



HAL
open science

Principal angles between subspaces and reduced order modelling accuracy in optimization

Bijan Mohammadi

► **To cite this version:**

Bijan Mohammadi. Principal angles between subspaces and reduced order modelling accuracy in optimization. Structural and Multidisciplinary Optimization, 2014, 50 (2), pp.237-252. 10.1007/s00158-013-1043-1 . hal-01054939

HAL Id: hal-01054939

<https://hal.science/hal-01054939>

Submitted on 10 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Principal angles between subspaces and reduced order modelling accuracy in optimization

Bijan Mohammadi

Struct. & Multi-Disciplinary Optimization. DOI: 10.1007/s00158-013-1043-1, 50/2, 237-252, 2014

Abstract The paper considers robust parametric optimization problems using multi-point formulations and addresses the issue of the approximation of the gradient of the functional by reduced order models. The question of interest is the impact of such approximations on the search subspace in the multi-point optimization problem. The mathematical concept used to evaluate these approximations is the principal angles between subspaces and practical ways to evaluate these are provided. An additional indicator is provided when a descent minimization algorithm is used. The approach appears also to be an interesting tool for uncertainty quantification of the design in the presence of models of increasing complexity. The application of these concepts is illustrated in the design of the shape of an aircraft robust over a range of transverse winds.

Keywords robust optimization · reduced order modelling · principal angles · uncertainty quantification

Mathematics Subject Classification (2000) 49N30 · 49M29 · 65Y05 · 65D30 · 14N20 · 15A18 · 76B75 · 76D55 · 90C52 · 93B51

1 Introduction

Reduced order models are widely used in both simulation and design. These can be built, for instance, by assimilation of high fidelity data or experimental results by a parametric model. Once the model built it can be used in an optimization procedure providing cheap estimation of the functional and its sensitivity with respect to the model parameters. It can also be locally used in trust region approaches where the reduced order model needs to be dynamically enriched. Reduced order modelling receives different denominations following the field of research it is issued from: learning in neural networks, fitting with least squares in experimental plans or higher degrees polynomials, reduced-order modeling with proper orthogonal decomposition, etc. Exhaustive referencing is therefore out of scope and we indicate a few review texts such as [1–3].

The performance of a system designed for given functioning conditions often seriously degrades when these conditions are modified. Typical situations of interest involve a few (typically one or two) parameters describing the functioning of the system. Multi-point optimization is widely used to address robustness issues in engineering and we showed [18,12] how to use such formulation to address the robustness issue when a few functioning parameters are involved. In particular, we showed how to introduce in this formulation what we would like for the outcome of the design through a 'target-based' weighting in the functional. We also discussed optimal sampling of the functioning parameters intervals and quantitative confidence levels on the quality of our search direction through Gram-Schmidt orthonormalization of the multi-point gradients. This naturally

B. Mohammadi
Université Montpellier II, Mathématiques, CC51, 34095 Montpellier, France
Tel.: +33-467143557
Fax: +33-467143558
E-mail: bijan.mohammadi@um2.fr

brought into light the question of the sampling size which needs to be monitored to maintain the calculation complexity low. The aim was to avoid the worst-case theoretical limitation indicating that the sampling size should be larger by one the size of the control space [20,18,19]. We showed that this is only necessary if all the associated gradients (i.e. evaluated at the sampling points) are linearly independent which is never the case in optimizations involving a state equation. Hence, we showed, again through Gram-Schmidt procedure, that large dimensional parametric optimization problems can be treated with very small sampling of the functioning parameters range with marginal losses on the gradient information.

However, one often uses approximation in the definition of the gradients. And this is the starting point of this paper. One would like to analyze the impact of these using reduced order models on the search space defined by the gradients of the functional at the different sampling points of the functioning parameters. The mathematical tool we use for this analysis is the principal angles between subspaces. Reduced order modelling also includes situations where lower accuracy and modelling complexity are accounted for in the linearization step than for the direct simulation chain used for the definition of the functional. This is often the situation where an approximate simulation chain is considered for linearization, dropping or approximating some of the ingredients in the direct chain. Of course, if the full gradient (i.e. based on the complete direct model) is accessible this discussion is pointless. But, calculation complexity and also use of black box simulation tools make that the linearization often takes place for a variant of the simulation loop.

After discussing algorithmic evaluations of the principal angles between the exact and approximate search spaces, we will compare the evolution of these through optimization iterations for a complex shape optimization problem for two scenarios of reduced order modelling. Even if non-intuitive, it will be shown why the modelling errors will have less impact on the design when using descent minimization algorithms. This approach appears to be a powerful tool to evaluate the pertinence of an increase in the modelling complexity and address related uncertainty quantification issues.

2 Robust parametric optimization

We are interested in a class of optimization problems where the cost function involves a parameter α not considered as a design parameter:

$$\min_{\mathbf{x} \in \mathbf{O}_{ad}} j(\mathbf{x}, \alpha), \quad \alpha \in \mathbf{I} \subset \mathbb{R}^n, \mathbf{O}_{ad} \subset \mathbb{R}^N, \quad n \ll N. \quad (1)$$

where \mathbf{x} is the actual design vector belonging to \mathbf{O}_{ad} the optimization admissible domain. Usually, the functioning parameters α are just a few. Typical situations of interest are where $n = 1$ or $n = 2$.

In [12] we showed how to use multi-point optimization to address such optimization problem. The aim is to remove during the optimization the dependency in α . This is done minimizing a functional $J(\mathbf{x})$ encapsulating this dependency:

$$J = \mu, \quad \text{such that} \quad \sigma \leq \sigma_0,$$

where

$$\mu = \frac{1}{\omega} \sum_{\alpha_l \in \mathbf{I}_m} \omega_l j(\mathbf{x}, \alpha_l), \quad \omega = \sum_{l=1}^m \omega_l, \quad (2)$$

with \mathbf{I}_m a given sampling of \mathbf{I} whose choice has been discussed in [12] as well as the weights ω_l which account for the kind of performance we eventually want for the design: constant performance over the functioning parameters ranges.

For the sake of simplicity and without loss of generality one can consider here both the sampling and the weighting to be uniform.

As in First-Order Second Moment (FOSM) methods [14], σ is to monitor the regularity of the final performance which should be as regular as possible with respect of α :

$$\sigma = \frac{1}{2} \sum_{\alpha_l \in \mathbf{I}_m} \langle \nabla_{\alpha} j(\mathbf{x}, \alpha_l), \nabla_{\alpha} j(\mathbf{x}, \alpha_l) \rangle, \quad (3)$$

where \langle, \rangle is the Euclidean scalar product. σ_0 is the level of variability for the initial design or of a reference configuration. As n is small, $\nabla_{\alpha} j$ is evaluated by finite differences on \mathbf{I}_m .

Such minimization problems have brought new interest to descent methods and this not only because of their lower computational complexity than the gradient free methods. Indeed, we will see that gradients are useful to see what should actually be the search space in a context of robust multi-point design. More precisely, the definition of the suitable search space is only possible after identification of the maximal free subspace in the space generated by the gradients of the functional at the sampling points \mathbf{I}_m . Indeed, as mentioned in the introduction, previous theory suggested that the size of the sampling should be larger than the dimension of the control space (i.e. $m = N + 1$) while we noted [18,12] that the dimension of the maximal free search subspace is usually much smaller than N .

3 A multi-point descent algorithm

We consider the following iterative descent algorithm for our constrained minimization problem involving a direct simulation chain linking the parameters (\mathbf{x}, α) to the state U solution of a state equation $F(U(q(\mathbf{x}, \alpha))) = 0$ and to a functional j is:

$$\left\{ \begin{array}{l} \text{Given } \mathbf{x}_0, 0 < \rho, 0 < \eta \ll 1, \mathbf{I}_m, p_{max}, 0 < \varepsilon \ll 1, \\ \text{optimization iterations } p = 1, \dots, p_{max} \\ \left\{ \begin{array}{l} -m \text{ parallel state equation solutions } F(U(q(\mathbf{x}_p), \alpha_l)) = 0, \alpha_l \in \mathbf{I}_m, \\ -m \text{ parallel evaluations of } j(\mathbf{x}_p, \alpha_l), \alpha_l \in \mathbf{I}_m, \\ -m \text{ parallel solutions of the adjoint state } V \text{ equation:} \\ \qquad \qquad \qquad V^t F_U(U(q(\mathbf{x}_p), \alpha_l)) = j_U^t, \alpha_l \in \mathbf{I}_m, \\ -m \text{ parallel evaluations of } \nabla_{\mathbf{x}} j(\mathbf{x}_p, \alpha_l) = j_{\mathbf{x}} + (V^t F_{\mathbf{x}})^t, \alpha_l \in \mathbf{I}_m, \\ -\text{define } d \text{ the descent direction, } d = \overline{\nabla_{\mathbf{x}} \mu} - \langle \overline{\nabla_{\mathbf{x}} \mu}, \overline{\nabla_{\mathbf{x}} \sigma} \rangle \overline{\nabla_{\mathbf{x}} \sigma} + \eta \overline{\nabla_{\mathbf{x}} \sigma}, \\ -\text{control parameter variation, } \mathbf{x}_{p+1} = \mathbf{x}_p - \rho d, \\ \text{Stop if } \|d\| \leq \varepsilon, \end{array} \right. \end{array} \right.$$

where $\bar{a} = a/\|a\|$ is the normalized vector a and

$$\nabla_{\mathbf{x}} \mu = \frac{1}{\omega} \sum_{\alpha_l \in \mathbf{I}_m} \omega_l \nabla_{\mathbf{x}} j(\mathbf{x}, \alpha_l), \quad (4)$$

$$\nabla_{\mathbf{x}} \sigma = \sum_{\alpha_l \in \mathbf{I}_m} \langle \nabla_{\alpha} j(\mathbf{x}, \alpha_l), \nabla_{\alpha \mathbf{x}} j(\mathbf{x}, \alpha_l) \rangle. \quad (5)$$

$\nabla_{\mathbf{x} \alpha} j(\mathbf{x}, \alpha_l)$ is obtained from $\nabla_{\mathbf{x}} j(\mathbf{x}, \alpha_l)$ by finite differences on \mathbf{I}_m , component by component, following what has been done for $\nabla_{\alpha} j$. The definition of the descent direction permits to make sure that both μ and σ decrease for small descent steps. Indeed, a first order development in \mathbf{x} gives:

$$\sigma(\mathbf{x}_{p+1}) - \sigma(\mathbf{x}_p) = \|\nabla_{\mathbf{x}} \sigma\| \overline{\nabla_{\mathbf{x}} \sigma} \cdot (\mathbf{x}_{p+1} - \mathbf{x}_p) = -\rho \eta \|\nabla_{\mathbf{x}} \sigma\| \leq 0,$$

and we have

$$\mu(\mathbf{x}_{p+1}) - \mu(\mathbf{x}_p) = \|\nabla_{\mathbf{x}} \mu\| \overline{\nabla_{\mathbf{x}} \mu} \cdot (\mathbf{x}_{p+1} - \mathbf{x}_p) = -\rho \|\nabla_{\mathbf{x}} \mu\| (1 - \zeta^2 + \eta \zeta),$$

where $\zeta = \langle \overline{\nabla_{\mathbf{x}} \mu}, \overline{\nabla_{\mathbf{x}} \sigma} \rangle$. Therefore, μ is also decreasing as $1 - \zeta^2 + \eta \zeta \geq 0$ for $|\zeta| \leq 1$ and $0 < \eta \ll 1$ as chosen in the algorithm.

Despite the natural presence of parallelism in this algorithm in the m independent evaluations of the state, functional and its gradient, computational complexity remains an issue. Possible solutions for the reduction of the sampling size have been presented in [12] together with the use of incomplete sensitivity concept in the evaluation of the gradients. This latter permits to avoid the solution of the m adjoint equations in the algorithm. This is particularly suitable when using

black-box state equation solvers not providing the adjoint of the state variables. We will mention these issues in section 7.

Beyond individual gradient accuracy (i.e. at each of the sampling point), what is important in these multi-point problems is the global search space defined by the ensemble of the gradient vectors. This means that one might tolerate higher error levels in each of the gradient defined at the different sampling point than for a single-point optimization situation as what is important is for the global search space to remain nearly unchanged. An interesting mathematical concept which permits to measure the deviation between two subspaces is the principal angles between subspaces.

4 Angles between subspaces

We use the mathematical concept of 'principal angles' between subspaces in the Euclidean space \mathbb{R}^N initially introduced by C. Jordan [4]. If the maximum principle angle between the two subspaces is small, the two are nearly linearly dependent. Geometrically, this is the angle between two hyperplanes embedded in a higher dimensional space.

Let us briefly recall the concept of principal angles and how to practically compute them [5,8]. For simplicity, suppose A and B are two subspaces of dimension k of \mathbb{R}^N , $N \geq 2k$, although this is not a prerequisite to define the principal angles. The k principal angles $\{\theta_i, i = 1, \dots, k\}$ are recursively defined as:

$$\cos(\theta_i) = \frac{\langle a_i, b_i \rangle}{\|a_i\| \|b_i\|} = \max \left\{ \frac{\langle a, b \rangle}{\|a\| \|b\|} : a \perp a_m, b \perp b_m; m = 1, \dots, i-1 \right\},$$

where $a_j \in A$ and $b_j \in B$.

The principal angles θ_i are between 0 and $\pi/2$. This is an important point and will be used later to take advantage of the positivity of the cosine of the angles. The procedure finds unit vectors $a_1 \in A$ and $b_1 \in B$ minimizing the angle θ_1 between them. It then takes the orthogonal complement of a_1 in A and b_1 in B and iterates. This procedure is not useful in practice as computationally inadequate. We would like to be able to find the angles θ_i from the inner products $\langle a_i, b_j \rangle$ of the elements of two bases of A and B [9]. This would be interesting in our multi-point optimization context where we can exhibit an orthonormal basis of the global search space for the multi-point optimization problem using Gram-Schmidt orthonormalization.

Now, let $\{a_i, i = 1, \dots, k\}$ and $\{b_i, i = 1, \dots, k\}$ be two arbitrary orthonormal bases for A and B . Orthonormal bases are easy to obtain through the Gram-Schmidt orthonormalization procedure. Consider M being the matrix of the projection operator Pr_A of B onto A defined by:

$$Pr_A(b_i) = \sum_{j=1}^k \langle b_i, a_j \rangle a_j, \quad M = (\langle b_i, a_j \rangle)_{i,j}.$$

The principal angles can be linked to this operator [9] through:

$$M = G\Sigma H^t,$$

where G and H are orthogonal matrices and $\Sigma = \text{diag}(\cos(\theta_i))$.

As G and H are orthogonal matrices, this is a Singular Vector Decomposition (SVD) of M . G and H are unknown at this point. But, we will show that we do not need them to get the θ_i . Otherwise, the approach will be again computationally useless.

We recall that the columns of G are the left-singular vectors of M and eigenvectors of MM^t and the columns of H are the right-singular vectors of M and eigenvectors of M^tM . And most important that $\cos^2(\theta_i)$ are the eigenvalues of $Pr_A^t Pr_A$ which writes in matrix form as: $M^tM = (G\Sigma H^t)^t(G\Sigma H^t) = H\Sigma^2 H^t$ with $\Sigma^2 = \text{diag}(\cos^2(\theta_i))$.

Therefore, to find the principal angles between subspaces A and B , knowing an orthonormal basis in each subspace, one should calculate M and find the eigenvalues of M^tM and take the square root of them. This last operation is valid as the angles are between 0 and $\pi/2$, and their cosine therefore always positive.

We presented the approach for subspaces of the same dimension k , but it is not necessary for the two subspaces to be of the same size in order to find the angles between them. We need

$N \geq 2k$ to be able to exhibit two orthogonal subspaces. If $N < 2k$, some principal angles necessarily vanish and for $N = k$ they all vanish. This procedure is still valid if the subspaces have different dimensions. The projection operator can be defined as well as its transpose and the eigenvalues of $M^t M$ are real as this is a symmetric square matrix.

In our optimization applications we always proceed first with a reduction in size of the search space using a sampling reduction size algorithm [12]. This makes the calculation of the whole set of eigenvalues feasible in terms of calculation complexity. However, if this is out of reach, one can evaluate the bounds on the angles to see the global pertinence of our reduced order models and gradient approximations. This can be done without an exact calculation of the all eigenvalues. It is sufficient to use the Gershgorin circle theorem to find these bounds as every eigenvalue of $M^t M$ lies within at least one of the Gershgorin discs $D((M^t M)_{ii}, R_i)$ centred on $(M^t M)_{ii}$ and with radius $R_i = \sum_{j \neq i} |(M^t M)_{ij}|$. And because $M^t M$ is symmetric, the eigenvalues being real, we only consider the intersection of the discs with the x -axis. Alternatively, the largest and smallest principal angles can be found using iterative power and inverse power methods applied to $M^t M$.

One should however be aware that these bounds might not be sufficiently sharp to discriminate between two reduced order models and decide, for instance, which one is the more adequate for sensitivity analysis. Figure 1 shows a typical sketch. It represents principal angles calculated between a first subspace generated by the exact gradients of a transport model for a 10 points sampling of one of the functioning parameters of the model and two subspaces generated by the sensitivities derived from two approximations of this model for the same sampling. Details of the models can be found in [13]. But the modelling problem is not of a main concern here. What is important is that if one only considers the first and last principal angles, model M_2 is found being a better approximation to be used in a linearization procedure. However, with the whole spectrum in hand the picture is quite different and M_1 appears be more suitable if one intends to use this reduced order model for sensitivity analysis.

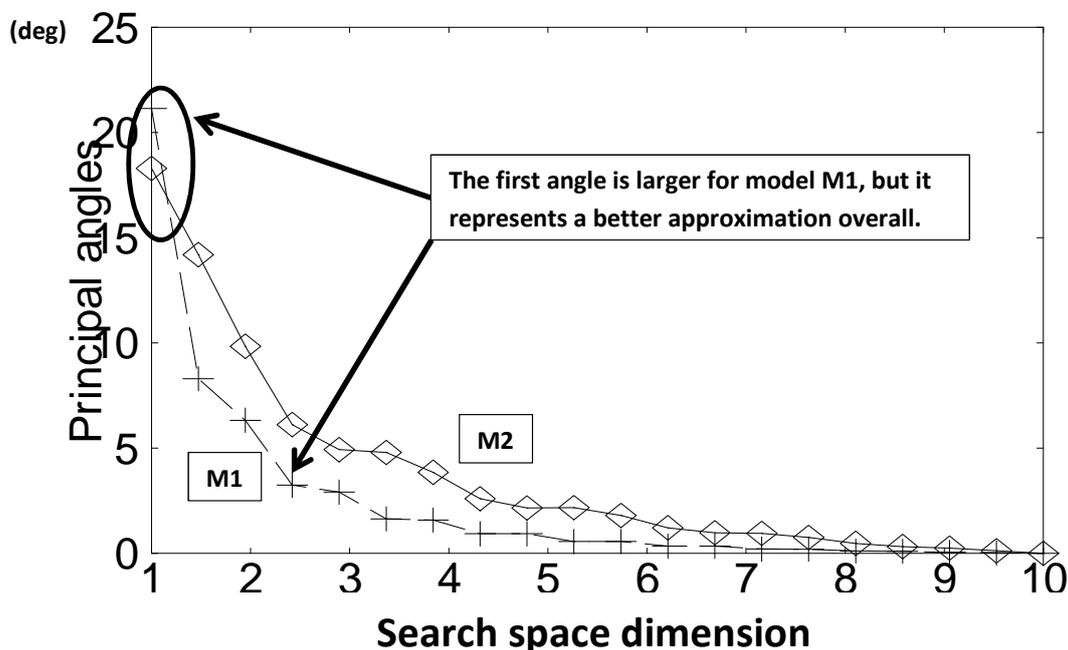


Fig. 1 Principal angles between the subspaces generated by an exact gradient calculation and the linearization of two reduced order models. This permits to quantify the pertinence of an approximation from the whole spectrum. Model M_1 is found to be a better approximation even with a first principal angle slightly larger than with M_2 .

Principal angles between multi-point search spaces are interesting to measure the pertinence of sensitivity definitions based on reduced order models. Indeed, the design will be unaffected

by a reduction in the model's complexity if the search subspaces, generated by the gradients at the sampling points of the functioning parameter interval and their approximations, remain the same. In the sequel we discuss the application of these ideas in our multi-point robust optimization context.

5 Definition of the subspaces

Let us define subspaces A and B in the context of our multi-point optimization problems defined in section 2. In our problem the size N is very large and is given by the dimension of the control space. But, the subspaces A and B are of dimension k and we have $2k \ll N$. For the sake of simplicity we consider that both subspaces are of the same dimension. These are free subspaces and we make sure by Gram-Schmidt orthonormalization that $\dim(A) = \dim(B) = k$.

Let us denote by j an approximation of the functional J . The orthonormalization is applied to a set of exact $G(\alpha_l)$ and incomplete $g(\alpha_l)$ gradient vectors with respect to the control variable \mathbf{x} evaluated at $\alpha_l \in \mathbf{I}_m$ the sampling of size m :

$$G(\alpha_l) = \nabla_{\mathbf{x}} j(\mathbf{x}, \alpha_l), \quad g(\alpha_l) = \nabla_{\mathbf{x}} J(\mathbf{x}, \alpha_l).$$

The choice of m is such that $k < m \ll N$. The issue of minimal sampling still providing an enough large search space has been discussed in [12].

So we have $A = \text{Span}(G(\alpha_l), \alpha_l \in \mathbf{I}_m)$ and $B = \text{Span}(g(\alpha_l), \alpha_l \in \mathbf{I}_m)$. Now, the orthonormal bases $\{a_i, i = 1, \dots, k\}$ and $\{b_i, i = 1, \dots, k\}$ are defined by the Gram-Schmidt orthonormalization procedure applied to $\{G(\alpha_l), \alpha_l \in \mathbf{I}_m\}$ and $\{g(\alpha_l), \alpha_l \in \mathbf{I}_m\}$. Hence, we will have $A = \text{Span}(a_i, i = 1, \dots, k)$ and $B = \text{Span}(b_i, i = 1, \dots, k)$ but with $k < m$.

The size k depends on the accuracy requested in the Gram-Schmidt procedure we recall briefly (in practice modified Gram-Schmidt is applied for numerical stability). This algorithm stops when a given tolerance is achieved and k will be the rank at which the rest after successive projections is found smaller than this tolerance:

$$\left\{ \begin{array}{l} \text{Given } \mathbf{I}_m, TOL, \\ a_1 = G(\alpha_1), \\ \text{iterations } k = 1, \dots \\ a_k = G(\alpha_k) - \sum_{j=1}^{k-1} \frac{\langle G(\alpha_k), a_j \rangle}{\langle a_j, a_j \rangle} a_j, \\ \text{until } \|a_k\| < TOL. \end{array} \right. \quad (6)$$

This gives us the basis $\{a_i, i = 1, \dots, k\}$. The same procedure is applied to g 's to get the basis $\{b_i, i = 1, \dots, k\}$. Again, the theory is valid even if the two bases do not have the same number of vectors for the same accuracy TOL (i.e. A and B have different dimension).

6 Evolution of the principal angles during optimization

We would like to use the principal angles concept to evaluate gradients approximation by reduced order modelling. These can be evaluated versus exact gradients when available or against each other. This former permits to estimate, for instance, how much deviation is caused in the search space by different levels of approximation in an incremental modelling process.

To implement these we modify the algorithm in section (3) introducing a few extra steps in order to dynamically evaluate during the optimization the principal angles between the subspaces generated by the set of exact and incomplete gradients over the functioning parameters sampling \mathbf{I}_m and described in section 5. We think this procedure is necessary because unlike in the single-point optimization, the approximation here needs to be valid over a range of the functioning parameters and one would like also to make sure that the approximation holds over the optimization history for the different control parameter values visited.

The algorithm given in section 3 is modified as:

$$\left\{ \begin{array}{l} \text{Given } \mathbf{x}_0, \rho, \eta, \mathbf{I}_m, p_{max}, \varepsilon, TOL, \\ \text{optimization iterations } p = 1, \dots, p_{max} \\ \begin{array}{l} -m \text{ parallel state equation solutions } F(U(q(\mathbf{x}_p), \alpha_l)) = 0, \alpha_l \in \mathbf{I}_m, \\ -m \text{ parallel evaluations of } j(\mathbf{x}_p, \alpha_l), \alpha_l \in \mathbf{I}_m, \\ -m \text{ parallel evaluations of } G_l(\alpha_l), \alpha_l \in \mathbf{I}_m, \\ -m \text{ parallel evaluations of } g_l(\alpha_l), \alpha_l \in \mathbf{I}_m, \\ \text{-define the maximal free subspaces } A \subset \text{Span}(G(\alpha_l), \alpha_l \in \mathbf{I}_m) \\ \quad \text{and } B \subset \text{Span}(g(\alpha_l), \alpha_l \in \mathbf{I}_m), \\ \text{-evaluate the principal angles between } A \text{ and } B, \\ \text{-define } d \text{ the descent direction as in 3,} \\ \text{-control parameter variation, } \mathbf{x}_{p+1} = \mathbf{x}_p - \rho d, \\ \text{Stop if } \|d\| \leq \varepsilon, \end{array} \end{array} \right.$$

As A and B are the 'approximate' largest free subspaces (in the sense given in section 5), they generate the respective gradient-based search spaces and we have $\dim(A) = \dim(\text{Span}(G(\alpha_l), \alpha_l \in \mathbf{I}_m))$ and $\dim(B) = \dim(\text{Span}(g(\alpha_l), \alpha_l \in \mathbf{I}_m))$ assuming that what falls beyond the threshold TOL is related to numerical artefacts and should not be considered as potential search directions. Indeed, we give in section 8 an example of how to numerically chose TOL . It is shown that its value should not be too small in order to avoid adding to the maximal free search subspace additional directions related to numerical noises and not containing any pertinent information. Avoiding these extra dimensions also improves the convergence of the optimization algorithm for the solution of our multi-point minimization problem. Indeed, extra gradients would make an admissible descent direction harder to find.

7 Sensitivity analysis

In the algorithms given in sections 3 and 6, at each iteration of minimization we need to provide $\nabla_{\mathbf{x}} j(\mathbf{x}_p, \alpha_l)$ for different values of α_l . This is computer intensive even if it is fully parallel. Our aim is to see how to reduce the complexity of this optimization problem using reduced order modelling. Beyond the computational complexity issue, the necessity for such alternatives also comes from the fact that it is not always possible to proceed with the linearization of the direct simulation chain used for the definition of j . Indeed, consider a general simulation loop linking the independent parameters (α, \mathbf{x}) to a functional j . As described in section 2, α is not a design parameter, but one would like the design to be robust over a range of α :

$$j : (\alpha, \mathbf{x}) \rightarrow (\alpha, q(\mathbf{x})) \rightarrow U(\alpha, q(\mathbf{x})) \rightarrow j(\alpha, \mathbf{x}, q(\mathbf{x}), U(\alpha, q(\mathbf{x}))), \quad (7)$$

where q and U are dependent variables. Suppose q is cheap to compute and U expensive. For instance, q represents geometrical quantities and U state related variables related to the physics of the problem and solution of some costly governing equation. The gradient of j with respect to \mathbf{x} is:

$$\nabla_{\mathbf{x}} j(\mathbf{x}, \alpha) = j_{,\mathbf{x}} + (j_{,q} + j_{,U} U_{,q}) q_{,\mathbf{x}}, \quad (8)$$

where $j_{,\mathbf{x}}, j_{,q}$ and $j_{,U}$ are easy to access and are usually provided by the user as external modules in an industrial simulation platform. $U_{,q}$ and $q_{,\mathbf{x}}$ are, on the other hand, difficult to access. The major part of the cost of this evaluation is due to $U_{,q}$ and one usually uses an adjoint method to make this cost independent of the size of the control space as recalled in algorithm 3. However, today's industrial simulation platforms are more and more based on black-boxes and commercial packages not enabling the user for a direct access to the source of the codes. Differentiating the simulation codes, by automatic differentiation [11] for instance, is therefore off the table. In the same way, it is quite unconceivable to develop in-house adjoint solvers when the cost function calculation relies on commercial packages. The only feasible gradient calculation approach with black-boxes is with finite difference approximation which has a cost proportional to the size of

the control parameter space. Despite this limitation, this technique remains the most widely used. Here, we consider it only as an alternative when an adjoint solution is not available and it is only used to validate sensitivity analysis based on reduced order modelling together with the ingredients presented above.

Suppose we have \tilde{q} and \tilde{U} two reduced order models for q and U expressed at iteration p of optimization as $\Phi\tilde{q}(\mathbf{x}_p) = q(\mathbf{x}_p)$ and $\Psi\tilde{U}(\alpha, \tilde{q}(\mathbf{x}_p)) = U(\alpha, q(\mathbf{x}_p))$. Φ and Ψ are scaling or transfer functions which will be frozen during linearization. The model involves control parameters values \mathbf{x}_p at iteration p of optimization. This indicates that the reduced order model is enriched during the iterations assimilating the new high-fidelity data by (7). Simple forms of the transfer functions are $\Phi = q/\tilde{q}$ and $\Psi = U/\tilde{U}$. We consider the approximate simulation loop j where only black-box and high complexity terms have been approximated:

$$j : (\alpha, \mathbf{x}_p) \rightarrow (\alpha, \Phi\tilde{q}(\mathbf{x}_p)) \rightarrow \Psi\tilde{U}(\alpha, \tilde{q}(\mathbf{x}_p)) \rightarrow j \left(\alpha, \mathbf{x}_p, \Phi\tilde{q}(\mathbf{x}_p), \Psi\tilde{U}(\alpha, \tilde{q}(\mathbf{x}_p)) \right). \quad (9)$$

We need both \tilde{q} and \tilde{U} as q is now often part of a black-box package. Actually, considering a shape optimization situation for instance, the q -code which includes the CAD manipulation tools (CAD: Computer Aided Design) is even more difficult to access than the U -code. An example of Φ linking 3D surface meshes and the corresponding CAD-based geometry definitions is given in the problem of section 8. Φ^{-1} describes the path used to go from the CAD shape definition to the mesh and for a given mesh \mathcal{M} , $\Phi(\mathcal{M})$ is defined in order to minimize the distance between \mathcal{M} and $\Phi^{-1}\Phi\mathcal{M}$ meshes. An example of these relations is shown in figure 2. The notation Φ^{-1} is formal and this operator is not built as the inverse of Φ (same for the operator Ψ and Ψ^{-1} below).

We consider the following linearization of (9) at iteration p of the optimization algorithms in 3 or 6:

$$\nabla_{\mathbf{x}} j(\mathbf{x}_p, \alpha) = j_{,\mathbf{x}} + (j_{,q} + j_{,U}(\Psi\tilde{U}_{,\tilde{q}}(\Psi^{-1}U)))\Phi\tilde{q}_{,\mathbf{x}}, \quad (10)$$

where $\tilde{U}_{,\tilde{q}}(\Psi^{-1}U)$ indicates the linearization of the reduced order model around the high-fidelity solution U restricted to the domain of definition of \tilde{U} . Again, the notation Ψ^{-1} is formal and this transfer operator is not built as the inverse of Ψ (see the multi-grid context in section 7.1.2).

7.1 Examples of reduced order models

The ingredients of the paper will be tested on two examples of low complexity models in the context of shape optimization. The first approximate modelling concerns Hadamard incomplete sensitivity formulation and the second is based on grid coarsening for sensitivity analysis. Below we describe both briefly using the notations above and refer to mentioned references for details of each approach. These two approximations represent two major classes of complexity reduction based either on a model simplification on the continuous level or use of lower accuracy in the discretization for the sensitivity analysis.

7.1.1 Model reduction on the continuous level

One example of gradient approximation useful in our shape optimization problems is Hadamard incomplete sensitivity which addresses the following context [10,15]:

- the cost function j and control \mathbf{x} have the same domain of definition $\mathbf{D}(\mathbf{x})$ (e.g. a shape and an aerodynamic coefficient defined over it),
- j is a product of functions such as $j(\mathbf{x}) = f(\mathbf{x}, q(\mathbf{x})) g(U(q(\mathbf{x})))$.

If these requirements hold, we can use an incomplete evaluation of this gradient, neglecting the sensitivity with respect to the state, leading to the approximation $\nabla_{\mathbf{x}} j = j_{,\mathbf{x}} + j_{,U} q_{,\mathbf{x}} = \nabla_{\mathbf{x}} f g$. This is very interesting as $\nabla_{\mathbf{x}} f$ can be analytically calculated in some situations. If not we will again need an approximate model \tilde{q} as described above leading to $\nabla_{\mathbf{x}} j = j_{,\mathbf{x}} + j_{,U}(\Phi\tilde{q}_{,\mathbf{x}})$. An example of such implementation is given in the problem of section 8 for the linearization of the normal to the shape with respect to the shape parameterization necessary to get the gradient of the aerodynamic coefficients, such as the lift and drag, with respect to the shape. Finally, the approach is also

interesting because the quantities involved are all locally defined on the domain of definition $\mathbf{D}(\mathbf{x})$ of \mathbf{x} and do not involve the full domain of definition of the state variable U or even q .

One can go one step further from incomplete sensitivities introducing simple algebraic relations in \tilde{U} [10]. Examples of such algebraic relations are, for instance, the Newton relation for the pressure distribution over the shape or wall functions describing the tangent velocity and temperature distribution along the direction normal to the wall.

Let us illustrate our purpose with the former relation. Designing a shape with minimum drag with respect to a parameter \mathbf{x} (scalar for clarity) involves an integral on the shape of $p(q(\mathbf{x}), \alpha)u_\infty(\alpha) \cdot n(\mathbf{x})$:

$$C_d(\mathbf{x}, \alpha) = \frac{1}{2\rho_\infty \|u_\infty(\alpha)\|^2} \int_{\text{shape}(\mathbf{x})} p(q(\mathbf{x}), \alpha)(u_\infty(\alpha) \cdot n(q(\mathbf{x}))) d\gamma, \quad (11)$$

where superscript ∞ indicates inflow conditions. The Newton or the cosine-square law for the pressure is: $p(\mathbf{x}, \alpha) = p_{tot}(\alpha)(u_\infty(\alpha) \cdot n(\mathbf{x}))^2$ where $p_{tot}(\alpha)$ is the total pressure function of the inflow conditions α (e.g. inflow Mach number and fluid density). We therefore have $p(\mathbf{x}, \alpha)u_\infty(\alpha) \cdot n(\mathbf{x}) = p_{tot}(\alpha)(u_\infty(\alpha) \cdot n(\mathbf{x}))^3$. Its derivative with respect to \mathbf{x} is $(pu_\infty(\alpha) \cdot n)_{,\mathbf{x}} = (pu_\infty(\alpha))_{,\mathbf{x}} \cdot n + p_{,\mathbf{x}}(u_\infty(\alpha) \cdot n) = 3p_{tot}(\alpha)u_\infty(\alpha)(u_\infty(\alpha) \cdot n)^2 n_{,\mathbf{x}}$. The first term in the sum is what we called above incomplete sensitivity and it worths $(pu_\infty(\alpha))_{,\mathbf{x}} = p_{tot}(\alpha)u_\infty(\alpha)(u_\infty(\alpha) \cdot n)^2 n_{,\mathbf{x}}$. We see that if the pressure is defined by the cosine-square law, the exact and incomplete derivatives only differ by a factor of 3 and have the same sign. Now what happens when the pressure is through the solution of the Euler equations for instance. The expression above can be rewritten as $pu_\infty(\alpha) \cdot n = p|u_\infty(\alpha)| \cos(\frac{u_\infty(\alpha)}{|u_\infty(\alpha)|} \cdot n)$.

The incomplete gradient is therefore $p(u_\infty(\alpha) \cdot n)_{,\mathbf{x}} = -p|u_\infty(\alpha)| \sin(\frac{u_\infty(\alpha)}{|u_\infty(\alpha)|} \cdot n) = 0$ when n is aligned with $u_\infty(\alpha)$. Otherwise, the incomplete sensitivity fails (e.g. in the area near the leading edge for an airfoil at no incidence). On the other hand, the model tells us that the pressure sensitivity with respect to shape variations vanishes if those are along the normal to the shape such that $n_{,\mathbf{x}} = 0$. But this is compatible with the pressure boundary condition $p_{,n} = 0$ one uses with the Euler equations as in the problem of section 8. We therefore expect the incomplete sensitivity to be a good approximation of the gradient if the shape deformation parameterization is along the normal to the shape. Finally, if \tilde{U} is the cosine-square law, Ψ is defined as a scaling term ratio of the pressure by the Euler equations over this expression.

7.1.2 Complexity reduction on the discrete level

Another widely used strategy for gradient approximation is using different space and time discretizations for the high-fidelity calculation with (7) and its approximation (9). This is particularly adapted to multigrid implementations [16, 6, 7] where $\Psi = I_H^h$ is the transfer (interpolation) operator from the coarse H to the fine level h . The two meshes can be separated by n intermediate levels and in this case $\Psi = I_{h_1}^h U_{h_2}^{h_1} \dots I_{h_n}^{h_{n-1}} I_H^{h_n}$. $\Psi^{-1} = R_h^H$ denotes the restriction operator from the fine h to the coarse level H . And in the presence of n intermediate levels $\Psi^{-1} = R_{h_n}^H R_{h_{n-1}}^{h_n} \dots R_{h_1}^{h_2} R_h^{h_1}$.

Hence, principal angles permit to evaluate the impact on the multi-point search space of the different transfer operators between grids. These operators concern, for instance, reconstruction by high order interpolations or different smoothing/filtering strategies to remove the high frequency features of the solution. The angles will tell how the introduction of a new operator impacts the search space defined by the multi-point gradients in a context of robust design.

7.1.3 Evaluation of an incremental or adaptive modelling procedures

Beyond mesh refinement, the most interesting application of this approach is the evaluation of the pertinence of an incremental or adaptive modelling procedure through the evolution of the associated search spaces generated by the gradients at the sampling of the functioning parameter. Indeed, the interest of an increase in modelling complexity, and therefore cost, should be evaluated through its true impact on the search space which is the truly important entity in a design process.

7.2 Defining the descent direction

We need to define d the descent direction in both algorithms given in 3 and 6. Either with an exact or incomplete evaluation of the gradients, one remarks that one should not use all the available vectors $G(\alpha_l)$ or $g(\alpha_l)$ for $\alpha_l \in \mathbf{I}_m$, but only those 'clearly' linearly independent (at given tolerance TOL) and generating subspaces A or B which are of dimension k such as $k < m$. This is because what falls below this tolerance threshold does not bring pertinent information on the problem and is rather due to numerical artifacts. With the approximate models, the descent direction d given in section 3 is evaluated with j instead of \tilde{j} giving \tilde{d} :

$$\tilde{d} = \overline{\nabla_{\mathbf{x}}\mu(j)} - \frac{\langle \overline{\nabla_{\mathbf{x}}\mu(j)}, \overline{\nabla_{\mathbf{x}}\sigma(j)} \rangle}{\|\overline{\nabla_{\mathbf{x}}\mu(j)}\|^2} \overline{\nabla_{\mathbf{x}}\sigma(j)} + \eta \overline{\nabla_{\mathbf{x}}\sigma(j)}. \quad (12)$$

To make sure we only consider the relevant part in the descent direction we proceed with a projection on subspace A or B when working respectively with the exact or incomplete gradients. Recalling that $\{a_i, i = 1, \dots, k\}$ and $\{b_i, i = 1, \dots, k\}$ are orthonormal bases of A and B , the descent directions then becomes:

$$Pr_A(d) = \sum_{i=1, \dots, k} \langle d, a_i \rangle a_i \quad \text{and} \quad Pr_B(\tilde{d}) = \sum_{i=1, \dots, k} \langle \tilde{d}, b_i \rangle b_i.$$

In a context of optimization, these are eventually the most important quantities. Therefore, instead of the principal angles between A and B we might consider the reduced order modelling pertinent if the deviation between these descent directions remains small. This is the case if the following holds through the optimization iterations:

$$\left| 1 - \frac{\langle Pr_A(d), Pr_B(\tilde{d}) \rangle}{\|Pr_A(d)\| \|Pr_B(\tilde{d})\|} \right| \ll 1. \quad (13)$$

This estimator avoids the calculation of the eigenvalues of $M^t M$ described in section 4. We show in section 8 an example of the evolution of this indicator during optimization iterations for a shape design problem.

8 Full aircraft shape optimization

In this section, we show how the pertinence of gradient approximations can be evaluated using the principal angles between the subspaces generated by the gradients of different approximate models and the exact gradient.

The numerical example concerns a shape optimization problem for a full aircraft in transonic cruise condition. For such configurations, an efficient single-point optimization is already a difficult task as it involves several software elements. It also requires coherent geometrical manipulation of the shape parameterization and the surface and volume meshes.

Several sources of variability exist for these problems where large deviations appear between the prescribed shape and the shape during the flight, which we suppose stationary. These may be due to a change in the weight of the aircraft or due to variability in the flight conditions. One example of the latter is when the aircraft cruises against transverse winds which are very common. Usually aircraft are designed for a range of angle of incidence. These designs are usually realized with the sideslip angle set to zero.

We previously used this problem [12] to illustrate robustness and optimal sampling issues in parametric optimization. And also how to provide quantitative confidence levels on the search direction compatible with a search space defined by parametric gradient evaluations. Here, we would like to go one step further and evaluate the impact of gradient approximations using principal angles between subspaces.

Let us briefly link our shape optimization procedure to the direct dependency chain (7) linking independent variables (α, \mathbf{x}) to geometry and state dependent variables (q, U) and to the cost function j and constraints $C_i, i = 1, \dots, n_c$.

8.1 Shape parameterization and other independent variables

In our dependency chain \mathbf{x} denotes a CAD-free parameterization [15] which does not require a priori local regularity assumptions on the shape as it is implicitly the case in CAD-based shape definitions. More precisely, \mathbf{x} represents shape deformations along the normal to the triangular faces of the surface mesh as shown in figure 2. This search space has a dimension $N \sim 5000$. This parameterization receives different denominations and belongs to the same class as node-based or free-form shape definitions. In all these approaches the regularity of the deformation needs to be controlled [21, 10]. This is because, unlike with a CAD definition, the shape $\partial\Omega$ of an object Ω and a gradient-based deformation do not belong to the same function space in terms of regularity and, actually, the second is always less regular. Let us illustrate this on a simple example with $J(\mathbf{x}) = \|\mathbf{Ax} - b\|^2$ taking $\mathbf{x} \in H^1(\partial\Omega)$, \mathbf{Ax} and b in $L^2(\partial\Omega)$. The gradient $J'_\mathbf{x} = 2A^T(\mathbf{Ax} - b)$ belongs to $H^{-1}(\partial\Omega)$. Therefore, any variation along $J'_\mathbf{x}$ will have less regularity than \mathbf{x} : $\delta\mathbf{x} = -\rho J'_\mathbf{x} = -\rho(2(\mathbf{Ax} - b)A) \in H^{-1}(\partial\Omega)$. We therefore need to project (or filter or smooth) into $H^1(\partial\Omega)$. But, why it is not the case with a CAD parameterization? Suppose the shape is described in a finite dimensional parameter space, as for instance with a polynomial definition of a surface (this is like a CAD parameterization). When we consider as control parameters the coefficients of the polynomial, changes in those do not change the regularity as the new shape will always belong to the same polynomial space. Sobolev inclusions give the key for the choice of the regularity operator with the CAD-free parameterization [10]. In our case, because we are using a piecewise linear discretization, a second-order elliptic system is sufficient. This operator will also be used below to enforces some of the geometrical constraints.

The parameter α is the sideslip angle inducing fully 3D effects on the flow around the plane making the consideration of a full aircraft necessary during the design. However, because the airplane geometry is symmetric spanwise, it is not necessary to consider a symmetric range for the transverse wind. We consider the sideslip angle α in $\mathbf{I} = [0, 10^\circ]$. In our multi-point analysis, we consider a uniform sampling \mathbf{I}_m of $[0, 10^\circ]$ with $m = 30$ points. The other flow conditions are given by a Mach number of 0.8 and an inflow incidence of zero degrees. Together with the sideslip angle these parameters fully describe a 3D inviscid flow around the aircraft.

Figure 2 shows the initial shape and a view of the initial triangular surface mesh. This correspondence is given by Φ^{-1} (described in section 7) while Φ permits to construct the corresponding optimized shape from this mesh and the shape deformation over it.

8.2 Geometric and state dependent variables

Let $q(\mathbf{x})$ denotes the auxiliary unstructured mesh related geometrical quantities. When the shape is modified, this change must be propagated through the mesh keeping it admissible and we need to recalculate all related geometrical quantities. Admissible and positive mesh deformation is achieved by a 3D torsional spring analogy method [24].

$U(q(\mathbf{x}), \alpha)$ denotes flow variables depending also on the extra parameter α not part of those involved in the definition of the shape. More precisely, $U(q(\mathbf{x}), \alpha) = (\rho, \rho\mathbf{u}, \rho E)^t$ represents the conservative flow variables solution of the Euler equations in conservation form. Our flow solver is based on a finite volume Galerkin method on unstructured tetrahedral meshes. The Roe [22] flux is used for the approximation of the advection operator together with MUSCL reconstruction with Van Albada limiters [23] in presence of shocks. We target steady solutions and use time marching to reach these. The time integration procedure is explicit and is based on a low-storage Runge-Kutta scheme (RK4). The details of this implementation are available in [10]. Of course, these are not the only choices possible and the literature on numerical methods for compressible flows is huge.

8.3 Cost function and constraints

We consider a classical aerodynamic shape optimization which aims at minimizing the drag coefficient $j = C_d$ given by formula (11). This minimization is usually performed under equality constraints on the lift and volume of the aircraft. Let us denote these by $C_1 = C_l - C_l^0 = 0$ and

$C_2 = V - V^0 = 0$ where superscript 0 indicates the corresponding values for the initial shape. The volume of an object Ω (here the aircraft) is expressed through the boundary integral formula:

$$V = \int_{\Omega} 1 = \int_{\Omega} \frac{1}{3} \nabla \cdot (\mathbf{X}) = \int_{\partial\Omega} \mathbf{X} \cdot \mathbf{n},$$

where $\mathbf{X} = (x_1, x_2, x_3)^t$ is the local coordinate. In addition to these, we consider some extra purely geometric constraints. Minimum surface regularity is prescribed for the wings leading edges. This is enforced through the CAD-free parameterization regularity operator. Other examples are the wings by-section thicknesses and the wings plan-form which are requested to remain unchanged. The latter does not introduce any difficulty but the by-section thickness is less obvious to enforce a posteriori. We define a by-section (figure 3) definition of the shape where the number of sections n_s is free (here $n_s = 100$) and depends on the complexity of the geometry. Each node in the parameterization is associated to a section Σ_i , and for each section, we define the maximum thickness Δ_i . This last operation requires a projective analysis based on the projection of the upper-surface nodes over the lower surface for each section. This constraint can therefore be expressed as: $C_3 = \sum_{i=1}^{n_s} |\Delta_i - \Delta_i^0|$.

8.4 Gradient of the functional and constraints

As we saw in algorithms given in sections 3 and 6, to built the multi-point descent direction, we need m evaluations of the state and functional and constraints sensitivities with respect to the shape. All the sensitivities are computed by automatic differentiation in reverse mode using `tapenade` [17].

Of course, the constraints can be accounted for by introducing a penalty term in the cost function: $J = j + \sum_{i=1,3} a_i C_i$, $a_i \in \mathbb{R}^+$. But this should be avoided when possible. Also, one classical technique to recover the lift during optimization is to change the flow incidence taking advantage of the linear relationship between the incidence and lift away from stall conditions. Suppose we do not want to use either penalty or such an approximation. An alternative would be to consider a locally admissible gradient orthogonal to $\Pi = \text{Span}(\nabla_{\mathbf{x}} C_i, i = 1, 2, 3)$ with $\dim(\Pi) \leq 3$. Let us denote by π an orthonormal basis of this subspace obtained by the Gram-Schmidt procedure applied to the gradients of the constraints. The admissible gradient will be used in algorithms 3 and 6 and is given by:

$$\nabla_{\mathbf{x}} j(\mathbf{x}, \alpha) \leftarrow \nabla_{\mathbf{x}} j - \langle \nabla_{\mathbf{x}} j, \pi \rangle \pi,$$

where \langle, \rangle indicates the scalar product over subspace Π . We need, therefore, to provide $\nabla_{\mathbf{x}} C_i, i = 1, 2, 3$ which require $\nabla_{\mathbf{x}} C_l$, $\nabla_{\mathbf{x}} V$ and $\nabla_{\mathbf{x}} \Delta_i$. $\nabla_{\mathbf{x}} j = \nabla_{\mathbf{x}} C_d$ and $\nabla_{\mathbf{x}} C_l$ receive a same treatment and we take advantage of the capability for multi-right-hand-side adjoint calculation of `tapenade` in reverse mode to access these two gradients without necessitating the solution of two separate adjoint problems. Both also enter the validity domain of incomplete sensitivities as presented in section 8.5. Our direct Euler code uses time marching to the steady solution with local time steps. The corresponding reverse (or adjoint) code can be used easily to produce the incomplete sensitivity by setting to one the number of reverse time iterations for the steady direct state variables. This is interesting as it permits to implement this approximation in existing adjoint based optimization platform with no extra coding. Another optimization of the reverse mode code comes from the fact that, our situations of interest being stationary in time, there is no interest in storing the forward states for backward integration. [25, 10].

Figure 4 shows the contours of a multi-point descent direction based on sensitivities evaluated on a fine and a coarse meshes using the state evaluated on the fine mesh and restricted on the coarse mesh. Below we compare the impact on the multi-point search space of gradient approximations using either the incomplete sensitivity concept or a two-level evaluation of the complete sensitivity.

8.5 Exact vs. incomplete sensitivities

Let us start with the evaluation of the impact on the multi-point search space for our shape optimization problem of a model reduction using the Hadamard incomplete sensitivity concept described in section 8.5. We think this approximation is valid because the cost function and the

constraints are either purely geometric or enter the validity domain of Hadamard incomplete sensitivity as they can be written as a boundary integral over the shape for a product of geometry by state functions. One interest of incomplete sensitivity as presented in section is that there is no need for field variable linearization as everything is defined on the shape. This locality issue is very important in parallel computing where data locality is aimed as much as possible to reduce the communications.

To evaluate the pertinence of this gradient approximation in robust parametric optimization we follow the algorithm given in section 6. In particular, at each iteration of optimization, we identify the size of the free subspaces generated by the exact and incomplete sensitivities. This is achieved using algorithm (6) and is done for a given tolerance level TOL .

Figure 5 shows the dimension of the maximal free search subspace found by the incomplete Gram-Schmidt orthonormalization algorithm (6) for the original shape for different values of the tolerance level TOL . Small values of this parameter lead to large dimension search spaces. But these do not necessarily contain pertinent physical information and are partly related to numerical artifacts and noises. One considers $TOL = 10^{-4}$ which gives a dimension of the search subspace less than 10. This choice appears also a posteriori pertinent from figure 7 where one sees that the dimensions of the maximal free search subspaces decrease during optimization to 4.

This analysis is enforced by figure 6 where one follows $\dim(\text{Span}(\nabla_{\mathbf{x}}j(\mathbf{x}, \alpha_k), \alpha_k \in \mathbf{I}_{30}))$ with $TOL = 10^{-4}$, $TOL = 10^{-6}$ and $TOL = 10^{-7}$ during optimization iterations where the descent directions have been built in the maximal free search subspaces obtained for $TOL = 10^{-4}$. We see that in all cases the dimension of the maximal free search subspace remains below 25 making 30 a safe choice for the sampling. Again, we consider that the dimensions above 10 are related to numerical artifacts and that these search directions should be avoided.

Figure 7 also shows the impact of gradient approximations on the dimension of the maximal free search subspaces during optimization. The evolution of the principal angles between the subspaces generated by the exact and the incomplete sensitivities shows a maximum angular deviation of around 30 degrees but with always some collinear directions between the subspaces. Most important the pertinence of the gradient approximation in the context of optimization is established following indicator (13) showing a deviation of about 15 degrees. This suggests, and this is surprising and non-intuitive, that descent methods will be less sensitive to modelling inaccuracy. Convergence histories for the first and second momentum of the functional over the 10 degrees sideslip angle range demonstrate that the minimization problem is properly addressed and that the choice of the descent direction d by (12) is both pertinent and robust validating the analysis presented in section 3.

8.6 Sensitivity evaluation on coarser meshes

The next complexity reduction approach is, as described in section 7.1.2, when using different discretizations of meshes for the state and sensitivity evaluations. One could simply use a direct solver on a fine mesh h , restrict the state U_h to a coarser mesh using operator $U_H = \Psi^{-1}U_h = R_h^H U_h$, use the adjoint solver to access the gradient G_H on this coarse mesh and interpolate back to the initial mesh with $G_h = \Psi G_H = I_H^h G_H$. Practical meshing and geometry handling issues are not of importance here and any method, based either on structured or unstructured meshes, can be considered.

Figure 8 shows the histories of Gram-Schmidt orthonormalizations during optimization to find the size of the maximal free search subspaces with the exact gradient on the two meshes. The principal angles between these subspaces during optimization show a larger deviation than when using the incomplete sensitivity evaluation on the same fine mesh in figure 7. This somehow recovers what is expected from the incomplete sensitivity theory: it is better to perform an incomplete evaluation of the sensitivity on a fine mesh avoiding any state approximation than an exact evaluation of the sensitivity on a coarse mesh, even with a state restricted from a fine mesh [10]. This can be seen as an a posteriori validation of the dominant importance of the geometric part in these sensitivities over the state dependency. This analysis is also confirmed by the histories of the the mean and variance during an optimization based on the gradients on the coarse mesh, which appear less regular than in figure 7. Comparison of the evolution of indicator (13) in the two

optimizations also confirms sometimes a larger deviation between the exact and approximate (12) descent directions in the present optimization. The descent direction is however eventually better for the second approach.

This evolution can be explained a posteriori as in both optimizations, the principal angles between the subspaces based on the exact and approximate gradients decrease during optimization (see the middle pictures in figures 7 and 8). But, this is even more visible for the second case where we eventually also have more common directions between the two maximal free search subspaces shown by vanishing principal angles. This means that the second approximate models eventually better coincides with the exact model and therefore explains why the optimization performances have been recovered (lower-left picture in figure 8).

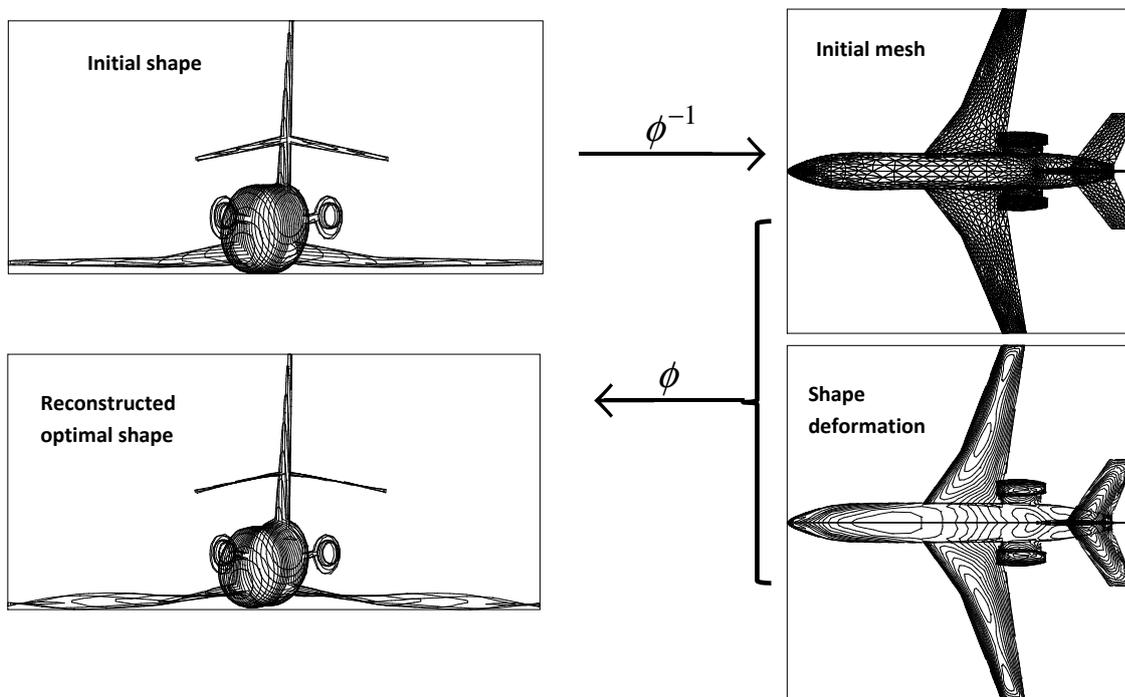


Fig. 2 Initial shape and a view of the initial triangular surface mesh. This correspondence is given by Φ^{-1} while Φ permits to construct the corresponding optimized shape from this mesh and the shape deformation over it.

9 Concluding remarks

The impact of different gradient approximations by reduced order models has been evaluated in the context of robust shape design by multi-point optimization. In particular, the deviations between the multi-point search spaces based on the exact evaluation of the gradients and these approximations have been analyzed using the concept of principal angles between subspaces. It has been shown how this can be seen as a singular value decomposition problem which can be reduced in our situation to an eigenvalue evaluation problem for a symmetric matrix of small size. The reduction in size has been made through the extraction of the pertinent information in the multi-point vector search space. This approach can therefore be seen as a tool to validate reduced order models in the context of robust optimization. Indeed, what is important is to make sure that model reduction does not introduce any major perturbation in the free search subspace. Finally,

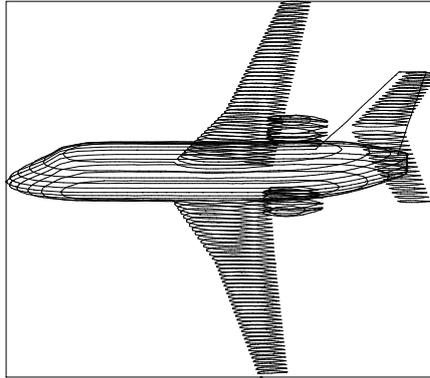


Fig. 3 An example of the spanwise slices of an aircraft used to enforce the by-section maximum thickness constraint.

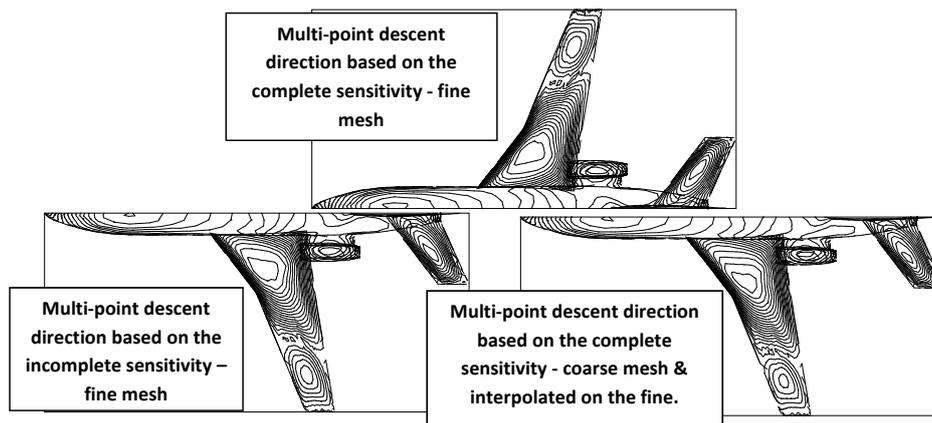


Fig. 4 Iso-contours of a multi-point descent direction based on the complete and incomplete sensitivities on the fine and coarse levels. The sensitivity on the coarse level has been evaluated using the state restricted from the fine to the coarse level with operator $\Psi^{-1} = R_h^H$ and the sensitivity transferred back to the fine level using interpolation operator $\Psi = I_H^h$.

it has been shown that the situation can be more favorable when using a descent minimization algorithm as the reduced order model gradients need then to verify a less constraining criteria.

Beyond the examples presented, the approach provides a valuable tool to a posteriori measure the 'distance' between models. In particular, this could be used with adaptive models. Indeed, in practice the high-fidelity model is not available or too expensive and one would like to be able to identify the 'right' level of complexity for the model to be used during design. This concerns both the continuous model and its discretizations. In that sense, we showed examples of the impact of grid refinements on the deviation between the search spaces generated by the sensitivities for the different meshes. Hence, the approach permits to quantify the pertinence of an increase in the modelling complexity through the deviation between the search spaces.

Acknowledgements The author would like to thank F. Gallard, M. Meaux and M. Montagnac from Airbus, EADS-IW and Cerfacs for their feedbacks.

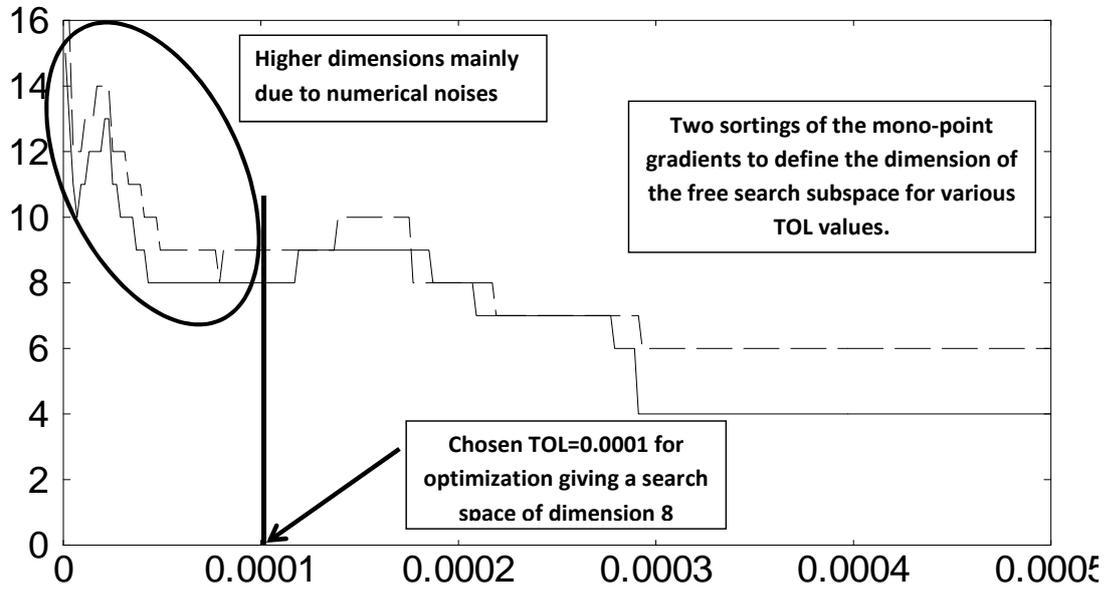


Fig. 5 Dimension of the maximal free search subspace versus the tolerance value TOL in the incomplete Gram-Schmidt orthonormalization algorithm (6) evaluated for the original shape. This evaluation helps choosing TOL in order to avoid non pertinent search directions due to numerical artifacts. The curves are for two different sortings of the gradients during orthonormalization.

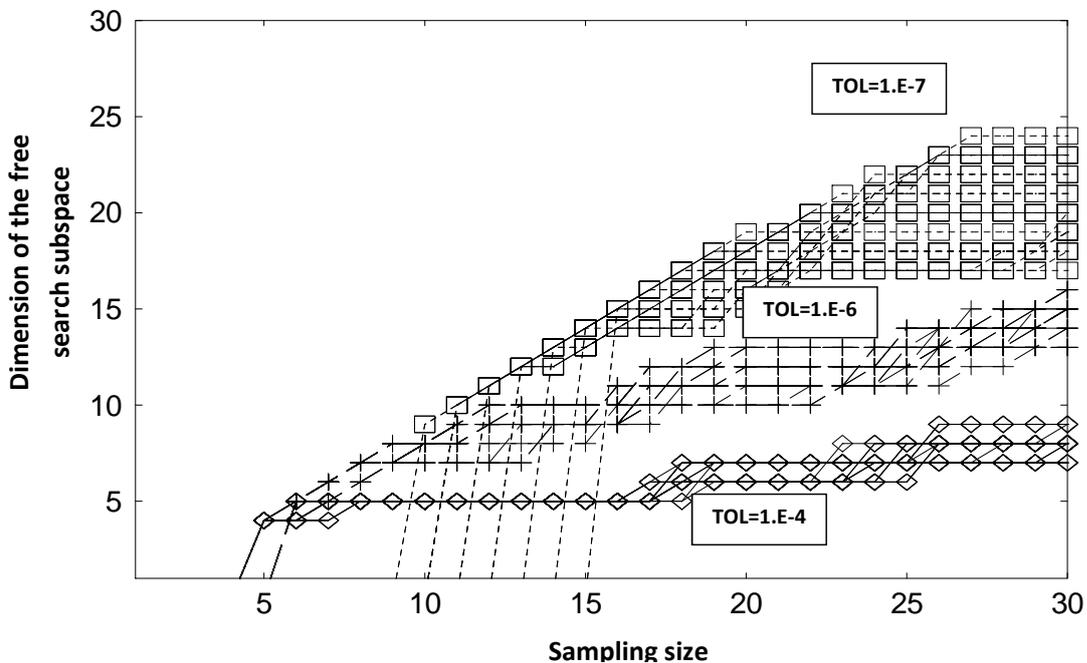


Fig. 6 Histories of Gram-Schmidt orthonormalization at optimization iterations for three levels of TOL . The dimension of the maximal free search subspace always remains below 25, making 30 a safe choice for the sampling.

References

1. Schilders, W. H. Van der Vorst, H. A. Rommes, J. *Model order reduction: Theory, research aspects and applica-*

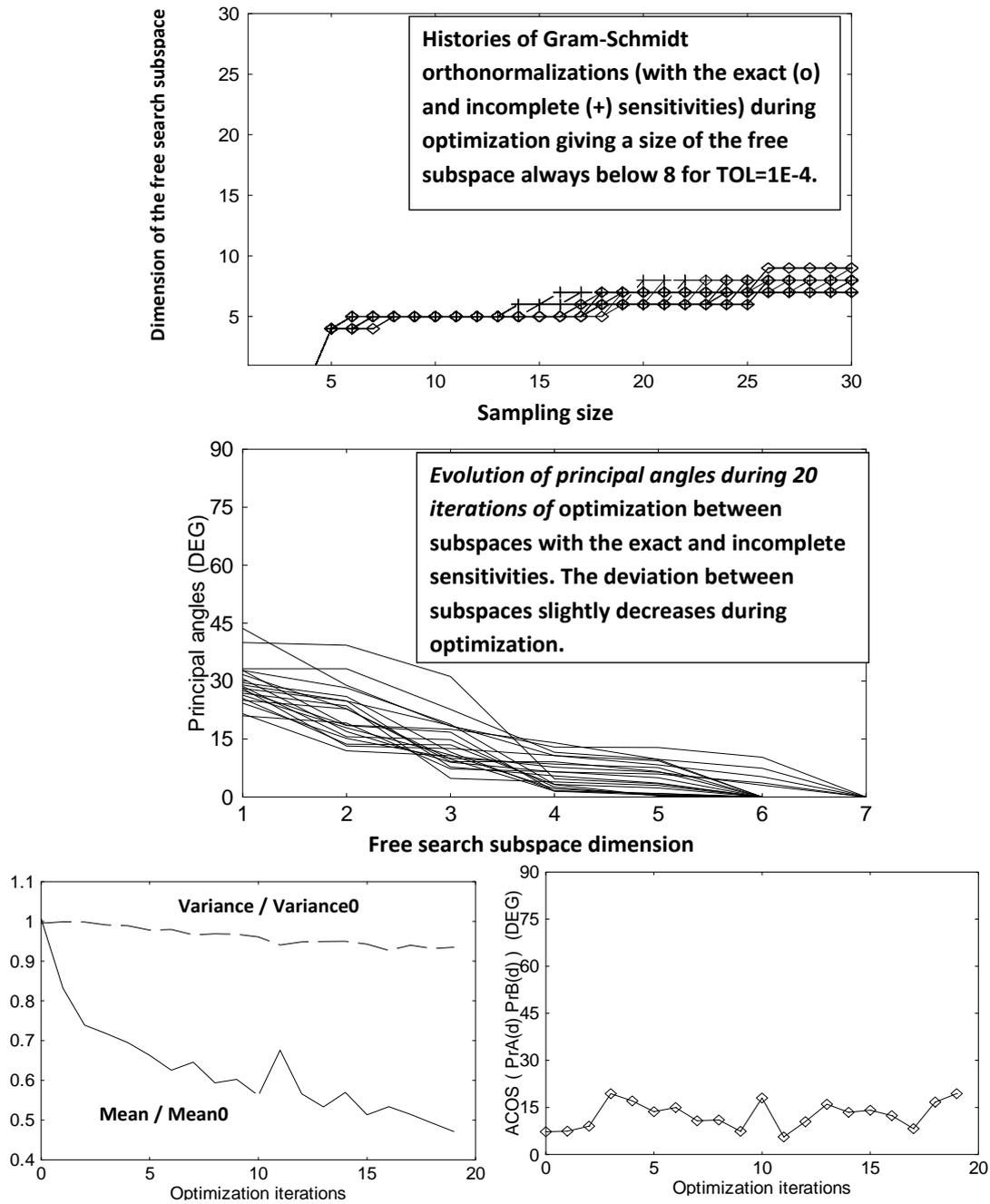


Fig. 7 Upper: histories of Gram-Schmidt orthonormalizations during optimization to find the size of the maximal free search subspaces with the exact and incomplete gradients. Middle: principal angles between these subspaces during optimization (a curve per iteration). Lower-left: optimization histories of the mean and variance of the functional over the 10 degrees sideslip angle range. Lower-right: angles between the two descent directions given by indicator (13).

- tions, Springer Math in Industry series Vol. 13, Berlin, (2008).
- Obinata, G. Andersonn B. *Model reduction for control system design*, Springer, Berlin (2000).
 - Qu, Z. *Model Order Reduction Techniques with Applications in Finite Element Analysis*, Springer, Berlin (2004).
 - Jordan, C. Essay on geometry in n dimensions, *Bull. Soc. Math. France*, **3**, 103:174 (1875).
 - Gluck, H. Warner, F. Great circle fibrations of the three-sphere, *Duke Math. J.*, **50**, 107-132 (1983).
 - Morano, E. Mavriplis, D. Venkatakrishnan, V. Coarsening strategies for unstructured multigrid techniques with application to anisotropic problems, *SIAM J. Sci. Comput.*, **20-2**, 393:415 (1998).

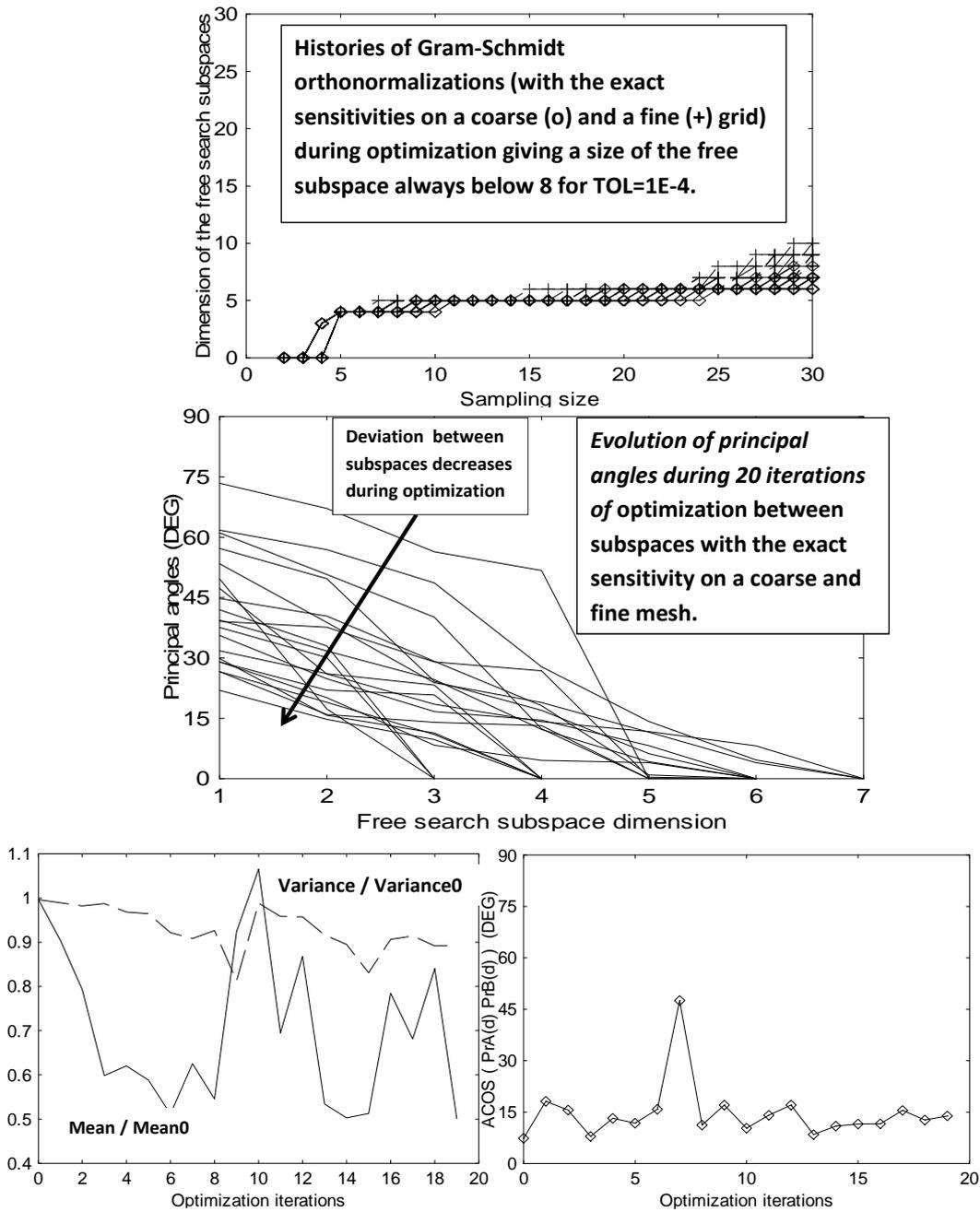


Fig. 8 Upper: histories of Gram-Schmidt orthonormalizations during optimization to find the size of the maximal free search subspaces with the exact gradient on these meshes. Middle: principal angles between these subspaces during optimization (a curve per iteration). Lower-left: optimization histories of the mean and variance of the functional over the 10 degrees sideslip angle range. Lower-right: angles between the two descent directions given by indicator (13).

7. Lallemand, M.-H. Steve, H. Dervieux, A. Unstructured multigridding by volume agglomeration: Current status, *Computers and Fluids*, **21**, 397:433 (1992).
8. Jiang, S. Angles between Euclidean subspaces, *Geometriae Dedicata*, **63(2)**, 113-121 (1996).
9. Shonkwiler, C. Poincare duality angles for Riemannian manifolds with boundary, PhD. thesis, Univ. Pennsylvania (2009).
10. Mohammadi B., Pironneau O. *Applied Shape Optimization for Fluids* (2nd Edition), Oxford Univ. Press, London

- (2009).
11. Griewank, A. *Computational derivatives*, Springer, New York, (2001).
 12. Mohammadi B. Reduced sampling and incomplete sensitivity for low-complexity robust parametric optimization, *Int. J. Num. Meth. Fluids*, DOI: 10.1002/fld.3798 (2013).
 13. Mohammadi B. Value at Risk for confidence level quantifications in robust engineering optimization, *optimal Control: Applications and Methods*, DOI: 10.1002/oca.2061 (2013).
 14. Melchers, R. E. *Structural Reliability Analysis and Prediction*. John Wiley and Sons, Chichester (1999).
 15. Mohammadi, B. Pironneau, O. Shape Optimization in Fluid Mechanics, *Annual Revue of Fluid Mechanics*, **36/1**, 255:279 (2004).
 16. Hackbush, W. *Multi-Grid Methods and Applications*, Springer Series in Computational Mathematics, Berlin, Vol. 4 (1985).
 17. Hascoet, L. Pascual, V. Tapenade 2.1 user's guide. INRIA Technical Report RT-300 (2004).
 18. Gallard, F. Mohammadi, B. Montagnac, M. Meaux, M. Robust parametric design by multi-point optimization. CERFACS Technical Report TR/CFD/12/33 (2012).
 19. Desideri, A. Multiple-gradient descent algorithm for multiobjective optimization. CRAS. **350/5**, 313:318 (2012).
 20. Li, W. Huyse, L. and Padula, S. Robust Airfoil Optimization to Achieve Consistent Drag Reduction Over a Mach Range, *Structural and Multidisciplinary Optimization*, **24/1**, 38-50 (2002).
 21. Firl, M. Wuchner, R. Bletzinger, K. Regularization of shape optimization problems using fe-based parametrization, *Structural and Multidisciplinary Optimization*, **47/4**, 507-521, (2013).
 22. Roe, P.L. Approximate Riemann solvers, parameters vectors and difference schemes, *J.Comp. Phys.* **43**, 357-372, (1981).
 23. Van Albada, G.D. and Van Leer, B. Flux vector splitting and Runge Kutta methods for the Euler equations, ICASE **84-27**, 1984).
 24. Farhat, C. Degand, C. A three-dimensional torsional spring analogy method for unstructured dynamic meshes, *Computers & Structures*, **80/3**, 305-316 (2002).
 25. Christianson, B. Reverse accumulation and implicit functions, *Optim Methods Software*, **9/4**, 307-322, 1998.