Numerical Methods for Uncertainty Quantification and Bayesian Update in Aerodynamics

Alexander Litvinenko and Hermann G. Matthies

Abstract. In this work we research the propagation of uncertainties in parameters and airfoil geometry to the solution. Typical examples of uncertain parameters are the angle of attack and the Mach number. The discretisation techniques which we used here are the Karhunen-Loève and the polynomial chaos expansions. To integrate high-dimensional integrals in probabilistic space we used Monte Carlo simulations and collocation methods on sparse grids. To reduce storage requirement and computing time, we demonstrate an algorithm for data compression, based on a low-rank approximation of realisations of random fields. This low-rank approximation allows us an efficient postprocessing (e.g. computation of the mean value, variance, etc) with a linear complexity and with drastically reduced memory requirements. Finally, we demonstrate how to compute the Bayesian update for updating a priori probability density function of uncertain parameters. The Bayesian update is also used for incorporation of measurements into the model.

1 Introduction

Nowadays, the trend of numerical mathematics is often trying to resolve inexact mathematical models by very exact deterministic numerical methods. The reason of this inexactness is that almost each mathematical model of a real world situation contains uncertainties in the coefficients, right-hand side, boundary conditions, initial data as well as in the computational geometry. All these uncertainties can affect the solution dramatically, which is, in its turn, also uncertain. The information of the interest is usually not the whole set of realisations of the solutions (too much data), but some other stochastic information: cumulative distribution function, probability density function, mean value, variance, quantiles, exceedance probability etc.

Alexander Litvinenko · Hermann G. Matthies
Institute of Scientific Computing, Technische Universität Braunschweig,
Hans-Sommer str. 65, 38106, Braunschweig, Germany
e-mail: wire@tu-bs.de
http://www.wire.tu-bs.de

B. Eisfeld et al. (Eds.): *Management & Minimisation of Uncert. & Errors*, NNFM 122, pp. 265–282.
DOI: 10.1007/978-3-642-36185-2_11 © Springer-Verlag Berlin Heidelberg 2013

During the last few years, one can see an increasing interest in numerical methods for solving stochastic computational fluid dynamic (CFD) problems [3, 8, 17, 19, 24, 26]. In this work we consider an example from aerodynamics, described by a system of Navier-Stokes equations with a k-w turbulence model. Uncertainties in parameters such as the angle of attack α and Mach number are modelled by random variables, uncertainties in the shape of the airfoil are modelled by a random field [14, 12]. Uncertain output fields such as pressure, density, velocity, turbulence kinetic energy are modelled by random fields as well. The lift, drag and moments will be random variables.

We assume that there is a solver which is able to solve the deterministic (without uncertainties) Navier-Stokes problem. In this work we used the TAU code (developed in DLR) with k-w turbulence model [4]. We also assume that spatial discretisation of the airfoil is given. Our job is the appropriate modelling of uncertainties and developing stochastic/statistical numerical techniques for further quantification of uncertainties. At the same time, due to the high complexity of the deterministic solver, we are allowed to use only non-intrusive stochastic methods such as Monte Carlo or collocation methods. So, we are interested in methods which do not require changes in the deterministic code.

The rest of the paper is structured as follows. In Section 3 we describe the problem and discretisation techniques, such as the Karhunen-Loève expansion (KLE) [16] and polynomial chaos expansion (PCE) of Wiener [25]. In Section 2.1 we explain how we model uncertainties in the parameters angle of attack and Mach number. Uncertainty in the airfoil geometry is described in Section 2.2. The low-rank response surface is presented in Section 4. To avoid large memory requirements and to reduce computing time, low-rank techniques for representation of input and output data (solution) were developed in Section 5. Section 7 is devoted to the numerical results, where we demonstrate the influence of uncertainties in the angle of attack α, in the Mach number Ma and in the airfoil geometry on the solution - drag, lift, pressure and absolute friction coefficients. The strongly reduced memory requirement for storage stochastic realisations of the solution is demonstrated as well. In Section 6 we demonstrate how to use the Bayesian update (BU) for improving the statistical description of the random airfoil geometry. Section 7 is devoted to other numerical experiments.

2 Statistical Modelling of Uncertainties

The problem to consider in this work is the stationary system of Navier-Stokes equations with uncertain coefficients and parameters:

$$\begin{aligned} &v(x,\omega)\cdot\nabla v(x,\omega) - \tfrac{1}{Re}\nabla^2 v(x,\omega) + \nabla p(x,\omega) = g(x)\ x\in\mathscr{G},\ \omega\in\Omega \\ &\nabla\cdot v(x,\omega) = 0 \end{aligned} \tag{1}$$

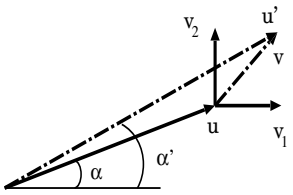

Fig. 1 Two random vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ model free-stream turbulence, $\mathbf{u}$ and $\mathbf{u}'$ old and new free stream velocities, α and α' old and new angles of attack

with some initial and boundary conditions. Here v is velocity, p pressure and g the right-hand side, the computational domain $\mathscr{G}$ is RAE-2822 airfoil with some area around. Examples of uncertain parameters are the angle of attack α and the Mach number Ma. Uncertainty in the airfoil geometry is modelled via random field (see Section 7).

2.1 Modelling of Uncertainties in Parameters

We assume that the free-stream turbulence in the atmosphere randomly and simultaneously changes the velocity vector or, what is equivalent the Mach number Eq. 4 and the angle of attack Eq. 3. One should not mix this kind of turbulence with the turbulence in the boundary layer reasoned by friction. It is assumed that turbulence vortices in the atmosphere are comparable with the size of the airplane. The free-stream turbulence in the atmosphere is modelled by two additionally axes-parallel velocity vectors $\mathbf{v}_1 := \mathbf{v}_1(\theta_1)$ and $\mathbf{v}_2 := \mathbf{v}_2(\theta_2)$ (Fig. 1), which have Gaussian distribution [13]. We model the free-stream turbulence via two random vectors (in 3D it will be three vectors) $\mathbf{v}_1$ and $\mathbf{v}_2$ which change α and Ma (see Fig. 1):

$$\mathbf{v}_1 = \frac{\sigma\theta_1}{\sqrt{2}} \quad \text{and} \quad \mathbf{v}_2 = \frac{\sigma\theta_2}{\sqrt{2}},$$

where θ_1 and θ_2 are two Gaussian random variables with zero mean and unit variance, $\sigma := \mathtt{I}u_\infty$, $\mathtt{I}$ the mean turbulence intensity and u_∞ the undisturbed free stream velocity beyond the boundary layer. This mean turbulence intensity is often used for characterising turbulence in a wind tunnel. For instance, $\mathtt{I} = 0.001$ means low turbulence, $\mathtt{I} = 0.002$ middle and $\mathtt{I} = 0.005$ high turbulence.
Denoting

$$\theta := \sqrt{\theta_1^2 + \theta_2^2},\ \mathbf{v} := \sqrt{\mathbf{v}_1^2 + \mathbf{v}_2^2},\ \beta := \operatorname{arctg}\frac{\mathbf{v}_2}{\mathbf{v}_1} \text{ and } z := \frac{\mathtt{I}\theta}{\sqrt{2}}, \tag{2}$$

and performing easy geometrical computations, obtain the new angle of attack and the new Mach number:

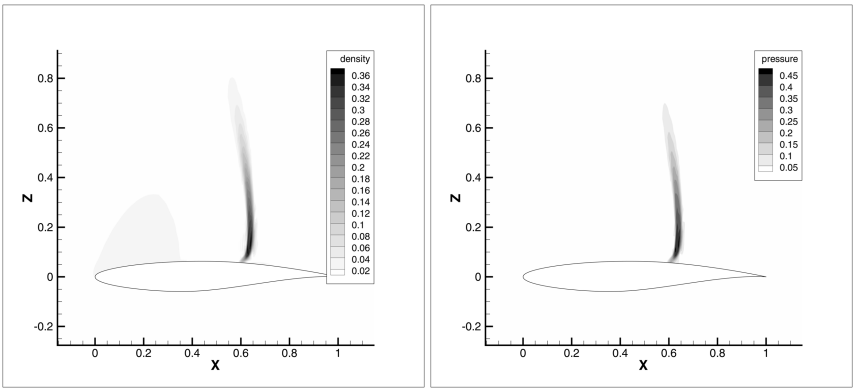

Fig. 2 (left) The difference $\Delta\rho := |\rho - \overline{\rho}|$ between the deterministic density $\rho := \rho(\alpha, Ma)$ and the mean density $\overline{\rho}$. (right) The same is for the pressure $\Delta p := |p - \overline{p}|$. Here $\overline{\rho} \in (0.5, 1.2)$ and $\overline{p} \in (0.7, 1.3)$.

$$\alpha^{'}(\theta_1, \theta_2) = \text{arctg}\frac{\sin\alpha + z\sin\beta}{\cos\alpha - z\cos\beta}, \tag{3}$$

$$Ma^{'}(\theta_1, \theta_2) = Ma\sqrt{1 + \frac{\mathrm{I}^2\theta^2}{2} - \sqrt{2}\mathrm{I}\theta\cos(\beta + \alpha)}. \tag{4}$$

Further we study how uncertainties in α and Ma spread into the solution. We note that uncertainties in α and in Ma can be modelled in a different way (see e.g. [23], [27]). >From the construction one can see that $\overline{Ma^{'}} := \mathbb{E}(Ma^{'}(\theta_1, \theta_2))$ and $\overline{\alpha^{'}} := \mathbb{E}(\alpha^{'}(\theta_1, \theta_2))$ are equal to the deterministic values Ma and α, here $\mathbb{E}(\cdot)$ is the mathematical expectation. In Fig. 2 (left) we compare the deterministic density $\rho(\alpha, Ma)$ with the $\overline{\rho} := \mathbb{E}(\rho(\alpha^{'}(\theta_1, \theta_2), Ma^{'}(\theta_1, \theta_2)))$ for the Case 9 ($\alpha = 2.79$, $Ma := 0.734$). In Fig. 2 (right) we do the same comparison for the deterministic pressure. One can see a large difference in the shock position. This large difference motivates us to model uncertainty in α and in Ma.

2.2 Modelling of Uncertainties in the Airfoil Geometry

We model uncertainties in the geometry of RAE-2822 airfoil via random boundary perturbations:

$$\partial\mathscr{G}_\varepsilon(\omega) = \{x + \varepsilon\kappa(x, \omega)n(x) : x \in \partial\mathscr{G}\}, \tag{5}$$

where $n(x)$ is the normal vector in a point x, $\kappa(x, \omega)$ a random field, $\mathscr{G}$ the computational geometry and $\varepsilon \ll 1$. We assume that the covariance function is of Gaussian type

$$\text{cov}_\kappa(\kappa_1, \kappa_2) = \sigma^2 \cdot \exp(-d^2), \quad d = \sqrt{|x_1 - x_2|^2/l_1^2 + |z_1 - z_2|^2/l_2^2},$$

Table 1 Statistics obtained for uncertainties in the airfoil geometry. We used the Gaussian covariance function, PCE of order $P = 1$ with $M = 3$ random variables and the sparse Gauss-Hermite grid with $n_q = 25$ points.

	mean	st. dev. σ	σ/mean
CL	0.8552	0.0049	0.0058
CD	0.0183	0.00012	0.0065

where $\kappa_1 = \kappa((x_1,0,z_1),\omega)$, $\kappa_2 = ((x_2,0,z_2),\omega)$ are two random variables in points $(x_1,0,z_1)$ and $(x_2,0,z_2)$. For numerical simulations we take the covariance lengths $l_1 = |\max_i(x_i) - \min_i(x_i)|/10$ and $l_2 = |\max_i(z_i) - \min_i(z_i)|/10$, standard deviation $\sigma = 10^{-3}$, $m = 3$ the number of KLE terms (see Eq. 6), the stochastic dimension $M = 3$ and the number of sparse Gauss-Hermite points (in 3D) for computing PCE coefficients in (Eq. 8) $n_q = 25$. In [13] one can see 21 random realisations of RAE-2822 airfoil.

Table 1 demonstrates the surprisingly small uncertainties (the last column) in the lift and in the drag — 0.58% and 0.65% correspondingly. A possible explanation can be small uncertain perturbations in the airfoil geometry.

3 Discretisation Techniques

In the following, $(\Omega,\mathscr{B},\mathbb{P})$ denotes a probability space, where Ω is the set of elementary events, $\mathscr{B}$ is the σ-algebra of events and $\mathbb{P}$ is the probability measure. The symbol ω always specifies an elementary event $\omega \in \Omega$.

The random field $\kappa(x,\omega)$ needs to be discretised both in the stochastic and in the spatial dimensions. One of the main tools here is the Karhunen-Loève expansion (KLE) [16]. By definition, KLE of a random field $\kappa(x,\omega)$ is the following series [16]

$$\kappa(x,\omega) = \overline{\kappa}(x) + \sum_{\ell=1}^{\infty} \sqrt{\lambda_\ell}\phi_\ell(x)\xi_\ell(\omega), \tag{6}$$

where $\xi_\ell(\omega)$ are uncorrelated random variables and $\overline{\kappa}(x)$ is the mean value of $\kappa(x,\omega)$, λ_ℓ and ϕ_ℓ are the eigenvalues and the eigenvectors of problem

$$T\phi_\ell = \lambda_\ell\phi_\ell, \quad \phi_\ell \in L^2(\mathscr{G}), \ell \in \mathbb{N}, \tag{7}$$

and operator T is defined like follows

$$T : L^2(\mathscr{G}) \to L^2(\mathscr{G}), \quad (T\phi)(x) := \int_{\mathscr{G}} \mathrm{cov}_\kappa(x,y)\phi(y)\mathrm{d}y,$$

where cov_κ is a given covariance function. Throwing away all unimportant terms in KLE, one obtains the truncated KLE, which is a sparse representation of the random field $\kappa(x,\omega)$. Each random variable ξ_ℓ can be approximated in a set of

new independent Gaussian random variables (polynomial chaos expansions (PCE) of Wiener [7, 25]), e.g.

$$\xi_\ell(\omega) = \sum_{\beta \in \mathscr{J}} \xi_\ell^{(\beta)} H_\beta(\theta(\omega)),$$

where $\theta(\omega) = (\theta_1(\omega), \theta_2(\omega), ...)$, $\xi_\ell^{(\beta)}$ are coefficients, H_β are multivariate Hermite polynomials, $\beta \in \mathscr{J}$ is a multiindex, $\mathscr{J} := \{\beta | \beta = (\beta_1, ..., \beta_j, ...), \beta_j \in \mathbb{N}_0\}$ is a multi-index set [18].

For the purpose of actual computation, truncate the polynomial chaos expansion after finitely many terms, e.g.

$$\beta \in \mathscr{J}_{M,P} := \{\beta \in \mathscr{J} \mid \gamma(\beta) \le M, \quad |\beta| \le P\}, \quad \gamma(\beta) := \max\{j \in \mathbb{N} \,|\, \beta_j > 0\}.$$

Since Hermite polynomials are orthogonal, the coefficients $\xi_\ell^{(\beta)}$ can be computed by projection

$$\xi_\ell^{(\beta)} = \frac{1}{\beta!} \int_\Theta H_\beta(\theta) \xi_\ell(\theta) \, \mathbb{P}(\mathrm{d}\theta).$$

This multidimensional integral over Θ can be computed approximately, for example, on a sparse Gauss-Hermite grid with n_q grid points

$$\xi_\ell^{(\beta)} \approx \frac{1}{\beta!} \sum_{i=1}^{n_q} H_\beta(\theta_i) \xi_\ell(\theta_i) w_i, \tag{8}$$

where weights w_i and points θ_i are defined from sparse Gauss-Hermite integration rule. After a finite element discretisation (see [10] for more details) the discrete eigenvalue problem (7) looks like

$$\mathbf{MCM}\phi_\ell = \lambda_\ell^h \mathbf{M}\phi_\ell, \quad \mathbf{C}_{ij} = \mathrm{cov}_\kappa(x_i, y_j). \tag{9}$$

Here the mass matrix $\mathbf{M}$ is stored in a usual data sparse format and the dense matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ (requires $\mathscr{O}(n^2)$ units of memory) is approximated in the sparse $\mathscr{H}$-matrix format [10] (requires only $\mathscr{O}(n \log n)$ units of memory) or in the Kronecker low-rank tensor format [9]. To compute m eigenvalues ($m \ll n$) and corresponding eigenvectors we apply the Lanczos eigenvalue solver [11, 22].

4 Low-Rank Response Surface

To compute statistics of the random (uncertain) solution (error-bars, quantiles, cumulative density function, etc) accurate enough, one needs a large sample size. Monte Carlo simulations are expensive. To decrease the computational costs we compute a, so-called, response surface — (multivariate) polynomial (see Eq. 10). The idea [13] is to construct a good response surface from few samples and then

to use the residual for its improvement. A motivation for this idea comes from the fact that in many software packages for solving engineering and physical problems it is impossible or very difficult to change the code, but it is possible to access the residual. Later on the computed response surface is used for very fast generation of a large sample.

Let $\mathbf{v}(x,\theta)$ be the solution (or a functional of the solution). It can be pressure, density, velocity, lift, drag etc. $\mathbf{v}(x,\theta)$ can be approximated in a set of new independent Gaussian random variables (truncated polynomial chaos expansions of Wiener [25])

$$\mathbf{v}(x,\theta(\omega)) \approx \sum_{\beta\in\mathscr{J}_{M,P}} \mathbf{v}_\beta(x) H_\beta(\theta) = [...\mathbf{v}_\beta(x)...][...H_\beta(\theta)...]^T, \tag{10}$$

where coefficients $\mathbf{v}_\beta(x)$ are computed as follows

$$\mathbf{v}_\beta(x) = \frac{1}{\beta!}\int_\Theta H_\beta(\theta)\mathbf{v}(x,\theta)\,\mathbb{P}(d\theta) \approx \frac{1}{\beta!}\sum_{i=1}^{n_q} H_\beta(\theta_i)\mathbf{v}(x,\theta_i)w_i, \tag{11}$$

The PCE representation in Eq. 10 was used to compute the mean and the variance of the pressure (see Fig. 3) for the Case 1 ($\alpha = 1.93$ and $\mathrm{Ma} = 0.676$, no shock). PCE coefficients are computed by the sparse Gauss Hermite grid with $n_q = 281$ nodes. Here the multidimensional integral over Θ is computed approximately, for example, on a sparse Gauss-Hermite grid [6, 2].

Fig. 4 demonstrates the mean of density and mean of pressure, computed again as in Eq. 10 for the Case 9 ($\alpha = 2.79$ and $\mathrm{Ma} = 0.734$, with shock).

Fig. 5 demonstrates the variance of density and variance of pressure, computed via Monte Carlo methods for the Case 9. One can see the largest uncertainty in the shock position.

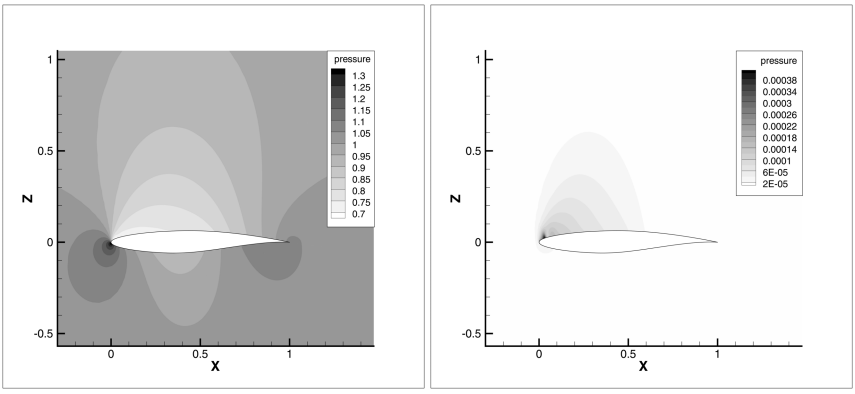

Fig. 3 (left) The mean pressure in Case 1; (right) The variance of the pressure in Case 1. Both are computed by the sparse Gauss Hermite grid with $n_q = 281$ nodes.

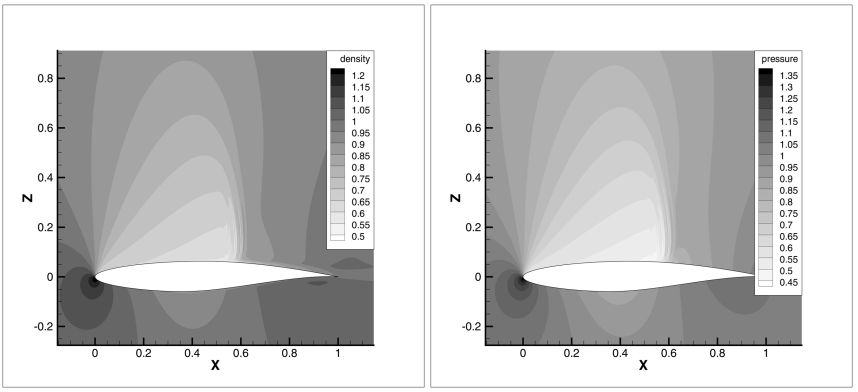

Fig. 4 (left) The mean density; (right) the mean pressure computed by PCE in the Case 9

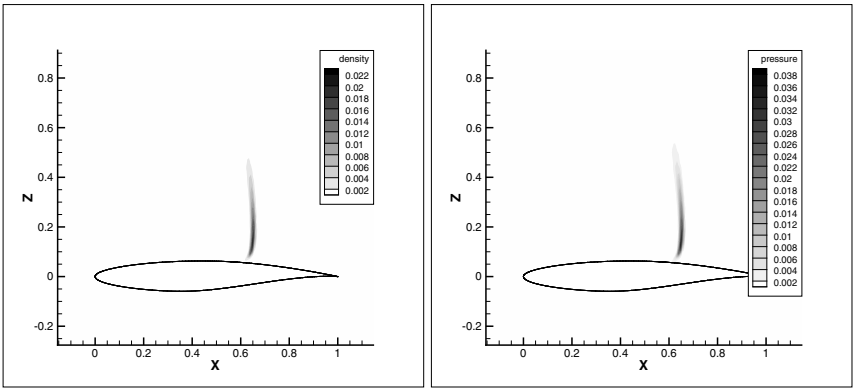

Fig. 5 (left) The variance of the density; (right) The variance of the pressure computed by MC in the Case 9.

Using the rank-k approximation of $[\mathbf{v}(x,\theta_1),...,\mathbf{v}(x,\theta_{n_q})]$, obtain

$$\mathbf{v}_\beta(x) = \frac{1}{\beta!}[\mathbf{v}(x,\theta_1),...,\mathbf{v}(x,\theta_{n_q})]\cdot[H_\beta(\theta_1)w_1,...,H_\beta(\theta_{n_q})w_{n_q}]^T \approx \mathbf{A}\mathbf{B}^T\mathbf{c}_\beta, \quad (12)$$

where $\mathbf{A}\in\mathbb{R}^{n\times k}$, $\mathbf{B}\in\mathbb{R}^{n_q\times k}$, $k\ll\min\{n,n_q\}$ and
vector $\mathbf{c}_\beta := \frac{1}{\beta!}[H_\beta(\theta_1)w_1,...,H_\beta(\theta_{n_q})w_{n_q}]^T$. The matrix of all PCE coefficients will be

$$[...\mathbf{v}_\beta(x)...] = \mathbf{A}\mathbf{B}^T[...\mathbf{c}_\beta...], \quad \beta\in\mathscr{J}_{M,P}. \quad (13)$$

Taking Eq. 10 and Eq. 13, obtain the final formula for the **low-rank response surface**

$$\mathbf{v}(x,\theta(\omega)) \approx \mathbf{A}\mathbf{B}^T[...\mathbf{c}_\beta...][...H_\beta(\theta)...]^T. \quad (14)$$

4.1 Update of the Low-Rank Response Surface via Computing the Residual

In real-world applications, the deterministic solver is very complicated and it is difficult or even impossible to change it, but one can often print out the norm of the residual. Assume that we already approximated the unknown solution by a response surface. Our response surface is approximation via multivariate Hermite polynomials like in Eq. 14, where coefficients are computed like in Eq. 13 with quadrature points θ_i, $i = 1..n_q$. The following algorithm updates the given response surface.

Algorithm: (**Update of the response surface**)

1. Take the next point θ_{n_q+1} and evaluate the response surface Eq. 14 in this point. Let $u(x, \theta_{n_q+1})$ be the obtained predicted solution.
2. Compute the norm of the residual $\|\mathbf{r}\|$ of the deterministic problem (e.g. evaluate one iteration). If $\|\mathbf{r}\|$ is small then there is no need to solve the expensive deterministic problem in θ_{n_q+1}, otherwise (if $\|\mathbf{r}\|$ is large) solve the deterministic problem and recompute $\mathbf{A}$, $\mathbf{B}^T$ and $\mathbf{c}_\beta$ in Eq. 12.
3. Go to item (1).

In the best case we never solve the deterministic problem again. In the worst case we must solve the deterministic problem for each θ_{n_q+i}, $i = 1,2,...$ To test this algorithm we computed the solution in Case 1 with 10000 TAU iterations (is usual number of iterations). Then, first, we computed the solution with the response surface (as described above) and, second, corrected it with 1000 TAU iterations. Then we compared both solutions and observed only a very small difference. Thus, the response surface reduced the number of needed iterations from 10000 to 1000. We note that the solution in Case 1 is smooth and there is no shock.

We tested this Algorithm also in the Case 9 (solution with a shock) and it failed. We pre-computed the solution by two different response surfaces (of order $P = 2$ and $P = 4$). Both response surfaces failed to produce a good result. For instance, we observed not only one shock, but many smaller shocks. Then we observed an increasing range of e.g. pressure (range $(-6;5)$ in contrast to $(0.5,1.3)$). It is similar when one tries to approximate a step function by a polynomial — the amplitude of oscillations grows up. Another negative effect which we observed during further iterating the solution, obtained from the response surface, was that the deterministic solver (TAU) produces "nan" after few iterations. A possible reason is that some important solution values, obtained from the response surface, are out of the physical range (e.g. negative density) and are non-realistic.

Thus, we can come to the conclusion that if the solution is smooth (e.g. as in Case 1) then response surface produces a good starting value. Otherwise, if the solution has a shock, the response surface produces a very poor approximation and further iterations do not help.

The computed solution $u(x, \theta_{n_q+1})$ can be used to update the response surface, i.e. to recompute the matrices $\mathbf{A}$, $\mathbf{B}$ and $[...\mathbf{c}_\beta...]$ and PCE coefficients (Eq. 13). Please note that this update works only in the case of the usage of embedded sparse grids or (Q)MC in Eq. 11.

5 Data Compression

A large number of stochastic realisations of random fields requires a large amount of memory and powerful computational resources. To decrease memory requirements and computing time we offer to use a low-rank approximation for all realisations of the solution [13]. This low-rank approximation allows us an effective postprocessing (computation of the mean value, variance, exceedance probability) with drastically reduced memory requirements (see Table 4). For each new realisation only the corresponding low-rank update will be computed (see, e.g. [1]). This can be practical when, e.g. the results of many thousands Monte Carlo simulations should be computed and stored. Let $\mathbf{v}_i \in \mathbb{R}^n$, $i = 1..Z$, be the solution vector (already centred), where Z is a number of stochastic realisations of the solution. Let us build from all these vectors the matrix $\mathbf{W} = (\mathbf{v}_1, ..., \mathbf{v}_Z) \in \mathbb{R}^{n \times Z}$ and consider the factorisation

$$\mathbf{W} = \mathbf{A}\mathbf{B}^T, \quad \text{where} \quad \mathbf{A} \in \mathbb{R}^{n \times k} \quad \text{and} \quad \mathbf{B} \in \mathbb{R}^{Z \times k}. \tag{15}$$

We say that matrix $\mathbf{W}$ is **a rank-k matrix** if the representation in Eq. 15 is given. We denote the class of all rank-k matrices for which factors $\mathbf{A}$ and $\mathbf{B}^T$ in Eq. 15 exist by $\mathscr{R}(k,n,Z)$. If $\mathbf{W} \in \mathscr{R}(k,n,Z)$ we say that $\mathbf{W}$ has a **low-rank representation**. The first aim is to compute a rank-k approximation $\mathbf{W}_k$ of $\mathbf{W}$, such that

$$\|\mathbf{W} - \mathbf{W}_k\| < \varepsilon, \quad k \ll \min\{n, Z\}.$$

The second aim is to compute an update for the approximation $\mathbf{W}_k$ with a linear complexity for every new coming vector $\mathbf{v}_{Z+1}$. Below in Section 5.1 we present the algorithm which performs this.

To get the reduced singular value decomposition we omit all singular values, which are smaller than a given level ε or, alternative variant, we leave a fixed number of largest singular values. After truncation we speak about *reduced singular value decomposition (denoted by rSVD)* $\mathbf{W}_k = \mathbf{U}_k \Sigma_k \mathbf{V}_k{}^T$, where $\mathbf{U}_k \in \mathbb{R}^{n \times k}$ contains the first k columns of $\mathbf{U}$, $\mathbf{V}_k \in \mathbb{R}^{Z \times k}$ contains the first k columns of $\mathbf{V}$ and $\Sigma_k \in \mathbb{R}^{k \times k}$ contains the k-biggest singular values of Σ.

The computation of such basic statistics as the mean value, the variance, the exceedance probability can be done with a linear complexity. The following examples illustrate computation of the mean value and the variance.

Let us take $\mathbf{A} := \mathbf{U}_k \Sigma_k$ and $\mathbf{B}^T := \mathbf{V}_k^T \in \mathbb{R}^{k \times Z}$. Denote the j-th row of matrix $\mathbf{A}$ by $\mathbf{a}_j \in \mathbb{R}^k$ and the i-th column of matrix $\mathbf{B}^T$ by $\mathbf{b}_i \in \mathbb{R}^k$. It is evident, that if $\mathbf{W}$ is given explicitly, one can compute the mean value and the variance just keeping in memory 2 vectors - the mean (variance) and the current value. Below we show how to compute the mean and the variance if only $\mathbf{A}$ and $\mathbf{B}$ are given.

1. One can compute the mean solution $\overline{\mathbf{v}} \in \mathbb{R}^n$ as follows

$$\overline{\mathbf{v}} = \frac{1}{Z}\sum_{i=1}^{Z} \mathbf{v}_i = \frac{1}{Z}\sum_{i=1}^{Z} \mathbf{A} \cdot \mathbf{b}_i = \mathbf{A}\overline{\mathbf{b}}, \tag{16}$$

Table 2 Rank-k approximation errors of the mean and of the variance of density in Case 1.

rank k	5	20
$\max_x \lvert\overline{\rho}(x) - \overline{\rho}_k(x)\rvert$	1.7e-6	4.2e-10
$\max_x \lvert\mathrm{var}(\rho)(x) - \mathrm{var}(\rho)_k(x)\rvert$	6.7e-5	2.3e-8

The computational complexity is $\mathcal{O}(k(Z+n))$, in contrast to $\mathcal{O}(nZ))$ for usual dense data format. As a demonstration we compute the mean.

2. One can compute the variance of the solution $\mathrm{var}(\mathbf{v}) \in \mathbb{R}^n$ by the computing the covariance matrix and taking its diagonal. First, one computes the centred matrix

$$\mathbf{W}_c := \mathbf{W} - \overline{\mathbf{v}}\mathbb{1}^T, \quad \text{where} \quad \overline{\mathbf{v}} = \mathbf{W}\cdot\mathbb{1}/Z, \text{ and } \mathbb{1} = (1,...,1)^T \in \mathbb{R}^Z. \tag{17}$$

Computing $\mathbf{W}_c$ costs $\mathcal{O}(k^2(n+Z))$ (addition and truncation of rank-k matrices). By definition, the covariance matrix is $\mathbf{C} = \frac{1}{Z-1}\mathbf{W}_c\mathbf{W}_c^T$. The reduced singular value decomposition of $\mathbf{W}_c$ is $(\mathbf{W}_c)_k = \mathbf{U}_k\boldsymbol{\Sigma}_k\mathbf{V}_k^T$, $\mathbf{U}_k \in \mathbb{R}^{n\times k}$, $\boldsymbol{\Sigma}_k \in \mathbb{R}^{k\times k}$ and $\mathbf{V}_k \in \mathbb{R}^{Z\times k}$ can be computed via the QR algorithm [5, 13]. Now, the covariance matrix can be written like

$$\mathbf{C} = \frac{1}{Z-1}(\mathbf{W}_c)_k(\mathbf{W}_c)_k^T \approx \frac{1}{Z-1}\mathbf{U}_k\boldsymbol{\Sigma}_k\boldsymbol{\Sigma}_k^T\mathbf{U}_k^T. \tag{18}$$

The variance of the solution vector (i.e. the diagonal of the covariance matrix $\mathbf{C}$ can be computed with the complexity $\mathcal{O}(k^2(Z+n))$.

Table 2 demonstrates the rank-5 and rank-20 approximations of the mean and of the variance of density. One can see that both rank-k approximation errors are very small, much smaller than e.g. the discretisation error or Monte Carlo error (by computing the mean value).

Lemma 0.1. *Let $\|\mathbf{W} - \mathbf{W}_k\|_2 \le \varepsilon$, and $\overline{\mathbf{u}}_k$ be a rank-k approximation of the mean $\overline{\mathbf{u}}$. Then a) $\|\overline{\mathbf{u}} - \overline{\mathbf{u}}_k\| \le \frac{\varepsilon}{\sqrt{N_s}}$, b) $\|\mathbf{W}_c - (\mathbf{W}_c)_k\| \le \varepsilon$, c) $\|\mathbf{C} - \mathbf{C}_k\| \le \frac{1}{N_s-1}\varepsilon^2$.*

Proof: Since $\overline{\mathbf{u}} = \frac{1}{N_s}\mathbf{W}\mathbb{1}$ and $\overline{\mathbf{u}}_k = \frac{1}{N_s}\mathbf{W}_k\mathbb{1}$, then

$$\|\overline{\mathbf{u}} - \overline{\mathbf{u}}_k\|_2 = \frac{1}{N_s}\|(\mathbf{W} - \mathbf{W}_k)\mathbb{1}\|_2 \le \frac{1}{N_s}\|(\mathbf{W} - \mathbf{W}_k)\|_2 \cdot \|\mathbb{1}\|_2 \le \frac{\varepsilon}{\sqrt{N_s}}.$$

Let $\mathbf{I} \in \mathbb{R}^{N_s\times N_s}$ be the identity matrix, then

$$\|\mathbf{W}_c - (\mathbf{W}_c)_k\|_2 \le \|\mathbf{W} - \mathbf{W}_k\|_2 \cdot \|\mathbf{I} - \frac{1}{N_s}\cdot\mathbb{1}\cdot\mathbb{1}^T\|_2 \le \varepsilon, \quad \text{and}$$

$$\begin{aligned}\|\mathbf{C} - \mathbf{C}_k\|_2 &\le \frac{1}{N_s-1}\|\mathbf{W}_c\mathbf{W}_c^T - (\mathbf{W}_c)_k(\mathbf{W}_c)_k^T\|_2 \\ &= \frac{1}{N_s-1}\|\mathbf{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T\mathbf{U}^T - \mathbf{U}_k\boldsymbol{\Sigma}_k\boldsymbol{\Sigma}_k^T\mathbf{U}_k^T\| \le \frac{1}{N_s-1}\varepsilon^2.\end{aligned}$$

5.1 Concatenation of Two Low-Rank Matrices

Let $\mathbf{A}$ and $\mathbf{B}$ such that $\mathbf{W}_k = \mathbf{A}\mathbf{B}^T$ be given. Suppose also that matrix $\mathbf{W}' \in \mathbb{R}^{n\times m}$ contains new m solution vectors. For a small m, computing the factors $\mathbf{C} \in \mathbb{R}^{n\times k}$ and $\mathbf{D} \in \mathbb{R}^{m\times k}$ such that $\mathbf{W}' \approx \mathbf{C}\mathbf{D}^T$ is not expensive. Now our purpose is to compute with a linear complexity the rank-k approximation of $\mathbf{W}_{\text{new}} := [\mathbf{W}\,\mathbf{W}'] \in \mathbb{R}^{n\times(Z+m)}$. To do this, we build two concatenated matrices $\mathbf{A}_{\text{new}} := [\mathbf{A}\,\mathbf{C}] \in \mathbb{R}^{n\times 2k}$ and $\mathbf{B}_{\text{new}}^T = \text{blockdiag}[\mathbf{B}^T\,\mathbf{D}^T] \in \mathbb{R}^{2k\times(Z+m)}$. Note that the difficulty now is that matrices $\mathbf{A}_{\text{new}}$ and $\mathbf{B}_{\text{new}}$ have rank $2k$. The rank k approximation of the new matrix $\mathbf{W}_{\text{new}}$ is done with a linear complexity $\mathcal{O}((n+Z)k^2+k^3)$ (for details see [13]).

6 Bayesian Update of the Uncertain Airfoil Geometry

We assume that the airfoil geometry contains random deformations (e.g. dents). A possible reason, for example, can be the influence of external forces. First our task is to parametrize all such deformations for all given airfoils. We offer to use random fields $\kappa(x,\omega)$, where ω is a vector of random parameters (see Section 2.2). The problem is that the probability density function of ω is unknown. We assume it a priori as Gaussian. If we could measure all given airfoils (from different airplains) then we could build a good parametrization model, but everything we can do is to measure airfoils only in a few points. This is our `knowledge`. The question now is how to incorporate this knowledge to our parametrization model? We can do this by using the Bayesian update. The Algorithm is described in [21, 20].

In Fig. 6 (left) you may see:

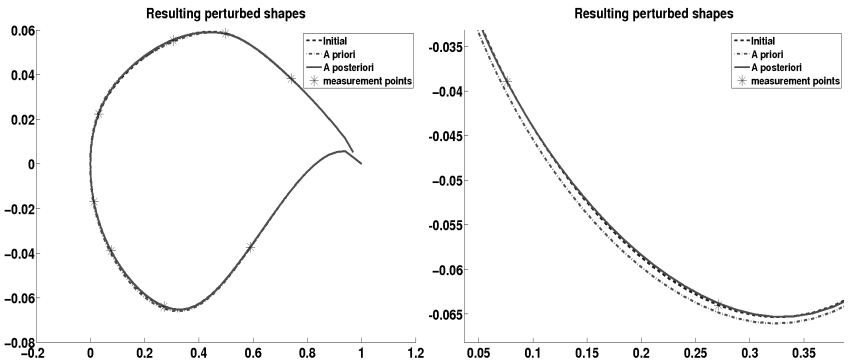

Fig. 6 (left) The truth airfoil (is in reality unknown), a priori (is our initial assumption) and a posteriori (the measurements are taken into account) airfoils. (right) Detailed RAE-2822 airfoil picture in interval [0.05, 0.35].

- The initial airfoil (dashed line). Is in real life, as a whole, unknown. One can measure it only in a few points.
- A priori realization (dash-dot line). One realization computed from the a priori model without any knowledge (without measurements). Usually does not coincide with the truth. Can be far away from the truth.
- A posteriori realization (solid line) is computed via the Bayesian update (see details of the Algorithm in [21, 20]) from the a posteriory model which takes into account the real data measured in the 8 measurement points (8 stars). Since large deformations are not allowed, all three curves are very similar. The detailed picture (in interval [0.05, 0.35]) is shown in Fig. 6 (right). One can see that the solid line (a posteriori model) is much closer to the measurement point (denoted by stars) than to the dash-dot line (a priori model).

7 Numerics

We demonstrate the influence of uncertainties in the angle of attack, the Mach number and the airfoil geometry on the solution (the pressure, density, lift, drag, lift and absolute skin friction coefficients). As an example we consider two-dimensional RAE-2822 airfoil. The deterministic solver is the TAU code with the k-w turbulence model. To quantify uncertainties we used the collocation method computed in nodes of sparse Gauss-Hermite grid. The Hermite polynomials are of order $P = \{1,2,4\}$ with M random variables (see Eq. 10).

The last column in Tables 3 on the left and on the right shows the measure of uncertainty σ/mean. It shows that 7.1% and 0.4% of uncertainties in α and in Ma correspondingly result in 4.4% and 16.3% (Table 3, on the right) of uncertainties in the lift CL and drag CD. For the comparison of different sparse grids see [13, 15].

In Fig. 7 we compare the cumulative distribution and density functions for the lift and drag coefficients, obtained via PCE and via 6300 Monte Carlo simulations. The response surface is PCE of order 1. There are 10^6 MC evaluations of the response surface. We see three very similar graphics. Thus, one can make the conclusion that the sparse Gauss-Hermite grid with a small number of points, e.g. $n_q = 13$, produces similar to MC results.

In Fig. 8 we compare the mean values computed by collocation and Monte Carlo methods for the Case 1. The collocation points are 281 nodes of two-dimensional

Table 3 Uncertainties in the input parameters ($\alpha(\theta_1,\theta_2)$ and $Ma(\theta_1,\theta_2)$) and in the solution (CL and CD). PCE of order 1 and sparse Gauss-Hermite grid with 137 points.

	mean	st. dev. σ	σ/mean
α	2.8	0.2	0.071
Ma	0.73	0.0026	0.004

$\Longrightarrow$

	mean	st. dev. σ	σ/mean
CL	0.85	0.0373	0.044
CD	0.0187	0.00305	0.163

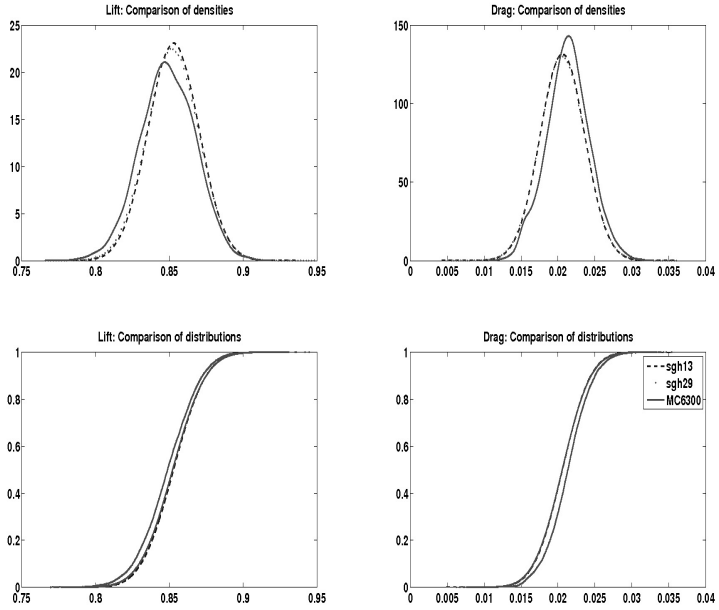

Fig. 7 Probability density functions (first row), cumulative distribution functions (second row) of *CL* (left) and *CD* (right). PCE is of order 1 with two random variables. Three graphics computed with 6360 MC simulations, $n_q = 13$ and $n_q = 29$ collocation points.

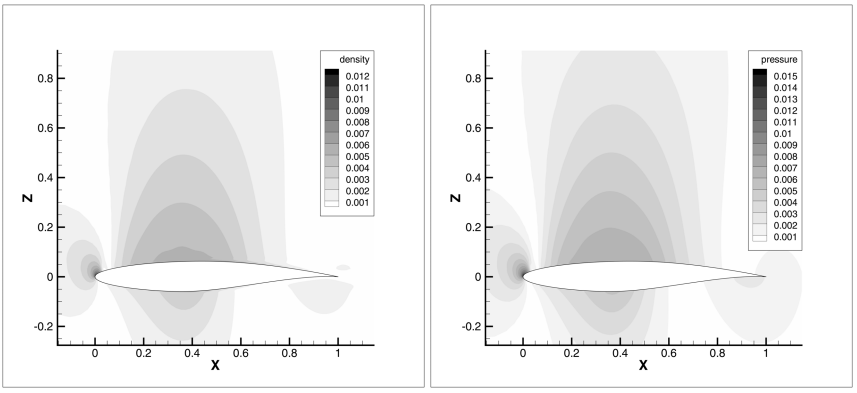

Fig. 8 (left) $\|\overline{\rho_{\mathrm{MC}}} - \overline{\rho_{\mathrm{SGH}}}\|$ and (right) $\|\overline{p_{\mathrm{MC}}} - \overline{p_{\mathrm{SGH}}}\|$. p_{SGH} was computed from the sparse Gauss Hermite grid with 281 nodes. $\rho_{\mathrm{SGH}} \in (0.65, 1.2)$, $p_{\mathrm{SGH}} \in (0.7, 1.3)$. Case 1.

sparse Gauss Hermite grid. One can see that the difference is very small compared to the corresponding physical values ρ_{SGH} and p_{SGH}.

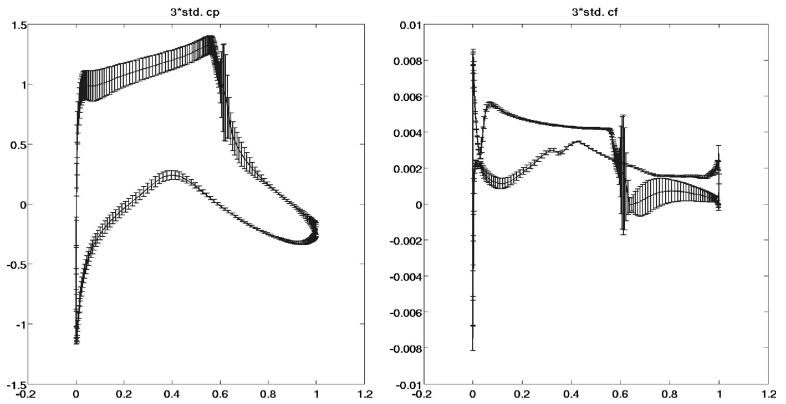

Fig. 9 3σ error bars in each point of the RAE-2822 airfoil surface for the pressure coefficient *cp* (left) and friction coefficient *cf* (right).

The graphics in Fig. 9 demonstrate 3σ error bars, σ the standard deviation, for the pressure *cp* and absolute skin friction *cf* coefficients in each surface point of the RAE-2822 airfoil. The data are obtained from 645 realisation of the solution. One can see that the largest uncertainty occurs at the shock ($x \approx 0.6$). A possible explanation is that the shock position is expected to change slightly with varying parameters α and *Ma*.

In Table 4 one can see relative errors of the rank-*k* approximations (in the Frobenius norm). Five solution matrices contain pressure, density, turbulence kinetic energy (tke), turbulence omega (to) and eddy viscosity (ev) in the whole computational domain with 260000 dofs. Additionally, one can also see much smaller memory requirement (dense matrix format costs 1.25GB). The column 7 shows the computing time required for the SVD-update (the Algorithm described in Section 5.1) and the the column 8 time required for the full SVD of the global matrix $\in \mathbb{R}^{260000\times 600}$ correspondingly. A possible explanation for the large computing time for the full SVD is the lack of memory and expensive swapping of data.

Table 4 Relative errors and computational requirements of rank-*k* approximations of the solution matrices $\in \mathbb{R}^{260000\times 600}$. Memory required for the storage of each matrix in the dense matrix format is 1.25 GB.

rank *k*	pressure	density	tke	to	ev	time, sec update	time, sec full SVD	memory MB
10	1.9e-2	1.9e-2	4.0e-3	1.4e-3	1.4e-3	107	1537	21
20	1.4e-2	1.3e-2	5.9e-3	3.3e-4	4.1e-4	150	2084	42
50	5.3e-3	5.1e-3	1.5e-4	9.1e-5	7.7e-5	228	8236	104

Table 5 Rank-k approximation errors of the variance of pressure and of the variance of density in Case 9

rank k	5	30
$\max_x \lvert \mathrm{var}(p)(x) - \mathrm{var}(p)_k(x) \rvert$	5.3e-3	1.6e-4
$\max_x \lvert \mathrm{var}(\rho)(x) - \mathrm{var}(\rho_k)(x) \rvert$	3.5e-3	8.8e-5

In Table 5 we provide the rank $k = \{5, 30\}$ approximation errors (in the maximum norm) of the variance of the pressure and of the density (compare with Fig. 5). The variances $\mathrm{var}(p)_k(x)$ and $\mathrm{var}(\rho_k)(x)$ were computed from the matrix $\mathbf{W} \in \mathbb{R}^{65568 \times 1521}$ as described in Section 5.

Further, we consider Case 1 ($\alpha = 1.93$ and $\mathrm{Ma} = 0.676$, no shock). Fig. 10 shows relative errors (for the Case 1) in the Frobenius and the maximum norms for pressure and density computed in 10 points of a two-dimensional sparse Gauss-Hermite grid. These relative errors compare the solution which we obtain after 10000 TAU iterations without any start value with the solution which we obtain after only 2000 TAU iterations with start values taken from the response surface (multivariate Hermite polynomials with $M = 2$ variables and of order $P = 2$). One can see that the errors are very small (of order 10^{-3}), i. e. the response surface produces a good approximation. We note that 10 chosen points are lying in a small neighbourhood of the point $\alpha = 1.93$ and $\mathrm{Ma} = 0.676$.

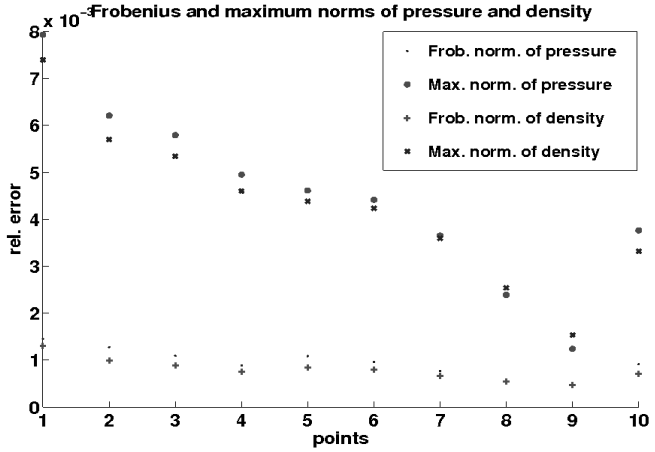

Fig. 10 Relative errors (Case 1) in the Frobenius and the maximum norms for the pressure and density

References

[1] Brand, M.: Fast low-rank modifications of the thin singular value decomposition. Linear Algebra Appl. 415(1), 20–30 (2006)

[2] Bungartz, H.-J., Griebel, M.: Sparse grids. Acta Numer. 13, 147–269 (2004)

[3] Chen, Q.-Y., Gottlieb, D., Hesthaven, J.S.: Uncertainty analysis for the steady-state flows in a dual throat nozzle. Journal of Computational Physics 204(1), 378–398 (2005)

[4] Gerhold, T., Friedrich, O., Evans, J., Galle, M.: Calculation of complex three-dimensional configurations employing the DLR-tau-Code. AIAA-0167 (1997)

[5] Golub, G.H., Van Loan, C.F.: Matrix computations, 3rd edn. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore (1996)

[6] Griebel, M.: Sparse grids and related approximation schemes for higher dimensional problems. In: Foundations of Computational Mathematics, Santander 2005. London Math. Soc. Lecture Note Ser., vol. 331, pp. 106–161. Cambridge Univ. Press, Cambridge (2006)

[7] Hida, T., Kuo, H.-H., Potthoff, J., Streit, L.: White noise - An infinite-dimensional calculus. Mathematics and its Applications, vol. 253. Kluwer Academic Publishers Group, Dordrecht (1993)

[8] Hosder, S., Walters, R., Perez, R.: A non-intrusive polynomial chaos method for uncertainty propagation in cfd simulations. In: 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, (AIAA 2006-891), pp. 209–236 (January 2006)

[9] Khoromskij, B.N., Litvinenko, A.: Data sparse computation of the Karhunen-Loève expansion. In: Numerical Analysis and Applied Mathematics: International Conference on Numerical Analysis and Applied Mathematics, AIP Conf. Proc., vol. 1048(1), pp. 311–314 (2008)

[10] Khoromskij, B.N., Litvinenko, A., Matthies, H.G.: Application of hierarchical matrices for computing the Karhunen-Loève expansion. Computing 84(1-2), 49–67 (2009)

[11] Lanczos, C.: An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. J. Research Nat. Bur. Standards 45, 255–282 (1950)

[12] Litvinenko, A., Matthies, H.G.: Sparse data representation of random fields. In: Proceedings in Applied Mathematics and Mechanics, PAMM, vol. 9, pp. 587–588. Wiley-InterScience (2009)

[13] Litvinenko, A., Matthies, H.G.: Low-rank data format for uncertainty quantification. In: Skiadas, C.H. (ed.) International Conference on Stochastic Modeling Techniques and Data Analysis Proceedings, Chania, Greece, pp. 477–484 (2010), `http://www.smtda.net/smtda2010proceedings.html`

[14] Litvinenko, A., Matthies, H.G.: Sparse data formats and efficient numerical methods for uncertainties quantification in numerical aerodynamics. In: IV European Congress on Computational Mechanics (ECCM IV): Solids, Structures and Coupled Problems in Engineering (2010), `http://www.eccm2010.org/complet/fullpaper_1036.pdf`

[15] Litvinenko, A., Matthies, H.G.: Uncertainties quantification and data compression in numerical aerodynamics. Proc. Appl. Math. Mech. 11(1), 877–878 (2011)

[16] Loève, M.: Probability theory I. Graduate Texts in Mathematics, 4th edn., vol. 45, 46. Springer, New York (1977)

[17] Mathelin, L., Hussaini, M.Y., Zang, T.A.: Stochastic approaches to uncertainty quantification in CFD simulations. Numer. Algorithms 38(1-3), 209–236 (2005)

[18] Matthies, H.G.: Uncertainty quantification with stochastic finite elements. Part 1. Fundamentals. In: Encyclopedia of Computational Mechanics. John Wiley and Sons, Ltd. (2007)
[19] Najm, H.N.: Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. In: Annual Review of Fluid Mechanics. Annu. Rev. Fluid Mech., vol. 41, pp. 35–52. Annual Reviews, Palo Alto (2009)
[20] Pajonk, O., Rosić, B.V., Litvinenko, A., Matthies, H.G.: A deterministic filter for non-gaussian bayesian estimation-applications to dynamical system estimation with noisy measurements. Physica D: Nonlinear Phenomena 241(7), 775–788 (2012)
[21] Rosic, B.V., Litvinenko, A., Pajonk, O., Matthies, H.G.: Sampling-free linear bayesian update of polynomial chaos representations. J. Comput. Physics 231(17), 5761–5787 (2012)
[22] Saad, Y.: Numerical methods for large eigenvalue problems. Algorithms and Architectures for Advanced Scientific Computing. Manchester University Press, Manchester (1992)
[23] Simon, F., Guillen, P., Sagaut, P., Lucor, D.: A gpc-based approach to uncertain transonic aerodynamics. Computer Methods in Applied Mechanics and Engineering 199(17-20), 1091–1099 (2010)
[24] Wan, X., Karniadakis, G.E.: Long-term behavior of polynomial chaos in stochastic flow simulations. Computer Methods in Applied Mechanics and Engineering 195(41-43), 5582–5596 (2006); John H. Argyris Memorial Issue. Part II
[25] Wiener, N.: The homogeneous chaos. American Journal of Mathematics 60, 897–936 (1938)
[26] Witteveen, J.A.S., Loeven, A., Bijl, H.: An adaptive stochastic finite elements approach based on newton-cotes quadrature in simplex elements. Computers & Fluids 38(6), 1270–1288 (2009)
[27] Witteveen, J.A.S., Doostan, A., Pecnik, R., Iaccarino, G.: Uncertainty quantification of the transonic flow around the rae2822 airfoil. Annual Research Briefs, Center for Turbulence Research, pp. 93–104. Stanford University (2009)