N_{funcs} is the number of functions evaluated, here 3: C_l for the objective, and C_d and C_m for the constraints. N_{iter} was 23 in this case. This SNOPT cost is compared to the cost of SEGO and MFSEGO in Table 7.

	HF DOE	LF DOE	HF Opt	LF Opt	Cost	Obj	Feasible	RMSCV
SNOPT (ref)	-	-	73	-	73	84.68	Yes	-
SEGO	40	-	60	-	100	89.188	Yes	8.8e-2
MFSEGO	24	964	18	63	47.135	84.67	Yes	4.9e-3

Table 7 Comparison of SEGO and MFSEGO for constrained optimization with 15 design variables. The reference value is obtained by SNOPT.

Table 7 shows that MFSEGO is more robust to the addition of new constraints. This is due to the fact that the low fidelity helps in the surrogate modeling of the C_m constraint as well. As a result, MFSEGO shows a 10 times lower constraint violation (RMSCV of 4.9e-3 for MFSEGO against 8.8e-2 for EGO). We should note that the small difference between SNOPT and MFSEGO in the objective should be attributed to the constraints tolerance. We should also point out that the first feasible solution found by EGO was after 52 HF iterations, whereas MFSEGO only needed 11 HF samples before finding its first feasible solution as shown in Figure 7 (vertical green dashed line). MFSEGO performed better than EGO and SNOPT in terms of cost. SNOPT needed 23 direct problems resolutions and 69 adjoint problems to find its optimal solution. In Figure 7 we show the evolution of the objective and constraint throughout the iterations of EGO and MFSEGO. The blue dashed line separates the initial DOE phase (random sampling) and the optimization phase. We show that MFSEGO has less constraints violations (less variations around the green dashed line denoting less constraint violations).

These three airfoil shape optimization problems demonstrated the good behaviour of the proposed multi-fidelity approach compared to classical Bayesian algorithms. Nevertheless, investigations will be carried on to increase both the Mach number, and the dimension of the design space.



(a) SEGO

(b) MFSEGO

Fig. 7 Comparison of objective and constraints function of iterations for SEGO (a) and MFSEGO (b) for optimization of C_d subject to C_l and C_m constraints with 15 design variables.

V. Conclusion

In this paper, we formulate an extension of Bayesian optimization algorithms to work with multi-fidelity information sources. This approach is successfully coupled to EGO and SEGO to solve both unconstrained and constrained problems. In this approach, the number of high-fidelity calls is limited as the low fidelity is used to favor the exploration phase and thus costly high-fidelity simulation is only dedicated to the exploitation phase. We demonstrate the efficiency of Bayesian optimization in several test cases by enhancing the design space exploration part. For both the unconstrained or constrained optimization problems, MFEGO and MFSEGO found better results at a 50% lower computational cost

compared to EGO. All problems have been treated within the SEGOMOE framework that is now extended with this new capability to handle multi-fidelity.

Future work will integrate the gradient information in the surrogate model and possibly in the devising of new criteria to allow for faster optimization. The extension to MDO workflow is also a target with studies on multi-fidelity coupling between two disciplines.

Acknowledgments

This work is part of the activities of ONERA - ISAE - ENAC joint research group and was partially supported by a ONERA internal project MUFIN dedicated to multi-fidelity. The authors would like to thank the ISAE-Supaero Foundation for its financial support and its role in making this project possible. The authors are also grateful to Rémi Lafage for his support with the SMT implementation and Rémy Priem for his support with the SEGOMOE framework. Mohamed Bouhlel and Joaquim Martins were funded by the Air Force Office of Scientific Research (AFOSR) MURI on "Managing multiple information sources of multi-physics systems," program officer Jean–Luc Cambier, award number FA9550-15-1-0038.

Appendix

A. Rosenbrock optimization

We present here the results for the Rosenbrock function (See Figure 8) as it is a particularly challenging function to optimize (the function is extremely flat around the optimum). This optimization problem also allows us to compare MFEGO algorithm to some of the literature's state-of-the-art multiple information source optimization algorithms (not presented here, refer to [3]).

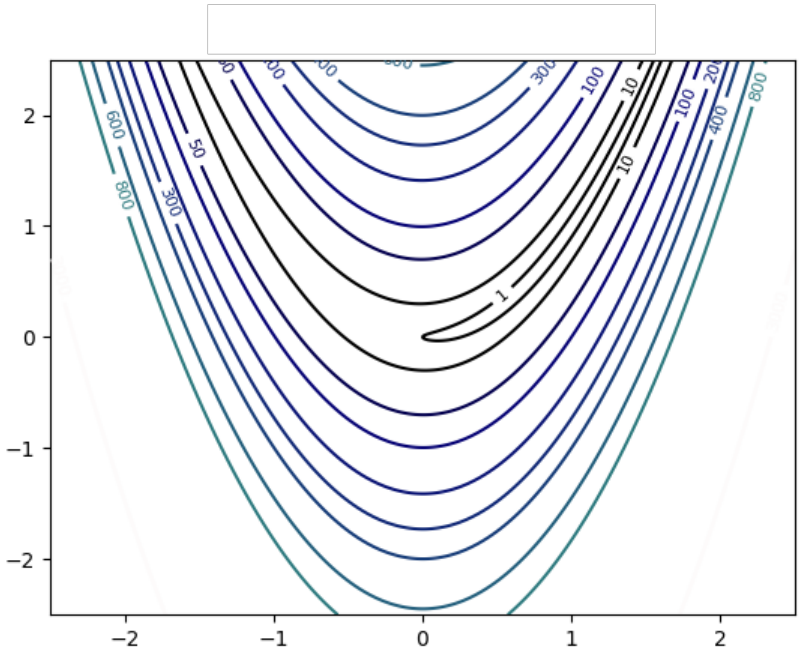
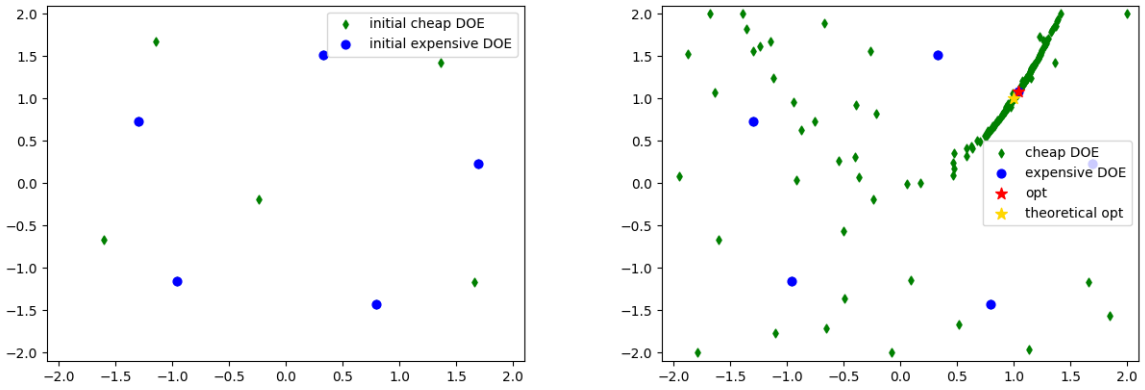


Fig. 8 Contour of the Rosenbrock function



(a) Initial DOE : 5 HF and 10 LF

(b) Final DOE : 5+1 HF and 10+200 LF

Fig. 9 MFEGO's inital and Final DOE of Rosenbrock function

We set up the optimization to match the one presented in [3] to have results that can be compared and to have a first order appreciation of the quality of the algorithm:

$$f_{HF}(x, y) = (1 - x)^2 + 100(y - x^2)^2,$$
$$f_{LF}(x, y) = f_{HF}(x, y) + 0.1 \sin(10x - 5y),$$

we assume a cost of 1000 for f_{HF} and 1 for f_{LF}.

The algorithm is started with a budget of 5 high-fidelity samples and 10 low-fidelity samples. Figure 9 shows MFEGO using the low fidelity to explore the space and more intensely the valley where the optimum is. It finally places a high-fidelity sample near the theoretical optimum of the function. It only needed one HF sample to do that.

This first result on a problem clearly cut for multiple information sources (assumption that only small variations exist between low and high fidelities) rather than multi-fidelities, was encouraging to carry on working on the algorithm and apply it to the aerodynamic shape optimization of an airfoil.

References

[1] Winkler, R. L., "Combining Probability Distributions from Dependent Information Sources," *Manage. Sci.*, Vol. 27, No. 4, 1981, pp. 479–488. doi:10.1287/mnsc.27.4.479, URL `https://doi.org/10.1287/mnsc.27.4.479`.

[2] Lam, R., Allaire, D. L., and Willcox, K. E., "Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources," *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2015, p. 0143.

[3] Poloczek, M., Wang, J., and Frazier, P. I., "Multi-Information Source Optimization," *NIPS*, 2017.

[4] Lewis, R., and Nash, S., "A multigrid approach to the optimization of systems governed by differential equations," *8th Symposium on Multidisciplinary Analysis and Optimization*, 2000, p. 4890.

[5] Huang, D., Allen, T. T., Notz, W. I., and Miller, R. A., "Sequential kriging optimization using multiple-fidelity evaluations," *Structural and Multidisciplinary Optimization*, Vol. 32, No. 5, 2006, pp. 369–382. doi:10.1007/s00158-005-0587-0, URL `https://doi.org/10.1007/s00158-005-0587-0`.

[6] Kennedy, M. C., and O'Hagan, A., "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 63, No. 3, 2001, pp. 425–464.

[7] 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.

[8] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *J. of Global Optimization*, Vol. 13, No. 4, 1998, pp. 455–492. doi:10.1023/A:1008306431147, URL `https://doi.org/10.1023/A:1008306431147`.

[9] Le Gratiet, L., "Multi-fidelity Gaussian process regression for computer experiments," Thesis, Université Paris-Diderot - Paris VII, Oct. 2013. URL https://tel.archives-ouvertes.fr/tel-00866770.

[10] Sasena, M., "Flexibility and efficiency enhancements for constrained global design optimization with Kriging approximations," Ph.D. thesis, University of Michigan, 2002.

[11] Bartoli, N., Kurek, I., Lafage, R., Lefebvre, T., Priem, R., Bouhlel, M., Morlier, J., Stilz, V., and Regis, R., "Improvement of efficient global optimization with mixture of experts: methodology developments and preliminary results in aircraft wing design," *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, At Washington DC*, 2016.

[12] Bartoli, N., Lefebvre, T., Dubreuil, S., Olivanti, R., Bons, N., Martins, J., Bouhlel, M.-A., and Morlier, J., "An adaptive optimization strategy based on mixture of experts for wing aerodynamic design optimization," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017, p. 4433.

[13] Bartoli, N., Lefebvre, T., Dubreuil, S., Olivanti, R., Priem, R., Bons, N., Martins, J. R. R. A., and Morlier, J., "Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design," *Aerospace Science and Technology*, 2019. (In press).

[14] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM review*, Vol. 47, No. 1, 2005, pp. 99–131.

[15] Forrester, A., Sobester, A., and Keane, A., *Engineering design via surrogate modelling: a practical guide*, John Wiley & Sons, 2008.

[16] Rasmussen, C. E., and Williams, C. K., *Gaussian processes for machine learning*, Vol. 1, MIT press Cambridge, 2006.

[17] Zhang, Y., Kim, N. H., Park, C., and Haftka, R. T., "Multifidelity Surrogate Based on Single Linear Regression," *AIAA Journal*, 2018, pp. 1–9.

[18] Bouhlel, M. A., Hwang, J. T., Bartoli, N., Lafage, R., Morlier, J., and Martins, J. R. R. A., "A Python surrogate modeling framework with derivatives," *Advances in Engineering Software*, 2019. doi:10.1016/j.advengsoft.2019.03.005.

[19] Močkus, J., *On Bayesian Methods for Seeking the Extremum*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1975, pp. 400–404. doi:10.1007/978-3-662-38527-2_55, URL https://doi.org/10.1007/978-3-662-38527-2_55.

[20] Li, J., Bouhlel, M. A., and Martins, J. R. R. A., "Data-based Approach for Fast Airfoil Analysis and Optimization," *Journal of Aircraft*, Vol. 57, No. 2, 2019, pp. 581–596. doi:10.2514/1.J057129.

[21] Lyu, Z., and Martins, J. R. R. A., "Aerodynamic Design Optimization Studies of a Blended-Wing-Body Aircraft," *Journal of Aircraft*, Vol. 51, No. 5, 2014, pp. 1604–1617. doi:10.2514/1.C032491.

[22] Lyu, Z., Kenway, G. K. W., and Martins, J. R. R. A., "Aerodynamic Shape Optimization Investigations of the Common Research Model Wing Benchmark," *AIAA Journal*, Vol. 53, No. 4, 2015, pp. 968–985. doi:10.2514/1.J053318.

[23] Yu, Y., Lyu, Z., Xu, Z., and Martins, J. R. R. A., "On the Influence of Optimization Algorithm and Starting Design on Wing Aerodynamic Shape Optimization," *Aerospace Science and Technology*, Vol. 75, 2018, pp. 183–199. doi:10.1016/j.ast.2018.01.016.

[24] Bons, N., He, X., Mader, C. A., and Martins, J. R. R. A., "Multimodality in Aerodynamic Wing Design Optimization," *AIAA Journal*, Vol. 57, No. 3, 2019, pp. 1004–1018. doi:10.2514/1.J057294.

[25] Drela, M., "XFOIL: An analysis and design system for low Reynolds number airfoils," *Low Reynolds number aerodynamics*, Springer, 1989, pp. 1–12.

[26] Kenway, G. K. W., Mader, C. A., He, P., and Martins, J. R. R. A., "Effective Adjoint Approaches for Computational Fluid Dynamics," *Progress in Aerospace Sciences*, 2019. (Submitted).

[27] "Aerodynamic Design Optimization Discussion Group Website," https://info.aiaa.org/tac/ASG/APATC/AeroDesignOpt-DG/default.aspx, 2014. Accessed: 2018-03-28.