# OPTIMAL AERODYNAMIC DESIGN UNDER SHAPE UNCERTAINTIES

VOLKER SCHULZ * AND CLAUDIA SCHILLINGS†

**Abstract.** The unavoidable presence of uncertainties poses several difficulties to the numerical treatment of optimization tasks. A novel approach towards stochastic distributed uncertainties is discussed for the specific application of shape uncertainties in aerodynamic design. Algorithmic approaches based on multiple-setpoint ideas in combination with one-shot methods are presented as well as numerical results.

**Key words.** Shape optimization, robust optimization, stochastic uncertainties.

**1. Introduction.** Aerodynamic design aims at providing shapes for parts of an aircraft such that an objective like the drag is minimized subject to important aerodynamic restrictions like maintaining a certain level of aerodynamic lift or pitching moment. The major computational burden is imposed by often expensive flow solvers. Therefore structure exploitation of these solvers is of utmost importance, when designing an efficient design algorithm. In particular one-shot techniques for the design problem are discussed in [1, 2, 3, 4].

Nevertheless, additional computational effort is required in handling unavoidable uncertainties arising in the formulation of the design tasks. Several approaches to this more complex problem class are discussed in [5, 6, 7] for scalar uncertainties. In this publication, we focus on distributed or function-valued uncertainties and develop a novel approach to efficient robust aerodynamic design in this context. In particular, we aim at robust designs for the case that the shape to be optimized is uncertain itself. From an application point of view, we look for shapes which give a good optimal model answer (in terms of drag minimization and lift preservation), even if the computed shape undergoes unforeseeable changes due to manufacturing inaccuracies or wear and tear. The major methodological approach exploits the truncated Karhunen-Loeve-expansion, uses sparse grid approaches and applies a goal oriented point of view towards the optimization objective in order to reduce the computational effort. A major algorithmic tool is the multiple-setpoint approach towards robustness issues

*Prof. Dr. V. Schulz, FB IV Department of Mathematics, University of Trier ,Universitätsring 15, 54296 Trier, Germany,(volker.schulz@uni-trier.de )

†Dipl.-Math. C. Schillings, , FB IV Department of Mathematics, University of Trier ,Universitätsring 15, 54296 Trier, Germany,(claudia.schillings@uni-trier.de )

established in [8, 9].

The paper is organized in the following way. The next section gives a characterization of the distributed uncertainties we are dealing with together with introductory application of the Karhunen-Loeve-expansion. Section 3 presents the discretized version the resulting robust shape optimization problem. Section 4 modifies this technique in a goal-oriented manner. Section 5 discusses sparse grid techniques. In section 6 the algorithmic working horse, the one-shot approach, is generalized to the robust problem formulation. Finally, section 7 presents numerical results for a standard transonic test case in a robust formulation.

**2. Mathematical description of the uncertainties.** To include geometry uncertainties in the aerodynamic shape optimization problem, we need a mathematical description of the perturbations on the shape. We want to avoid a parametrization of the uncertainties which would lead to a reduction of the space of perturbed geometries. Therefore, we choose a stochastic approach. The perturbations of the profile are modeled as a random field $\psi : \Gamma, \Omega \to \mathbb{R}$, defined on a probability space $(\Omega, Y, P)$ and on the shape of the airfoil $\Gamma$. In each point $x$ of the shape $\Gamma$, the uncertainty is described by a random variable $\psi(x, \cdot) : \Omega \to \mathbb{R}$. Additionally, the second order statistics, the mean value and the covariance function, are given to fully describe the random field.

Since we expect no perturbations, the mean value of the random field $\psi$ is equal to 0 and the squared exponential covariance function describes the interaction between the random variables on the shape:

$$\mathbb{E}\left(\psi\left(x,\zeta\right)\right) = \psi_0\left(x\right) = 0 \qquad \forall x \in \Gamma \tag{2.1}$$

$$Cov\left(x,y\right) = b^2 \cdot \exp\left(-\frac{\|x-y\|^2}{l^2}\right) \qquad \forall x,y \in \Gamma \tag{2.2}$$

The parameter $l$ determines how quickly the covariance falls off and $b$ controls the magnitude of the bumps. Then, a perturbed geometry is given as

$$v\left(x,\zeta\right) = x + \psi\left(x,\zeta\right) \cdot \vec{n}\left(x\right) \qquad \forall x \in \Gamma, \zeta \in \mathcal{Q} \tag{2.3}$$

where $\vec{n}$ is the unit vector in $x$ normal to the profile $\Gamma$. Figure 2.1 shows a realization of the

random field $s$ and the corresponding, resulting pertubed shape Figure 2.2.

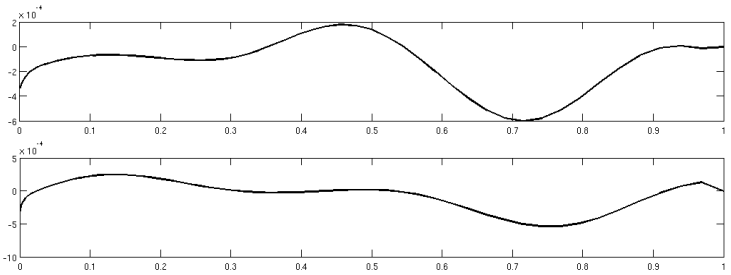

FIG. 2.1. *one realization of the random field $\psi$: perturbations on the upper side of the profile (above) and on the lower side (below)*

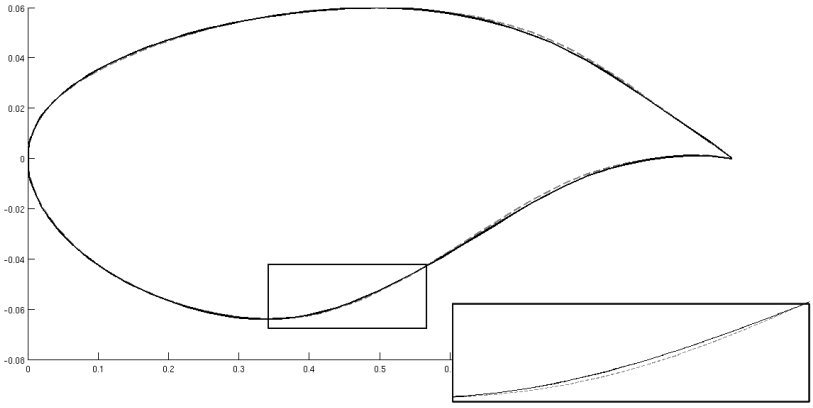

FIG. 2.2. *resulting perturbed geometry compared with the original shape (in green)*

As we need to compute statistics of the flow depending on the uncertainty in our optimization algorithm, we have to approximate the infinite dimensional probability space in a finite number of random variables. In the next subsection, we will introduce the Karhunen-Loève-Expansion which provides an approximation of the random field $\psi$ for the numerical evaluation of such statistics.

**2.1. Karhunen-Loève-Expansion.** The Karhunen-Loève-Expansion, also known as Proper Orthogonal Decomposition, represents the random field as a infinite linear combination of orthogonal functions chosen as the eigenfunctions of the covariance function [25],

[27]. The Karhunen-Loève-Expansion of the Gaussian random field $\psi$ is given as:

$$\psi(x,\zeta) = \psi_0(x) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} z_i(x) Y_i(\zeta) \tag{2.4}$$

$$= \sum_{i=1}^{\infty} \sqrt{\lambda_i} z_i(x) Y_i(\zeta) \qquad x \in \Gamma, \zeta \in \Omega \tag{2.5}$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_i \geq \ldots \geq 0$ and $z_i$ are the eigenvalues and eigenfunctions of the covariance function $Cov$ which is symmetric and positive definite by definition. The deterministic eigenfunctions $z_i$ are obtained from the spectral decomposition of the covariance function via solution of

$$\int_{\Gamma} Cov(x,y) z_i(y) = \lambda_i z_i(x). \tag{2.6}$$

Having the eigenpairs, the uncorrelated Gaussian random variables $Y_i$ in equation 2.5 can be expressed as

$$Y_i(\zeta) = \frac{1}{\sqrt{\lambda_i}} \int_{\Gamma} \psi(x,\zeta) z_i(x) \, \mathrm{d}x \qquad j = 1, 2, \ldots \tag{2.7}$$

with zero mean and unit variance, i.e. $\mathbb{E}(Y_i) = 0$ and $\mathbb{E}(Y_i Y_j) = \delta_{ij}$ for $j = 1, 2, \ldots$ [13]. In the special case of a Gaussian random field, uncorrelated implies independent which is an important property we will need later on for the Sparse Grid.

Truncating now the Karhunen-Loève-Expansion after a finite number of terms, we obtain the approximation of the random field $\psi$

$$\psi_d(x,\zeta) = \sum_{i=1}^{d} \sqrt{\lambda_i} z_i(x) Y_i(\zeta) \qquad x \in \Gamma, \zeta \in \Omega. \tag{2.8}$$

The corresponding covariance function is given by

$$Cov_d(x,y) = \sum_{i=1}^{d} \lambda_i z_i(x) z_i(y). \tag{2.9}$$

In [19], it is shown that the eigenfunction basis $\{z_i\}$ is optimal in the sense that the mean square error resulting from the truncation after the d-th term is minimized.

The following approximation error representation is then obtained by Mercer's theorem

$$\lim_{d\to\infty}\left\{\sup_{\Gamma}\int_{\Omega}\left(\psi-\psi_d\right)^2 d\mathcal{P}\left(\zeta\right)\right\}=\lim_{d\to\infty}\left\{\sup_{\Gamma}\left(\sum_{j=d+1}^{\infty}\lambda_j z_j^2\right)\right\}=0. \tag{2.10}$$

So, $\psi_d$ may provide a suitable approximation of $\psi$, if the eigenvalues decay sufficiently fast and $d$ is large enough [13]. If one assumes a Gaussian covariance function (cf. 2.2), the eigenvalues will exponentially decay towards zero. The proof of this behaviour of the eigenvalues can be found e.g. [29]. This paper also provides a fast algorithm based on a kernel independent fast multipole method to compute the Karhunen-Loève approximation. Another approach to solve the large eigenvalue problem arising from the Karhunen-Loève-Expansion can be found in [23]. They introduce a Krylov subspace method with a sparse matrix approximation using sparse hierarchical matrix techniques to solve it.

**3. Robust shape optimization problem.** The usual single setpoint aerodynamic shape optimization problem can be described in the following rather abstract form

$$\min_{y,p} f(y,p) \tag{3.1}$$

$$\text{s.t. } c(y,p)=0 \tag{3.2}$$

$$h(y,p)\geq 0 \tag{3.3}$$

We think of the equation (3.2) as the discretized outer flow equation around, e.g., an airfoil described by geometry parameter $p\in\mathbb{R}^{n_p}$. The vector $y$ is the state vector (velocities, pressure,...) of the flow model (3.2) and we assume that (3.2) can be solved uniquely for $y$ for all reasonable geometries $p$. The objective in (3.1) $f:(y,p)\mapsto f(y,p)\in\mathbb{R}$ typically is the drag to be minimized. The restriction (3.3) typically denotes lift or pitching moment requirements. To make the discussion here simpler, we assume a scalar valued restriction, i.e., $h(y,p)\in\mathbb{R}$. The generalization of the discussions below to more than one restriction is straight forward. In contrast to previous papers on robust aerodynamic optimization, we treat the angle of attack as an fixed parameter which is not adjusted to reach the required lift (cp. e.g. Ref.[22], Ref.[21], Ref.[28] ).

Uncertainties arise in all aspects of aerodynamic design. However, we want to limit the discussion here to uncertainties which cannot be avoided at all before constructing an aircraft. We distinguish two types of uncertainties: uncertainties with respect to the flight conditions and geometry uncertainties. The comparison of different robust formulation considering scalar valued uncertainties in the flight conditions can be found in [7].

In this paper, we will focus on geometry uncertainties. We mean by this the case that the real geometry deviates from the planned geometry. The sources for deviations from the planned geometry may lie in manufacturing, usage and wearing of the aircraft or wheather conditions (e.g., ice crusts). In order to compute a shape which is robust to small perturbations of the shape itself, we choose a semi-infinite formulation of the aerodynamic shape optimization influenced by stochastic disturbances modeled as random fields. This results in the following robust optimization problem

$$\min_{y,p} \int_{\Omega} f(y,p,\psi\left(\zeta\right))d\mathcal{P}(\zeta) \tag{3.4}$$

$$\text{s.t. } c(y,p,\psi\left(\zeta\right)) = 0\,,\ \forall\zeta \in \Omega \tag{3.5}$$

$$h(y,p,\psi\left(\zeta\right)) \geq 0\,,\ \forall\zeta \in \Omega \tag{3.6}$$

The semi-infinite reformulation 3.7-3.9 aims at optimizing the average objective function but maintaining the feasibility with respect to the constraints everywhere. Thus, it aims at an average optimal and always feasible robust solution. To ensure the feasibility over the whole range of variations, we have to consider infinitely many inequality constraints, one for each realizations of the random field $s$ of perturbations. This definition of robustness can also be found in Ref. [7], in Ref.[21] and in Ref.[24]. For the numerical treatment of complicated design tasks, one has to approximate the integral in the objective (3.4). The integral in (3.4) can be efficiently evaluated by a Gaussian quadrature for small stochastic dimensions, where the quadrature points $\{\psi^i\}_{i=1}^N$ are the roots of a polynomial belonging to a class of orthogonal polynomials. Due to the exponential growth of the effort with increasing dimension, the full tensor product Gaussian quadrature rule should be replaced in the higher dimensional case by Smolyak type algorithms which use a recursive contribution of lower-order tensor products

to estimate the integral Ref.[30]. We will discuss this method in the next section. Therefore, we can reformulate problem (3.4-3.6) in an approximate fashion in the form of a multiple set-point problem for the set-points $\{\psi^i\}_{i=1}^N$:

$$\min_{y_i,p} \sum_{i=1}^{N} f(y_i, p, \psi^i)\omega_i \tag{3.7}$$

$$\text{s.t. } c(y_i, p, \psi^i) = 0\,,\ \forall i \in \{1, \ldots, N\} \tag{3.8}$$

$$h(y_i, p, \psi^i) \geq 0\,,\ \forall i \in \{1, \ldots, N\}. \tag{3.9}$$

This approximation supports a generalization of the optimization method used to solve the usual single setpoint aerodynamic shape optimization problem very well. For each realization of the uncertainty, the update in the design variables can be computed completely independently, so this formulation suggests a parallel implementation of the optimization method which we will introduce in section 6.

**4. Model reduction using a goal-oriented Karhunen-Loève basis.** The evaluation of the objective function in the robust optimization problem (3.4) requires the computation of the mean, i.e. the computation of the integral of the random field with respect to its probability measure. Applying the introduced Karhunen-Loève-Approximation, the objective function can be written as the following d-dimensional integral

$$\mathbb{E}\left(f(y, p, \psi\left(x, \zeta\right))\right) = \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} (f(y, p, s\left(x, \zeta_1, \ldots, \zeta_d\right))\mathrm{d}\gamma_1\left(\zeta_1\right) \cdots \mathrm{d}\gamma_1\left(\zeta_d\right) \tag{4.1}$$

where $\mathrm{d}\gamma_1\left(\zeta_i\right)$ is the one-dimensional Gaussian measure. So, one term more in the truncated Karhunen-Loève expansion to increase the approximation accuracy results in an integral of one-dimension higher. In order to reduce the computational effort, the orthogonal basis functions $\{z_i\}$ will be chosen goal-oriented, i.e. the individual impact of the eigenvectors on the target functional will be taken into account. This method is well established in the model reduction methods of dynamic systems and the adaptive mesh refinement (cf. [10]). The idea is to develop an error indicator for the individual eigenvectors reflecting the influence on the drag. The introduced error analysis of the Karhunen-Loève-Expansion in section 2.1

only gives the approximation error of the random field $\psi$, but not of the function of interest $f\left(y,p,\psi\right)$. We propose to use sensitivity information to capture the local sensitivities of the drag with respect to the eigenvectors

$$\eta_i := \frac{\mathrm{d}f}{\mathrm{d}z_i} = -\lambda^\top \frac{\partial c}{\partial z_i} + \frac{\partial f}{\partial z_i}, \qquad \forall i = 1,\dots,d \tag{4.2}$$

where $\lambda$ solves the adjoint equation. The adjoint equation is independent of $i$, hence it has to be solved only once and the indicator $\eta_i$ is numerically cheap to evaluate. Now, the reduced basis $\{\hat{z}_i\}$ can be automatically selected, the eigenvector $z_i$ with a large value $\eta_i$ have to be kept in the reduced basis, whereas a small value indicates that the basis vector can be rejected from the basis.

**5. Adaptive Sparse Grid for high-dimensional integration.** The mean value of the robust optimization problem depending on the current design vector is required in each iteration of the optimization algorithm. Since we cannot solve this integral analytically, we have to approximate it in appropriate, efficient way. Several possibilities can be found in the literature, the most common are: Monte-Carlo simulation , respectively general Sampling methods, full Tensor grid interpolation and Sparse Grid interpolation. Their efficiency depends on the dimension $d$ of the probability space $\Omega_d$ and on the properties of the integrand $f(y,p,\psi_d)$. Each of these methods provides an approximation $\mathbb{E}_N$ of the mean value $\mathbb{E}(f)$ by evaluating the function $f(y,p,\psi_d)$ in $N$ integration points $\zeta_1,\cdots,\zeta_N$ and summing the results $f(y_i,p,\psi_d(.,\zeta_i))$ multiplied with the weights $\omega_1,\cdots,\omega_N$ up

$$\mathbb{E}_N = \sum_{i=1}^{N} \omega_i \cdot f(y_i,p,\psi_d(.,\zeta_i)) \tag{5.1}$$

The Sampling methods randomly select realizations of the uncertainties in the given probability space and take some kind of average of the function values at these points which converges to the exact value of the integral due to the law of large numbers. The advantage of this approach consists of the straightforward implementation, the algorithm only needs the underlying integration space as input and function evaluations at the randomly selected points. But on the other hand, the expected convergence rate $O(N^{-\frac{1}{2}})$ requires a large num-

ber of function evaluations to ensure a given error tolerance. In our application, one function evaluation is very expensive since the solution of the flow equation, Euler or Navier Stokes equation, is needed. So, the Sampling methods, even the improved methods which use additional information in order to select the realizations, are not an appropriate choice in our case to compute the mean value in our optimization problem.

Another possibility obtaining the objective value is the full tensor grid quadrature derived from the full tensor product of the one-dimensional interpolation formulas. Constructing the multi-dimensional interpolation, we first consider the following one-dimensional interpolation formula in order to approximate a function $h : [-1, 1] \rightarrow \mathbb{R}$:

$$Q^i(h) = \sum_{j=1}^{N} h(Y_j^i) \cdot a_j^i \tag{5.2}$$

with the set of interpolation points $X^i = \{Y_j^i | Y_j^i \in [-1, 1], j = 1, 2, \ldots, m_i\}$ , $m_i$ is the number of elements of the set $X^i$ and $a_j^i \equiv a_j(Y_j^i)$ are the interpolation functions.

$$(Q^{i_1} \otimes \cdots \otimes Q^{i_d})(f) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_d=1}^{m_d} f(Y_{j_1}^{i_1}, \ldots, Y_{j_d}^{i_d}) \cdot (a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_d}^{i_d}) \tag{5.3}$$

This generalization of the one dimensional formula to the full tensor interpolation (5.3) provides an approximation of $f : [-1, 1]^d \rightarrow \mathbb{R}$ by evaluating the function $f$ on the regular mesh $X^{i_1} \times \cdots \times X^{i_d}$. Considering the difference formulas defined by

$$\Delta^i := Q^{i+1} - Q^i \tag{5.4}$$

$$Q^0 := 0 \tag{5.5}$$

5.3 can be reformulated as

$$(Q^{i_1} \otimes \cdots \otimes Q^{i_d})(f) = \sum_{i_1, \ldots, i_d \leq k} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right)(f) \tag{5.6}$$

The mean value is then derived from the following equation:

$$\mathbb{E}_N(f) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_d=1}^{m_d} f(Y_{j_1}^{i_1}, \ldots, Y_{j_d}^{i_d}) \cdot \int_{[-1,1]^d} (a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_d}^{i_d})(\mathbf{Y}) d\mathbf{Y} \tag{5.7}$$

The approximation error for functions with bounded derivatives up to order $r$ has a behaviour of $O(N^{-\frac{r}{d}})$ [15]. Thus, due to the exponential growth of the effort with increasing dimension, this method is not suitable for high stochastic dimensions, which is the case in our application. To circumvent this curse of dimensionality, we apply a Sparse Grid method in order to preserve the accuracy of the tensor grid quadrature, but avoiding the exponential growth of interpolation nodes.

The underlying idea of Sparse Grids was originally found by the Russian mathematician Smolyak [30]. The sparse interpolant is given as [31]:

$$S(k,d)(f) = \sum_{k-d+1 \leq |\mathbf{i}| \leq k} (-1)^{k-|\mathbf{i}|} \cdot \binom{d-1}{k-|\mathbf{i}|} \cdot (Q^{i_1} \otimes \cdots \otimes Q^{i_d})(f) \tag{5.8}$$

with $k \geq d$, $\mathbf{i} \in \mathbb{N}^d$ multi-index and $|\mathbf{i}| = \sum_{j=1}^{d} i_j$. The index $i_j$ indicates the order in the $j$-th dimension, so the algorithm combines only those one-dimensional quadrature formulas whose indices fullfill the constraint that the total sum across all dimensions is greater or equal than $k-1+d$ and smaller or equal than $k$. Using incremental interpolation formulas $\Delta^i$, 5.8 can be transformed to

$$\begin{aligned} S\left(k,d\right)\left(f\right) &= \sum_{|\mathbf{i}| \leq k} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right)\left(f\right) \\ &= S\left(k-1,d\right)\left(f\right) + \sum_{|\mathbf{i}|=k} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right)\left(f\right) \end{aligned}$$

with $\Delta^i = Q^{i+1} - Q^i$, $Q^0 \equiv 0$ and $S(d-1,d) \equiv 0$. The collection of all the interpolation points

$$\mathcal{H}(k,d) = \bigcup_{k-d+1 \leq |\mathbf{i}| \leq k} \left(X^{i_1} \times \cdots \times X^{i_d}\right) \tag{5.9}$$

is called a Sparse Grid of level $k$.

The derivation of the Sparse Grid suggests the use of nested interpolation functions due to the recursive construction. In the literature, the most popular choice of the collocation points is the Clenshaw-Curtis grid at the non-equidistant extrema of the Chebyshev polynomials and the underlying interpolation formula is the Chebyshev-Gauss-Lobatto formula.

**5.1. Adaptive Sparse Grid.** Since the function evaluations are very expensive in our application, we use an adaptive Sparse Grid strategy in order to further reduce the number of grid points but conserving the approximation quality. The presented isotropic Smolyak algorithm is effective for problems whose input data uniformly depend on all dimensions. But the convergence rate deteriorates for highly anisotropic problems, such as those appearing when the input random variables come from a Karhunen-Loève-Expansion as in our application [16]. The reduction of computational effort can be achieved by using spatially adaptive or dimension-adaptive refinement [31], [18]. In order to develop adaptive schemes during the cubature process, the interpolation error can be used as an adaptivity indicator. Therefore, nested cubature formulas are useful since they allow the error evaluation based on the difference of two subsequent formulas. Due to the fact that in our application the mean value is computed by the Sparse Grid interpolation, this target value is also used as an error indicator for the adaptivity. The dimension-adaptive quadrature method tries to find important dimensions and adaptively refines in this with respect to given error estimators. This leads to an approach which is based on generalized sparse grid index sets [18]. This strategy allows to employ every nested interpolation formulas, so it can be chosen problem dependent, e.g. in our application depending on the distribution of the random variables. On the other hand, the locally refined Sparse Grid gives more flexibility in the adaptive procedure, but requires equidistant support nodes.

Below, we introduce a locally adaptive hierarchical Sparse Grid approach using piecewise multilinear hierarchical basis functions following closely [26], [31]. Due to the straightforward implementation of the refinement, we choose the linear hat functions as interpolation basis functions which are also well established in the adaptive mesh refinement [11]. Hence,

the interpolation formulas in the one dimensionl case are defined by

$$a_j^i(Y) = \begin{cases} 1 - \frac{1}{2}(m_i - 1) \cdot \left|Y - Y_j^i\right|, & \text{if } \left|Y - Y_j^i\right| < \frac{2}{m_i - 1} \\ 0, & \text{otherwise.} \end{cases} \tag{5.10}$$

with the support nodes $Y_j^i = \begin{cases} 0 & \text{for } j = 1, m_i = 1 \\ 2 \cdot \frac{j-1}{m_i - 1} - 1 & \text{for } j = 1, \ldots, m_i, m_i > 1 \end{cases}$ and

$m_i = \begin{cases} 1, & \text{for } i = 1 \\ 2^{i-1} + 1 & \text{for } i > 1 \end{cases}$.

Due to the fact that the support nodes are nested (e.g. $X^i \subset X^{i+1}$) the Sparse Grid can be rewritten in the following hierarchical form

$$S(k,d)(f) = S(k-1,d)(f) + \sum_{|\mathbf{i}|=k} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right)(f) \tag{5.11}$$

$$= S(k-1,d)(f) + \Delta S(k,d)(f) \tag{5.12}$$

with

$$\begin{aligned} \Delta S(k,d)(f) = \sum_{|\mathbf{i}|=k} \sum_{\mathbf{j} \in B_\mathbf{i}} \underbrace{\left(a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_d}^{i_d}\right)}_{a_\mathbf{j}^\mathbf{i}} \cdot \\ \cdot \underbrace{\left(f\left(Y_{j_1}^{i_1}, \ldots, Y_{j_d}^{i_d}\right) - S(k-1,d)(f)\left(Y_{j_1}^{i_1}, \ldots, Y_{j_d}^{i_d}\right)\right)}_{w_\mathbf{j}^\mathbf{i}} \end{aligned} \tag{5.13}$$

where $B_\mathbf{i} := \{\mathbf{j} \in \mathbb{N}^\mathbb{N} : Y_{j_l}^{i_l} \in X_\Delta^{i_k} \text{ for } j_l = 1, \ldots, m_\Delta^{i_l}, k = 1, \ldots, d\}$ is a new set of multi-indices consistent with the multivariate hierarchical basis $\{a_\mathbf{j}^\mathbf{i} : \mathbf{j} \in B_\mathbf{l}, \mathbf{l} \leq \mathbf{i}\}$, $X_\Delta^i = X^i \setminus X^{i-1}$, $m_i^\Delta = \# X_\Delta^i = m_i - m_{i-1}$.

Thus, the objective function in our application can be approximated by the following rather abstract expression:

$$f(p, \psi_d(\zeta)) = \sum_{|\mathbf{i}| \leq k} \sum_{\mathbf{j} \in B_\mathbf{i}} w_\mathbf{j}^\mathbf{i}(p) \cdot a_\mathbf{j}^\mathbf{i}(\zeta) \tag{5.14}$$

The mean value of the objective function can be then computed as:

$$\mathbb{E}_N\left(f\left(p\right)\right) = \sum_{|\mathbf{i}|\leq k} \sum_{\mathbf{j}\in B_{\mathbf{i}}} w_{\mathbf{j}}^{\mathbf{i}}\left(p\right) \cdot \int_{\Omega} a_{\mathbf{j}}^{\mathbf{i}}\left(\psi_d\left(\zeta\right)\right) dP\left(\zeta\right) \tag{5.15}$$

Instead of using the hierarchical surplus $w_{\mathbf{j}}^{\mathbf{i}}$ as an error indicator for the adaptivity (cf. [26], [31]), we suggest to adapt the grid checking the following expression:

$$\tilde{w}_{\mathbf{j}}^{\mathbf{i}} := w_{\mathbf{j}}^{\mathbf{i}} \cdot \int_{\Omega} a_{\mathbf{j}}^{\mathbf{i}}\left(\psi_d\left(\zeta\right)\right) dP\left(\zeta\right) \tag{5.16}$$

Since it is not necessary to exactly interpolate the drag depending on the uncertainty in the optimization loop in our application, the adaptivity indicator $\tilde{w}_{\mathbf{j}}^{\mathbf{i}}$ only measures the difference between the value of the mean inserting a new point $Y_{\mathbf{j}}^{\mathbf{i}}$ of the current level of interpolation and the corresponding value of the mean at the previous interpolation level. The underlying algorithm in order to construct the adaptive Sparse Grid which is then used for the optimization can be found in [26], [31].

**6. One-shot aerodynamic shape optimization and its coupling to robust design.** Novel one-shot aerodynamic shape optimization in the form (3.1-3.3) have been introduced in [3, 2] . They have the potential of fast convergence in only a small multiple of cpu-time compared to on flow simulation. These methods are based on approximate reduced SQP iterations in order to generate a stationary point satisfying the first order KKT optimality conditions.

In this context, a full SQP-approach reads as

$$\begin{bmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yp} & h_x^\top & c_x^\top \\ \mathcal{L}_{py} & \mathcal{L}_{pp} & h_p^\top & c_p^\top \\ h_x & h_p & 0 & 0 \\ c_x & c_p & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta y \\ \Delta p \\ \Delta \mu \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} -\mathcal{L}_y^\top \\ -\mathcal{L}_p^\top \\ -h \\ -c \end{pmatrix}, \quad \begin{pmatrix} y^{k+1} \\ p^{k+1} \\ \mu^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ p^k \\ \mu^k \\ \lambda^k \end{pmatrix} + \tau \cdot \begin{pmatrix} \Delta y \\ \Delta p \\ \Delta \mu \\ \Delta \lambda \end{pmatrix} \tag{6.1}$$

The symbol $\mathcal{L}$ denotes the Lagrangian function. We assume that the lift constraint $h$ is active at the solution, which is the reason that we formulate is rather as an equality condition in the single setpoint case. The approach (6.1) is not implementable in general, because one usually starts out with a flow solver for $c(y, p) = 0$ and seeks a modular coupling with an

optimization approach, which does not necessitate to change the whole code structure, as would be the case with formulation (6.1). A modular but nevertheless efficient alternative is an approximate reduced SQP approach as justified in [4] .

$$\begin{bmatrix} 0 & 0 & 0 & A^\top \\ 0 & B & \gamma & c_p^\top \\ 0 & \gamma^\top & 0 & 0 \\ A & c_p & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta y \\ \Delta p \\ \Delta \mu \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} -\mathcal{L}_y^\top \\ -\mathcal{L}_p^\top \\ -h \\ -c \end{pmatrix}, \quad \begin{pmatrix} y^{k+1} \\ p^{k+1} \\ \mu^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ p^k \\ \mu^k \\ \lambda^k \end{pmatrix} + \tau \cdot \begin{pmatrix} \Delta y \\ \Delta p \\ \Delta \mu \\ \Delta \lambda \end{pmatrix} \tag{6.2}$$

where

$$\gamma = h_p^\top + c_p^\top \alpha\,, \text{ such that } A^\top \alpha = -h_x^\top$$

The matrix $A$ denotes an appropriate approximation of the system matrix $c_x$, which is used in the iterative forward solver. An algorithmic version of this modular formulation is given by the following steps

(1) generate $\lambda^k$ by performing $N$ iterations of an adjoint solver with right hand side $f_y^\top(y^k, p^k)$ starting in $\lambda^k$

(2) generate $\alpha^k$ by performing $N$ iterations of an adjoint solver with right hand side $h_y^\top(y^k, p^k)$ starting in $\alpha^k$

(3) compute approximate reduced gradients

$$g = f_p^\top + c_p^\top \lambda^{k+1}\,, \quad \gamma = h_p^\top + c_p^\top \alpha^{k+1}$$

(4) generate $B_{k+1}$ as an approximation of the (consistent) reduced Hessian

(5) solve the QP

$$\begin{bmatrix} B & \gamma \\ \gamma^\top & 0 \end{bmatrix} \begin{pmatrix} \Delta p \\ \mu^{k+1} \end{pmatrix} = \begin{pmatrix} -g \\ -h \end{pmatrix}$$

(6) update $p^{k+1} = p^k + \Delta p$

(7) compute the corresponding geometry and adjust the computational mesh

(8) generate $y^{k+1}$ by performing $N$ iterations of the forward state solver starting from an interpolation of $y^k$ at the new mesh.

This highly modular algorithmic approach is not an exact transcription of equation (6.2), but is shown in [4] to be asymptotically equivalent and to converge to the same solution. The overall algorithmic effort for this algorithm is typically in the range of factor 7 to 10 compared to a forward stationary simulation.

Now we generalize this algorithmic framework to the semi-infinite problem formulation (3.4-3.6). Numerical approaches to this problem class have already been proposed in [8, 9].

For the sake of simplicity, we restricted the formulation to a problem with two set-points coupled via the objective, which is a weighted sum of all set-point objectives (weights: $\omega_1, \omega_2$), and via the free optimization variables $p$, which are the same for all set-points. The generalization to more setpoints (i.e., the adaptive Sparse Grid points below) is then obvious. The lift constraint is formulated for the smallest value $\psi^{\min}$ of all grid points. The corresponding Lagrangian in our example is

$$\mathcal{L}(y_1, y_2, p, \lambda_1, \lambda_2) = \sum_{i=1}^{2} \omega_i f_i(y_i, p, \psi^i) + \sum_{i=1}^{2} \lambda_i^\top c_i(y_i, p, \psi^i) + \mu h(y_{\min}, p, \psi^{\min}) \quad (6.3)$$

The approximate reduced SQP method above applied to this case can be written in the following form

$$\begin{bmatrix} 0 & 0 & 0 & 0 & A_1^\top & 0 \\ 0 & 0 & 0 & 0 & 0 & A_2^\top \\ 0 & 0 & B & \gamma_1 & c_{1,p}^\top & c_{2,p}^\top \\ 0 & 0 & \gamma_1 & 0 & 0 & 0 \\ A_1 & 0 & c_{1,p} & 0 & 0 & 0 \\ 0 & A_2 & c_{2,p} & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta y_1 \\ \Delta y_2 \\ \Delta p \\ \Delta \mu \\ \Delta \lambda_1 \\ \Delta \lambda_2 \end{pmatrix} = \begin{pmatrix} -\mathcal{L}_{y_1}^\top \\ -\mathcal{L}_{y_2}^\top \\ -\mathcal{L}_p^\top \\ -h \\ -c_1 \\ -c_2 \end{pmatrix} \quad (6.4)$$

We notice that the linear sub-problems involving matrices $A_i^\top$ are to be solved independently, and therefore trivially in parallel. The information from all these parallel adjoint problems is

collected in the reduced gradient

$$g = \sum_{i=1}^{2} \omega_i f_p^\top + \sum_{i=1}^{2} c_p^\top \lambda_i$$

Next, the solution of optimization step

$$\begin{bmatrix} B & \gamma_1 \\ \gamma_1^\top & 0 \end{bmatrix} \begin{pmatrix} \Delta p \\ \mu^{k+1} \end{pmatrix} = \begin{pmatrix} -g \\ -h \end{pmatrix}$$

is distributed to all approximate linearized forward problems

$$A_i \Delta y_i + c_{i,p} \Delta p = -c_i \,,$$

which can then again be performed in parallel.

**7. Numerical results.** We investigate the problem discussed in Ref.[3] , i.e. the shape optimization of a RAE2822 profile in transonic Euler flow, by the use of the CFD software FLOWer provided by DLR within a one-shot framework. The block-structured FLOWer code solves the three-dimensional compressible Reynolds-averaged Navier-Stokes equation in integral form and provides different turbulence models. The equations are solved by a finite-volume method with second order upwind or central space discretization. The discrete equations are integrated explicitly by multistage Runge-Kutta schemes, using local time stepping and multigrid acceleration. In our example, the space is discretized by a 133×33 grid, see Fig 7.1 and the profile is described by 129 grid points. For parametrization, the airfoil is decomposed into thickness and camber distribution. Then, only the camber of the airfoil is parametrized by 21 Hicks-Henne functions and the thickness is not changed during the optimization process.

The geometry uncertainties are characterized by a Gaussian random field and the follow-

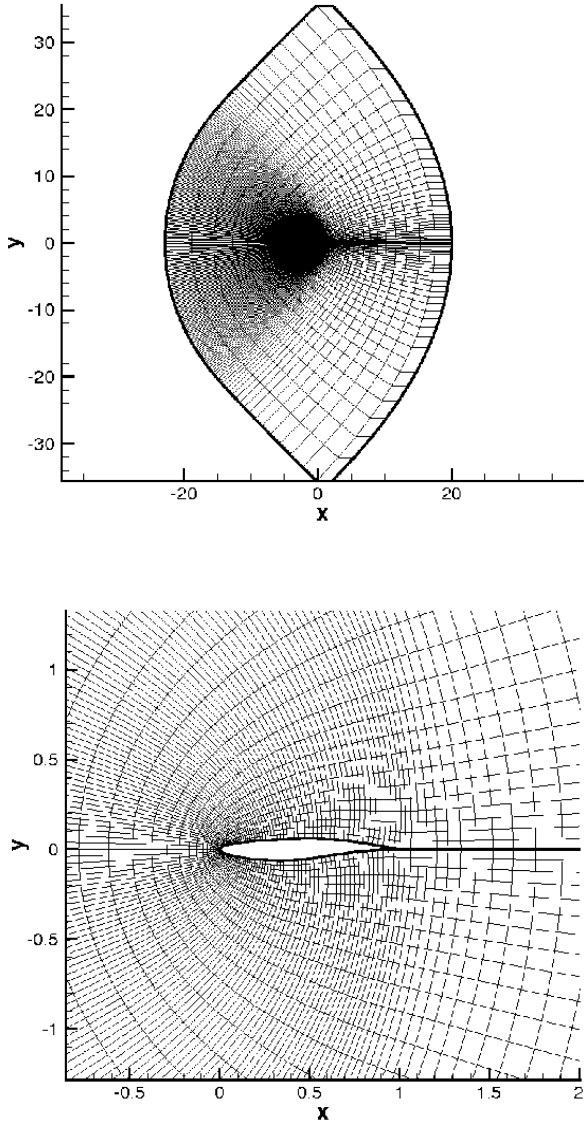


FIG. 7.1. *grid for the RAE2822 airfoil: the total geometrical plane (above) and zoom around the airfoil (below)*

ing second order statistics

$$\mathbb{E}\left(\psi\left(x,\omega\right)\right)=0 \qquad \forall x\in\Gamma \tag{7.1}$$

$$Cov\left(x,y\right)=\left(0.005\right)^{2}\cdot\exp\left(-\frac{\left\|x-y\right\|^{2}}{\left(0.1\right)^{2}}\right) \qquad \forall x,y\in\Gamma. \tag{7.2}$$

One realization of the random field and the resulting perturbed geometry is shown in Figure 2.1 and Figure 2.2. Representing the random field for the numerical treatment $\psi$ in a finite number of independent random variables using the Karhunen-Loève-Expansion, one has to solve the eigenvalue problem 2.6. In our two-dimensional testcase, the discretization of the profile leads to a matrix of size $(129 \times 129)$, so the eigenvalues and eigenvectors can be computed by common methods. The generalization to a three-dimensional application would lead to a much more complex task, hence more sophisticated solvers which can be found e.g. in [23] are required. The distribution of the eigenvalues of the given random field 7.2-7.2 is shown in the next figure 7.2. As stated before, the eigenvalues exponentially converge towards

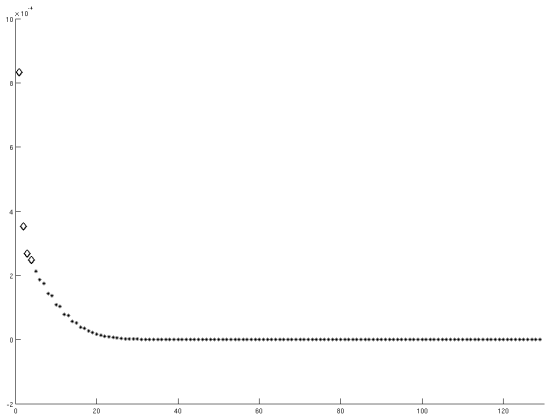


FIG. 7.2. *distribution of the eigenvalues of the given random field* $\psi$

0. For the first numerical results, we have considered only the first four eigenvalues and eigenvectors to represent the random field $\psi$ of perturbations. The corresponding eigenvectors are shown in Figure 7.3 Using the truncated Karhunen-Loève representation, the mean value of the drag is then computed by

$$\mathbb{E}\left(f(y,p,\psi_4\left(x,\omega\right))\right) = \int_{\mathbb{R}}\int_{\mathbb{R}}\int_{\mathbb{R}}\int_{\mathbb{R}}(f(y,p,\sum_{i=1}^{4}\sqrt{\lambda_i}z_i\left(x\right)Y_i\left(\omega_i\right))\mathrm{d}\gamma_1\left(\omega_1\right)\cdots\mathrm{d}\gamma_1\left(\omega_4\right) \tag{7.3}$$

The random variables $Y_i$ are uncorrelated and therefore independent, so one has to approximate a four-dimensional integral in the optimization problem. To further reduce the computational effort, we investigate the influence of the individual eigenvectors in order to reject

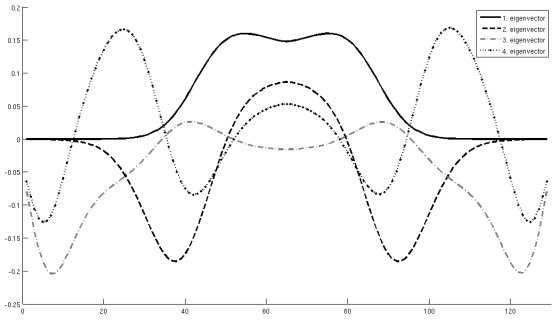


FIG. 7.3. *first four eigenvectors of the given random field* $\psi$

those eigenvectors from the reduced basis which have no impact on the target functional. As

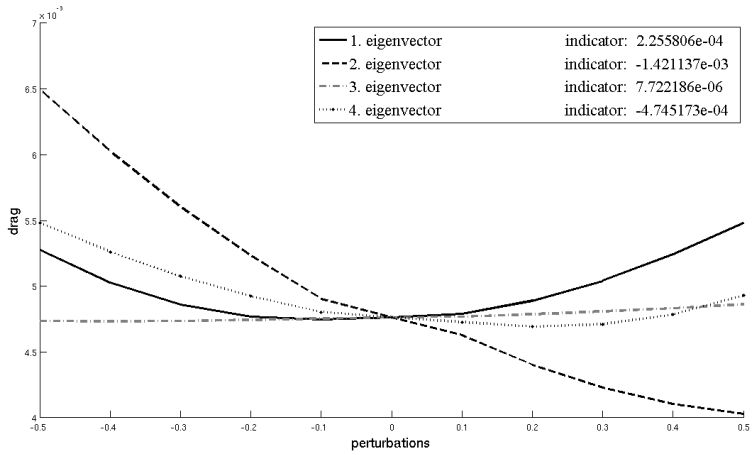


FIG. 7.4. *drag performance of the first four eigenvectors on the target functional*

Figure 7.4 shows, the third eigenvector has no impact on the objective function, hence it can be rejected from the Karhunen-Loève basis and the dimension of the integral is reduced. This behaviour is also reflected by the introduced indicator. Consequently, the mean value is given by

$$\begin{aligned} &\mathbb{E}\left(f(y,p,\psi_4^{reduced}(x,\omega))\right) = \\ &= \int_{\mathbb{R}}\int_{\mathbb{R}}\int_{\mathbb{R}}(f(y,p,\sum_{\substack{i=1\\ i\neq 3}}^{4}\sqrt{\lambda_i}z_i(x)\,Y_i(\omega_i))\mathrm{d}\gamma_1(\omega_1)\,\mathrm{d}\gamma_1(\omega_2)\,\mathrm{d}\gamma_1(\omega_4) \end{aligned} \tag{7.4}$$

If one approximate the expected value 7.4 using a full tensor grid interpolation 5.3, 729 grid

points will be needed to reach the error tolerance of $10^{-4}$. The resulting full grid is shown in Figure 7.5. The Sparse Grid method can reduce the computational effort by a factor of 10

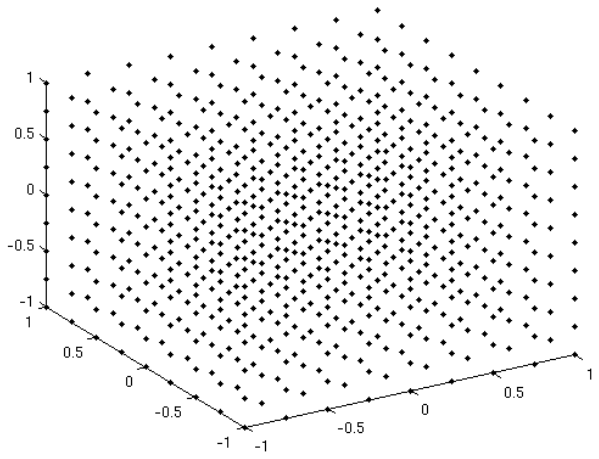


FIG. 7.5. *full Tensor grid with 729 grid points*

maintaining the same approximation quality. The corresponding grid is depicted in Figure 7.6. Since we want to compare the efficiency of the different introduced methods, we have

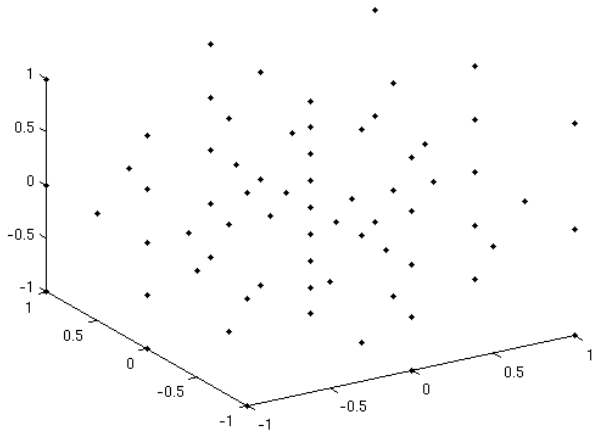


FIG. 7.6. *Sparse Grid with 69 grid points*

chosen multilinear hierarchical basis function as ansatzfunctions for all grids. Last, an adaptive Sparse Grid was constructed in order to compute the objective function. As Figure 7.7 shows, the number of grid points can again be reduced from 69 grid points to 35 grid points,

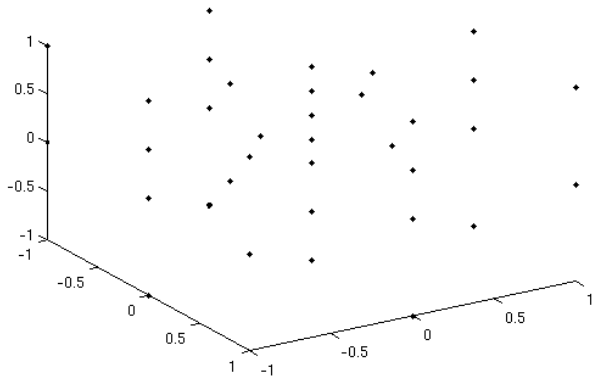


FIG. 7.7. *adaptive Sparse Grid with 35 grid points*

i.e. almost by the factor of 2. Since the optimization requires the evaluation of the mean value in each iteration, this reduction by factor 20 compared with the full grid takes place in each step of the optimization algorithm and hence significantly speed up the whole algorithm. The construction of the adaptive Sparse Grid although needs some additional function evaluations in order to compute the adaptivity indicator, but this amount of computational effort occurs outside the optimization loop, i.e. these costs are negligible.

Now, the next two Figures 7.8, 7.9 compare the results of the robust optimization and of the single setpoint optimization, i.e. without considering any uncertainties in the optimization. The drag and lift performance is plotted against the 35 perturbed geometries and the dashed line in Fig. 7.8 indicates the mean value of the drag. The line in-between the points of the single setpoint optimization in grey and the robust optimization in black points out the better function value of the two solutions. The robust optimization improves the mean value of the target functional and also shows at the same time to a better lift performance over the whole range of perturbations, whereas the single setpoint optimization is infeasible in more than the half of the considered grid points. Summing it all up, it can be said that the robust optimization leads to a better lift to drag ratio than the single setpoint optimization and the resulting profile is more robust against small perturbations of the shape itself. Last, we will compare the different resulting shape in Figure 7.10. Although we have assumed only small perturbations of the shape itself (cf. Fig. 2.2), the robust shape differs quite strongly from the

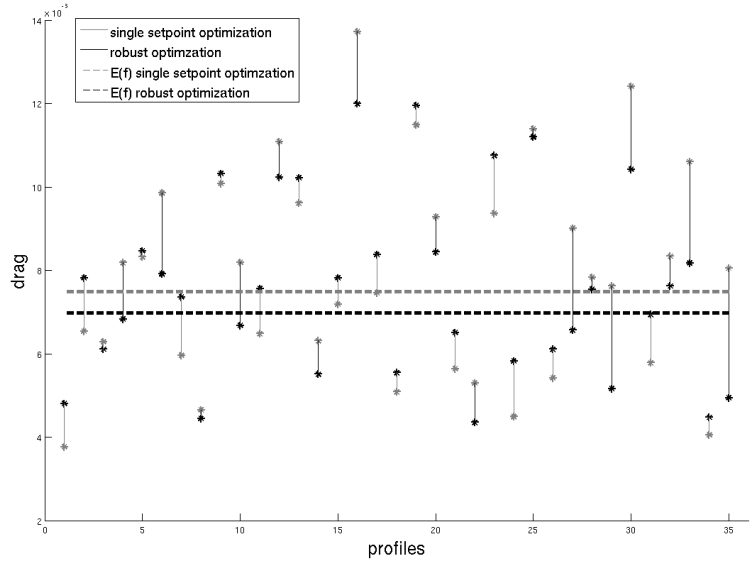


FIG. 7.8. *drag performance of the 35 perturbed geometries*

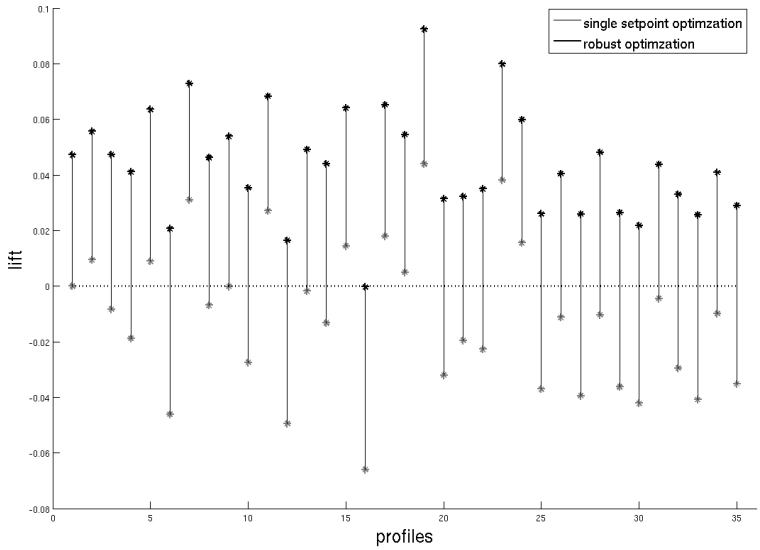


FIG. 7.9. *lift performance of the 35 perturbed geometries*

single setpoint optimization. So, this observation shows the importance of robust design in the aerodynamical framework to ensure a good performance even under uncertain conditions.

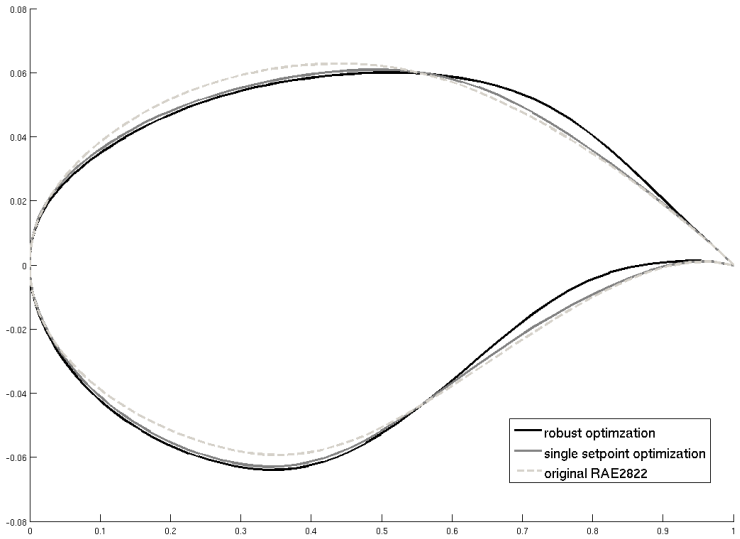


FIG. 7.10. *comparison of the optimized shapes*

**8. Conclusions.** Robust design is an important task to make aerodynamic shape optimization relevant for practical use. It is also highly challenging because the resulting optimization tasks become much more complex than in the usual single set-point case. Espespially in the case of the geometric uncertainty, the approximation of the random field describing the perturbations of the geometry leads to a very high dimensional optimization task. The dimension of the probability space was efficiently reduced by a goal-oriented choice of the Karhunen-Loève basis. Furthermore, adaptive Sparse Grid techniques and one-shot methods have been succesfully generalizied to the semi-infinite formulation of the shape optimization problem in order to reduce the amount of computational effort in the resulting robust optimization. The numerical results show that even small deviations from the planned geometry have a significant effect on the drag and lift coefficient, so that geometry uncertainties have to be taken into account in the aerodynamic design optimization problem to ensure a robust solution. The introduced methods can significantly reduce the costs of the robust optimization, so that robust design becomes numerically tractable in the aerodynamical framework.

**Acknowledgements.** This research has been supported by BMWA within the collaborative effort MUNA.

REFERENCES


[1] V. Schulz and I. Gherman One-Shot Methods for Aerodynamic Shape Optimization In N. Kroll, D. Schwamborn, K. Becker, H. Rieger and F. Thiele (eds.) *MEGADESIGN and MegaOpt German Initiatives for Aerodynamic Simulation and Aircraft Design*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 107, Springer, 2009

[2] S.B. Hazra and V. Schulz. Simultaneous pseudo-timestepping for aerodynamic shape optimization problems with state constraints. *SIAM Journal on Scientific Computing*, Vol. 28, No. 3, 2006, pp. 1078-1099.

[3] S.B. Hazra, V. Schulz, J. Brezillon, and N. Gauger. Aerodynamic shape optimization using simultaneous pseudo-timestepping. *Journal of Computational Physics*, Vol. 204, No. 1, 2005, pp. 46-64.

[4] I. Gherman. *Approximate Partially Reduced SQP Approaches for Aerodynamic Shape Optimization Problems*. PhD thesis, University of Trier, 2007.

[5] M. Putko, P. Newman, A. Taylor, and L. Green Approach for uncertainty propagation and robust design in CFD using sensitivity derivatives AIAA 2001-2528, 2001.

[6] M. Putko, P. Newman, and A. Taylor Employing Sensitivity Derivatives for Robust Optimization under Uncertainty in CFD 9th ASCE Special Conference on Probabilistic Mechanics and Structural Reliability, 2004.

[7] V. Schulz, C. Schillings On the nature and treatment of uncertainties in aerodynamic design *AIAA Journal*, Vol. 47, No. 3, 2009, pp. 646-654

[8] H.G. Bock, W. Egartner, W. Kappis, and V. Schulz. Practical shape optimization for turbine and compressor blades. *Optimization and Engineering*, Vol. 3, 2002, pp. 395-414

[9] H. G. Bock, E. Kostina, A. Schäfer, J. P. Schlöder, and V. Schulz. Multiple set point partially reduced SQP method for optimal control of PDE. In W. Jäger, R. Rannacher, and J. Warnatz, editors, *Reactive Flows, Diffusion and Transport*. Springer, Berlin, 2007, pp. 147-176

[10] R. Becker, M. Braack, D. Meidner, R. Rannacher, and B. Vexler Adaptive finite element methods for PDE-constrained optimal control problems *Reactive Flows, Diffusion and Transport, Rolf Rannacher et. al. (eds.)*, Springer Verlag, Berlin, pp. 177-205, 2006

[11] R. Becker, H. Kapp, R. Rannacher Adaptive finite element methods for optimization problems *Proc. Conf. Numer. Analysis 1999 (D.F. Griffiths and G.A. Watson, eds.)*, London, pp. 21-42, 2000.

[12] H.J. Bungartz Finite Elements of Higher Order on Sparse Grids *Shaker Verlag*, Aachen, Germany, 1998

[13] A. Borzì, G. von Winckel Multigrid methods and sparse-grid collocation techniques for parabolic optimal control problems with random coefficients *SIAM Journal on Scientific Computing* Vol. 31, No. 3, 2009, pp. 2172-2192

[14] M. Putko, P. Newman, A. Taylor, and L. Green. Approach for uncertainty propagation and robust design in CFD using sensitivity derivatives. In *Proceedings of the 15 th AIAA Computational Fluid Dynamics Conference; June 11-14 2001, Anaheim CA*, 2001. AIAA Paper 2001-2528.

[15] P. Davis, P. Rabinowitz. Methods of numerical integration *Academic Press*, Second Edition, Orlando, Florida, 1984, pp. 11-15 and 188-190

[16] M.S. Eldred, C.G. Webster, P. Constantine Evaluation of Non-Intrusive Approaches for Wiener-Askey Generalized Polynomial Chaos *AIAA Proceedings of the 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (10th AIAA Non-Deterministic Approaches Conference)*, 2008-1892.

[17] J. Garcke, M. Griebel Classification with sparse grids using simplicial basis functions *Intelligent Data Analysis*, Vol. 6, No. 6, 2002, pp.483-502.

[18] T. Gerstner, M. Griebel Dimension-adaptive tensor-product quadrature *Computing*, Vol. 71, No. 1, 2003, pp.65-87.

[19] R. Ghanem, P.D. Spanos Stochastic Finite Element: A Spectral Approach *Springer*, New York, 1991.

[20] S.P. Huang, S.T. Quek, K.K. Phoon Convergence study of the truncated Karhunen-Loeve expansion for simulation of stochastic processes *International Journal for Numerical Methods in Engineering*, Vol. 52, No. 9, 2001, pp. 1029-1043

[21] L. Huyse, S.L. Padula, R.M. Lewis, et. al Probabilistic approach to free-form airfoil shape optimization under uncertainty. *AIAA Journal*, Vol. 40, No. 9, 2002, pp. 1764-1772.

[22] W. Li, L. Huyse, S. Padula Robust airfoil optimization to achieve drag reduction over a range of Mach numbers. *Structural and Multidisciplinary Optimization*, Vol. 24, No. 1, 2002, pp. 38-50.

[23] B.N. Khoromskij, A. Litvinenko, H.G. Matthies Application of hierarchical matrices for computing the KarhunenLove expansion *Computing*, Vol. 84, No 1-2, 2009, pp. 49-67

[24] L. Wu, S. Padula Using high resolution design spaces for aerodynamic shape optimization under uncertainty. *NASA Technical Publication*, 213003, 2004.

[25] K. Karhunen Zur spektraltheorie stochastischer Prozesse *Annales Academiae Scientarum Fennicae*, 34, 1946.

[26] A. Klimke, B. Wohlmuth Algorithm 847: Spinterp: piecewise multilinear hierarchical sparse grid interpolation in MATLAB *ACM Trans. Math. Softw.*, Vol. 31, No. 4, 2005, pp. 561-579

[27] M. Loève Probability Theory *Van Nostrand*, Princeton, New Jersey, 1955

[28] S.L. Padula, C.R Gumbert, W. Li Aerospace applications of optimization under uncertainty. *Optimization and Engineering*, Vol. 7, No. 3, 2006, pp. 317-328.

[29] P. Frauenfelder, C. Schwab, R.A. Todor Finite elements for elliptic problems with stochastic coefficients *Computer Methods in Applied Mechanics and Engineering, Selected papers from the 11th Conference on The Mathematics of Finite Elements and Applications*, Vol. 194, No. 2-5 4 February 2005, pp. 205-228

[30] S.A. Smolyak Quadrature and interpolation formulas for tensor products of certain classes of functions *Soviet Mathematics Doklady*, Vol. 4, 1963, pp. 240-243

[31] X. Ma, N. Zabaras An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential *Journal of Computational Physics*, Vol. 228, No. 8, 2009, pp. 3084-3113