

DATA-DRIVEN GEOMETRIC FILTRATION FOR AERODYNAMIC SHAPE OPTIMIZATION

DAVIDE CINQUEGRANA¹ AND EMILIANO IULIANO²

¹ Computational Fluid Dynamics Lab, Fluid Mechanics Unit,
via Maiorise snc 80143 Capua (ITA)
d.cinquegrana@cira.it, www.cira.it

² Multi-Disciplinary Optimization Lab, Fluid Mechanics Unit,
via Maiorise snc 80143 Capua (ITA)
e.iuliano@cira.it, www.cira.it

Key words: Design Space Reduction, Aero Shape optimization, Proper Orthogonal Decomposition, Curse of Dimensionality, Geometric Filtration

Abstract. The aim of the paper is to investigate the geometric filtration by Proper Orthogonal Decomposition in order to reduce the dimensionality of the design space in an aerodynamic shape optimization problem. Thanks to the capability of re-ordering the data according to decreasing variance, the Proper Orthogonal Decomposition extracts and filters the basic features of the dataset so as to formulate the optimization problem into a new, shrunk design space. The capability of the new optimal bases in representing different airfoil shapes is critically analyzed, as well as the proper choice of the new design space bounds. Furthermore, an updating algorithm is designed to adapt the basis during the optimization. A series of optimization results are shown for a two-dimensional test case by using a genetic algorithm and surrogate-based optimization. Both native geometry parameterization and the filtered one are used to highlight benefits and drawbacks of the proposed approach. Results show that, by employing the filtered parameterization, it is possible to achieve and even enhance the design performance attained with the classical parameterization, whatever the optimizer employed.

1 Introduction

In the framework of global optimization approaches for aerodynamic design, a shape is usually parameterized by acting on the design variables (DV) whose number and bounds generate the design space. However, when increasing the geometry complexity of the problem at hand, the number of design variables may grow up significantly, giving rise to the well-known 'curse of dimensionality'[1].

This may hamper the success of the optimization process, as large design spaces may lead to multi-modal and noisy landscapes. As a consequence, the problem of searching for a global solution may result hard or even intractable, independently of the search method, i.e. deterministic or stochastic or meta-heuristic.

In surrogate-based optimization (SBO), the meta-model that mimics the behavior of the objective function is made explicitly dependent on the design variables. From literature, it is known that, whatever the meta-model, a loss in prediction quality is associated to an increase of dimensionality[2]. On the other hand, sensitivity analysis often shows that not all the design variables have the same influence on the objective function behavior. As a matter of fact, a reduction of the design space is envisaged to preserve the main features of the target function landscape while avoiding the high-dimensional noise.

Another issue is related to the huge amount of CPU required to face with an industrial aerodynamic design. Indeed, geometric complexities and non-linear flow features (e.g., flow separation, shock waves, shock-boundary layer interaction) still require to use fine mesh size and large number of CFD iterations to be correctly solved. An even more crucial bottleneck is represented by the search of a solution with a global algorithm, which may require a large number of CFD evaluations. Therefore, it is mandatory to reduce the problem complexity at each level of the design chain by means of the parallelization of the CFD solver, the usage of surrogate or reduced order models to evaluate the objective function and the reduction of the design space.

In literature, design space reduction (DSR) is treated with different strategies to shrink the bounds of the search volume like trust region approaches [3], heuristic, move limits[4], Variable-Complexity Response Surface Modeling Method (VCRSM), and Concurrent Sub Space Optimization (CSSO) [5]. For example, VCRSM as proposed by Zahir[6] suggests a preliminary exploration of the design space with a low fidelity model, then selects the new reduced bounds with a pattern search coupled with high fidelity simulation and integration of data. CSSO is a subspace coordination procedure: here the data generated by the subspace optimizers is not uniformly centered about the current design, but instead follows the descent path of the subspace optimizers[5, 7]. The common feature of these techniques is the reduction of the search volume and the associated relieving effect, mostly if the optimization is driven by surrogate models.

In another perspective, DSR can be viewed also as a reduction of the inherent dimensionality of the problem, i.e. a technique to identify a lower dimension that effectively and efficiently drives the problem at hand. The aim is to learn from a given sampling set of the original design space and to map it to a lower dimensional space while keeping the most important features of the original design problem. Various strategies have been proposed in literature to seek an equivalent lower dimensional representation, even if some losses are introduced. Proper orthogonal decomposition (POD), kernel POD, locally linear embedding, Laplacian eigenmaps, isomap and semidefinite embedding are just some examples [8, 9].

The aim of the paper is to apply a geometric filter on the design variables set by performing a variance analysis based on Proper Orthogonal Decomposition. Exploiting the intrinsic properties of POD, the geometric data are re-arranged and ranked according to their relative importance, thus deriving a new parameterization in a transformed space. The beneficial aspect of this transformation is two-fold as it brings a quantitative and qualitative filtering at the same time: first, as a result of POD ranking and truncation,

the new design space is shrunk and the effective size of the problem at hand is reduced (quantitative filtering); second, the new geometric modes are orthogonal and, hence, independent [10]: this removes all possible spurious coupling between original design variables (qualitative filtering). The advantages of the linear technique to data reduction have been investigated in recent literature, showing good performances if compared to other non linear techniques [11, 8]. Concerning how to feed the filter, recent works proposed to set up the training geometry database with airfoil shapes from aeronautical libraries [12, 13] and, specifically, to select them according to the operating points of the optimization problem.

The present authors already proposed the application of the DSR technique to two- and three-dimensional surrogate-based optimization: in particular, in the first investigation [14] the optimization of an airfoil in subsonic viscous flow was studied; in the second [15], the DSR was applied to an industrial case, i.e. a wing shape optimization in multi-point, transonic and viscous conditions. The last application represents quite a novelty with respect to recent papers [16, 17, 18, 12].

In this perspective, the present paper represents a step inwards the full understanding of its potential and exploitation. Indeed, a more pronounced emphasis is put on a twofold aspect: on one hand, the reduced design variables setting and adaptation during the optimization process, thus adjusting the reduced basis with the ongoing search results; on the other hand, addressing the low dimensional mapping in a more local sense, that is driving it to promising and narrow regions of the reduced design space. Both improvements are effective in confining the randomness and the dispersion of the initial geometry database, obtained through a pseudo-random Latin Hypercube Sampling.

The paper is organized as follows: the next section will provide details about the DSR technique based on POD; afterwards, the optimization process and the POD basis updating strategy are presented; finally, a quite extensive section is devoted to the presentation of the results of the GA-based and surrogate-based optimizations.

2 Geometric Data Reduction by POD

Proper Orthogonal Decomposition is a technique to compute a linear basis of vectors which is optimal in some sense. In other words, it provides the best representation of a given dataset in a different reference frame originated by the POD basis vectors. A detailed mathematical formulation for the derivation can be found in [19, 20]. Here, the fundamental concepts and relations are recalled. The dataset is organized in a snapshot matrix $\mathbf{S} = \{S_1, S_2, \dots, S_n\}$, which collects the available data in a column-wise manner, where n is the number of experiments or numerical simulations. Here, it is assumed that each snapshot element S_k is obtained by varying a set of parameters $\mathbf{w} = \{w_1, \dots, w_m\}$ which, in the present case, represent the design variables of the optimization problem at hand. The combination of the sets $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n\}$ and $\{S_1, S_2, \dots, S_n\}$ represent the so-called training dataset of database.

Computing a POD basis consists in searching a set of orthonormal vectors $\phi_1, \phi_2, \dots, \phi_n$

such that the following relation holds:

$$S_j = \bar{S} + \sum_{i=1}^n \alpha_i(\mathbf{w}_j) \phi_i = \bar{S} + \sum_{i=1}^{\hat{n}} \alpha_i(\mathbf{w}_j) \phi_i + \epsilon_{\hat{n}}^j = \bar{S} + \sum_{i=1}^{\hat{n}} (S_j, \phi_i) \phi_i + \epsilon_{\hat{n}}^j$$

where \bar{S} is a base solution [21, 20] (i.e. an average field), the scalars α_i are the unknown POD coefficients, $\hat{n} \leq n$ and the error ϵ is optimal", i.e. the smallest possible. Indeed, for any set of orthonormal vectors $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \dots, \boldsymbol{\psi}_n$ the optimality condition is expressed as:

$$\epsilon_{\hat{n}} = \sum_{j=1}^n S_j - \bar{S} - \sum_{i=1}^{\hat{n}} (S_j, \phi_i) \phi_i \leq \sum_{j=1}^n S_j - \bar{S} - \sum_{i=1}^{\hat{n}} (S_j, \psi_i) \psi_i$$

In practice, the POD decomposition is obtained by taking the singular value decomposition (SVD) of $\mathbf{P} = \{S_1 - \bar{S}, S_2 - \bar{S}, \dots, S_n - \bar{S}\}$:

$$\mathbf{P} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T = \mathbf{U} \begin{pmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_M \\ 0 & \cdots & 0 \end{pmatrix} \mathbf{V}^T \quad (1)$$

with the singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_M \geq 0$. The POD basis vectors, also called POD modes or eigen-functions, are the first n column vectors of the matrix \mathbf{U} , while the POD coefficients $\alpha_i(\mathbf{w}_j)$ are obtained by projecting the snapshots onto the POD modes:

$$\alpha_i(\mathbf{w}_j) = (S_j - \bar{S}, \phi_i) \quad (2)$$

Usually, the singular values become small rapidly and a small number of basis vectors are adequate to reconstruct and approximate the snapshots as they preserve the most significant ensemble energy contribution. In this way, POD provides an efficient mean of capturing the dominant features of a multi-degree of freedom system and representing it to the desired precision by using the relevant set of modes. The reduced order model is derived by projecting the original model onto a reduced space spanned by only some of the proper orthogonal modes or POD eigenfunctions. This process realizes a kind of lossy data compression by truncating the POD sum to $\hat{n} < n$

$$S_j \simeq \bar{S} + \sum_{i=1}^{\hat{n}} \alpha_i(\mathbf{w}_j) \phi_i \quad (3)$$

A criterion to truncate the sum in equation 3 is usually based on the total information content, which is given by the following equation:

$$E_{\hat{n}} = \frac{\sum_{j=1}^{\hat{n}} \sigma_j^2}{\sum_{i=1}^n \sigma_i^2} \quad (4)$$

Of course, if $\hat{n} = n$ then $E_{\hat{n}}$ is equal to unity and all modes are preserved. The amount of E_p can be defined as a spectral energy content. Often the threshold value is set to 0.99, i.e. 99% of the information content is retained, so that $E_{\hat{n}} \geq 0.99$ and $E_{\hat{n}-1} < 0.99$. The complex, non linear behavior of the original model (e.g., a CFD parametric model) is thus transformed to a linear sum that is still able to optimally represent the original dataset while employing a reduced number of degrees of freedom (i.e. \hat{n} scalars α_k and modes $\phi_k, k = 1, \dots, \hat{n}$).

In the present context, the POD approximation is applied to geometry coordinates data (i.e., airfoil or wing shapes) coming from a classical geometry parameterization, here referred to as native" parameterization. Given m the dimension of the design space and a set of n design vectors $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n\}$, the geometry parameterization can be seen as a correspondence which transforms a generic design vector in a set of p geometric coordinates:

$$\mathbf{w} \in \mathbb{R}^m \rightarrow \mathbf{s} \in \mathbb{R}^p$$

It is assumed that the parametric geometry is defined as a finite set (of size p) of (x, y, z) coordinates in a Cartesian space and that the snapshot vector \mathbf{s} contains only the z -coordinate as the (x, y) data are fixed when varying the design vector \mathbf{w} :

$$\mathbf{s}_i = \mathbf{s}(\mathbf{w}_i) = \{z_{i,1}, \dots, z_{i,p}\}^T$$

Once built the snapshot matrix $\mathbf{S} = [\mathbf{s}_1, \dots, \mathbf{s}_n]$, it is decomposed into a mean and a deviation matrices, as mentioned above :

$$\mathbf{S} = \bar{\mathbf{S}} + \Phi A$$

where $\bar{\mathbf{S}} = \{\frac{1}{n} \sum_1^n \mathbf{s}_i, \dots, \frac{1}{n} \sum_1^n \mathbf{s}_i\}$ is the mean matrix having the same column replicated n times, Φ is the POD modes matrix containing n spatial eigen-functions of the matrix $\mathbf{C} = (\mathbf{S} - \bar{\mathbf{S}})^T(\mathbf{S} - \bar{\mathbf{S}})$ and A is the matrix of the corresponding scalar coefficients

$$\mathbf{A} = \{\mathbf{a}_1, \dots, \mathbf{a}_n\} \tag{5}$$

$$\mathbf{a}_1 = \{\alpha_1(\mathbf{w}_1), \dots, \alpha_1(\mathbf{w}_n)\}^T \tag{6}$$

$$\vdots$$

$$\mathbf{a}_n = \{\alpha_n(\mathbf{w}_1), \dots, \alpha_n(\mathbf{w}_n)\}^T$$

In analogy to what has been explained above, a reduced number of bases can be selected in order to meet a minimum required information content. Hence, the original snapshot vectors can be approximated by:

$$\mathbf{S} \approx \bar{\mathbf{S}} + \tilde{\Phi} \tilde{A} \tag{7}$$

where $\tilde{\Phi}$ and \tilde{A} are obtained by Φ and A by considering only the $\tilde{n} < n$ first (and hence most important) basis vectors and corresponding modal coefficients.

This procedure is also referred to as geometric filtration as the most significant components of the geometric modification are extracted, so as to neglect possible spurious effects. Indeed, by condensing the geometric data into the first modes, the core of the

geometric filtration consists in interpreting the corresponding modal coefficients as transformed design variables. By acting directly on the \tilde{n} POD modal coefficients instead of on the original m design variables, an optimization algorithm can benefit from the dimensionality reduction in having $\tilde{n} < m$. As it will be made clearer in the next sections, the bounds of the newly-defined design variables strongly depends on the composition of the training dataset and can be set according to engineering considerations concerning the maximum allowable shapes or in a more smart way, that is adapting them to the optimization history collected so far.

3 Optimization methods

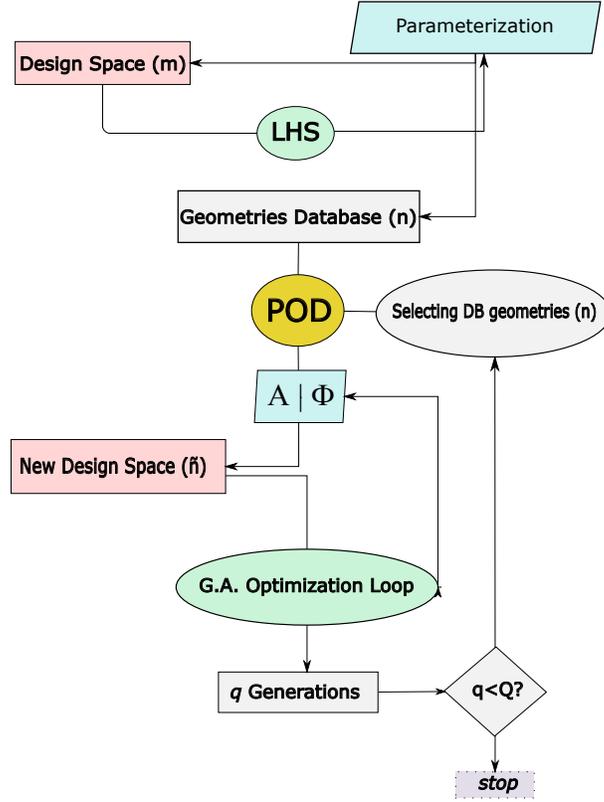
In the following, two optimization methods are presented which share the common objective of working onto the space spanned by the POD modes but differ in theoretical aspects. The first one relies on a genetic algorithm whose evolution is stopped several times to recompute and update the POD basis; the second is assisted by local surrogate models and trust region approaches to accelerate the search process.

3.1 GA-based optimization

In this section the shape optimization method assisted by POD-driven design space reduction is introduced. For the sake of clarity, it is conceived as two nested loops: the outer loop is devoted to the POD-based geometric filtration, while in the inner loop the optimization algorithm explores the reduced design space resulting from the outer loop.

The work-flow in figure 1 illustrates the whole optimization process. The procedure may work in two modes, with and without the POD basis updating stage. In the first case, the optimization search is stopped after a certain number of generations, a selection is made among the candidates computed so far, a new training database is obtained and the eigen-modes are re-computed and updated. In the second case, a single, full-length optimization is performed on the reduced design space originated by the initial POD basis. The following steps can be identified:

1. design space initialization: the original design space of size m , generated by the native design variables, is initially sampled by means of Latin Hypercube Sampling (LHS) strategy and n samples are collected in the ensemble database $\{\mathbf{w}_1, \dots, \mathbf{w}_n\}$;
2. parameterization: by using the native parameterization, the design space samples are converted into geometric data and the snapshot matrix $\mathbf{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_n\}$ is collected;
3. geometric filtration: the POD basis of matrix \mathbf{S} is computed and the modes $\phi_i, i = 1, \dots, n$ are stored. A reduced design space is generated by the first \tilde{n} transformed design variables $\alpha_i, i = 1, \dots, \tilde{n}$. This phase is aimed only at collecting geometry shapes and generating the POD basis and coefficients, thus it does not foresee any objective function computation. The truncation of the POD basis is crucial in reducing the design variables number with respect to the original parameterization.


Figure 1: Optimization Flowchart with POD Updating

4. optimization: this step is devoted to searching a global optimum with an evolutionary algorithm (EA). Given $f : \mathbf{s}(\mathbf{w}) \in \mathbb{R}^p \rightarrow f \in \mathbb{R}$ the objective function, the minimization problem is set in the reduced form as

$$\begin{aligned}
 & \min_{\alpha_1, \dots, \alpha_{\hat{n}}} f[\mathbf{s}(\alpha_1, \dots, \alpha_{\hat{n}})] \\
 & \text{subject to} \quad lb_i < \alpha_i < ub_i, \quad i \in \{1, \dots, \hat{n}\} \\
 & \quad \quad \quad \mathbf{s}(\alpha_1, \dots, \alpha_{\hat{n}}) = \hat{\mathbf{s}} + \sum_{i=1}^{\hat{n}} \alpha_i \phi_i \\
 & \quad \quad \quad lb_i = \min_{\mathbf{a}_i} \alpha_i \\
 & \quad \quad \quad ub_i = \max_{\mathbf{a}_i} \alpha_i
 \end{aligned}$$

The lower and upper bounds (lb_i and ub_i) of the reduced design variables α_i are obtained by taking the minimum and maximum value among the elements of the coefficient vectors \mathbf{a}_i as specified by equation 5 and 6. The search process is carried out by means of an evolutionary algorithm from the in-house optimization library ADGLIB[22, 23]. The selection operator is a fixed-step random walk, while classical crossover and bit mutation operators are applied to generate new individuals. The elitist strategy is used to keep the best fit individual within the current population.

If the POD basis updating is selected, the algorithm is let evolve up to q generations. Given n_q the total number of updates foreseen over the entire loop and Q the total number of generations to be performed (a pre-defined parameter of the EA), q is computed as:

$$q = \frac{Q}{n_q + 1} \quad (8)$$

5. update check: if POD basis update is selected, the loop continues with step 6; otherwise, a termination condition occurs;
6. generation check: check if the current number of generations q has exceeded the total budget Q , if the answer is negative, then n geometries are selected among the candidates evaluated by the optimizer by using a rank selection with elitism. A new snapshot matrix \mathbf{S} is then collected;
7. geometric filtration update: the POD basis is re-computed on the new training database and the reduced design variables are updated;
8. new optimization: with a new design space and POD basis, the EA is restarted (back to step 4) by keeping constant the number of the design variables. In this new stage of EA optimization, it is possible to read an initial population (the very first one that was used to train the first POD basis, transformed in the current design space), or to let to the optimizer sample the new design space from scratch.

The geometric filter synthesizes the available data in a different space with fewer variables, realizing *de facto* a new parameterization. This approach differs from literature approaches[13] where existing libraries of airfoil shapes are employed to feed the filter. Here, the initial airfoil database is quite general as it is derived by setting the bounds of the native design space and sampling it, while the only *a priori* choice is limited to the baseline airfoil.

The POD basis updating is introduced to avoid an over-dependence of the reduced design space (and hence of the optimization results) from the initial LHS database. In fact, as it will be made clearer in the following section, the optimum geometric features may be too far to be reached with the eigen-modes extracted from a space-filling sampling: to cope with this, a natural choice would be to enlarge the bounds of the reduced design space, but the optimization cost would rise with no guarantee to find the global minimum. As a consequence, feeding the geometric filter with a blend of sub-optimal samples and randomly chosen candidates provides the DSR model with much more information about the optimization history and the potential directions of improvement.

3.2 Surrogate-based optimization

In order to save computational time, a surrogate-based optimization approach is also implemented to replace the GA-based module. The procedure is very similar to the one described in the previous section, the only differences being in steps 4 to 6. The definition of the minimization problem is identical but a multi-start surrogate-based local approach

(SBLO) replaces the evolutionary search. In the following sections, the local approach will be firstly introduced, then the multi-start procedure will be detailed.

3.2.1 Trust region optimization

In SBLO [24], a trust region approach is used to manage the minimization process and maintain acceptable accuracy between the surrogate model and the true model. This is realized by limiting the range over which the surrogate model is trusted according to a surrogate accuracy indicator. The process iteratively minimizes the surrogate model prediction by moving the center and modifying the bounds of the trust region. The surrogate model is trained by means of an outer sampling of predefined size (here, the samples are the POD modal coefficients). At the end of each approximate minimization, the candidate optimum point is validated using the true model. If sufficient decrease has been obtained in the true model, the trust region is re-centered around the candidate optimum point and the trust region will either shrink, expand, or remain the same size depending on the accuracy with which the surrogate model predicted the true model decrease. If sufficient decrease has not been attained, the trust region center is not updated and the entire trust region shrinks by a user-specified factor. The cycle then repeats with the construction of a new surrogate model, a minimization, and another test for sufficient decrease in the true model. This cycle continues until convergence is attained. The acceptance logic of the k -th iteration step is based on the evaluation of the so called trust region ratio ρ^k :

$$\rho^k = \frac{f(x_c^k) - f(x_{opt}^k)}{\hat{f}(x_c^k) - \hat{f}(x_{opt}^k)} \quad (9)$$

where x_c^k is the current trust region center, x_{opt}^k is the location of the optimum value predicted by the surrogate, f is the true response function and \hat{f} is the surrogate model prediction. The trust region ratio measures the ratio of the actual improvement to the improvement predicted by optimization on the surrogate model. According to its value, the k -th iteration is accepted or not and the trust region is reduced, retained or enlarged. The following criteria are used here:

- $\rho^k < 0$: the surrogate accuracy is poor, the step is rejected and the trust region is shrunk;
- $0 < \rho^k < 0.25$: the surrogate accuracy is marginal, the step is accepted and the trust region is shrunk;
- $0.25 < \rho^k < 0.75$: the surrogate accuracy is moderate, the step is accepted and the trust region is retained;
- $0.75 < \rho^k < 1.25$: the surrogate accuracy is good, the step is accepted and the trust region is expanded;

- $\rho^k > 1.25$: the surrogate accuracy is moderate, the step is accepted and the trust region is retained;

Three stopping criteria have been implemented and the search is terminated when at least one of them is satisfied:

- Maximum number of iterations and trust regions evaluations (=20);
- Minimum size of the trust region (=0.001);
- Minimum evaluation error $|\hat{f}(x_{opt}^k) - f(x_{opt}^k)| = 0.001$

3.2.2 Multi-start local optimization

As the SBLO is local in nature, hence it strongly depends on the starting point, a multi-start procedure has been implemented to provide a global character. Given the set of training samples, a cluster analysis is performed by using the k -means algorithm [25] which assign each sample to a specific cluster and computes the location of each cluster center. As the k -means algorithm requires to know the number of clusters, in the present study 5 clusters are generated. Finally, each cluster center is used as starting point of the surrogate-based local approach, so that 5 local optimizations are performed. At the end of the multi-start process, n geometries are selected among the full set of candidates evaluated during the optimization by using a roulette wheel algorithm with elitism (i.e., the best element of the set is always included). A new snapshot matrix \mathbf{S} is then collected and a new POD basis is computed, so that the procedure described in section 3.1 is continued from step 7.

4 Results and discussion

The present section discusses the results of a quite extensive numerical test campaign carried out with the DSR approach. The capability and flexibility of the method in reconstructing a generic airfoil geometry, the dependence from the size of the training database, considerations about the choice of the ranges for the reduced design variables are analyzed in previous works and not show there.

The closing sections 4.1.1 and 4.1.2 are devoted to the detailed description of the DSR updating strategy driven by GA and SBO evolution history. The aim is to reduce the randomness and the dispersion of the *a priori* LHS sampling. It will be shown how this strategy operates a compression of the volume of search and that this is beneficial for the optimizer.

4.1 Shape optimization

In this section the shape optimization problem is introduced in terms of design variables, design points, constraints and objective function. The baseline shape is the NACA 0012 airfoil and the geometry is parameterized by means of a NURBS approach[26].

The flow conditions include prescribed angle of attack AoA , Mach number, Reynolds number as shown in Table 1. A single design point in subsonic regime is selected and the

Table 1: Free-stream conditions

M_∞	Re_∞	$AoA[^\circ]$
0.3	6.0×10^6	2.0

evaluation of the objective function is carried out by using the Xfoil code by Drela [27], an high order panel method with fully coupled viscous/inviscid interaction correction. The laminar to turbulent transition is let free on both upper and lower side of the airfoil.

A classical objective in an aerodynamic shape optimization problem is to maximize the aerodynamic efficiency, while preserving some geometrical characteristics and aerodynamic performance as constraints. Here an equivalent formulation is followed as the objective is to minimize the inverse of the aerodynamic efficiency with constraints on minimum airfoil sectional area, minimum pitching moment and an additional constraint to control the derivative of airfoil curvature. More specifically, the objective function is defined as:

$$f_{obj} = \left(\frac{C_d + C_{d,t} + C_{d,g}}{C_l} \right) \frac{C_{d,0}}{C_{l,0}} + \max \left\{ 0, 0.01 [P(\chi') - P(\chi'_0)] \right\} \quad (10)$$

where:

$$\begin{aligned} C_{d,t} &= 0.01 \max(0, C_{m_y,0} - C_{m_y}) \\ C_{d,g} &= \max(0, A_0 - A) \end{aligned}$$

and $C_{d,0} = 0.0057$ and $C_{l,0} = 0.233$ are the drag and lift coefficients of the NACA 0012 at the selected subsonic design point, while $C_{m_y,0} = -0.05$ is the minimum allowed pitching moment coefficient. The problem is normalized with respect to the value of the objective function for the baseline airfoil, which is equal to 1. The aerodynamic constraint on pitching moment is formulated as a penalty on drag ($C_{d,t}$), thus representing a sort of trim drag: if the minimum value of the pitching moment is exceeded, the penalty applied will be 1 drag count per 0.01 in ΔC_m . The geometric constraint ($C_{d,g}$) imposes that the airfoil cross sectional area A should not be smaller than $A_0 = 0.0822$, i.e. the area of the NACA 0012 airfoil.

Furthermore, an additional constraint is introduced in equation 10 in order to control geometric oscillations that the NURBS parameterization may introduce. The optimizer uses a penalty function to activate this constraint when it exceeds a threshold value, even if it is well know that it might present some difficulties [28]. The penalty is based on the local airfoil curvature, χ , and its derivative, χ' with respect to the curvilinear abscissa. A numerical parameter $P(\chi')$ is used to measure the frequency and magnitude of changes in curvature over the entire airfoil, defined as:

$$P(\chi') = \sum_2^{p-1} \left[0.5 \left(\chi'_{i+1} - \chi'_{i-1} \right) \right]^2 \quad (11)$$

where p , as mentioned above, is the number of points describing the airfoil shape. The threshold value is $P(\chi'_0) = 250.0$ and it is established from baseline analysis and airfoil aerodynamics experience.

Two kind of parameterizations are used in the following analyses: the NURBS approach with 14 control points, referred to as full DV or native parameterization, and the filtered DSR approach with variable training set size and number of retained modes. The results of several optimization runs will be shown: each run is performed with the same problem definition and optimizer parameters but different design spaces. In the following section, the POD basis updating strategy is introduced.

4.1.1 On-line POD-DSR updating: GA-based optimization

An adaptive approach is proposed in this section to refine the choice of the training snapshots and update the POD basis iteratively while the optimization process is running. The aim is to enhance the POD basis by adding new information captured during the EA evolution and not included in the initial database, thus activating a sort of local search within the global optimization. Indeed, in evolutionary optimization it is quite common to discover crucial geometric features along the design path which are recognized as milestones by the optimizer itself. Crossover operator and elitism are just some examples of how offspring inherit those significant feature from parents. Therefore, when using POD-DSR during optimization, a natural choice is to update the POD basis to screen and include newly discovered characteristics into the reduced parameterization.

A first set of tests performed by using the optimization data from DSR-I cases with $\hat{n} = 8$. Recalling equation 8, the updating frequency n_q indicates how often the updating algorithm is executed while q is the number of generations collected prior to update the POD basis. Taking as reference $Q = 1000$ generations, here three different options are explored:

- $n_q = 1$ and $q = \frac{Q}{n_q+1} = \frac{Q}{2}$
- $n_q = 3$ and $q = \frac{Q}{n_q+1} = \frac{Q}{4}$
- $n_q = 7$ and $q = \frac{Q}{n_q+1} = \frac{Q}{8}$

As explained in section 3 and referring also to figure 1, the POD-DSR update is an on-line approach. When the evolution has reached q generations, a selection of n candidates among the collected dataset is made by means of a roulette wheel with elitism (i.e., the best element of the database is always included), based on the objective function values, and the resulting geometry population is converted into the new snapshot matrix. Then, the POD basis is recomputed and the updated reduced design variables/design space are used for another q generations. The selection by means of roulette wheel avoids to search a narrow design space, prone to local minima entrapment, as the selected subset is broad in terms of objective function values. The candidates selection and POD-DSR basis update is repeated every q generations, according to the updating frequencies n_q . The roulette

wheel approach introduces a further source of randomness that may affect the results, hence for each q the optimization tests are repeated 5 times in order to measure the mean and standard deviation of the minimum f_{obj} .

Figure 2 illustrates the histories of the minimum objective function: the gray curve is the envelope of the DSR-I run with $\hat{n}=8$ and represents the reference level. Depending on the value of q , the POD-DSR updating histories (averaged over 5 repetitions) depart from this curve at the first updating point: for instance, the green curve represent the case with $n_q = 1$ and starts from 15,000 evaluated elements, that is from $\frac{Q}{2}$ with $Q = 1000$ and a population size of 30. Likewise, blue and red curves are representative of cases with $n_q = 3$ and $n_q = 7$, respectively, and deviates from the reference history at 7,500 and 3,750 elements. The POD-DSR curves are depicted as piecewise linear for ease of representation as they connect the best values found before each POD basis update. The standard deviation of minimum f_{obj} over 5 repetitions is also reported at updating points as error bars.

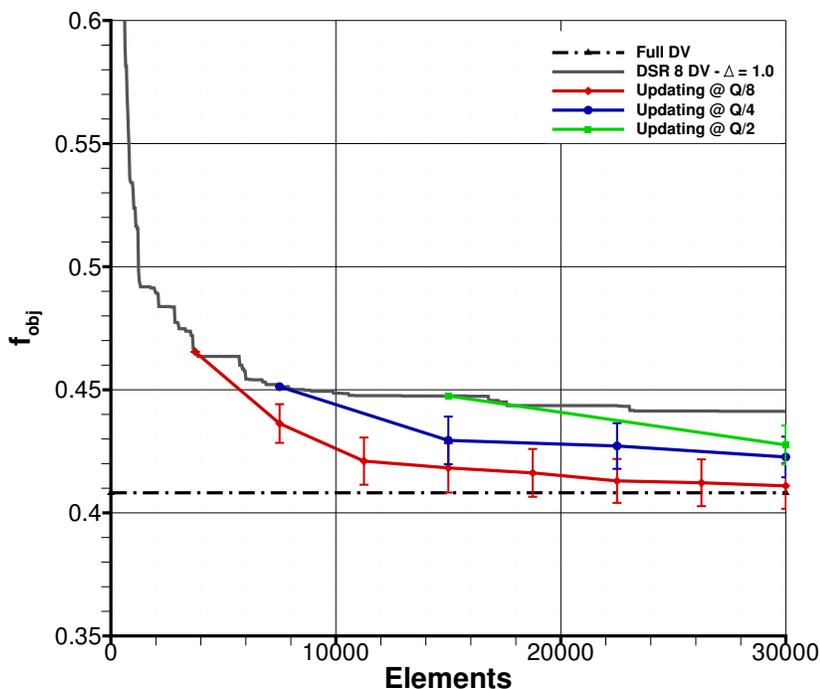


Figure 2: Updating POD-DSR with $\hat{n} = 8$

By increasing q , the average minimum f_{obj} decreases significantly with respect to the DSR-I test results up to reach the minimum of the full NURBS optimization. At the same time, looking at the standard deviation, the random component seems to play a considerable role and this can be related to the geometry selection algorithm. For

example, while the average of the POD–DSR run with $n_q = 7$ is slightly worse than the full approach, the variance over the repetitions is quite large as the updating strategy may perform better or worse than DSR-I run with almost the same probability.

In order to assess the role played by the selection algorithm in the updating step, figure 3 shows the empirical cumulative distribution function of f_{obj} over the first 15,000 elements evaluated by the optimizer. Values with $f_{obj} \gg 1$ indicate strong violations of the curvature constraint, i.e. completely unfeasible and highly wavy shapes. The plot compares the distribution of f_{obj} at an intermediate point of the optimization history with no update ($n_q = 1$), two updates ($n_q = 3$) and four updates ($n_q = 7$). The figure shows that the probability to have $f_{obj} < 1$ (representing the baseline performance) is 30% for $q = 500$, 38% for $q = 250$ and 65% for $q = 125$: a more frequent update implies an improvement of the quality of the population while preserving the diversity.

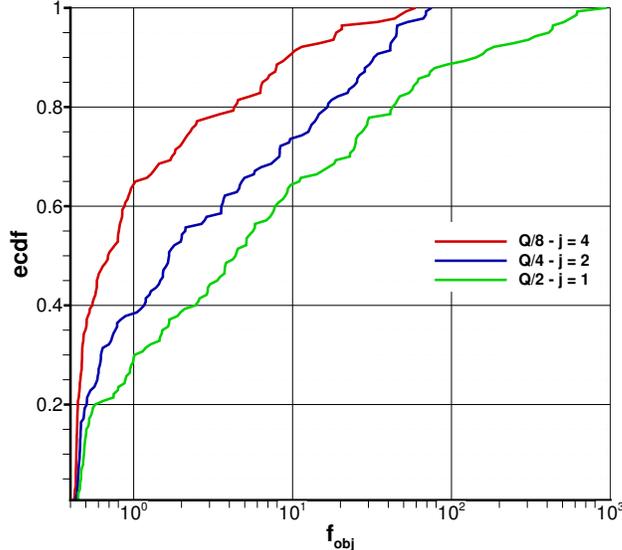


Figure 3: Empirical cumulative distribution function

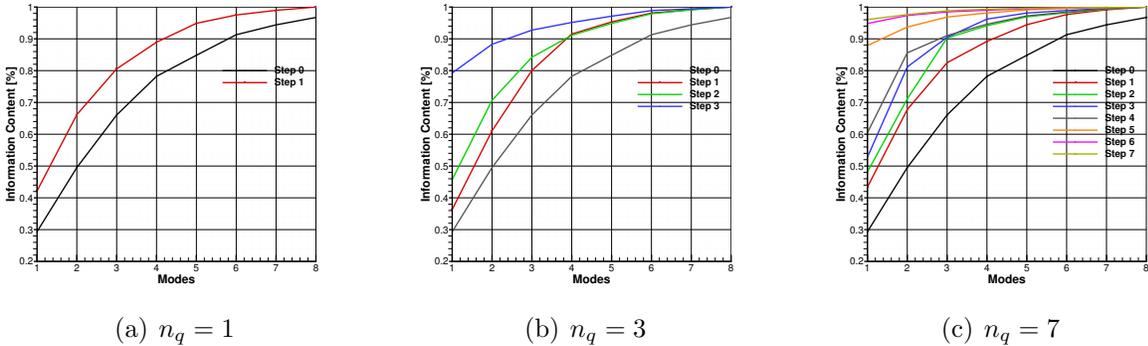
Table 2 resumes the aerodynamic performance of the best airfoils found at each repetition of the highest updating frequency case ($q = 0.125Q$).

The relative information content associated to the i -th reduced DV is σ_i^2 , i.e. the variance retained by the corresponding mode ϕ_i , and it is associated to the product $\langle \alpha_i^2 \rangle = \mathbf{a}_i \mathbf{a}_i^T$ product, hence strictly linked to the DSR DV ranges. For that reason, the higher the variance of the modes, the larger the design space ranges associated to that modes. After the repeated POD updating process, some noticeable differences with respect to the case where no update is applied can be observed: the average airfoil, \bar{S} , is no longer symmetrical; the set of eigenvectors Φ is fed with enhanced geometric features and synthesizes better shapes as showed in Figure 3; smaller variance is related to each mode, i.e. the ranges of variation of the new DVs are shrunk.

Table 2: Objective function and aerodynamic data of best airfoils for $n_q = 7$ POD-DSR optimization, 5 repetitions

Run ID	f_{obj}	C_l	$C_d[\text{dc}]$	C_m	$\frac{C_l}{C_d}$	$P(\chi') - P(\chi'_0)$
Run 1	0.4033	0.4401	43.1	-0.0499	102.03	2.10
Run 2	0.4151	0.4187	42.2	-0.0444	99.12	1.14
Run 3	0.4277	0.3986	41.4	-0.0409	96.18	1.12
Run 4	0.4041	0.4474	43.9	-0.0500	101.81	-4.01
Run 5	0.4021	0.4425	43.2	-0.0500	102.32	-3.01

In order to clarify how the variance σ_i^2 is reducing after POD updates, figure 6 shows the cumulative information content of the POD basis at each update step. The data refer to a single run for each value of q , selected among the 5 repetitions. By increasing the number of updates, the first modes are able to explain a greater amount of the data variance, thus allowing for a reduction of the effective dimensions of the problem while the optimization is in progress.


Figure 4: Information content during the DSR update

Resuming the discussion in the previous section, it is interesting to verify how the reduced design space volume changes when updating the POD basis. To this aim, Figure 5 shows the averaged (over the five optimization repetitions) hyper-volume ratio $\nu(j, \hat{n})/\nu(0, \hat{n})$ with $\hat{n} = 8$, j ranging in $\{0, n_q\}$ and $q = (0.125, 0.25, 0.5) \cdot Q$: at each update j , the volume contracts at constant rate and, increasing the updating frequency, it becomes even smaller. To sum up, figures 3, 6 and 5 provide three different views of the same notion: the data-driven update pushes the search of the global optimum in more local and promising regions of the design space.

4.1.2 On-line POD-DSR updating: SBO optimization

In this section the results obtained with the surrogate-based optimization (section 3.2) are shown. Starting from the same training database of 140 airfoil geometries, the process is described in Figure 1, where the GA optimizer is replaced by the multi-start surrogate-based (SB) local optimizer.

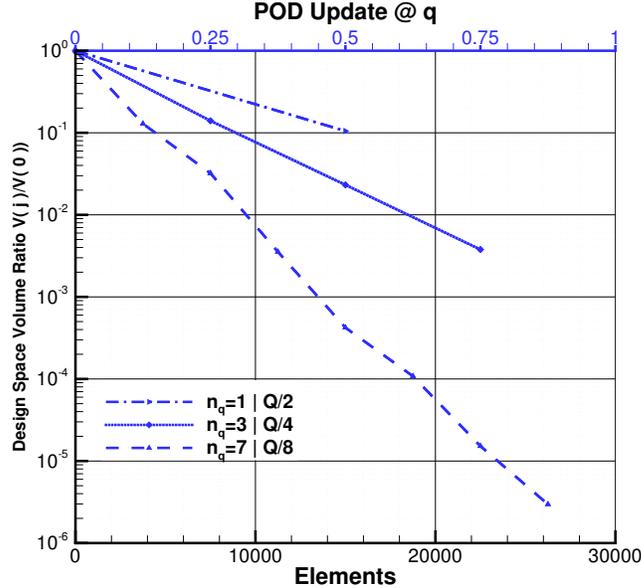


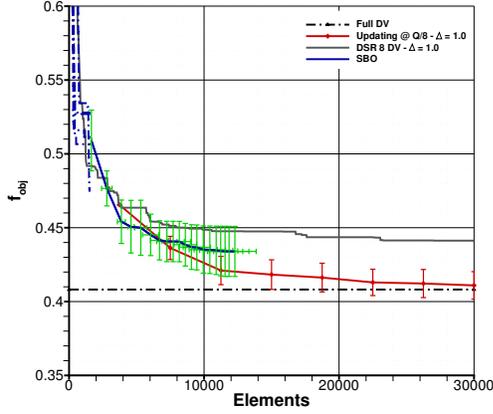
Figure 5: Volume ratio with POD updating

The number of reduced design variables (8) is kept equal to the previous case while the number of POD basis updates is increased up to 20. In order to obtain statistical information, each run is repeated five times. Figure 6(a) shows the evolution of minimum value of f_{obj} , evaluated with the aerodynamic solver, as a function of the number of samples evaluated during the process. The SBO curve (blue line) is the average of f_{obj} over the five repetition and it is characterized by a standard deviation on both axes: on the y-axis the deviation of the f_{obj} values due to repetitions is reported, while on the x-axis the deviation is due to the fact that the number of evaluated samples in each SBO optimization stage is not constant with each repetition. In average, the SBO is bounded by the GA-based optimizations with (red curve) and without (black curve) updating. Moreover, the progress towards the optimum location is somewhat accelerated in the first steps with respect to the GA-based optimization and then tends to reach a plateau around a sub-optimal value.

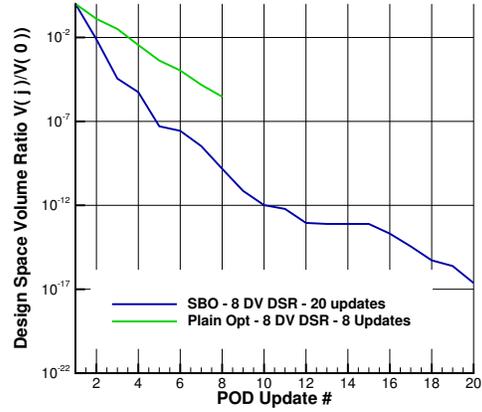
Figure 6(b) shows the compression of design space volume obtained with SBO and compared to DSR GA-based optimization. With SBO, the POD update steps are increased, hence the absolute value reached at the end of the optimization is well below the GA approach. Furthermore, comparing the volume compression level at fixed update step, it is evident that the SBO proceeds faster than GA.

5 Conclusions

The work proposed an investigation on Design Space Reduction (DSR) by means of POD technique, aiming at reducing the design variables number in a shape optimization problem.



(a) Progress of f_{obj}



(b) Progress of DSR volume

Figure 6: Results of SBO with POD updating

Introducing an updating algorithm of the POD basis, driven by a selection of data coming from the on-going optimization, some weak points emerged by previous investigations can be improved and optimal results have been shown.

The two-dimensional test case, due to its relatively cheap computational cost, has allowed a detailed investigation to prove the capability of the reduced parameterization involved in the optimization frame, conducted with both GA-based and SBO approach, showing promising results.

REFERENCES

- [1] Bellman, R., *Adaptive control Processes: A Guided Tour*, Princeton University Press, 1961.
- [2] Queipo, N., Haftka, R., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P., “Surrogate-based analysis and optimization,” *Progress in Aerospace Sciences*, Vol. 41, 2005, pp. 1–28.
- [3] Wujek, B. A. and Renaud, J. E., “New Adaptive Move-Limit Management Strategy for Approximate Optimization, Part 1,” *AIAA Journal*, Vol. 36, No. 10, Oct. 1998, pp. 1911–1921.
- [4] Toropov, V., va, F., Markine, V., and d, H., “Refinements in the multi-point approximation method to reduce the effects of noisy structural responses,” *Multidisciplinary Analysis Optimization Conferences*, American Institute of Aeronautics and Astronautics, Sept. 1996, pp. –.
- [5] Simpson, T., Poplinski, J., Koch, P. N., and Allen, J., “Metamodels for Computer-based Engineering Design: Survey and recommendations,” *Engineering with Computers*, Vol. 17, No. 2, Jul 2001, pp. 129–150.
- [6] Zahir, M. K. and Gao, Z., “Variable-fidelity optimization with design space reduction,” *Chinese Journal of Aeronautics*, Vol. 26, No. 4, 2013, pp. 841 – 849.
- [7] Koziel, S., Bekasiewicz, A., and Leifsson, L., “Multi-Objective Design Optimization of Planar Yagi-Uda Antenna Using Physics-Based Surrogates and Rotational Design Space Reduction,” *Procedia Computer Science*, Vol. 51, No. Supplement C, 2015, pp. 885 – 894, International Conference On Computational Science, ICCS 2015.
- [8] Sorzano, C. O. S., Vargas, J., and Montano, A. P., “A survey of dimensionality reduction techniques,” *arXiv preprint arXiv:1403.2877*, 2014.
- [9] Ghodsi, A., “Dimensionality reduction a short tutorial,” .
- [10] Toal, D. J., Bressloff, N., Keane, A., and Holden, C., “Geometric filtration using proper orthogonal decomposition for aerodynamic design optimization,” *AIAA Journal*, Vol. 48, No. 5, May 2010, pp. 916–928.
- [11] van der Maaten, L., Postma, E. O., and van den Herik, H. J., “Dimensionality Reduction: A Comparative Review,” 2008.
- [12] Poole, D. J., Allen, C. B., and Rendall, T. C. S., “Metric-Based Mathematical Derivation of Efficient Airfoil Design Variables,” *AIAA Journal*, Vol. 53, No. 5, Jan. 2015, pp. 1349–1361.

- [13] Masters, D. A., Taylor, N. J., Rendall, T. C. S., Allen, C. B., and Poole, D. J., “Geometric Comparison of Aerofoil Shape Parameterization Methods,” *AIAA Journal*, Vol. 55, No. 5, Jan. 2017, pp. 1575–1589.
- [14] Cinquegrana, D. and Iuliano, E., “Geometric Data reduction in aeroshape Optimization,” *23rd Conference of the Italian Association of Aeronautics and Astronautics*, edited by Carrera and Cinefra, AIDAA, November 2015.
- [15] Cinquegrana, D. and Iuliano, E., “Efficient Global Optimization of a Transonic Wing with Geometric Data Reduction,” *AIAA AVIATION Forum*, American Institute of Aeronautics and Astronautics, June 2017, pp. –.
- [16] Viswanath, A., J. Forrester, A. I., and Keane, A. J., “Dimension Reduction for Aerodynamic Design Optimization,” *AIAA Journal*, Vol. 49, No. 6, June 2011, pp. 1256–1266.
- [17] Berguin, S. H. and Mavris, D. N., “Dimensionality Reduction Using Principal Component Analysis Applied to the Gradient,” *AIAA Journal*, Vol. 53, No. 4, Oct. 2014, pp. 1078–1090.
- [18] Ghoman, S. S., Wang, Z., Chen, P. C., and Kapania, R. K., “Hybrid Optimization Framework with Proper-Orthogonal-Decomposition-Based Order Reduction and Design-Space Evolution Scheme,” *Journal of Aircraft*, Vol. 50, No. 6, Sept. 2013, pp. 1776–1786.
- [19] Sirovich, L., “Turbulence and the dynamics of coherent structures,” *Q. Appl.Math.*, Vol. XLV, No. 3, October 1987, pp. 561–571.
- [20] Berkooz, G., Holmes, P., and Lumley, J., “The Proper Orthogonal Decomposition in The analysis of turbulent Flows,” *Ann. Rev. Fluid Mech.*, Vol. 25, 1993, pp. 539–575.
- [21] Lucia, D., King, P., and Beran, P., “Reduced order Modeling of a two dimensional flow with moving shocks,” *Computers & Fluids*, Vol. 32, 2003, pp. 917–938.
- [22] Vitagliano, P. L. and Quagliarella, D., “A Hybrid genetic algorithm for constrained design of wing and wing-body configurations,” *Evolutionary Methods for Design, Optimization and Control Applications to Industrial and Societal Problems*, edited by G. Bugeđa, J. A. Désidéri, J. Périaux, M. Schoenauer, and G. Winter, International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain, 2003.
- [23] Quagliarella, D., Iannelli, P., Vitagliano, P. L., and Chinnici, G., “Aerodynamic shape design using hybrid evolutionary computation and fitness approximation,” *AIAA 1st Intelligent Systems Technical Conference*, American Institute of Aeronautics and Astronautics (AIAA), Chicago, IL, Sept. 2004, AIAA Paper 2004-6514.

- [24] Alexandrov, N., Lewis, R., Gumbert, C., Green, L., and Newman, P., “Optimization with variable-fidelity models applied to wing design,” *38th Aerospace Sciences Meeting and Exhibit*, Aerospace Sciences Meetings, American Institute of Aeronautics and Astronautics, Jan. 2000.
- [25] Lloyd, S., “Least squares quantization in PCM,” *IEEE Transactions on Information Theory*, Vol. 28, No. 2, March 1982, pp. 129–137.
- [26] Martin, M., Valero, E., Lozano, C., and Andres, E., “Gradient Calculation for Arbitrary Parameterizations via Volumetric NURBS: The Control Box Approach.” *5th European Conference for Aeronautics and Space Sciences (EUCASS)*, 2013.
- [27] Drela, M., “XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils,” *Low Reynolds Number Aerodynamics*, edited by T. Mueller, Vol. 54 of *Lecture Notes in Engineering*, Springer Berlin Heidelberg, 1989, pp. 1–12.
- [28] Kolla, M. L., Yokota, J. W., Lassaline, J. V., and Fejtek, I., “Curvature constraints for airfoil shape optimization in turbulent flow,” *Canadian Aeronautics and Space Journal*, Vol. 54, No. 1, 2008, pp. 1–7.