Backward uncertainty propagation in shape optimization

Bijan Mohammadi

BACKWARD UNCERTAINTY PROPAGATION IN SHAPE OPTIMIZATION

BIJAN MOHAMMADI

MONTPELLIER UNIVERSITY
MATHEMATICS, CC51, 34095 MONTPELLIER, FRANCE
BIJAN.MOHAMMADI@UMONTPELLIER.FR

ABSTRACT. We aim at quantifying the impact of state uncertainties in shape optimization. This provides confidence bounds for the optimal solution. The approach is presented for inverse designs where the target is assumed uncertain. No sampling of a large dimensional space is necessary and the approach uses what is already available in a deterministic gradient-based inversion algorithm. Our proposal is based on the introduction of directional quantile-based extreme scenarios knowing the Probability Density Function (PDF) of the target data. We use these scenarios to define a matrix having the structure of the covariance matrix of the optimization parameters. We compare this construction to another one using the gradient of the functional by an adjoint method. The paper goes beyond inverse design and shows how to apply the method to general optimization problems. The ingredients of the paper are illustrated on a model problem with the Burgers equation and on the optimization of the shape of an aircraft. Overall, the computational complexity is comparable to the deterministic case.

1. INTRODUCTION

Forward and backward uncertainty propagation [1, 4, 25] are obviously of great importance with a huge literature dedicated to uncertainty quantification (UQ). Forward propagation aims at defining, for instance, a probability density function for a functional $j(\mathbf{x})$ knowing those of optimization parameters $\mathbf{x}$ [18, 27, 21]. This can be done, for instance, through Monte Carlo simulations or a separation between deterministic and stochastic features using Karhunen-Loeve theory (polynomial chaos theory belongs to this class) [14, 44, 15, 42]. Backward propagation aims at reducing models bias or calibrating models parameters knowing the probability density function of j [40, 23, 5]. This can be seen as a minimization problem and Kalman filters [24] give, for instance, an elegant framework for this inversion assimilating the uncertainties on the observations.

In this work we discuss one aspect of UQ where the target state u^* used in an inverse problem is uncertain. This is the case, for instance, minimizing $j(\mathbf{x}) = \|u(\mathbf{x}) - u^*\|$ to reduce the distance between a model state $u(\mathbf{x})$ and observations. This work therefore rather belongs to the class of backward uncertainty propagation.

In the past, we showed how to quantify epistemic and aleatory uncertainties through geometric characterizations of global sensitivity spaces built using adjoint-based gradients of the functional available in existing optimization platforms [30]. In the presence of different modeling or solution methods, principal angles between

---

these sensitivity subspaces permit to measure the deviation due to a change in the modeling [31]. We also showed how these ingredients can be used in the context of multi-point robust analysis of a system to define worst-case scenarios for its functioning. To this end we combined a multi-point search direction with the probabilistic features of the control parameters through their quantiles [22, 33]. These ingredients permit to define the concept of directional uncertainty quantification and directional extreme scenarios (DES) [32]. Global sensitivity spaces can then be built for these extreme scenarios and the above geometric characteristics permit again to measure the impact of the variability of the parameters of the problem. The main interest of this construction is to account for the variability of the parameters in large dimension without a sampling of the corresponding parameter space.

To go beyond directional extreme scenarios, we combined the previous ingredients with Ensemble Kalman Filters (EnKF) [24, 2, 12, 13, 9, 10] which are efficient to account for the variability in the observations in inversion. The outcome is what we called Ensemble Directional Extreme Scenarios (EDES) which provides a more accurate and exhaustive sampling of the boundary of the uncertainty domain. We showed how to link the necessary ensemble size to the geometry of global sensitivity spaces in inversion [34].

Despite these approaches avoid the sampling of a large dimensional space and despite EnKF, indeed, is an elegant solution for the inversion of uncertain data, the computation cost remains high and the procedures difficult to simply explain in engineering environments. In this work, we would like to simplify our procedures and propose low-complexity uncertainty information on the design variables still using the ingredients of existing gradient-based optimization platforms. We illustrate our purpose in the context of inversion of uncertain data.

Targeting uncertain data is a realistic situation as the acquired data are usually uncertain. It is therefore interesting to be able to quantify the impact of this uncertainty on the inversion results. An important information will be the sensitivity of the design to a given level of uncertainty on the data at some location. Indeed, if this sensitivity is low, this would be an indication that a more accurate acquisition there is unnecessary.

Considering the target as uncertain is also interesting because we do not always have existence of a solution for an inversion problem as u^* is not necessary solution of the state equation making an exact or deterministic inversion inversion pointless. Also, the approach permits to go beyond inversions based on least square minimization involving a mean state (flow) target.

Finally, the uncertainty in measurements is also an interesting way to account for the presence of variability in the state (e.g. due to the presence of turbulence in the flow). More generally, as a model or numerical procedure are by nature imperfect and partial, we can consider this uncertainty as a representation and estimation of these imperfections. These imperfections are even more present in inverse problems where one cannot afford the same level of resolution than for a single simulation. We therefore need to be able to quantify the impact of these weaknesses. The approach presented here is therefore also useful to account for epistemic uncertainties related to possible model or solution procedures deficiencies.

The ingredients of the paper are illustrated on a model problem with the Burgers equation and a shape optimization problem for an aircraft.

2. General settings

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

$$(1) \qquad \min_{\mathbf{x} \in \mathbf{O}_{ad}} j(\mathbf{x}, u^*), \ u^* \in \mathbf{I} \subset \mathbb{R}^p, \mathbf{O}_{ad} \subset \mathbb{R}^n.$$

where $\mathbf{x}$ is the design vector belonging to $\mathbf{O}_{ad}$ the optimization admissible domain. This is a very general context and we visited it to address robustness issues in optimization with respect to $\mathbf{x}$ and u^* [33, 31, 32]. In this work we are interested in functionals j of the form:

$$(2) \qquad j(\mathbf{x}, u^*) = \frac{1}{2} \|\Pi u(\mathbf{x}) - u^*\|^2,$$

where the state $u(\mathbf{x}) \in \mathbb{R}^N$ is solution of a state equation $F(u(\mathbf{x})) = 0$ and operator $\Pi : \mathbb{R}^N \to R^p$ (typically a linear interpolation operator) makes the state available at data locations. Inverse problems are in this class [20, 40].

3. Uncertainties on data

We assume that the data u^* are uncertain and given by their probability density functions here assumed Gaussian $\mathcal{N}(\mu_i, \sigma_i^2)$, $i = 1, ..., p$ with mean μ_i and variance of σ_i^2.

3.1. **Propagation of the uncertainties.** The simplest way to measure the effect of these uncertainties on the inversion result is to proceed with Monte Carlo simulation. This implies a sampling of the variation domain of the data consistent with their PDF. This means we proceed with M independent inversions for M data sets defined by independent choices compatible with the PDF of u^* given by:

$$\mathcal{N}(\mu_i, \sigma_i^2) \to (u_i^*)^m, \quad i = 1, ..., p, \quad m = 1, ..., M.$$

These independent inversions will produce M optimal control parameters $\mathbf{x}_{opt}^m$, $m = 1, ..., M$ from which statistical moments can be defined (typically the mean and variance) with a rate of convergence in $M^{-1/2}$ independent of the size p. Such generation of scenarios is already very demanding even when involving only a direct simulation chain. In our problem, each of the scenarios involves an inversion, each requiring several solutions of the direct and adjoint problems. This complexity makes that this approach is clearly out of the table even if the calculations are independent and can be carried out in parallel.

As mentioned in the introduction, Ensemble Kalman Filters, also using a reduced number of parallel scenarios, provide an elegant solution to assimilate these uncertainties. We can cast our inversion, even if it is gradient-based, into the EnKF formulation as shown in [34]. Considering the underlying global sensitivity space generated by the gradients of the functional for the different scenarios, we showed that EnKF is only efficient if the dimension of this space is small compared to the dimension of the design space. Otherwise, EnKF is not feasible with a reduced ensemble size as this size needs to be larger than the dimension of the global sensitivity space. Obviously in this case the calculation complexity is again a burden.

3.2. **Low-complexity uncertainty evaluation.** In the sequel, we discuss two low-complexity constructions of $Cov_{\mathbf{x}}$ the covariance matrix of the control parameters from Cov_{u^*} the covariance matrix of the data. We want these constructions to have a cost comparable to a deterministic inversion and we want to avoid any sampling of a large dimension space.

We work in the context of a deterministic gradient based minimization algorithm using an adjoint formulation to access $\nabla_{\mathbf{x}} j$ [31, 30, 32] with j given by (2).

4. α-QUANTILE

Consider a random variable v with its PDF known (either analytic or tabulated). The tail of the PDF can be characterized defining for a given probability level ($0 < \alpha < 1$) the following threshold value:

$$\mathrm{VaR}_\alpha = \inf\{l \in \mathbb{R} : P(v > l) \leq 1 - \alpha\}.$$

Different α-quantile are available. One very well known is the Value at Risk (VaR) which has been widely used in financial engineering as a measure of risk of loss on a given asset [22]. We do not need the time dependency issue here but it is interesting as it permits to account for possible improvement of measurement accuracy as discussed in [33].

4.1. **Bounding the uncertainty domain.** We would like to use the concept of α-quantile (we call in the sequel VaR) to define a closed domain of variation for the uncertain data [33]. Given a threshold $0 \leq \alpha < 1$, a data $u_i^*, i = 1, ..., p$ belongs to the interval $[\mu_i + \mathrm{VaR}_\alpha^-, \mu_i + \mathrm{VaR}_\alpha^+]$ with $\mathrm{VaR}_\alpha^- \leq 0 \leq \mathrm{VaR}_\alpha^+$ with probability α.

As we consider Gaussian probability density functions we have $\mathrm{VaR}_\alpha^- = -\mathrm{VaR}_\alpha^+$ and the values at risk are explicitly known:

$$\mathrm{VaR}_{0.99}(N(0,1)) = 2.33, \quad \text{and} \quad \mathrm{VaR}_{0.95}(N(0,1)) = 1.65,$$

and $\mathrm{VaR}_\alpha(N(0, \sigma^2)) = \sigma^2 \mathrm{VaR}_\alpha(N(0,1))$.

We have therefore, with probability α, an uncertainty domain for the data given by:

$$B_\alpha(\mu) = \Pi_{i=1}^p [\mu_i - 1.65\sigma_i^2, \mu_i + 1.65\sigma_i^2] \subset \mathbb{R}^p$$

This is a large domain and we do not want to proceed with any sampling of it.

4.2. **Directional Extreme Scenarios (DES).** However, using the sensitivity of the functional with respect to the data we can identify two directional extreme sets of data corresponding to the intersection of $B_\alpha(\mu)$ and $d = \mu + t\, \partial j / \partial u^*,\ t \in \mathbb{R}$. Let us call these two data sets $(u^*)^\pm$ defined by:

$$(3) \qquad\qquad (u^*)^\pm = \mu \pm 1.65\,\sigma_i^2\,\left(\frac{\partial j/\partial u^*}{\|\partial j/\partial u^*\|}\right)_i.$$

To measure of the impact of this variability on the result of the inversion, we proceed with two minimization with the target data given by $(u^*)^\pm$ starting from $\mathbf{x}^* = \mathbf{x}_{opt}(u^* = \mu)$. Let us call $(\mathbf{x}^*)^\pm$ the results of these inversions.

We assume monotonic behavior of the outcome of the inversion with respect to the data:

$$(4) \qquad\qquad \|\mathbf{x}^*(\mu) - \mathbf{x}^*(\nu)\| \nearrow \quad \text{if} \quad \|\mu - \nu\| \nearrow .$$

This assumption is reasonable and means that larger deviations in data sets bring larger variations in the outcome of the optimization. This also suggests that the

maximum deviation for the results of the inversion due to the uncertainty on the data can be estimated through: $X^{\pm} = (\mathbf{x}^*)^+ - (\mathbf{x}^*)^-$. We consider the following matrix as an approximation of the covariance matrix $Cov_{\mathbf{x}^{\pm}}$ [43] for $\mathbf{x}$:

$$(5) \quad Cov_{\mathbf{x}^{\pm}} = \mathbb{E}((X^{\pm})(X^{\pm})^t) - \mathbb{E}(X^{\pm})\mathbb{E}(X^{\pm})^t \sim (X^{\pm})(X^{\pm})^t - (\overline{X^{\pm}})(\overline{X^{\pm}})^t,$$

with $\overline{X^{\pm}} = ((\mathbf{x}^*)^+ + (\mathbf{x}^*)^- - 2\mathbf{x}^*)/2$.

The concept of Directional Extreme Scenarios has been presented in [31, 32] with applications to robust shape optimization in aeronautics, atmospheric dispersion and also to quantify the sensitivity of littoral erosion to uncertainties in bottom sand characteristics [36].

4.3. **Algorithm.** Let us summarize the steps to take in an existing optimization platform to introduce the ingredients above.

1. Start minimizing $j(\mathbf{x}, u^*)$ for $u^* \in \mathbb{R}^p$. Let us call $\mathbf{x}_{opt}(u^*)$ the result of this deterministic inversion. We use a gradient-based global minimizer [19, 39] with an adjoint formulation for $\nabla_{\mathbf{x}}j$ shortly recalled in section 5 and 7 [31, 30, 37].

2. Define two extreme sets of data $(u^*)^{\pm} = B_{\alpha}(\mu) \cap d$ with $d = \mu + t\, \partial j(\mathbf{x}_{opt}(u^*))/\partial u^*$.

3. Minimize j for $(u^*)^{\pm}$ giving $\mathbf{x}_{opt}^{\pm} = Argmin_{\mathbf{x} \in \mathbf{O}_{ad}} j(u(\mathbf{x}), (u^*)^{\pm})$.

We have therefore three solutions for our optimization: $\mathbf{x}_{opt}(u^*)$ corresponding to the classical deterministic minimization and $\mathbf{x}_{opt}((u^*)^{\pm})$ which can be used to provide confidence bounds for the deterministic inversion.

5. Sensitivity evaluation

During optimization, we need the gradient of the functional with respect to optimization parameter $\nabla_{\mathbf{x}}j(\mathbf{x}, u(\mathbf{x}))$ [38, 37]. This is obtained through an adjoint formulation. Let us recall the approach for a generic state equation $F(u(\mathbf{x})) = 0$. The gradient of j with respect to $\mathbf{x}$ writes:

$$\nabla_{\mathbf{x}}j = \frac{\partial j}{\partial \mathbf{x}} + \left((\frac{\partial j}{\partial u})^t \frac{\partial u}{\partial \mathbf{x}} \right)^t = \frac{\partial j}{\partial \mathbf{x}} + \left((\frac{\partial j}{\partial u})^t (\frac{\partial F}{\partial u})^{-1} \frac{\partial F}{\partial \mathbf{x}} \right)^t = \frac{\partial j}{\partial \mathbf{x}} + \left(v^t \frac{\partial F}{\partial \mathbf{x}} \right)^t,$$

where we have introduced the adjoint variable v solution of:

$$(6) \qquad\qquad v^t \frac{\partial F}{\partial u} = (\frac{\partial j}{\partial u})^t.$$

In cases the governing equations are self adjoint (i.e. $\frac{\partial F}{\partial u} = (\frac{\partial F}{\partial u})^t$), one can use the corresponding solver with $\frac{\partial j}{\partial u}$ as the right-hand side and simply solve:

$$\frac{\partial F}{\partial u} v = \frac{\partial j}{\partial u}.$$

Also, if F is linear, $\frac{\partial F}{\partial u}$ is a constant operator independent of u. The interest of the adjoint formulation is that the cost of getting $\nabla_{\mathbf{x}}j$ becomes independent of the size of $\mathbf{x}$. But, the problem with the adjoint approach is that, except for the two situations we mentioned (linear or self adjoint state equations), it needs the development (and maintenance) of a new code. This is why, as discussed in sections 8 and 10, we use automatic differentiation when possible. We give an example of continuous adjoint derivation for a time dependent nonlinear problem in section 7.

In multi-criteria problems like the one shown in section 8, where the functional j is minimized under equality or inequality constraints $C_{i=1,...,q}$, we need to solve an adjoint problem for the functional and each of the active constraints (needed to express

the first order KKT conditions). This can be seen as a block diagonal matrix inversion with all blocks similar and the right-hand side given by $(\partial_u j, \partial_u C_1, ..., \partial_u C_q)^t$ if we have q constraints active. We will describe how automatic differentiation in reverse mode with multiple right-hand side capacity can be used to address this problem. Otherwise, deflation techniques for linear systems with multiple right-hand sides can be applied [41, 28] taking advantage of the fact that the blocks being the same the Krylov decomposition needs to be conducted only once.

j is the least square deviation at data location between model and data. $\partial_u j$ in the right-hand side of (6) is a vector of size N and can be obtained writing:

$$j(\mathbf{x}, u^*) = \frac{1}{2}\|\Pi u - u^*\|^2 = \frac{1}{2} < \Pi u - u^*, \Pi u - u^* >$$

$$= \frac{1}{2} < \Pi^t \Pi u, u > - < \Pi^t u^*, u > + \frac{1}{2} < u^*, u^* >,$$

and we have $\partial_u j = \Pi^t \Pi u - \Pi^t u^*$. On the other hand, the sensitivity of j with respect to the data $\partial_{u^*} j$ needed in (3) is a vector of size p given by $\partial j / \partial u^* = -(\Pi u - u^*)$.

6. From the adjoint to the covariance matrix of the optimization parameters

Let us discuss how to take advantage of our adjoint calculation leading to $\nabla_\mathbf{x} j$ the gradient of the functional with respect to the optimization parameters to estimate the covariance matrix of these latter.

Let us start establishing the expression for the covariance matrix of $\mathbf{x}$ considered as a vector of zero-mean random variables. Denote, for simplicity, by u the model solution (also zero mean valued: $u \leftarrow u - \mu$) at data location and suppose it is linked to the parameters through a linear model: $u = L\mathbf{x}$. The covariance matrix for u is therefore:

$$Cov_u = \mathbb{E}(uu^t) = \mathbb{E}(L \mathbf{x}\mathbf{x}^t L^t) = L \mathbb{E}(\mathbf{x}\mathbf{x}^t) L^t = L \; Cov_\mathbf{x} \; L^t.$$

If the dependency u with respect to the parameter $\mathbf{x}$ is nonlinear the analysis holds for the linearized model. Introducing $\mathcal{J} = \partial u / \partial \mathbf{x}$ we have:

$$Cov_u = \mathcal{J} \; Cov_\mathbf{x} \; \mathcal{J}^t.$$

To get $Cov_\mathbf{x}$ we need therefore to invert this expression and because the amount of data can be large and probably impossible to exactly fit, we proceed with a least-square formulation looking for $Cov_\mathbf{x}$ minimizing:

$$\frac{1}{2} < \mathcal{J} \; Cov_\mathbf{x} \; \mathcal{J}^t, \mathcal{J} \; Cov_\mathbf{x} \; \mathcal{J}^t > - < Cov_u, \mathcal{J} \; Cov_\mathbf{x} \; \mathcal{J}^t > .$$

First order optimality condition with respect to $Cov_\mathbf{x}$ gives:

$$\mathcal{J}^t \mathcal{J} \; Cov_\mathbf{x} \; \mathcal{J}^t \mathcal{J} - \mathcal{J}^t \; Cov_u \; \mathcal{J} = 0,$$

which leads to

$$Cov_\mathbf{x} = (\mathcal{J}^t \mathcal{J})^{-1} \; \mathcal{J}^t \; Cov_u \; \mathcal{J} \; (\mathcal{J}^t \mathcal{J})^{-1},$$

and eventually, to

$$(7) \qquad Cov_\mathbf{x} = \mathcal{J}^{-1} \; Cov_u \; \mathcal{J}^{-t} = \left(\mathcal{J}^t \; Cov_u^{-1} \; \mathcal{J}\right)^{-1}.$$

To get $Cov_\mathbf{x}$ and knowing Cov_u, it is therefore sufficient to evaluate $\mathcal{J} = \partial u / \partial \mathbf{x}$. The second expression in (7) is interesting as it involves the inversion of a $n \times n$ matrix and gives a least square sense to the inversion of $N \times n$ matrices. Also, if

the optimization is successful and model u and data u^* close, we can use the fact that data are usually independent and use the covariance matrix of the observation instead of Cov_u:

$$Cov_u \sim Cov_{u^*},$$

which is then diagonal and its inversion straightforward.

The question is, therefore, how to efficiently evaluate $\mathcal{J} = \partial u / \partial \mathbf{x}$. The model at data locations Πu is obtained applying, for instance, a linear interpolation operator Π to the model solution u on the mesh. Therefore, we have:

$$\mathcal{J} = \Pi \frac{\partial u}{\partial \mathbf{x}}.$$

Now recall that $\nabla_{\mathbf{x}} j$ is available and has been computed with an adjoint approach in section 5. We now use it to access $\partial u / \partial \mathbf{x}$ without extra calculation:

$$\nabla_{\mathbf{x}} j = \frac{\partial j}{\partial \mathbf{x}} + \left((\frac{\partial j}{\partial u})^t \frac{\partial u}{\partial \mathbf{x}} \right)^t = \frac{\partial j}{\partial \mathbf{x}} + \left((\frac{\partial j}{\partial u})^t \Pi^{-1} \mathcal{J} \right)^t,$$

The first terms in the right-hand sides vanishes for (2) but is non-zero if, for instance, a Tykhonov regularization term is introduced in the functional [40]. This leads to:

$$(\frac{\partial j}{\partial u})^t \Pi^{-1} \mathcal{J} = (\nabla_{\mathbf{x}} j - \frac{\partial j}{\partial \mathbf{x}})^t,$$

and eventually to,

$$(8) \qquad \mathcal{J} = \Pi \, (\frac{\partial j}{\partial u})^{-t} \, (\nabla_{\mathbf{x}} j - \frac{\partial j}{\partial \mathbf{x}})^t.$$

$(\partial j / \partial u)^{-t}$ is a line vector of size N with components given by the inverse of those of $(\partial j / \partial u)$ divided by N in order to have $(\partial j / \partial u)^{-t} . (\partial j / \partial u) = 1$.

Alternatively, to avoid numerical difficulties with small components of $(\partial j / \partial u)$, (8) can again be seen in a least square sense with the inverse of a normal matrix involved:

$$(9) \qquad \mathcal{J} = \Pi \, \left((\frac{\partial j}{\partial u})(\frac{\partial j}{\partial u})^t \right)^{-1} \, \frac{\partial j}{\partial u} \, (\nabla_{\mathbf{x}} j - \frac{\partial j}{\partial \mathbf{x}})^t.$$

This expression involves the inverse of the information matrix $((\partial j / \partial u)(\partial j / \partial u)^t)$. One should be aware that the numerical condition of this matrix can be very poor. We do not discuss this issue here but typically the Bunch and Kaufman [3] algorithm should be used in order to account for this possible deficiency. In particular, if rank deficiency is detected the Moore-Penrose inverse should be used based on the eigenvalue decomposition of the information matrix [7].

Under the hypothesis of the validity of the physical model, this analysis gives indications on the level of backward sensitivity of the optimization parameters with respect to the model solution at data locations which is also the sensitivity with respect of the deviation between the model and data at the data locations (as the data are independent of the optimization parameters):

$$\frac{\partial u}{\partial \mathbf{x}} = \frac{\partial (u - u^*)}{\partial \mathbf{x}}.$$

7. Inversion with the Burgers equation

Let us apply the concepts presented above for the quantification of some of the uncertainties for the solution of the following Burgers equation with a distributed control in its right-hand side:

$$\partial_t u + \partial_y \frac{u^2}{2} = \mathbf{x}(y)u,$$

(10)
$$u(t, -1) = 1, u(t, 1) = -0.8,$$

$$u(t = 0, y) = u^0(y) = -\frac{1 + 0.8}{2}y + \frac{1 - 0.8}{2},$$

where the control $\mathbf{x}(y)$ is set to $\mathbf{x}(y) = \mathbf{x}^*(y) = 0.3y$. The steady solution (we denote by $u^\infty = u(t \to \infty, y)$) of (10) is piecewise parabolic and has a jump at $y = s$:

(11)
$$u^\infty(y) = 0.15y^2 + 1 - 0.15 \quad \text{for } y < s,$$
$$u^\infty(y) = 0.15y^2 - 0.95 \quad \text{for } y > s,$$

and the shock position is found by asking for the flux to have no jump:

(12)
$$(u^\infty)_s^- = -(u^\infty)_s^+ \quad \text{therefore} \quad s = -\sqrt{\frac{1}{3}}.$$

We solve equation (10) with a centred scheme for first and second order spatial derivatives after introduction of a numerical viscosity given by $\nu_h = 0.5hu_{max}$ where $u_{max} = \max(|u_{i-1}|, |u_i|, |u_{i+1}|)$ on a 100 points uniform mesh (i.e. $h = 0.02$). Time integration is based on a third order Runge-Kutta scheme with local times step given by h/u_{max}. These are very basic choices and much sophisticated schemes could have been envisaged but this is not central to our discussion.

One example of inverse design is when we would like to recover the control $\mathbf{x}(y)$ minimizing the functional j given by:

(13)
$$j(\mathbf{x}(y)) = \int_0^\infty \int_{-1}^1 J(\mathbf{x}(y), t, y) dy dt,$$

where the control $\mathbf{x}(y) \in \mathbb{R}^{100}$ is defined by its 100 values of the mesh points. One typical situation in inverse problems is with $J = \frac{1}{2}(u(\mathbf{x}(y), t) - u^*(y))^2$ to measure the distance between the state variable and the target state u^*. If we choose for u^* the piecewise parabolic solution (11) the existence of the solution is guaranteed with $\mathbf{x}^* = 0.3y$.

Now, suppose that the target u^* is uncertain and defined, for instance, as $u^*(y) = u^\infty(y)(1 + \mathcal{N}(0, \sigma^2(y)))$ with the local uncertainty on target data known. *The question we are interested in is how much this affects the inversion ? In other words, we would like to have confidence bounds around the solution $\mathbf{x}^*(y)$ which would be obtained (at best) by an inversion in a deterministic context where $\sigma(y) = 0$.*

Of course, the control can be recovered knowing u^* in regions where the solution u has no shocks and where we can therefore write:

(14)
$$u^*\mathbf{x}^*(y) = \partial_t(u^*) + u^*\partial_y(u^*) = u^*\partial_y(u^*).$$

The second equality is because we are looking for a control independent of the time. So $\mathbf{x}^* = \partial_y(u^*)$ which is known in regions where u_d is smooth and we recover the target control. Solving this problem in the context of uncertain measurement u^* also permits to a posteriori give a sens to $\partial_y(u^*)$.

Let us first establish the adjoint formulation described in section 5 for our problem to obtain $\nabla_{\mathbf{x}} j$ at a cost independent of the size of $\mathbf{x}$.

We denote by T_∞ the time at which the steady solution is reached by our direct solver. We voluntarily present the approach in a time dependent context with time marching even if in this case none of the target control nor the state feature time dependency. This is to remain close of what is done in flow solvers, such as the one used in section 8, where time marching to steady state is used. The approach remains obviously valid if the control is function of time.

7.0.1. *The adjoint formulation.* The derivative of $j = \frac{1}{2T_\infty} \int_{(-1,1)\times(0,T_\infty)} J$ with respect to $\mathbf{x}$ writes:

$$\nabla_{\mathbf{x}} j = \frac{1}{2T_\infty} \int_{(-1,1)\times(0,T_\infty)} (J_{\mathbf{x}} + J_u u_{\mathbf{x}}) dt dy.$$

For our functional targeting u^*, $J_{\mathbf{x}} = 0$. From the linearized Burgers equation we have:

$$(15) \qquad \partial_t u_{\mathbf{x}} + \partial_y(u u_{\mathbf{x}}) - \mathbf{x}(y) u_{\mathbf{x}} = 0, \;\; u_{\mathbf{x}}(y, t = 0) = -\frac{1}{0.6}, \;\; u_{\mathbf{x}}(y = \pm 1, t) = 0.$$

One introduces the adjoint state v solution of the backward equation:

$$(16) \qquad \partial_t v + u \partial_y v - \mathbf{x}(y) v = J_u, \qquad v(y, t = T_\infty) = v(y = \pm 1, t) = 0,$$

and write

$$(17) \quad \frac{1}{2T_\infty} \int_{(-1,1)\times(0,T_\infty)} J_u u_{\mathbf{x}} dt dy = \frac{1}{2T_\infty} \int_{(-1,1)\times(0,T_\infty)} (\partial_t v + u \partial_y v - \mathbf{x}(y) v) u_{\mathbf{x}} dt dy$$

$$= -\frac{1}{2T_\infty} \int_{-1}^{1} (u^0)_{\mathbf{x}} \, v(y, t = 0) dy = -\frac{1}{2T_\infty} v(y, t = 0),$$

where $(u^0)_{\mathbf{x}} = \delta(y)$ formally indicates the sensitivity with respect to the independent variable $\mathbf{x}$ of the initialization of the dependency chain $\mathbf{x} \to u^0 \to u \to J$. The adjoint state v has no shock because its time boundary condition is continuous and the characteristics integrated backward never cross the shock [16, 38].

7.1. **Directional extreme scenarios.** Suppose we have proceeded with a deterministic inversion with a gradient method using our adjoint-based gradient as presented in section 4.3 and obtained the solution $\mathbf{x}^*(y)$. We would like now to establish the two extreme scenarios presented in sections 4.2. It is reasonable to assume $\sigma^2(y) \sim |\partial_y^2 u|$. The second derivative can be established in regions where u is regular (piecewise parabolic). This gives us a piecewise constant σ over the domain. Using this distribution, we define our extreme scenarios data from (3):

$$(18) \qquad (u^*)^\pm = u^*(y) \pm 1.65 \, \sigma^2(y) \left(\frac{\partial j / \partial u^*}{\|\partial j / \partial u^*\|} \right)_i,$$

and proceed with two inversions with $u^* = (u^*)^\pm$ as described in section 4.3 giving $(\mathbf{x}^*)^\pm$. As in section 4.2, we introduce $X^\pm = (\mathbf{x}^*)^+ - (\mathbf{x}^*)^-$ and also consider the difference:

$$(19) \qquad \varepsilon(y) = \max(|(\mathbf{x}^*)^+ - \mathbf{x}^*|, |(\mathbf{x}^*)^- - \mathbf{x}^*|, |(\mathbf{x}^*)^+ - (\mathbf{x}^*)^-|),$$

which appears here to coincide with $|X^\pm|$ which is compatible with the monotonicity hypothesis made is section 4.2. This gives an indication of how much deviation the
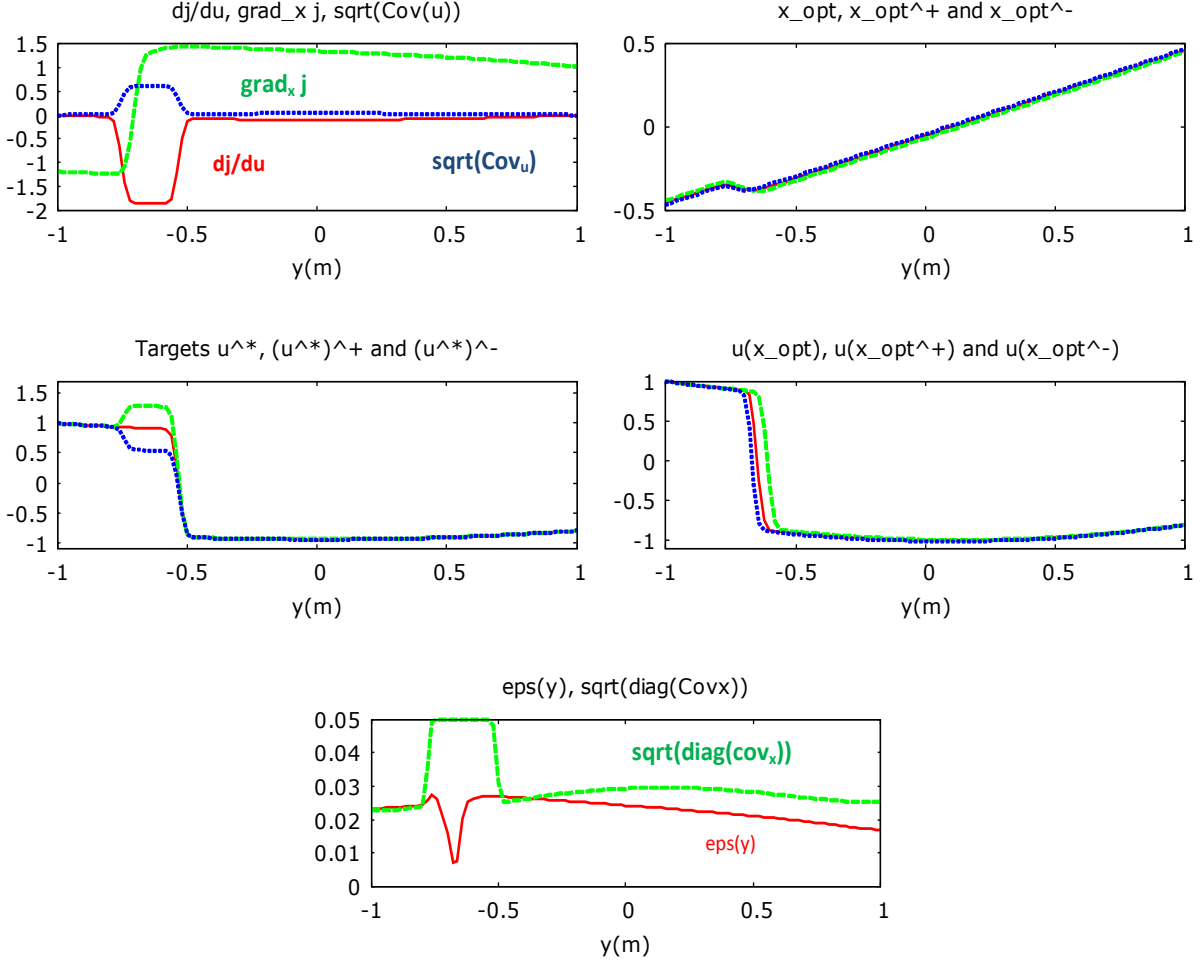
FIGURE 1. Incremental construction of two approximations of the covariance matrix of the control parameters from the covariance of the data for the inversion with the Burgers equation of section 10.

uncertainty in the data can produce on the inversion solution. We also compare the following matrix:

$$(20) \qquad Cov_{\mathbf{x}^{\pm}} = (X^{\pm})(X^{\pm})^t - (\overline{X^{\pm}})(\overline{X^{\pm}})^t,$$

to the construction using the adjoint operation in 6 and discussed in the next section. Figure 1 illustrates this incremental construction. In particular, it shows a comparison between the estimation of the impact of the data variability on the control parameters by (19) and $(diag(Cov_{\mathbf{x}}))^{1/2}$ by (21) given in the next section.

7.2. **Adjoint-based** $Cov_{\mathbf{x}}$. Let us compare the previous construction to the outcome of the linear analysis presented in section 6 where the covariance matrix of $\mathbf{x}$ is linked to the covariance matrix of u^*:

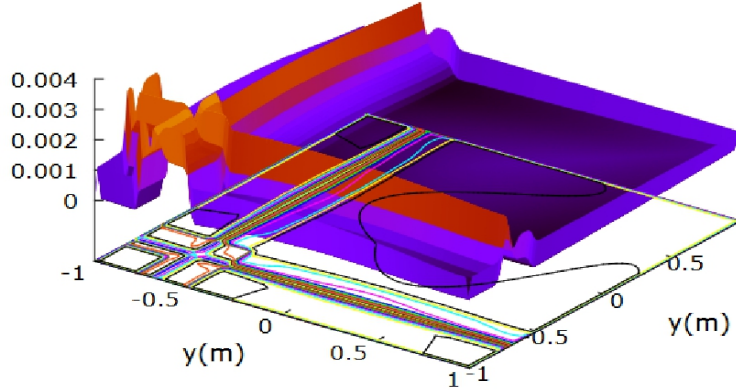$$(21) \qquad Cov_{\mathbf{x}} = \left(\mathcal{J}^t \, Cov_{u^*}^{-1} \, \mathcal{J}\right)^{-1},$$

FIGURE 2. $|Cov_{\mathbf{x}\pm} - Cov_{\mathbf{x}}|$ given by (20) and (21).

where $\mathcal{J}$ comes from (9) with $\partial j/\partial \mathbf{x} = 0$ and $\Pi = Id$:

$$\mathcal{J} = \left((\frac{\partial j}{\partial u})(\frac{\partial j}{\partial u})^t \right)^{-1} \frac{\partial j}{\partial u} \ (\nabla_{\mathbf{x}} j)^t,$$

with

$$\nabla_{\mathbf{x}} j = -\frac{1}{2T_\infty} v(y, t = 0) \quad \text{and} \quad \frac{\partial j}{\partial u}(y) = \frac{1}{2T_\infty} \int_0^{T_\infty} (u(\mathbf{x}(y), t) - u^*(y))dt.$$

Figure 2 shows $|Cov_{\mathbf{x}\pm} - Cov_{\mathbf{x}}|$ measuring the deviation between (20) and (21). One can see from pictures 1 and 2 that the bound information from (20) gives reasonable estimation of the variability for the control parameters away from shock regions where the level of the uncertainty appears being well predicted. The two approaches are complementary. The extreme scenarios approach providing fast estimation of the variability in regions where the solution is smooth.

8. Full aircraft shape optimization

Let us now apply the concepts presented in the previous sections in a context of shape optimization of an aircraft in transonic cruise condition. Several sources of variability exist, for instance, due to a change in the weight of the aircraft during the flight because of fuel consumption or due to variability in the flight conditions.

In the sequel we consider an aircraft flying at a Mach number of 0.85 and zero inflow and sideslip incidences. These parameters fully describe a 3D inviscid flow around the aircraft.

8.1. **Inverse design.** Inverse shape design aims at finding a shape realizing a given target state minimizing a functional of the form:

$$(22) \qquad j(\mathbf{x}, p^*) = \frac{1}{2}\|\Pi p(\mathbf{x}) - p^*\|^2,$$

where p denotes the pressure distribution over the shape of the aircraft and Π indicates where target pressure data are available. We can therefore use what has been presented to give uncertainty estimation on $\mathbf{x}$ from known uncertainty on target pressure data p^*. Of course, other variables can be considered instead of the pressure distribution.

In the sequel, we would like to go beyond inverse design and use our analysis in a wider shape optimization framework. Let us start describing our platform.

8.2. **A shape optimization platform.** We work in the framework of an existing shape optimization platform which has been previously presented in [31, 30, 37]. We use, in particular, three of its simulation codes for the shape parameterization, for the fluid dynamic calculations and for the shape sensitivity analysis of aerodynamic coefficients. This is a very standard and generic situation and these particular choices are not central to our discussion.

Let us briefly recall our direct dependency chain linking independent variables $\mathbf{x}$ to dependent variables (q, U) describing geometrical entities and state variables and to the cost function C_d and constraints $C_{i=1,...,4}$:

$$(23) \qquad \mathbf{x} \to q(\mathbf{x}) \to U(q(\mathbf{x})) \to (C_d, C_{i=1,...,4})(\mathbf{x}, q(\mathbf{x}), U(q(\mathbf{x}))).$$

8.2.1. *Shape parameterization and geometrical entities.* In (23) $\mathbf{x}$ denotes a CAD-free parameterization [37, 38] which does not require a priori local regularity assumptions on the shape as it is implicitly the case in Computer Aided Design (CAD) based shape definitions. More precisely, $\mathbf{x}$ represents shape deformations along the normal to the triangular faces of the surface mesh as shown in figure 3. For the problem discussed here this search space has a dimension $n \sim 16000$. This is very challenging, especially when addressing backward uncertainty propagation and defining the covariance matrix $Cov_{\mathbf{x}}$ in section 8.4.

This parameterization receives different denominations and belongs to the same class as node-based or free-form shape definitions. In all these approaches the regularity of the deformation needs to be monitored [29, 37]. This parameterization is intermediate in term of generality between CAD based definitions of shapes and fully free topological optimization choices where both the regularity and topology of the shape free. It is also interesting because it permits non intuitive and bio-inspired shapes to be introduced not necessarily compatible with the initial CAD definition of the geometry. An example of shape deformation produced by our optimization procedure is shown in figure 5.

Need for regularity control comes from the fact that, unlike with a CAD definition, the shape $\partial\Omega$ of an object Ω and a gradient-based deformation do not belong to the same function space in terms of regularity and, actually, the second is always less regular. Let us illustrate this on a simple example with $J(\mathbf{x}) = \|A\mathbf{x} - b\|^2$ taking $\mathbf{x} \in H^1(\partial\Omega)$, $A\mathbf{x}$ and b in $L^2(\partial\Omega)$. The gradient $J'_{\mathbf{x}} = 2A^T(A\mathbf{x} - b)$ belongs to $H^{-1}(\partial\Omega)$. Therefore, any variation along $J'_{\mathbf{x}}$ will have less regularity than $\mathbf{x}$: $\delta\mathbf{x} = -\rho J'_{\mathbf{x}} = -\rho(2(A\mathbf{x} - b)A) \in H^{-1}(\partial\Omega)$. We therefore need to project (or filter or smooth) into $H^1(\partial\Omega)$. But, why this is not the case with a CAD parameterization? Suppose the shape is described in a finite dimensional parameter space, as for instance with a polynomial definition of a surface (this is like a CAD parameterization). When we consider as control parameters the coefficients of the polynomial, changes in those do not change the regularity as the new shape will always belong to the same polynomial space. Sobolev inclusions give the key for the choice of the regularity operator with the CAD-free parameterization [37]. In our case, because we are using a piecewise linear discretization, a second-order elliptic system is sufficient. This operator is also used to monitor the regularity of the wing leading edge in order to avoid sharp edges.
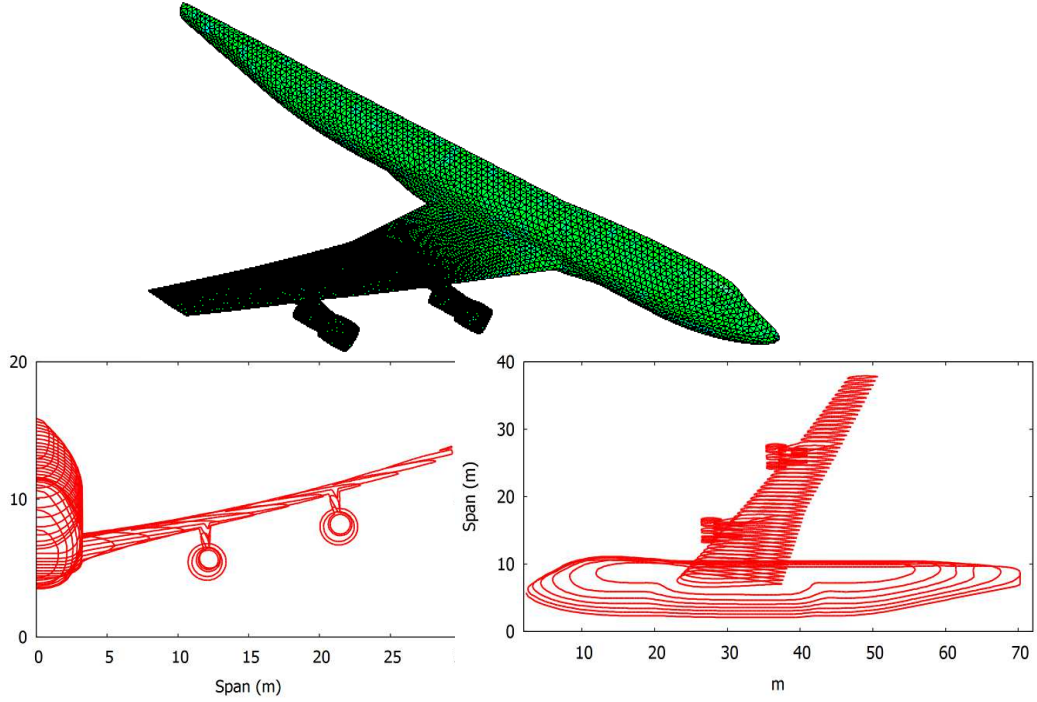
FIGURE 3. CAD-free shape parameterization (upper) and by-section definitions of the shape for geometric constraints enforcement.

$q(\mathbf{x})$ denotes the auxiliary unstructured mesh related geometrical quantities. When the shape is modified, this change must be propagated through the mesh keeping it admissible and we need to recalculate all related geometrical quantities. Admissible and positive mesh deformation is achieved by a 3D torsional spring analogy method [11].

8.2.2. *Flow solver.* In (23) $U(q(\mathbf{x}))$ denotes flow variables. More precisely, $U(q(\mathbf{x})) = (\rho, \rho\vec{u}, \rho E)^t$ represents the conservative flow variables solution of the Euler equations in conservation form where, T being the temperature, the total energy is given by $E = C_v T + \|\vec{u}\|^2/2$ and the pressure by the state law $p = \rho RT$ with R the perfect gas constant.

The details of the implementation of the flow solver are available in [37]. It is based on a finite volume Galerkin method on unstructured tetrahedral meshes [8]. Of course, other choices are possible for the flow solver and the literature on numerical methods for compressible flows is huge. This is not central to our discussion. We target steady solutions and use time marching with local time steps to reach these. The local time steps are proportional to $h/(\|\vec{u}\|+c)$ with h the local mesh size and c the local speed of sound. The time integration procedure is explicit and is based on a low-storage Runge-Kutta scheme. To illustrate the level of convergence to the steady state we use during optimization a typical convergence history is shown in figure 4. This means that, to increase the computational efficiency, we only use partial convergence for the state equations. In particular, the sufficient level of convergence retained for the aerodynamic coefficients is when the flow solver iterations only

modify their third digits. This is achieved with about 100 RK iterations starting from a uniform solution distribution. During optimization a new calculation for a new shape is always initialized with the previously available solution reducing even more this number (we proceed with typically 20 RK iterations).
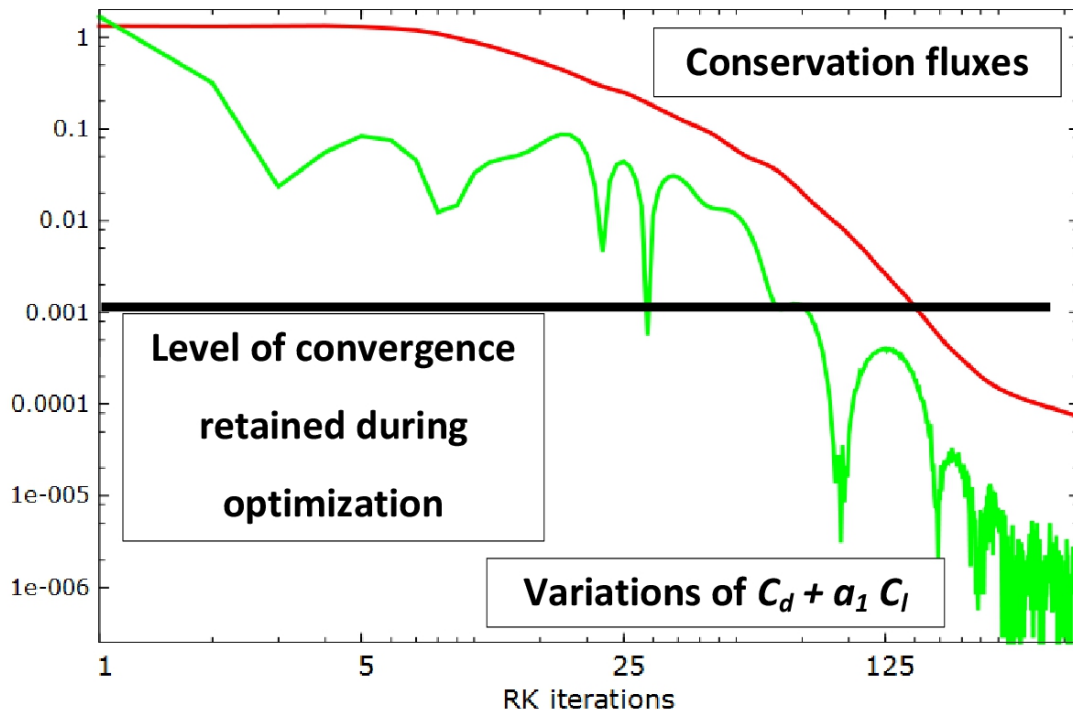

FIGURE 4. Normalized histories during time marching to steady state of $\|\partial_t U\|$ (U being the conservation variables) and $|\partial_t(C_d + a_1 C_l)|$.

The mesh used here has around 200K nodes which is obviously insufficient to reach full mesh resolution for the flow around an aircraft, even in a context of inviscid modeling. As mentioned in the introduction, we should consider that in practice our modeling capability and our computational resources will always be limited. The reverse uncertainty propagation is also a way to quantify the impact of this lack of resolution on the design.

8.2.3. *Optimization problem.* We consider a classical aerodynamic problem where two main quantities of interest are the drag C_d and lift C_l coefficients:

$$(24) \qquad C_d(\mathbf{x}) = \frac{1}{2\rho_\infty \|\vec{u}_\infty\|^2} \int_{shape(\mathbf{x})} p(q(\mathbf{x}))(\vec{u}_\infty.\vec{n}(q(\mathbf{x}))d\gamma,$$

where superscript ∞ indicates inflow conditions. The lift coefficient is evaluated with formula (24) where u_∞ is replaced by $\vec{u}_\infty^\perp$ in the boundary integral. Aircraft performance analysis concerns its payload and range. These are directly linked to the aerodynamic coefficients of the aircraft called the lift (conditioning the payload) and drag (conditioning the fuel consumption) coefficients. The lift coefficient often appears through an inequality $C_l - C_l^{target} \geq 0$ or equality constraint $C_1 = |C_l^{target} - C_l(p(q(\mathbf{x})))|$ with C_l^{target} a target performance. Let us consider this second situation.
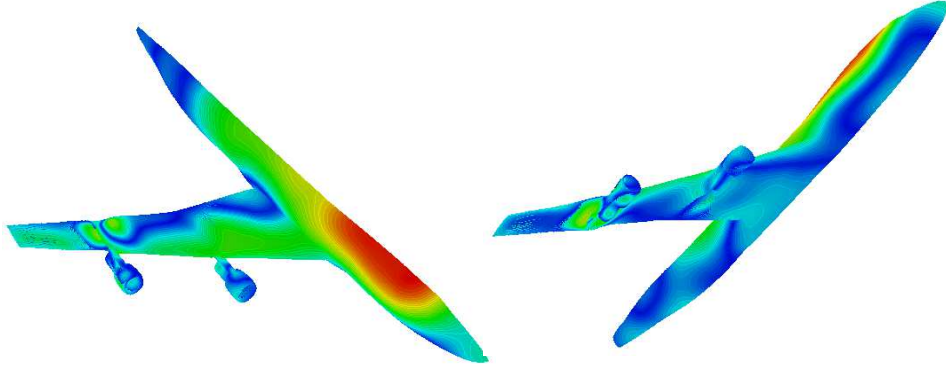
FIGURE 5. Shape deformation for a full aircraft optimization optimizing its payload and range under geometric constraints based on search direction $\nabla_{\mathbf{x}} C_d - < \nabla_{\mathbf{x}} C_d, \pi > \pi$ described in section 5.

Structural efficiency and necessity of useful free volume also implies the consideration of geometric criteria such as a constraint on the volume V of the aircraft or its by-section definition. As for the lift coefficient, this gives a constraint of the form $C_2 = |V^{target} - V(q(\mathbf{x}))|$. The volume of an object Ω (here the aircraft) is expressed through the boundary integral formula: $V = \int_\Omega 1 = \int_\Omega \frac{1}{3} \nabla.(\vec{X}) = \int_{\partial\Omega} \vec{X}.\vec{n}$, where $\vec{X} = (x_1, x_2, x_3)^t$ is the local coordinate over the shape.

The last geometric constraint concerns the local wing by-section thickness which is is prescribed. We define by-section (figure 3) definitions of the shape where the number of sections n_s is free (here $n_s = 50$) and can be adapted to account for the complexity of the geometry. Each node in the parameterization is associated to a section Σ_i, and for each section, we define the maximum thickness Δ_i. This last operation requires the projection of the upper-surface nodes over the lower surface for each section. This constraint is expressed as: $C_3 = \sum_{i=1}^{n_s} |\Delta_i(q(\mathbf{x})) - \Delta_i^{target}|$.

An alternative solution which is much simpler to implement is to only enforce a local volume constraint in each section Σ_i using the volume formula above: $V(\Sigma) = \int_{\Sigma_i} 1 = \int_\Omega 1 \; \chi_{\Sigma_i} = \int_\Omega \frac{1}{3} \nabla.(\vec{X}) \; \chi_{\Sigma_i} = \int_{\partial\Omega} \vec{X}.\vec{n} \; \chi_{\partial\Sigma_i}$, where χ is an indicator function ($\chi = 1$ if the point is in section Σ_i and $\chi = 0$ otherwise). Testing if a point is in Σ_i is easy and only requires an interval-based coordinate check, spanwise in this situation.

Let us add a data assimilation criteria to this multi-criteria optimization: $C_4 = \frac{1}{2}\|\Pi p(\mathbf{x}) - p^*\|^2$. We discuss in the sequel how to use this final expression to account for state variability in the design.

8.2.4. *Gradient of the functional and constraints.* The minimization algorithm we consider uses the gradients of functional and constraints with respect to the shape. These are also necessary for sensitivity reverse propagation as described in section 6. In our approach, all the sensitivities are computed by automatic differentiation (AD) in reverse mode using `tapenade` [17]. A brief description of AD in direct and reverse modes is given in appendix. This is how `tapenade` generates a computer program for $\nabla_{\mathbf{x}} j$ from the program computing $j(\mathbf{x})$.

Of course, the constraints can be accounted for by introducing a penalty term in the cost function: $j = C_d + \sum_{i=1,4} a_i C_i, \; a_i \in \mathbb{R}^+$. But this should be avoided when

possible. We will use it, however, below for the definition of the two directional extreme scenarios.

One classical technique to recover the lift during optimization is to change the flow incidence taking advantage of the linear relationship between the incidence and lift away from stall conditions. Suppose we do not want to use either penalty or such approximations. An alternative would be to consider a locally admissible gradient orthogonal to $\mathbf{A} = Span(\nabla_{\mathbf{x}} C_i, i = 1, ..., 4)$ with $\dim(\mathbf{A}) \leq 4$. Let us denote by π an orthonormal basis of this subspace obtained by the Gram-Schmidt procedure applied to the gradients of the constraints. The admissible gradient is given by $\nabla_{\mathbf{x}} C_d - < \nabla_{\mathbf{x}} C_d, \pi > \pi$, where $<, >$ indicates the scalar product over subspace $\mathbf{A}$. We need, therefore, to provide $\nabla_{\mathbf{x}} C_d$, $\nabla_{\mathbf{x}} C_l$, $\nabla_{\mathbf{x}} \|\Pi p(\mathbf{x}) - p^*\|^2$, $\nabla_{\mathbf{x}} V$ and $\nabla_{\mathbf{x}} \Delta_i$. The three former require the adjoint of the state equation and we take advantage of the capability for multi-right-hand-side adjoint calculation of `tapenade` in reverse mode to access these gradients without necessitating the solution of three separate adjoint problems. Our direct Euler code uses time marching to the steady solution with local time steps. An optimization of the reverse mode of AD comes from the fact that, our situations of interest being stationary in time, there is no interest in storing the forward states for backward integration [6, 37, 35].
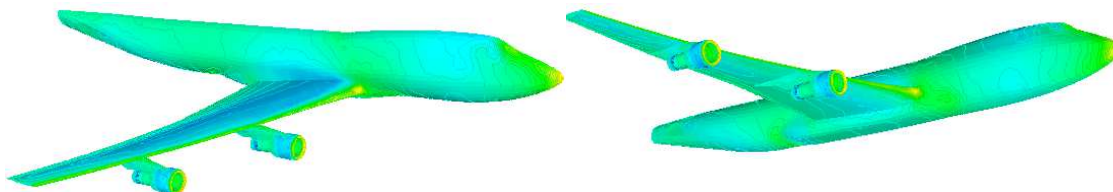

FIGURE 6. Pressure $p(q(\mathbf{x}^*))$ distribution over the shape which will be considered as the target pressure $p^*(\mathbf{x}^*)$ in reverse uncertainty propagation in section 8.3.

8.3. **Reverse propagation and DES.** Let us discuss how to use our reverse propagation techniques to quantify the uncertainty on $\mathbf{x}$. Suppose we have an optimal shape $\mathbf{x}^*$ provided by our optimization platform. We consider the pressure distribution over this shape as the target pressure p^*: $p^* = p^*(\mathbf{x}^*) = p(q(\mathbf{x}^*))$ which means that C_4 vanishes. This pressure distribution is shown in figure 6. To quantify the robustness of the design we want to see how much a perturbation of this pressure distribution, for instance due to external flow perturbations, impacts the design.

From now we follow the path of what we did in section 7.

8.3.1. *State variability:* The first step is to know or assume the distribution of the state variability through its local variance distribution $\sigma^2(\vec{X})$ at each point $\vec{X}$ on the shape. We can, as in section 7.1, assume that the target state uncertainty is proportional to local state variation measured, for instance, through the norm of the Hessian matrix H: $\sigma^2(\vec{X}) \sim \|H(p^*(\mathbf{x}^*))\|$. One can also assume that this variability is similar to the uncertainty one observes in practice (experimental or in flight). It is indeed well known that the flow distribution is quite stable in the cockpit and over the first and business class siting area where the flow is nearly potential.

On the other hand, flow variability increases spanwise (easy to see from wings tips motions) and also toward the tail of the aircraft (flying coach once makes this easy to understand). These are due, among others, to separated turbulent flows instabilities and fluid-structure interactions. These remarks permit to define a rough geometric distribution of $\sigma^2(\vec{X})$ over the shape as shown in figure 7.
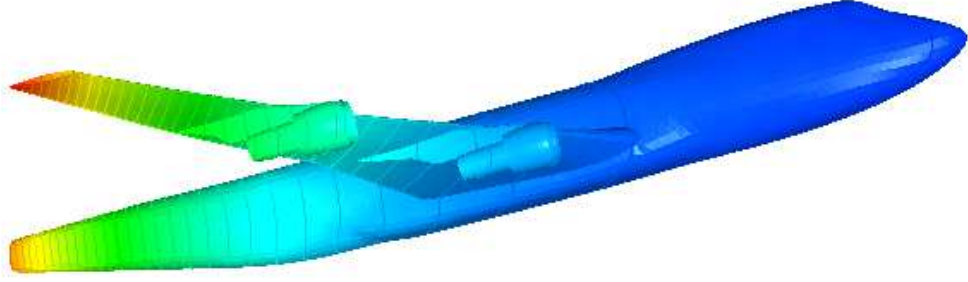

FIGURE 7. Variability field $\sigma^2(\vec{X})$ distribution for $\vec{X}$ on the shape.

8.3.2. *Target Directional Extreme Scenarios:* Consider a pressure distribution $p^* = p(\mathbf{x}^*)$ available everywhere on the shape (i.e. $\Pi = Id$ in (22)). The next step is to define the target extreme scenarios from p^* using (3). We need to define $\partial j / \partial p^*$. We consider here the weighted definition of $j = C_d + \sum_{i=1}^{4} a_i C_i$ (with $a_1 = 0.1$, $a_2 = a_3 = a_4 = 0$) involving only the drag and lift coefficients. C_4 is inactive and the two geometric constraints do not involve any state quantities. We have:

$$\frac{\partial j}{\partial p^*} = \frac{\partial C_d}{\partial p^*} + a_1 \frac{\partial C_l}{\partial p^*}.$$

From their definition in section 8.2.3 we have linearity with respect to the pressure of the aerodynamic coefficients. Linearizing in $p^* = p(\mathbf{x}^*)$ we have a distribution over the shape:

$$\frac{\partial j}{\partial p^*} = \frac{1}{2\rho_\infty \|\vec{u}_\infty\|^2} (\vec{u}_\infty + a_1 \vec{u}_\infty^\perp).\vec{n}.$$

Using this expression, we define two extreme target scenarios as:

(25) $$(p^*)^\pm(\vec{X}) = p^*(\vec{X}) \left(1 \pm 1.65 \; \sigma^2(\vec{X}) \; \frac{\partial j / \partial p^*}{\|\partial j / \partial p^*\|} \right),$$

and shown in figure 8. One can see that a same level of state uncertainty σ (from figure 7) does not produce the same impact on the target pressure $(p^*)^\pm$. This is related to the fact that $(\vec{u}_\infty + a_1 \vec{u}_\infty^\perp).\vec{n}$ is a geometric quantity decreasing, for instance, toward the tail of the aircraft.

8.3.3. *Inversion targeting the DES:.* As in section 7 our construction ends targeting these scenarios of pressure distributions looking for the shapes realizing them best. This is done considering our multi-criteria optimization problem described in 8.2.4 but this time with $C_4 = \frac{1}{2} \|p(\mathbf{x}) - (p^*)^\pm\|^2$ active and p^* is respectively taken as $(p^*)^+$ and $(p^*)^-$. These two optimizations give two solutions $(\mathbf{x}^*)^+$ and $(\mathbf{x}^*)^-$. Figure 9 shows $X^\pm = (\mathbf{x}^*)^+ - (\mathbf{x}^*)^-$ measuring the deviation between the two shapes.

Important deviations are predicted on the upper surface of the wing after the second engine and along the leading and trailing edges.
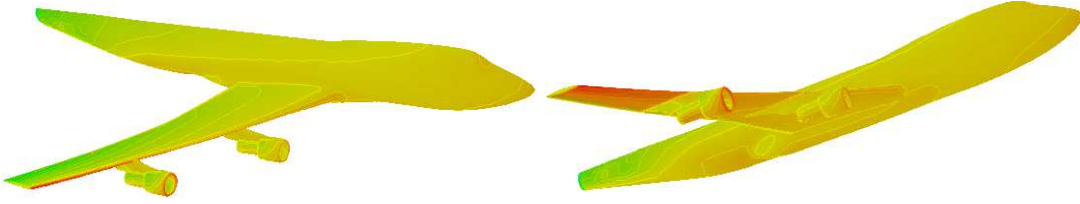

FIGURE 8. $p^*\sigma^2 \frac{\partial j/\partial p^*}{\|\partial j/\partial p^*\|}$ to build directional extreme scenarios with (25).
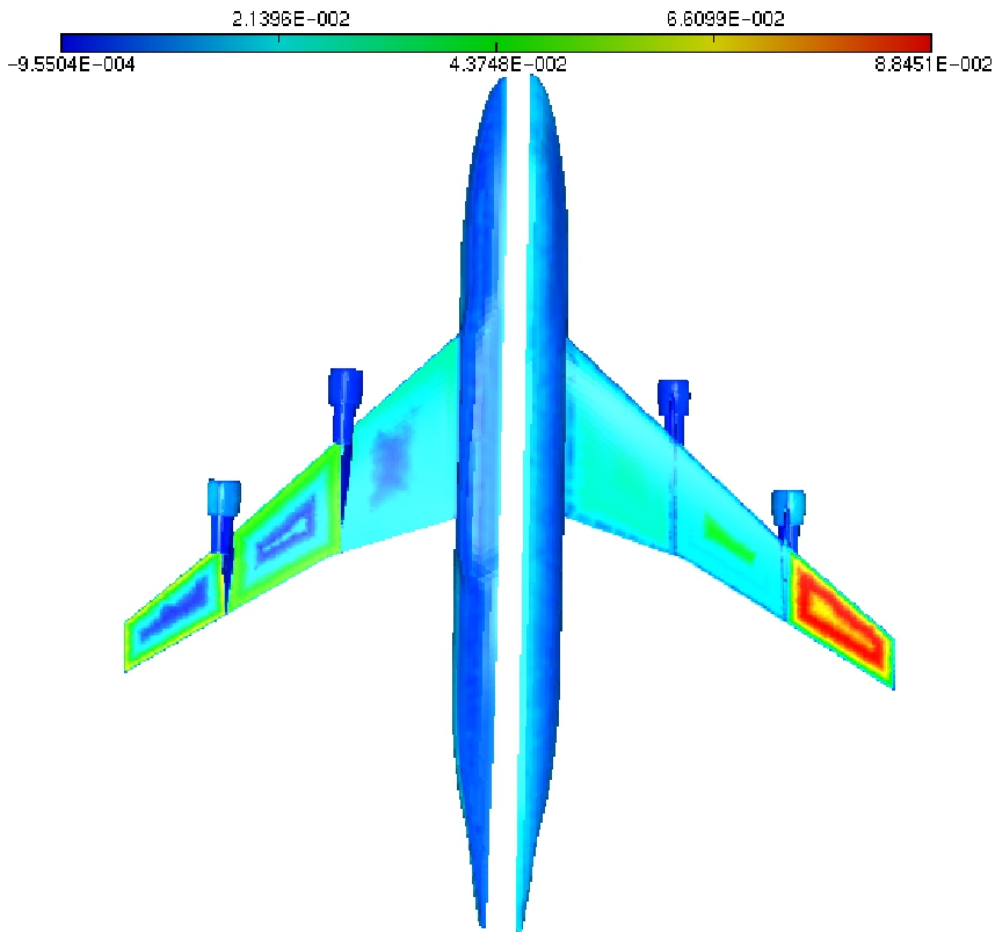

FIGURE 9. $X^{\pm} = (\mathbf{x}^*)^+ - (\mathbf{x}^*)^-$ (meter) on the lower (left) and upper (right) surfaces of the aircraft measuring the distance between two shapes after inversions targeting the two directional extreme scenarios pressure distributions (18).

8.4. **Reverse $Cov_{\mathbf{x}}$ construction.** As in section 7.2, let us compare the previous construction to the outcome of the linear analysis presented in section 6 to build an estimation of $Cov_{\mathbf{x}}$ from Cov_{p^*}. As mentioned in section 6 we assume p^* as
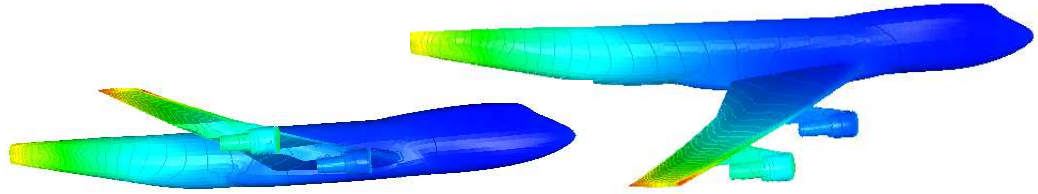
FIGURE 10. Assumption of $Cov_{p^*}(= p^*\sigma^2\ I)$ used in getting $Cov_{\mathbf{x}}$ from (26).
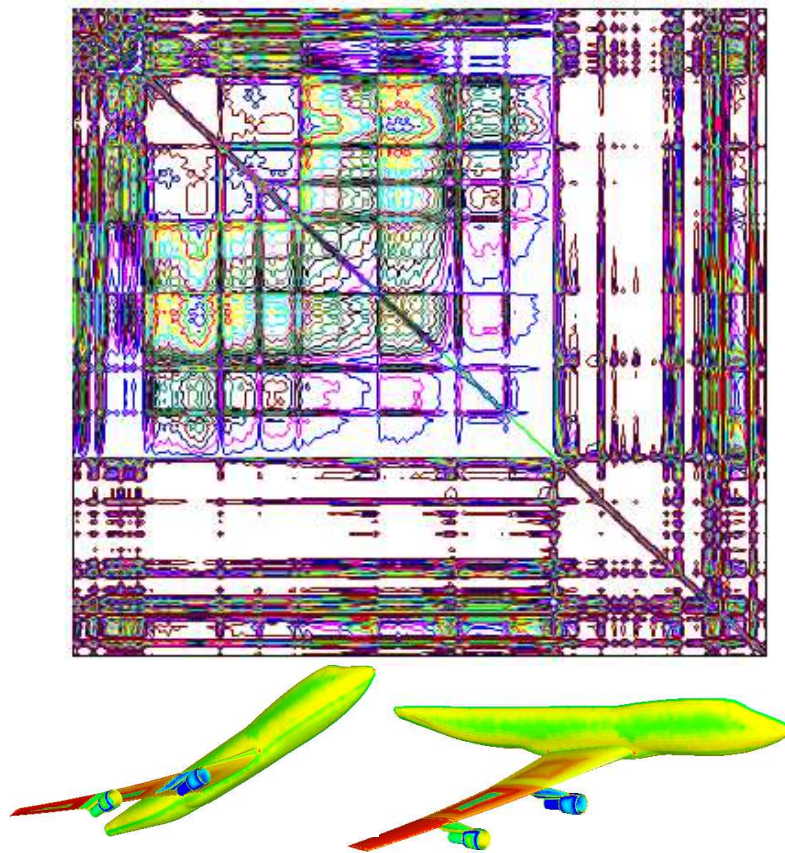

FIGURE 11. Graph of $Cov_{\mathbf{x}}$ from (26). This is a $n \times n = 16K \times 16K$ matrix. Three main regions of interaction (wing, wing-body, tail) can be identified. Iso-contours show an example of correlations in the design parameters between a point in the wing tip region and the rest of the aircraft (this is a line of $Cov_{\mathbf{x}}$).

independent measurements with a therefore diagonal covariance matrix $Cov_{p^*} = p^*\sigma^2\ I$ (shown in figure 10) with σ^2 shown in figure 7. We have:

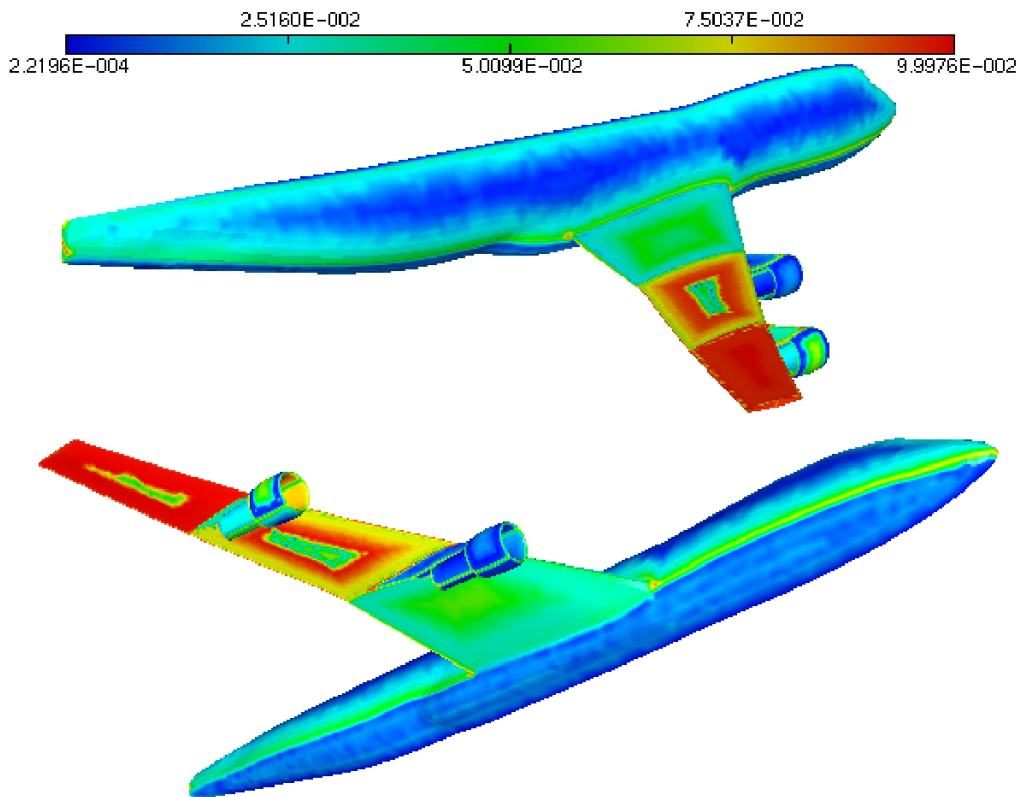$$(26) \qquad Cov_{\mathbf{x}} = \left(\mathcal{J}^t\ Cov_{p^*}^{-1}\ \mathcal{J}\right)^{-1},$$

FIGURE 12. $(diag(Cov_{\mathbf{x}}))^{1/2}$ (meter).

where $\mathcal{J}$ comes from (9) with $\partial j/\partial \mathbf{x} = 0$ as there is no direct dependency between $\mathbf{x}$ and C_d or $C_{i=1,...,4}$ and again with $\Pi = Id$ as p^* is available everywhere on the shape. The inversion in (26) is made, after putting the matrix $\mathcal{J}^t\,Cov_{p^*}^{-1}\,\mathcal{J}$ in L^tDL form, assuming it exists[1], following:

```
Covx(n,n) = 1/D(n,n);
For j = n-1:1
Covx(j+1:n,j) = -Covx(j+1:n,j+1:n)*L(j+1:n,j);
Covx(j,j+1:n) = transpose(Covx(j+1:n,j));
Covx(j,j) = 1/D(j,j) - transpose(L(j+1:n,j)) * Covx(j+1:n,j);
Done
```

If requested and avoiding a full inversion, one could access to selected elements of the matrix following [26].

The graph of the $n \times n = 16K \times 16K$ matrix $Cov_{\mathbf{x}}$ is given in figure 11. As we use unstructured meshes, two close nodes are not necessary close in the matrix and nodes in a region are not simple to locate on the graph. Three main regions of interactions can be identified: the wing, the wing-body and the tail regions. Main interactions are between the wing and the wing-body junction regions. As shown, picking a point and plotting the corresponding line in $Cov_{\mathbf{x}}$ illustrates the regions which will be impacted (due to state variability) by a change in the design of the shape of the

---

[1]We consider symmetric matrices A for which the following decomposition exists $A = L^tDL$ with $D_{jj} = A_{jj} - \sum_{k=1}^{j-1} L_{jk}^2 D_{kk}$ and $L_{i>j} = \frac{1}{D_{jj}} \left(A_{ij} - \sum_{k=1}^{j-1} L_{ik}L_{jk}D_{kk} \right)$.
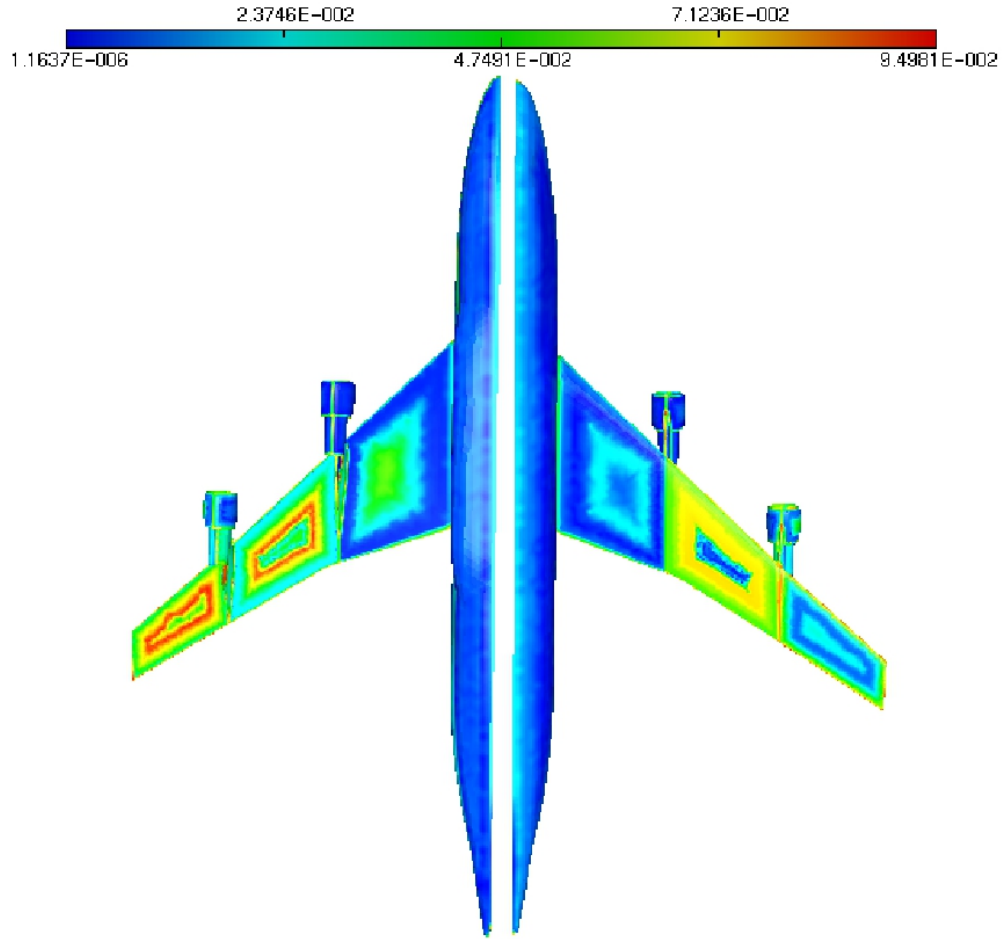
FIGURE 13. $|diag(Cov_{\mathbf{x}})^{1/2} - diag(Cov_{\mathbf{x}\pm})^{1/2}|$ (meter) on the lower (left) and upper (right) surfaces of the aircraft.

aircraft at this point. One sees, for instance, that the wing-body interaction goes beyond just the wing region.

Figure 12 shows $(diag(Cov_{\mathbf{x}}))^{1/2}$ (in meter) which gives the standard deviation of the design due to state variability. Together with figure 9 these are eventually the most important information. The uncertainty margin can reach 0.1 m in some regions. This is slightly superior to the values reached by the extreme scenarios analysis presented in figure 9. This construction features much richer correlations between design parameters. We notice that to increase their robustness in variable flight conditions (which means reducing the correlations) the two engines pylon, fairing and air intakes should be receiving different designs following their positions on the wing. Eventually, figure 13 shows $|diag(Cov_{\mathbf{x}})^{1/2} - diag(Cov_{\mathbf{x}\pm})^{1/2}|$ measuring the differences between the two constructions of the design standard deviation. Large differences exist in some regions between the two constructions which can be of the order of the maximum deviation predicted.

9. Concluding remarks

A low-complexity solution for reverse propagation of aleatory uncertain target data in inverse design has been presented. Two approximations of the covariance matrix of the optimization parameters have been discussed. These provide uncertainty quantification analysis for the inversion solution with confidence margins on the design parameters in very large design spaces. None of the approaches require a sampling of a parameter space and both take advantage of existing ingredients in deterministic gradient-based inversion algorithms. In both approaches the overall calculation complexities remain comparable to the deterministic situation.

The constructions also permit to account for epistemic uncertainties considering a model or solution procedure always imperfect. Hence, seeing the associated error as uncertainty these reverse propagations provide a quantification of the impact of these weaknesses on the design.

Most importantly the ingredients of the approach are simple to explain and do not require a deep mathematical background. Something which is a limitation today to the progress of uncertainty analysis when using high fidelity tools. Indeed, because of their computational complexity, simple statistical approaches are difficult to directly deploy in the framework of high fidelity simulations and more intrusive approaches require major reformulation of existing models and heavy extra coding.

The application of the method to the optimization of the shape of an aircraft indicates that the optimization problem should be restated in order for the design to gain in robustness. For instance, the covariance distribution over the shape of the aircraft shows that for the design to be robust in variable flight conditions the engines pylon, fairing and air intakes should have different shapes following their position on the wing. Our current effort is to account for this extra constraint in the optimization problem.

References

[1] AIAA. Guide for the verification and validation of computational fluid dynamics simulations. *AIAA*, G-077, 1998.

[2] B. Anderson and J. Moore. *Optimal Filtering.* Prentice-Hall, NY, 1979.

[3] J. R. Bunch and L. Kaufman. Some stable methods for calculating inertia and solving symmetric linear systems. *Mathematics of Computation*, 31(137):163–179, 1997.

[4] J. Caers. *Modeling Uncertainty in the Earth Sciences.* Wiley-Blackwell, 2011.

[5] G. Casella and R. Berger. *Statistical Inference, 2nd Ed.* Duxbury Press, London, 2001.

[6] B. Christianson. Reverse accumulation and implicit functions. *Optimization Methods and Software*, 9/4:307–322, 1998.

[7] P. Courrieu. Fast computation of moore-penrose inverse matrices. *CoRR*, abs/0804.4809, 2008.

[8] A. Dervieux. Steady euler simulations using unstructured meshes. *VKI Lecture series, Revised version published in Partial Differential Equations of hyperbolique type and Applications, World Scientic*, 1985/04:23–64, 1985.

[9] G. Evensen. Advanced data assimilation for strongly nonlinear dynamics. *Monthly Weather Review*, 125:1342–1354, 1997.

[10] G. Evensen. *Sequential Data Assimilation for Nonlinear Dynamics: The Ensemble Kalman Filter In Ocean Forecasting: Conceptual basis and applications.* Springer-Verlag, Heidelberg, 2002.

[11] C. Farhat and C. Degand. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computers & Structures*, 80/3:305–316, 2002.

[12] A. Gelb. *Stochastic Processes and Filtering Theory.* Academic Press, NY, 1970.

[13] A. Gelb. *Applied Optimal Estimation.* M.I.T Press, Boston, 1974.

[14] R. Ghanem and A. Doostan. On the construction and analysis of stochastic models: characterization and propagation of the errors associated with limited data,. *J. of Comput. Phys.*, 217:63–81, 2006.

[15] R. Ghanem and P. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer Verlag, New York, 1991.

[16] MA. Giles and NA. Pierce. Analytic adjoint solutions for the quasi-one-dimensional euler equations. *Journal of Fluid Mechanics*, 426:327–345, 2001.

[17] L. Hascoet and V. Pascual. Tapenade user's guide. In *INRIA Technical report*, pages 1–31. INRIA, 2004.

[18] P.G. Hoel. *Introduction to Mathematical Statistics*. John Wiley, London, 1971.

[19] B. Ivorra and B. Mohammadi. Optimization strategies in credit portfolio management. *Journal of Global Optimization*, 43/2:415–427, 2009.

[20] M. Peyret J. Chery, B. Mohammadi and C. Joulain. Plate rigidity inversion in southern california using interseismic gps velocity field. *Geophys. J. Int.*, 187/2:783–796, 2011.

[21] R. O. Onez J. T. Spooner, M. Maggiore and K.M. Passino. *Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques*. John Wiley, New York, 2002.

[22] Ph. Jorion. *Value at Risk: The New Benchmark for Managing Financial Risk*. McGraw-Hill, New York, 2006.

[23] M. Ghil K. Ide, P. Courtier and A. Lorenc. Unified notation for data assimilation: operational, sequential and variational. *Journal of the Meteorological Society of Japan*, 75/1B:181–189, 1997.

[24] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82:35–45, 160.

[25] K.C. Kapur and L.R. Lamberson. *Reliability in Engineering Design*. John Wiley & Sons, New York, 1997.

[26] Lin Lin, Chao Yang, Juan C. Meza, Jianfeng Lu, Lexing Ying, and Weinan E. Selinv—an algorithm for selected inversion of a sparse symmetric matrix. *ACM Trans. Math. Softw.*, 37(4):40:1–40:19, February 2011.

[27] H.R. Lindman. *Analysis of Variance in Complex Experimental Designs*. Freeman, New York, 1974.

[28] A. Dywayne A. Nicely M. Abdou, R.B. Morgan and W. Wilcox. Deflated and restarted symmetric lanczos methods for eigenvalues and linear equations with multiple right-hand sides. *SIAM Journal on Scientific Computing*, 32/1:129–149, 2010.

[29] R. Wuchner M. Firl and K. Bletzinger. Regularization of shape optimization problems using fe-based parametrization. *Structural and Multidisciplinary Optimization*, 47/4:507–521, 2013.

[30] B. Mohammadi. Reduced sampling and incomplete sensitivity for low-complexity robust parametric optimization. *Int. J. Num. Meth. Fluids*, 73/4:307–323, 2013.

[31] B. Mohammadi. Principal angles between subspaces and reduced order modeling accuracy in optimization. *Structural and Multidisciplinary Optimization*, 50/2:237–252, 2014.

[32] B. Mohammadi. Uncertainty quantification by geometric characterization of sensitivity spaces. *Compt. Meth. Appl. Mech. Eng.*, 280:197–221, 2014.

[33] B. Mohammadi. Value at risk for confidence level quantifications in robust engineering optimization. *Optimal Control: Applications and Methods*, 35/2:179–190, 2014.

[34] B. Mohammadi. Ensemble kalman filters (enkf) and geometric characterization of sensitivity spaces for uncertainty quantification in optimization. *Computer Methods in Applied Mech. & Eng.*, 290:228–249, 2015.

[35] B. Mohammadi. Parallel reverse time integration and reduced order models. *J. of Computational Mathematics*, 2:17–33, 2015.

[36] B. Mohammadi and F. Bouchette. Extreme scenarios for the evolution of a soft bed interacting with a fluid using the value at risk of the bed characteristics. *Computers & Fluids.*, 89:22–46, 2014.

[37] B. Mohammadi and O.Pironneau. *Applied Shape Optimization for Fluids (2nd Edition)*. Oxford Univ. Press, Oxford, 2009.

[38] B. Mohammadi and O. Pironneau. Shape optimization in fluid mechanics. *Annual Revue of Fluid Mechanics*, 36/1:255–279, 2004.

[39] P. Redont and B. Mohammadi. Improving the identification of general pareto fronts by global optimization. *CRAS*, 347:1–8, 2009.

[40] A. Tarantola. *Inverse problem theory and methods for model parameter estimation*. SIAM, N.Y., 1987.

[41] A. Curioni V. Kalantzis, C. Bekas and E. Gallopoulos. Accelerating data uncertainty quantification by solving linear systems with multiple right-hand sides. *Numerical Algorithms*, 62/2:637–653, 2014.

[42] X. Wan and G.E. Karniadakis. Multi-element generalized polynomial chaos for arbitrary probability measures. *SIAM J. Sci. Comput.*, 28/3:901–928, 2006.

[43] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. ISBN 0-387-40272-1. Springer, 2004.

[44] D. Xiu. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press, 2010.

10. Appendix: Automatic Differentiation of computer codes

We would like to give a brief description of automatic or algorithmic differentiation methods which permit to compute derivatives in discret level from a computer code linking the independent variables to the functional.

Consider the problem of finding $j'(x)$ when $j(x)$ is given by a computer program.

10.1. **The direct mode of AD.** Because the program is made of differentiable lines, j' can be computed by differentiating every line and adding them to the computer program immediately above each line. For instance,

$$
\begin{array}{ll}
\textbf{Program for } j\ . & \textbf{Lines to add} \\
u = (1+x)*\log(x) & du = (1+x)*dx/x + \log(x) \\
z = u + \cos(x) & dz = du - \sin(x)*dx \\
j = u*z & dj = du*z + u*dz.
\end{array}
$$

If this new program is run with `x=x0, dx=1, du=0, dz=0, dj=0`, then dj is the derivative of j with respect to x at $x0$. This is called the direct mode of AD.

10.2. **The reverse mode of AD.** The reverse mode of AD is similar to the continuous adjoint method and aims to provide the gradient with a cost independent of the number of variables in the program. Let us interpret this mode introducing the Lagrangian of the code above by associating to each variable in the program a dual variable p, except for the last line for which $p = 1$ (each line of a computer code is seen as an equality constraint and the final line as the cost function):

$$
(27) \qquad L = p_1[u - (1+x)\log(x)] + p_2[z - u - \cos(x)] + j - uz
$$

Stationarity with respect to intermediate variables in reverse order (z, u) gives

$$
\begin{aligned}
\frac{\partial L}{\partial z} &= 0 = u + p_2 \\
\frac{\partial L}{\partial u} &= 0 = z - p_2 \cos(x) + p_1.
\end{aligned}
$$

This gives p_2 first, and then p_1, and then dj/dx is

$$
j' = \frac{\partial L}{\partial x} = p_2 u \sin(x) - p_1(\log(x) + \frac{1+x}{x}).
$$

This is different from the direct mode in term of complexity because whatever the number of independent variables, the adjoint variables p_i are evaluated only once. A powerful technique to avoid the Lagrange method is to use reverse accumulation and this is how `Tapenade` works. More precisely, for each assignment $y = y + f(x)$, the dual expression is $p_x = p_x + f'p_y$ with p_x and p_y the dual variables associated to x and y. Hence, for $(p_x = 0, p_y = 1)$ as initialization, this gives $p_x = f'$. The previous example becomes

$$p_u = p_z = p_x = 0, \ p_j = 1,$$
$$p_u = p_u + zp_j, \ p_z = p_z + up_j,$$
$$p_u = p_u + p_z, \ p_x = p_x - \sin(x)p_z,$$
$$j' = p_x + (\log(x) + \frac{1+x}{x})p_u.$$

This approach can be used to directly write 'by hand' the adjoint code. This can also be seen as an alternative to deriving the continuous adjoint and programming it.