Comparison of Gradient and Response Surface Based Optimization Frameworks Using Adjoint Method

J. Laurenceau *, M. Meaux †

Airbus, Toulouse, 31060, France

This paper deals with aerodynamic shape optimization using an high fidelity solver. Due to the computational cost and restitution time needed to solve the RANS equations, this type of optimization framework must improve the solution using very few objective function evaluations despite the high number of design variables. The choice of the optimizer is thus largely based on its speed of convergence. The quickest optimization algorithms use gradient information to converge along a descent path departing from the baseline shape to a local optimum. Within the past few decades, numerous design problems were successfully solved using this method. In our framework, the reference algorithm uses a quasi-Newton gradient method and an adjoint method to inexpensively compute the sensitivities of the functions with respect to shape variables. As usual aerodynamic functions show numerous local optima when varying shape, a more global optimizer can be beneficial at the cost of more function evaluations. More recently, the use of expensive global optimizers became possible by implementing response surfaces between optimizer and CFD code. In this way, a Kriging based optimizer is described. This optimizer proceeds in iteratively refining at up to three points per iteration by using a balancing between function minimization and error minimization. It is compared to the reference algorithm on two drag minimization problems. The test cases are 2D and 3D lifting bodies parameterized with six to more than forty design variables driving deformation of meshes with Hicks-Henne bumps. The new optimizer effectively proves to converge to lower function values without prohibitively increasing the cost. However, response surfaces are known to become inefficient when dimension increases. In order to efficiently apply this response surface based optimizer on such problems, a Cokriging method is used to interpolate gradient information at sample locations.

Nomenclature

$C(x)$	Sampling refinement criterion
Cd	Drag coefficient
Cl	Lift coefficient
Cp	Pressure coefficient
c	Mean chord length
$\mathcal{D}$	Domain of design variables
F_s	Exact function at samples $[N]$
$F(x)$	Objective function
$H(x)$	Hessian matrix of the objective function
L	Likelihood estimate
N	Order of the correlation matrix / quantity of information
n_{aug}	Number of gradient augmented samples
n_{dv}	Number of design variables
n_{eval}	Number of function evaluations
n_{grad}	Number of gradient evaluations
n_{iter}	Number of iterations of the optimization process
n_{pop}	Optimizer population size
n_s	Number of samples
$\mathcal{P}$	Set of possible optima on the response surface

*Ph.D. Student, Aerodynamics department, Airbus France, 316 route de Bayonne.
†Engineer, Aerodynamics department, Airbus France, 316 route de Bayonne.

R	Correlation matrix $[N \times N]$
$r(x)$	Correlation vector $[N]$
$\mathcal{S}$	Domain of sample points
$SCF(.,.)$	Spatial correlation function
$S(x)$	Standard error
s^i	i-th sample
x	Vector of design variables $[n_{dv}]$
$Z(x)$	Stochastic process
β	Zero order regression model
σ^2	Model variance
θ	SCF correlation coefficients $[n_{dv}]$
$\|\|.\|\|$	Euclidian norm; 2-norm

Subscripts

i	$\in [1, n_s]$
j	$\in [1, n_s]$
k	$\in [1, n_{iter}]$
c	$\in [1, n_{eval}]$
r	$\in [1, n_{pop}]$
v	$\in [1, n_{dv}]$

Superscripts

ini	initial sampling database
$\hat{.}$	approximated value
ref	current best value

I. Introduction

In the field of aerodynamic aircraft design, the functions studied are very sensitive to small changes on the shape and it is then particularly hard for designers to reach an optimal solution by trial-and-error. Shape optimization tools are thus particularly favoured by aerodynamicians. These tools are completely automatic process capable of running by themselves computer expensive numerical simulation given some degrees of freedom on the geometry and a figure of merit qualifying the performance of each shape. In the context of detailed design, it was decided that the same level of fidelity of the analysis code should be used for absolute performance assessment and for shape optimization purpose. As computational cost of CFD problems generally increases with computational resources (size of meshes can be adpated to available computational power), the typical restitution time required for one flow anaylsis cannot be sufficiently decreased to enable the use of expensive global optimization algorithms requiring thousands objective function evaluations such as genetic algorithms. This drawback can be circumvent by using the shape given by studies performed during conceptual and preliminary design phases as a starting point for a gradient based local optimization. The first aerodynamic shape optimizations by Hicks et al.[1] demonstrate the efficiency of this type of process, still in use nowadays[2–5] in combination with efficient gradient calculation techniques based on adjoint method maintaining the cost of the optimization independent of the number of design variables.

Despite its speed of convergence, gradient based algorithms are known to lack design space exploration and are easily trapped by local optima. As the non-linear physical phenomena occuring in transonic flows imply numerous local optima on aerodynamic functions (drag, lift, momentum), a significant gain is expected by the use of more global optimizers. The use of such optimizers for applications driven by analysis of full Navier-Stokes equations was made possible more recently by the use of surrogate models (or response surfaces)[6–9] approximating the expensive CFD function by an inexpensive to evaluate black-box model. Moreover, the use of response surfaces coupled with a multi-objective global optimizer enables to have a clearer view of possibles trade-offs between concurrent objective functions[10] and facilitates multidisciplinary analysis,[11,12] whereas gradient based algorithms are limited to mono-objective problems and solved multi-objective problems only by using pre-determined equivalence coefficients between functions.

In the section II of this paper is presented the high-fidelity optimization suite OPTaliA and the gradient based optimizer *DOT*. This algorithm uses a classical quasi-Newton method and is taken as the

reference optimizer. Numerous application were successfully conducted using this optimization framework,[5,12] but to tackle the limitations underlined previously a response surface based optimizer was developped using a Kriging method and is presented in section III. The performance of this optimizer should largely exceed the reference on low-dimensional problems by using a multi-criteria refinement process enabling to run multiples CFD runs in parallel. As surrogate models are known to become inefficient as dimension increased, a third optimizer using a Cokriging (gradient-enhanced Kriging) formulation is presented. It uses a sample limited Cokriging approach. This formulation was set up to overcome the large computational cost needed to build a surrogate model interpolating high dimensional gradient vectors.

The last section (section IV), compares the two new response surface based optimizers to the gradient reference on a low-dimensional drag reduction problem considering 6 design variables on a RAE2822 airfoil and a high dimensional test problem considering 45 design variables on a wing.

II. Optimization suite

The software OPTaliA, internally developed at Airbus, is used to perform aerodynamic shape optimization. This high-fidelity optimization suite can improve aerodynamic performance of an aircraft by changes in the external shape (planform variables fixed) and is adapted to the work done during the detailed design phases.

A. Common optimization framework

A general optimization framework, represented in Figure 1, has been set up in OPTaliA, in order to implement various type of optimizers (gradient, genetic, response surface). From the global point of view, the optimization process can be interpreted as a succession of two main tasks: evaluation and optimization. Within the evaluator, the function value and if needed the gradient value corresponding to n_{pop} shapes are computed. The population size depends on the optimization algorithm chosen, but if it is superior to one the evaluator performs the simulations of all shapes simultaneously by running multiple jobs on high performance computers. A large population can reduce the restitution time of the optimization process, at the risk of saturating computational resources. Once all shapes have been evaluated, the optimizer proposes a new population of shapes by using the new information on functions and the next iteration begins. In addition to optimizer internal stopping criteria, the convergence is

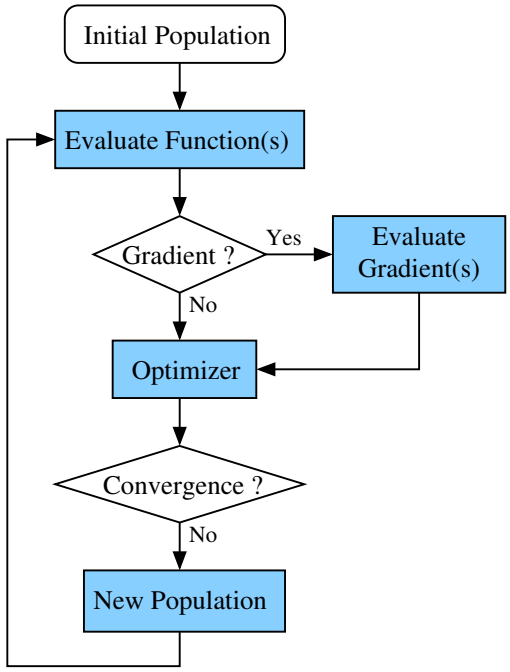

Figure 1. General Optimization Process

forced at the OPTaliA level when the number of iterations or the number of function evaluations exceeds a given threshold, $n_{iter} \leq 100, n_{eval} \leq 200$.

One of the challenges in aerodynamic shape optimization is to manage running efficiently evaluator

and optimizer automatically in batch mode. More particularly, the evaluator itself is a complex process requiring large computational resources.

B. Evaluator for CFD functions

1. *Shape parameterization and mesh deformation*

The shape parameterization consists in applying Hicks-Henne sinusoidal bumps on a surface skin of an initial block-structured mesh. Each bump is defined by three shape variables driving the amplitude, the position and the width expansion. The direction of the deformation can be either along the vector locally normal to the surface or along a fixed vector (vertical axis). This type of deformation was initially developed by Hicks et al.[1] for numerical optimization of airfoils. When applied on a bidimensional surface corresponding to a three dimensional shape, a linear propagation of the bump is done in the second direction using fixed propagation distances.

Once computed, the vector field of deformation at the surface skin is propagated to the volume mesh using a mixed integral / transfinite interpolation method.[5] The integral method is used to compute deformation of nodes defining boundaries between blocks and then the transfinite interpolation computes deformation inside each block in parallel.

An analytical linearization of the shape parameterization and mesh deformation modules enables to inexpensively compute (in terms of CPU time) the sensitivity of the surface mesh and the sensitivity of the volume mesh with respect to design variables.

2. *Flow simulation*

Flow analysis were performed with the *elsA*[13] software developed by Onera. The flow is simulated by solving the Reynolds Averaged Navier-Stokes (RANS) equations associated with the one-equation Spalart-Allmaras turbulence model on block structured meshes using a cell-centered finite volume approach. The second order Roe's upwind scheme with the Van Albada limiter is used as spatial scheme coupled with an implicit time resolution. Multigrid and local time stepping techniques are used to converge more quickly.

One of the main requirement from designers is to obtain the same results when using the CFD solver inside or outside the automatic optimization tool. As hysterisis phenomena are common when dealing with transonic flows, it imposes that the same initial flow condition (uniform flow) are used for all simulations during the optimization. So, the computational cost of CFD simulations cannot be reduced by using a restart strategy using the flow solution corresponding to the previous shape. The computational cost of the optimization grows linearly with the number of function evaluations.

The sensitivity of the objective function with respect to the shape variables is computed using the discrete adjoint method[14] of *elsA*. For an explicit presentation of the adjoint system solved the reader is referred to Peter et al.[15] and Meaux et al.[5] This method enables to compute the sensitivity of a single function with respect to n_{dv} design variables at the cost of one linear system resolution (same size as the RANS system). The gradient vector is thus computed using approximatively the same computational time (factor ≈ 1.5) as one direct flow simulation. For typical aerodynamic problems considering hundreds of design variables and a few functions (lift, drag), this is a considerable improvement over the classical method of finite differences requiring as many flow solutions as design variables.

3. *Aerodynamic function computation*

The objective function chosen is the far-field pressure drag,

$$F = Cd_p^{ff} = Cd_{vp} + Cd_w + Cd_i. \tag{1}$$

The friction drag, Cd_f, is excluded from the objective function as it does not significantly change considering small amplitudes of deformation. The wetted surface is kept unchanged (almost) by the shape deformation. The far-field code used is *ffd41*[16] developed by Onera. The two main advantages of this approach are its ability to decompose pressure drag into physical components (wave drag, induced drag, viscous pressure drag) and its accuracy through a filtering of non-physical drag (spurious drag).

The post-processing module can also compute the sensitivity of the drag with respect to the flow variables and with respect to the mesh with an analytical formulation.

As aerodynamicians work at fixed lift rather than at fixed angle of attack, a more realistic problem should also take into account a fixed lift coefficient as a constraint for the optimizer and include the angle of attack as a design variable, but the present work compares two very different type of algorithms and the comparison must not be biased by different methods of constraints handling.

C. Gradient based optimizer, reference optimizer (DOT-BFGS)

The reference optimizer is gradient based and uses the classical quasi-Newton BFGS method (Broyden-Fletcher-Goldfarb-Shanno) from the DOT (Design Optimization Tools)[17] library. Description of one internal step of this optimizer is given in Figure 2. As all gradient optimizers, it converges along a descent path until no improvement is achieved during one optimizer iteration or if the gradient norm is null. Firstly, the algorithm determines a descent direction, d_k, using the evolution of the gradient vector during the last two internal iterations. Once the direction is computed, a linear search aiming at computing the norm of displacement giving the best improvement is performed. The linear search is driven by a mono-dimensional polynomial interpolation and requires successive function evaluations. This type of optimizer is intrinsically sequential as it follows a single descent path. For the optimization

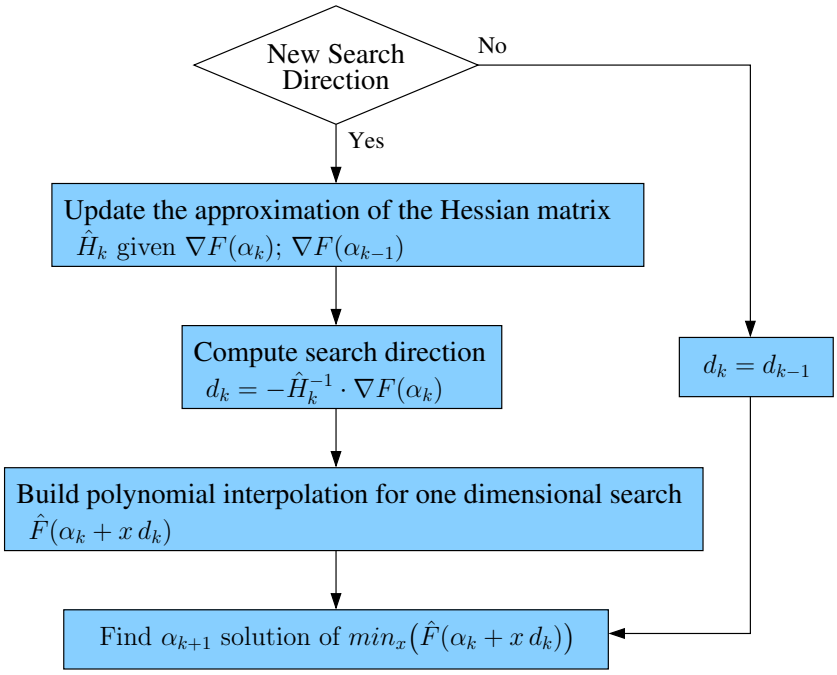

Figure 2. Inside the gradient based optimizer box

framework, only one set of design variables is handled by process iteration. The process population contains only one individual ($n_{pop} = 1$).

In terms of quantity of information, the quasi-Newton gradient algorithm proposes the next set of variables by using only the information about the current internal iteration. The internal iteration contains information about the descent direction (computed using evolution of two gradient values) plus some function evaluations (usually no more than ten). This optimization algorithm proposes a new shape based on $N = 2n_{dv} + 10$ scalar informations on the unknown function. Even if the approximated Hessian matrix, $\hat{H}_k$, is more and more accurate as the number of internal iterations increases, the algorithm does not retain all the information known about the function but focuses on the information in the vicinity of the current shape.

D. Performance of optimization algorithms

The properties of optimization algorithms are described by two opposite notions, exploration and exploitation. The exploration denotes the ability of the optimizer to avoid being trapped by the nearest optimum, whereas the exploitation denotes how quickly the optimizer find the nearest optimal point. Gradient based algorithms are well known to be very efficient in exploitation and poor in exploration.

The figure of merit measuring the restitution time needed for an optimization to converge is the number of process iterations, n_{iter}. The figure of merit measuring the computational cost of an optimization is the total number of function and gradient evaluations, $n_{eval} + n_{grad}$. These quantities both qualify the exploitation properties of the optimizer. One has to notice that for more precision, a distinction is made between iteration at the process level and iteration at the optimizer level. The number of internal iterations of the optimizer does not directly intervene to assess performance and is not reported (for DOT it corresponds to the number of gradient calculations).

The values denoted by the superscript *ref* are reference values obtained by comparing the value(s) of the objective function at a given process iteration to all previous evaluations, so as the quantity $F_c - F_c^{ref}$ is positive or null. As gradient algorithms converge by following a descent path, the final *ref* point is effectively the last computation, but that is not true for all optimizers.

The exploration can be measured in terms of variation of the function value during the optimization, $\sum(F_c - F_c^{ref})$, and in terms of variation of the design variables during the optimization, $\sum ||x_c - x_c^{ref}||$. In fact, as the variables are defined on very different ranges and are not scaled to measure the exploration, the measure of variation of the objective function is more representative.

III. Response Surface based optimizer (RS Kriging and RS Cokriging)

Surrogate modeling tools are now widely available within engineering. Basically, these tools enable to inexpensively approximate a function on a continous domain $\mathcal{D}$ given a database of n_s samples, $\mathcal{S}$. Once built, the surrogate model can give numerous information on the true function through: graphical plots giving trends, sensitivity of the function with respect to each variable, or the value and location of the minimum when coupled with an optimizer.

Although it is possible to build a globally reliable response surface when considering three or four design variables, a phenomenon described by Bellman[18] as the 'curse of dimensionality' prevents the use of global response surfaces on high dimensional spaces. In fact, this phenomenon also discards the use of global optimization algorithms on such problems. A previous study[19] varying dimension from one to six showed that the complexity of aerodynamic functions cannot be represented with as few as 200 function evaluations and recommends the use of local response surfaces. Local response surface based optimizers depart from a space filling sampling giving rough trends of the function and internally use an expensive global optimizer to optimize one or multiple sampling refinement criteria, $C_r(x)$, in order to iteratively increase the surface accuracy around locations of possible optima (cf. Figure 3).

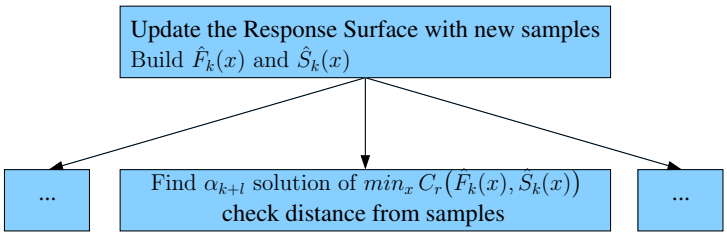

Figure 3. Inside a general response surface based optimizer box

This type of optimizer supposes that an efficient surrogate model can be built in high-dimensional search spaces (one hundred variables).

A. Building response surfaces with Kriging and Cokriging

Kriging[20] (DACE formulation) was chosen to build the response surface $\hat{F}(x)$ for its ability to approximate accurately non-linear functions and to guarantee a null error at samples.

1. *Kriging*

The Kriging method is from the statistical point of view the best linear interpolator. It is formed by a constant term, $\hat{\beta}$, representing a mean of the function at samples, F_s, plus a linear combination of basis function interpolating each sample built up as a stochastic process, $Z(x)$,

$$\hat{F}(x) = \hat{\beta} + Z(x) \approx F(x). \tag{2}$$

The basic assumption behind Kriging is that the covariance of the function is linked to the spatial correlation and this correlation is maximum when distance between points is null and decreases with distance,

$$cov[Z(s^i), Z(s^j)] = \hat{\sigma}^2 R_{ij}. \tag{3}$$

The correlation matrix R is dense symmetric positive definite (order $N = n_s$) with ones along the diagonal and results from evaluations of a Spatial Correlation Function,

$$R_{ij} = SCF(s^i, s^j) = \prod_v scf_v(|s_v^i - s_v^j|). \tag{4}$$

The basis function are the directional function (scf_v) linking the distance to the correlation between two points. A cubic spline function depending of one hyperparameter driving the directional strength of the correlation , θ_v, was chosen,

$$scf_v(x) = \begin{cases} 1 - 6(x\theta_v)^2 + 6(x\theta_v)^3 \, , \, x < \frac{1}{2\theta_v} \\ 2(1 - x\theta_v)^3 \, , \, \frac{1}{2\theta_v} \leq x < \frac{1}{\theta_v} \\ 0 \, , \, x \geq \frac{1}{\theta_v} \end{cases} \tag{5}$$

In comparison to the traditional gaussian function, the cubic spline gives generally a better conditioned correlation matrix. In fact, the gaussian function decreases exponentially with the distance but tends toward zero without reaching it, whereas the spline function is null after a threshold value fixed by the hyperparameter.

Contrary to the radial basis function method using a different function radius for each basis function, the Kriging method defines this radius by direction. Thus, an appropriate comparative study (domain scaling) of each θ_v values enables to assess the influence of each variable on the function.

Spatial correlation functions must respect at least two properties, be equal to one for a null distance and decrease with distance. These properties lead Kriging models to be unstable when sample points are clustered. If the distance between two samples is too small, their corresponding columns in the correlation matrix will be almost the same implying an ill-conditioned matrix. Space filling sampling methods like Latin Hypercube Sampling (LHS) are thus recommended when using Kriging.

To evaluate Kriging at an unknown location a vector of correlation r between sample points and the unknown is computed,

$$r_i(x) = SCF(x, s^i). \tag{6}$$

Once the correlation matrix inverted, the Kriging, $\hat{F}$, can finally be evaluated at a new point x,

$$\hat{F}(x) = \hat{\beta} + r^t(x)R^{-1}(F_s - 1\hat{\beta}). \tag{7}$$

One should notice, that only two samples are needed to build a Kriging response surface, whereas a linear polynomial interpolator requires at least $n_{dv} + 1$ samples to determine polynomial coefficients. In addition, the predicted uncertainty of the Kriging function or standard error, $\hat{S}(x)$, can be computed,

$$\hat{S}(x) = \hat{\sigma} \left(1 \, - \, r^t(x)R^{-1}r(x) \, + \, \frac{\left(1 - 1R^{-1}r(x)\right)^2}{1^tR^{-1}1} \right)^{\frac{1}{2}} \geq 0. \tag{8}$$

2. Kriging fit

As Kriging models directly interpolate the samples, they are not fitted by minimizing the least square residual error at samples as polynomial regression models. The standard error of Kriging is null at samples and increases with distance. Kriging are fitted by minimizing their predicted errors, more precisely by maximizing the logarithm of their likelihood estimates. This optimization problem is known as the Maximum Likelihood Estimate (MLE) problem.

For the parameters β and σ^2 analytical expressions maximizing this value are known,

$$\hat{\beta} = (1^tR^{-1}1)^{-1}1^tR^{-1}F_s, \tag{9}$$

$$\hat{\sigma}^2 = \frac{1}{n_s}(F_s - 1\hat{\beta})^tR^{-1}(F_s - 1\hat{\beta}). \tag{10}$$

The correlation parameters (θ_v) are solution of the following MLE optimization problem,

$$MLE = max_\theta\big(\ln(L)\big). \tag{11}$$

It is solved using a gradient based optimization algorithm initialized by an appropriate guess as described by Laurenceau et al.[19] Each likelihood evaluation requires the computation of the determinant of the correlation matrix,

$$\ln(L(\theta)) = -\frac{1}{2}\left(n_s\big(\ln(\hat{\sigma}^2) + \ln(2\pi) + 1\big) + \ln|R|\right), \tag{12}$$

and the resolution of the optimization problems requires up to one thousand likelihood evaluations.

During the iterative refinement process, the hyperparameters are refitted at each update of the sample database.

3. Sample limited indirect Cokriging, gradient enhanced Kriging

A Cokriging model[21] interpolates the function and the gradient at each sample location. As Cokriging models include more information on the true function than Kriging models, they need fewer samples to achieve a given level of accuracy. Moreover a comparison varying the dimension of the problem and the number of samples on an aerodynamic test case[19] has proven that the vectorial information provided by the gradient is more beneficial for high-dimensional problems.

For its flexibility, the formulation retained to build gradient enhanced Kriging is the indirect Cokriging formulation.[21] It does not change the Kriging formulation described previously because the gradient interpolation is performed through a sample database augmentation scheme. Instead of directly using the gradient information, it is used to add one point per direction at each sample using a first order Taylor approximation,

$$F(s^{n_s+iv}) = F(s^i) + \frac{\partial F(s^i)}{\partial x_v} 10^{-4} range_v(\mathcal{S}). \tag{13}$$

After using the gradient information at each sample, the augmented database contains $N = n_s(n_{dv}+1)$ samples.

The computational cost of Kriging model fit or evaluation depends mainly on the order of the correlation matrix, N. On a standard Intel Pentium4 2.8GHz processor, the estimated computation time for one correlation matrix inversion is,

$$t_{user} \approx 3.10^{-9} N^3 \ seconds, \tag{14}$$

and a total of several hundreds matrix inversions are done for each response surface.

The computational cost of the response surface (fit and evaluation) is considered negligible compared to CFD function evaluations for matrix order up to $N \approx 400$. However, as the computational cost of Cokriging depends on the number of variables, it becomes unusable for high-dimensional problems. It is then necessary to use altered augmentation schemes. Liu[21] proposed a direction limited augmentation scheme using only one augmentation point per sample, $N = 2n_s$. Practically, this strategy implies a loss of information. As gradients of CFD functions are computed with an adjoint method, the complete gradient vector is computed even if only one term of the vector is needed.

This is why a sample limited augmentation scheme is preferred, as suggested by Kim et al.[22] on their aerodynamic problem depending on six variables. It consists in using the complete gradient information but only for a fixed number of samples denoted n_{aug}. The augmented matrix order is then $N = n_s + n_{aug} n_{dv}$. Basically, using $n_{aug} = 10$ enables to abide by the rule of thumb recommending the use of at least 10 samples per direction to have a correct response surface. Practically, it is used to adapt the Cokriging computational cost to the number of variables.

B. Global optimization on the response surface

As stated in Figure 3, another optimization problem has to be solved inside the response surface based optimizer box. Some sampling refinement criteria, $C_r(x)$, must be optimized to exploit the information contained in the response surface and find a suitable new set of design variables. As the criteria are inexpensively evaluated, a global optimization strategy was adopted for their minimization. In order to effectively find the global minimum even for high dimensional problems, a strategy using multiple optimizers and multiple runs is used. This strategy proceeds in building a set, $\mathcal{P}$, of possible optima of the criterion given by various methods and finally retains only the minimum of this set.

Firstly, the set of possible optima, $\mathcal{P}$, is initialized by the minimum from the sample database.

Secondly, a binary coded genetic algorithm from David Caroll [a] is used with a population size of 100 individuals and a maximum number of generations equal to 1000. A convergence test was implemented to stop this algorithm when no improvement on the function is made during 10 consecutive iterations. This algorithm is ran 100 times using different seed values. Each run enables to increment a new point in the set of possible optima $\mathcal{P}$.

Thirdly, a pseudo-random exploration is performed on $100 \cdot 10^3$ sites. The minimum of the exploration phase is used to initialize a gradient descent (quasi-newton BFGS method). Once converged the minimum of the gradient algorithm is incremented in the set $\mathcal{P}$. This step is repeated 10 times with different seed values for the exploration.

Finally, a last gradient descent is performed departing from the minimum point of the set of possible optima $\mathcal{P}$.

[a] CU Aerospace, `http://www.cuaerospace.com/carroll/ga.html`

The global optimization on the response surface is necessary to assess the efficiency of the response surface based optimizer. In fact less caution would have been possible during this internal optimization, but as this step should be as efficient on the 6 design variables test case as on the 45 design variables test case, robustness was privileged in detriment of convergence speed. As no bias can be attributed to this internal optimization, any loss of performance of the response surface based optimizer when increasing dimension can be attributed to a loss of accuracy of the response the surface.

C. Sampling refinement criteria

A variety of sampling refinement criteria exists. Some are designed to reduce the global error of the response surface,[19] and some are designed to minimize the true function modelised by the response surface. Sasena[23] compared a variety of criteria.

1. Definition

The most basic criteria exploits only information on the function,

$$C_1(x) = \hat{F}(x). \tag{15}$$

As stated in the Surrogate Management Framework,[24,25] optimization algorithms that refine iteratively only at the predicted minimum ('SEARCH' step) cannot efficiently find an optimum. A spatial exploration is necessary ('POLL' step) to ensure convergence. Using the standard error (Eqn. 8), it is possible to compute criteria able to locate minimum of the function taking into account uncertainty of the model.

Each refinement criterion proposes its own manner to balance exploitation of low function values with exploration of high uncertainty values. The Expected Improvement (EI)[6] is the most common when using Kriging. It is a statistical criterion maximizing the probability of improvement over the best sample value. The maximum of the EI function indicates the predicted location of the global minimum of the true function. It has proven to be an efficient refinement criterion on low dimensional problems.[6,9,26] Despite that, the EI criterion was not retained here as it tends to cluster points around predicted minimum before exploring other locations. Moreover, it has quasi-null gradients on large parts of the domain making it impossible to use gradient algorithms to find its maximum (Figure 4). This criterion is then hard to maximize.

The Lower Confidence Bounding (LCB) criterion of Cox et al.[27] was preferred as it tends to give more weight to the uncertainty. The LCB directly uses a balancing coefficient b between function and error to perform a linear combination,

$$LCB_b(x) = \hat{F}(x) - b\hat{S}(x),\ b \in \mathcal{R}^*. \tag{16}$$

The minimum of LCB indicates the expected location of the minimum of the true function. The standard error of Kriging being always positive, the error balancing coefficient, b, can be positive or negative. The values tested by Cox et al. are $b = 2.0$ and $b = 2.5$, using a different optimization process (single refinement criterion) on low-dimensional analytical functions. In this work, the values chosen are $b = 1.0$ and $b = -1.0$,

$$C_2(x) = LCB_1(x) = \hat{F}(x) - \hat{S}(x), \tag{17}$$

$$C_3(x) = LCB_{-1}(x) = \hat{F}(x) + \hat{S}(x). \tag{18}$$

On Figure 4, the minimum of the surrogate model F indicates an already sampled location, different from the optimum on the true function. Using the standard error (error bars), the minimum of the LCB_1 function will explore a zone of higher uncertainty whereas the EI criterion indicates a location very close to the minimum of the Kriging function corresponding to an already sampled location.

2. Distance check

Each location indicated by the minimization of a sampling refinement criteria must be validated by performing a distance computation with respect to already computed locations. The main purpose of this validation is to avoid expensive CFD computations on shapes already well known. The secondary purpose is to ensure a good conditioning of the Kriging correlation matrix. The point is rejected if its minimal distance with respect to already computed samples fall below a fixed threshold,

$$x^{ref} \text{ accepted if, } min_i \left(\sum_v \frac{|x_v^{ref} - s_v^i|}{range_v} \right) > n_{dv} \cdot \epsilon. \tag{19}$$

The value of the threshold, ϵ, should be adapted to the level of convergence expected from the optimizer i.e. the number of iterations authorized. The value chosen here is $\epsilon = 10^{-6}$.

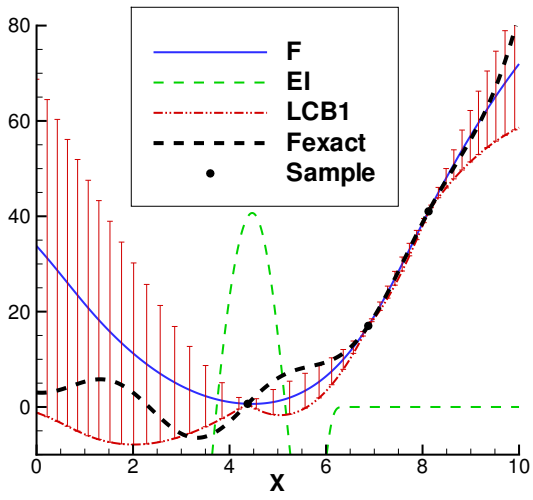

Figure 4. Expected Improvement and Lower Confidence Bounding Function on a one dimensional example approximated using three samples

D. Kriging/Cokriging based response surface optimizer

The new Kriging/Cokriging based optimizers (Figure 5) are initialized using a space filling sampling method for the first iteration. The refinement process begins at the second iterations after n_s^{ini} function evaluations. It uses for the exploitation of the best function value a refinement at the predicted minimum of the function. In order to ensure exploration of the domain and try to converge to the global optimum, the LCB function is used with two different error balancing coefficients. This framework runs simultaneously up to three CFD runs ($n_{pop} \leq 3$) per iteration. As the sampling refinement includes exploratory steps, this optimizer is able to manage very coarse initial sample database (10 samples for 45 variables) at the cost of more iteration of the process. The use of large initial sampling database can also save time as the n_s^{ini} evaluations are done simultaneously.

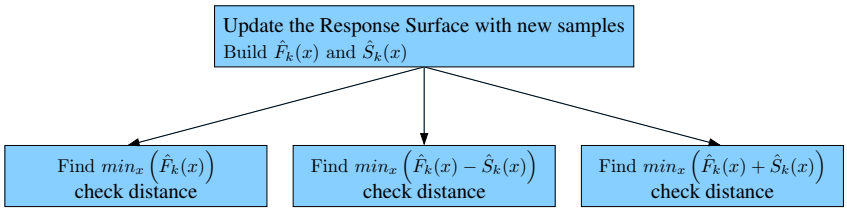

Figure 5. Inside the Kriging/Cokriging based optimizer box

Convergence is said to be reached when all three proposed refinements are close too already sampled locations or if the function value cannot be improved during 20 iterations.

IV. Results

In the following section, a comparison between the two new Kriging and Cokriging based optimizers and the reference gradient optimizer is made on two test cases. This comparison was not made on analytical test functions common in the optimization community because it cannot be representative of real CFD functions: multimodal (lots of local optima), high number of more or less independent variables and noise. The no-free-lunch theorem for optimization by Wolpert et al.[28] states that the ultimate optimization algorithm does not exist. If an optimizer is the most competitive for a given set of problems, it is always possible to find another set of problems to contradict this fact. Wolpert et al. recommendations are then to use a wide variety of test problems in order to compare general-purpose optimization algorithms, and to incorporate problem-specific knowledge into the optimizer as much as possible. Our test cases are then directly performed within OPTaliA. Moreover, the problems are defined as representative simplifications (no constraints, less design variables, coarse meshes) of real industrial problems handled by Airbus designers.

A. Airfoil drag minimization considering 6 design variables

The first test problem is about a RAE2822 airfoil at a Mach number $M = 0.729$, an angle of attack 2.31^o. The chord length is one meter and the Reynolds number value is $Re = 7.10^6$. The C-mesh is formed of 73×458 nodes with its boundary layer refinement. The restitution time for one flow simulation is 400 seconds using two processors AMD Opteron 275 (2.2 Ghz) to perform 400 steady iterations. The objective function considered is the far field pressure drag. Two bumps ($n_{dv} = 6$) are applied to deform the upper surface in the direction of the vertical axis. Only positive deformations are authorized and the maximum amplitude of one bump is 5 millimeters. The domain of variation for the position of the bumps authorizes recovering.

The complexity of the optimization problem is low as only one function and six design variables are considered. Moreover, the ranges on variables are restrained to positive deformations in order to limit the complexity of the objective function on the domain and thus the number of possible optima and to limit the impact on lift.

The same initial database is used for Kriging and Cokriging response surfaces. It contains 10 samples, $n_s^{ini} = 10$, including the baseline shape and nine space filling samples distributed by a Sobol method. The sample database augmentation for the sample-limited Cokriging includes the information of 10 gradient vectors, $n_{aug} = 10$, providing a quantity of information equivalent to 60 additional samples.

The starting point for the gradient algorithm corresponds to large equidistributed bumps of null amplitudes.

On Figure 6 are represented the baseline and the optimized shapes and associated pressure coefficient values. The pressure distribution of the baseline shape demonstrates the presence of a shock wave (drag production) at the upper surface of the airfoil (between 50% and 60% of the chord length). This shock wave is attenuated (suppressed) by the deformation proposed by the different optimizers. It appears that the reference optimizer (DOT-BFGS) converged to a different optimal deformation. Whereas both response surface optimizers find an optimal shape by distributing the two bumps at two different locations (leading edge and location of the shock), the gradient optimizer has been trapped by the high sensivity of the drag to deformation near the shock wave and has tried to superpose both bumps. This illustrates the fact that even considering simple shape variables (only two bumps), functions studied in aerodynamic shape optimization exhibit multimodal behaviour and strong trends (high gradient values) especially when considering transonic flows. The low complexity of this problem does not prevent the gradient to be trapped by a local optima.

In terms of final function value, both response surface optimizers largely outperform the gradient optimizer. It appears that even the purely exploratory space filling samples ($n_{eval} \leq 10$) managed to improve the baseline shape and to outperform the gradient. Both response surfaces algorithms give very close results in terms of shape deformation and final objective function value confirming their enhanced global exploration capabilities. One can see from Table 1 that contrary to the RS optimizers, DOT-BFGS does not manage to completely suppress the shock and slightly increases the viscous pressure drag by its larger deformation near the trailing edge.

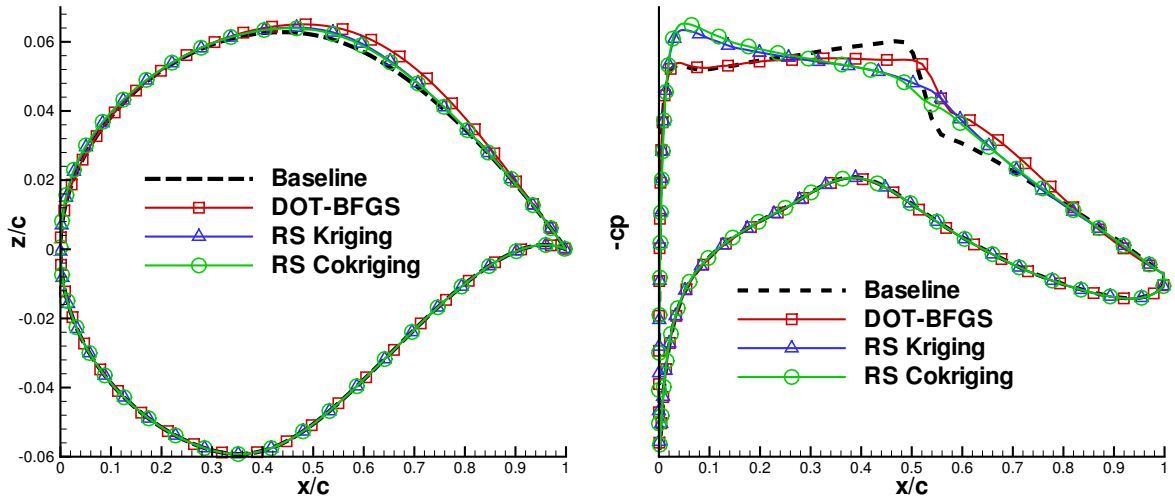

Figure 6. Optimized shape and pressure distribution of an RAE2822 airfoil

Before totally discarding the reference optimizer, it is necessary to look at the computational cost

	Baseline	DOT-BFGS	RS Kriging	RS Cokriging
$Cd_p = F$	100	84.6 (-15%)	76.7 (-23%)	76.9 (-23%)
Cd_w	100	16.5	0.0	3.0
Cd_{vp}	100	100.7	95.3	94.5
Cl	100	103.2	102.1	102.2

Table 1. Aerodynamic performance summary for RAE2822 optimizations

(total number of evaluations $n_{eval}+n_{grad}$) and restitution time (number of process iterations n_{iter}) of the different optimizations represented on the Figure 7 and summarized in Table 2. The worst algorithm in terms of final function value, DOT-BFGS, is also the quickest and less expensive. It seems coherent with the fact that this method performs very few exploration of the domain in terms of variation in function value and variables value. The additional cost of RS optimizers is attenuated by the parallelization of the optimization process and stays acceptable in an industrial context. It is important to note that the RS optimizers reached a plateau of convergence in as few iterations as the gradient optimizer, but the RS optimizers managed to continue domain exploration from this reference value. The additional gradient information used by the Cokriging RS seems not absolutely necessary, certainly due to the low dimensionality of the problem. It effectively produces a more accurate surrogate model as can be deduced from the lower values of exploration compared to the Kriging RS When considering the the last process iteration of DOT-BFGS ($n_{iter} = 41$ on Figure 7), the function value given by the gradient optimizer is a lot larger than function values given by both RS based optimizers. Moreover both RS optimizers have reached a plateau of convergence. One could then say that in terms of resitution time the RS based optimizers are better than the gradient reference.

On figures Figure 7 and Figure 10, the lines stand for *ref* values of the objective function and the symbols represent the current values of the function.

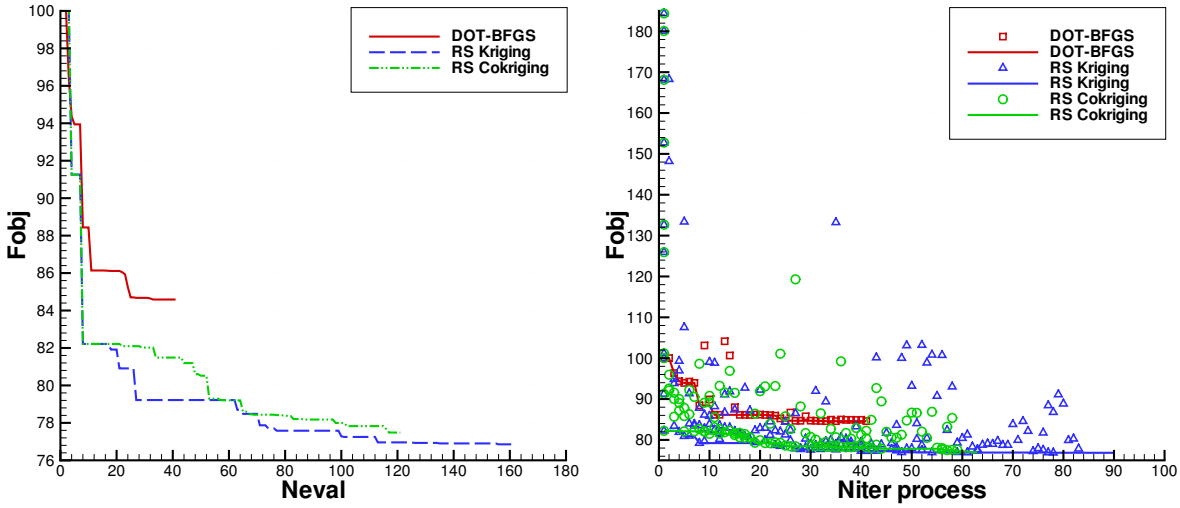

Figure 7. Optimization convergence history on the RAE2822 test case

The experience of numerous airfoil optimization with OPTaliA demonstrated that it is impossible to increase indefinitely the number of design variables on bidimensional airfoils without having highly correlated design variables. In fact, by using a decomposition of optimal shape deformation into a sum of Hicks-Henne bumps, one can see that the optimum of airfoil drag minimization problems are generally obtained through a sum of no more than five bumps. Increasing the number of bumps on a bidimensional wing section is generally equivalent to a parameterization using less bumps with higher ranges in amplitude of bumps. By considering only two bumps, one can guarantee that there is no redundancy between design variables and that the complexity of the optimization problem is effectively represented by the number of design variables. In Figure 8, a simple shape deformation obtained with 2 Hicks-Henne bumps of maximal amplitude equal to 3.0 millimeters is reconstructed using a very complex combination of 20 Hicks-Henne recovering bumps of maximum amplitude equal to 0.5 millimeters.

	DOT-BFGS	RS Kriging	RS Cokriging
F^{ref} final value	84.6 (-15%)	76.7 (-23%)	76.9 (-23%)
n_{iter} ; $n_{eval} + n_{grad}$; n_s^{ini}	41; 41+6; 0	90; 162+0; 10	63; 121+10; 10
$\sum \lvert F_c - F_c^{ref} \rvert$	56.2	1503	490.0
$\sum \lvert F_c - F_c^{ref} \rvert / n_{iter}$	1.37	16.7	7.78
$\sum \lvert F_c - F_c^{ref} \rvert / n_{eval}$	1.37	9.28	4.05
$\sum \lvert\lvert x_c - x_c^{ref} \rvert\rvert / n_{dv}$	0.565	65.4	41.0
$\sum \lvert\lvert x_c - x_c^{ref} \rvert\rvert / n_{iter} / n_{dv}$	$1.38\,10^{-2}$	$72.7\,10^{-2}$	$65.1\,10^{-2}$
$\sum \lvert\lvert x_c - x_c^{ref} \rvert\rvert / n_{eval} / n_{dv}$	$1.38\,10^{-2}$	$40.4\,10^{-2}$	$33.9\,10^{-2}$

Table 2. Optimizers performance summary on the RAE2822 test case

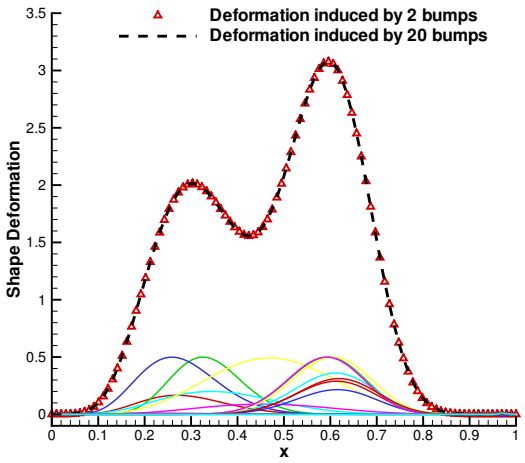

Figure 8. Equivalence between 2 Hicks-Henne bumps and 20 Hicks-Henne bumps of lower amplitude

B. Wing drag minimization considering 45 design variables

In order to increase the number of design variables with as less redundancy between variables as possible a three dimensional shape is considered, the AS28 wing in cruise condition, at Mach number $M = 0.8$ and angle of attack $AoA = 2.2^o$. The wing span is $b = 25$ meters and the mean chord value is $c = 5.4$ meters, giving a Reynolds number $Re = 40 \cdot 10^6$. The structured mesh (Figure 9) is formed of 4 blocks containing a total of $500 \cdot 10^3$ nodes with its boundary layer refinement. The restitution time for one flow simulation is 4400 seconds using two processors AMD Opteron 275 (2.2 Ghz) to perform 500 steady iterations.

The 45 shape parameters correspond to 15 Hicks-Henne bumps on the upper surface. The vertical bumps are distributed by group of three in five spanwise sections. At each section, the three bumps are equidistributed and a recovering is authorized by groups of two bumps. In order to ensure a wide diversity of possible shapes (maximum degree of freedom) with this parameterization, the linear spanwise expansion of each sectional deformation is stopped either at the closest boundary either at the closest deformed section. Only positive deformations are authorized and the maximum amplitude of one bump is 50 millimeters. The complexity of this problem is thus increased by considering more independent design variables and also by increasing the relative range of possible deformation, $\frac{A^{max}}{c}$ from five to ten as represented by optimal shapes on Figures 14-17.

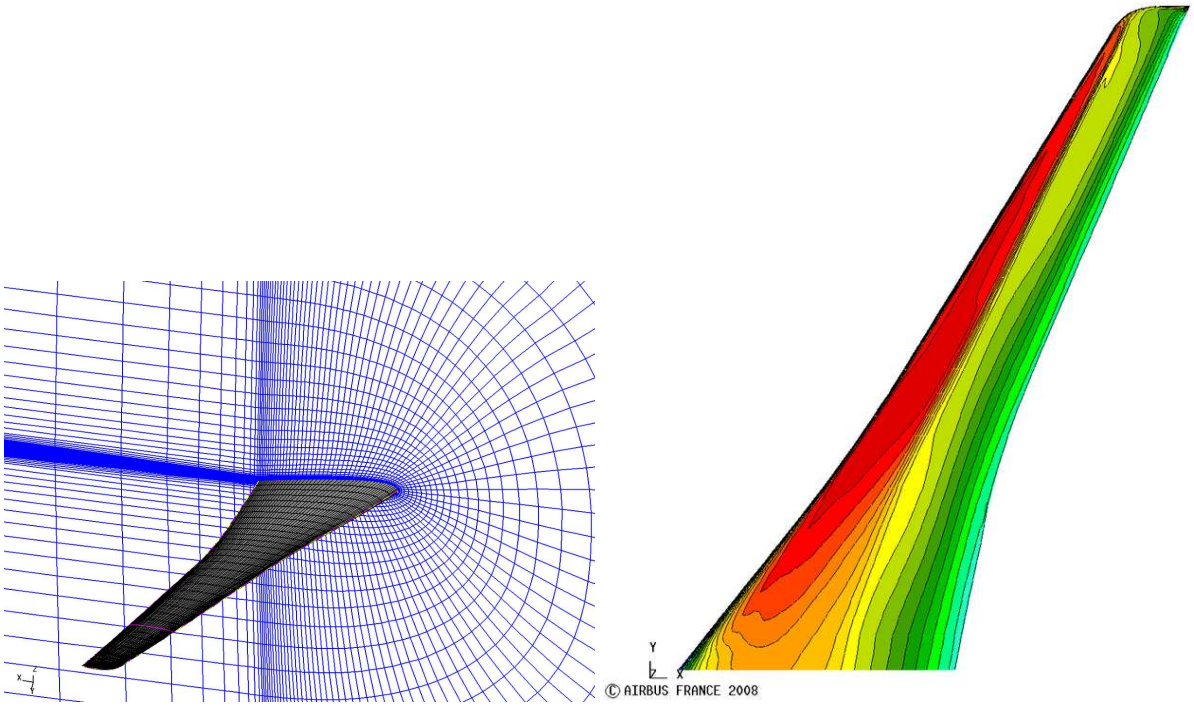

Figure 9. Mesh and pessure coefficient of the AS28 wing baseline shape

The RS based optimizers were intialized with only 11 samples, $n_s^{ini} = 11$, and the number of samples is even inferior to the number of variables. The initial database was formed of the baseline configuration plus 10 space filling samples (Latin Hypercube Sampling method). The Cokriging sampling was augmented by including the information of 5 gradient vectors, $n_{aug} = 5$, giving 225 additional scalar values. The starting point for the gradient algorithm corresponds to large equidistributed bumps of null amplitudes.

In cruise condition a shock wave appears on the upper surface of the baseline configuration. The Table 3 shows that most of optimizers drag reduction cames from minimization of the wave drag and leads to shapes with very weak shocks as can be verified on pressure distribution (cf. Figures 11-13).

Figure 10 presents the convergence history of the optimizations. The gradient based algorithm keeps the same properties as with the previous test problem. It converges using few process iterations and very few objective function evaluations (even fewer than on the six dimensional problem). Despite its speed, it gives the lowest improvement of the objective function.

For the RS based optimizations, the influence of the very coarse initial sampling can directly be observed by comparing exploration values of Kriging and Cokriging optimizers (Table 4). As the Kriging response surface is less accurate, the Kriging based optimizer performs three times more exploration and needs more iterations to converge to its final value. The sampling refinement becomes almost similar

	Baseline	DOT-BFGS	RS Kriging	RS Cokriging
$Cd_p = F$	100	95.1 (-4.9%)	94.6 (-5.4%)	93.5 (-6.5%)
Cd_w	100	28.3	28.3	11.2
Cd_{vp}	100	99.2	98.3	98.7
Cd_i	100	103.2	102.9	103.7
Cl	100	102.9	102.3	103.2

Table 3. Aerodynamic performance summary for AS28 wing optimizations

to a space filling refinement, but a significant improvement of the function is achieved and the Kriging optimizer outperforms the gradient reference. Somehow, its convergence is stopped because no improvement is achieved during the last 20 iterations whereas the algorithm continues its domain exploration characterized by the large widespreading of symbols on Figure 10. The Kriging based optimizer seems to need more iterations to robustly converge. The Cokriging based algorithms does not suffer this problem, as very few improvement is achieved during the last 50 iterations despite the continous exploration of the domain. By looking at the tridimensional views of optimal deformation fields given by Kriging (Figure 12) and Cokriging (Figure 13) based optimizers, it appears that both fields are very similar but the Kriging shapes corresponds in fact to an intermediate shape explored during the Cokriging based optimization.

The sample limited Cokriging based reached the best final objective function value at the cost of more function evaluations compared to DOT-BFGS. Despite the additionnal computational cost, the Cokriging based optimizer outperforms the gradient reference in terms of restitution time. When considering the last process iteration of DOT-BFGS ($n_{iter} = 26$ on Figure 10), the function value given by the gradient optimizer is larger than the function value given by the Cokriging based optimizer. However, the RS Cokriging optimizer has not yet reached its plateau at this iteration and needs three times more iterations to converge.

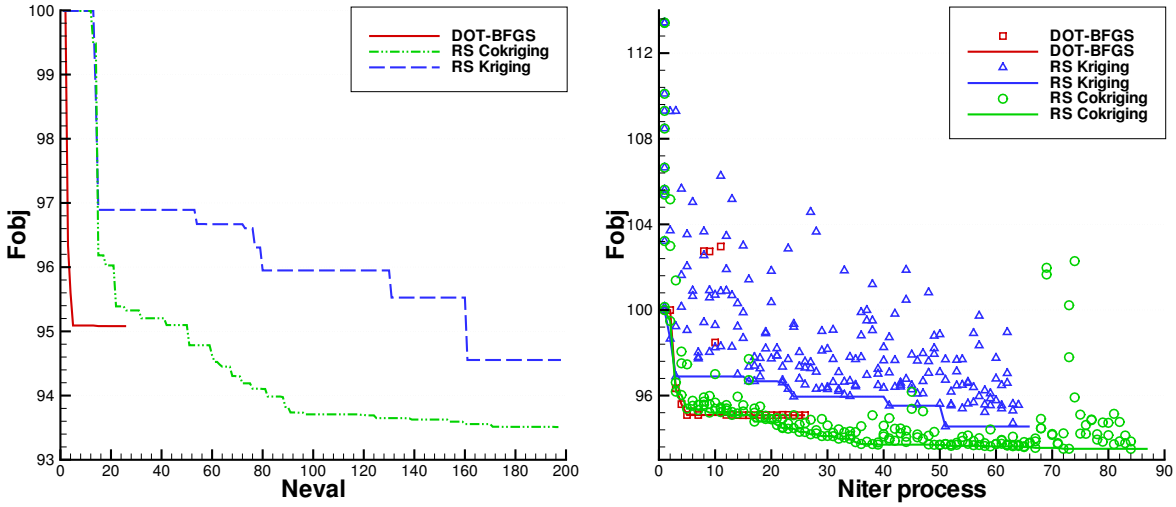

Figure 10. Optimization convergence history

On Figure 18 are plotted approximated drag polar for the baseline configuration and the three optimized shapes. The approximation is done using a second order polynomial method fitted with three samples for each shape represented by symbols on the figure. This graph shows that even after changing the angle of attack of the new shapes to obtain the same lift coefficient as with the baseline shape, the aerodynamic performance keep the same classification, gradient and Kriging based optimization give equivalent improvements and the Cokriging based optimal shape is best. Despite the improvement at cruise conditions, it is not surprising to observe that the performances at low lift flight conditions are decreased by all optimizations as it was not taken into account in the formulation of the optimization problem. As expected, the less the drag is reduced, the less low lift performances are impacted and the shape given by the gradient optimization is better than the shapes given by both RS based optimzations

	DOT-BFGS	RS Kriging	RS Cokriging
F^{ref} final value	95.1 (-4.9%)	94.6 (-5.4%)	93.5 (-6.5%)
n_{iter} ; $n_{eval} + n_{grad}$; n_s^{ini}	26; 26+4+0	66; 198+0; 11	88; 197+5; 11
$\sum \lvert F_c - F_c^{ref} \rvert$	26.8	534.4	205.5
$\sum \lvert F_c - F_c^{ref} \rvert / n_{iter}$	1.03	8.10	2.34
$\sum \lvert F_c - F_c^{ref} \rvert / n_{eval}$	1.03	2.70	1.04
$\sum \lvert\lvert x_c - x_c^{ref} \rvert\rvert / n_{dv}$	4.79	232.1	73.5
$\sum \lvert\lvert x_c - x_c^{ref} \rvert\rvert / n_{iter} / n_{dv}$	0.18	3.52	0.84
$\sum \lvert\lvert x_c - x_c^{ref} \rvert\rvert / n_{eval} / n_{dv}$	0.18	1.17	0.37

Table 4. Optimizers performance summary on the AS28 wing test case

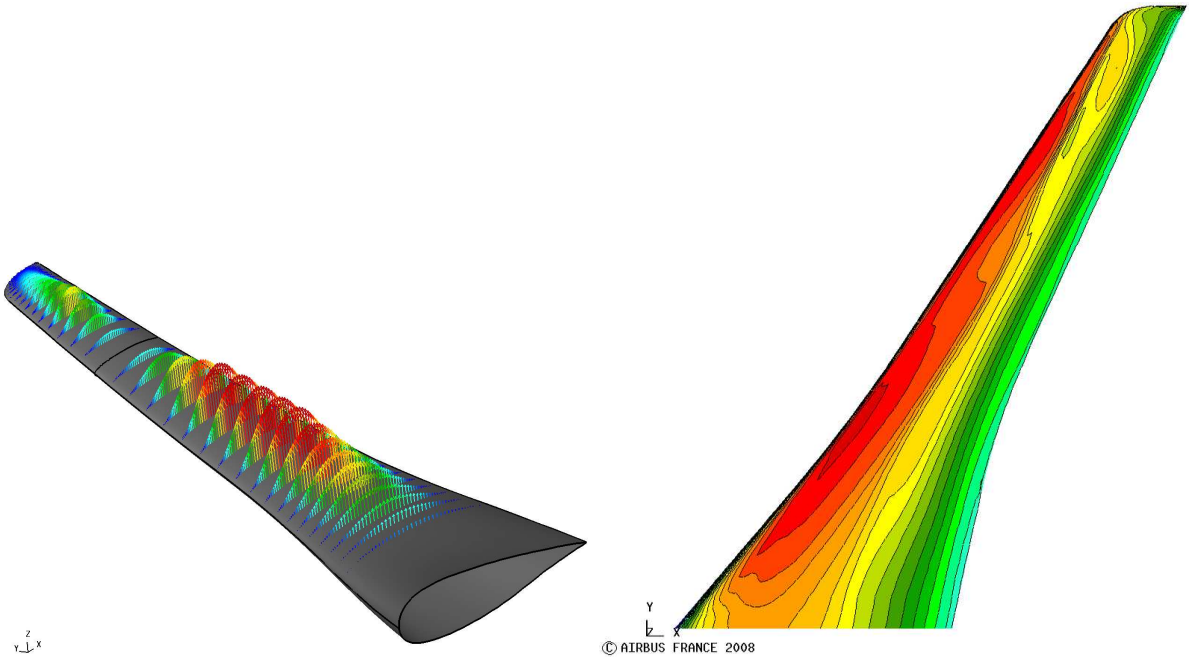

Figure 11. Optimum given by DOT-BFGS for the AS28 wing. Left: vectorial field of deformation. Right: pressure coefficient.

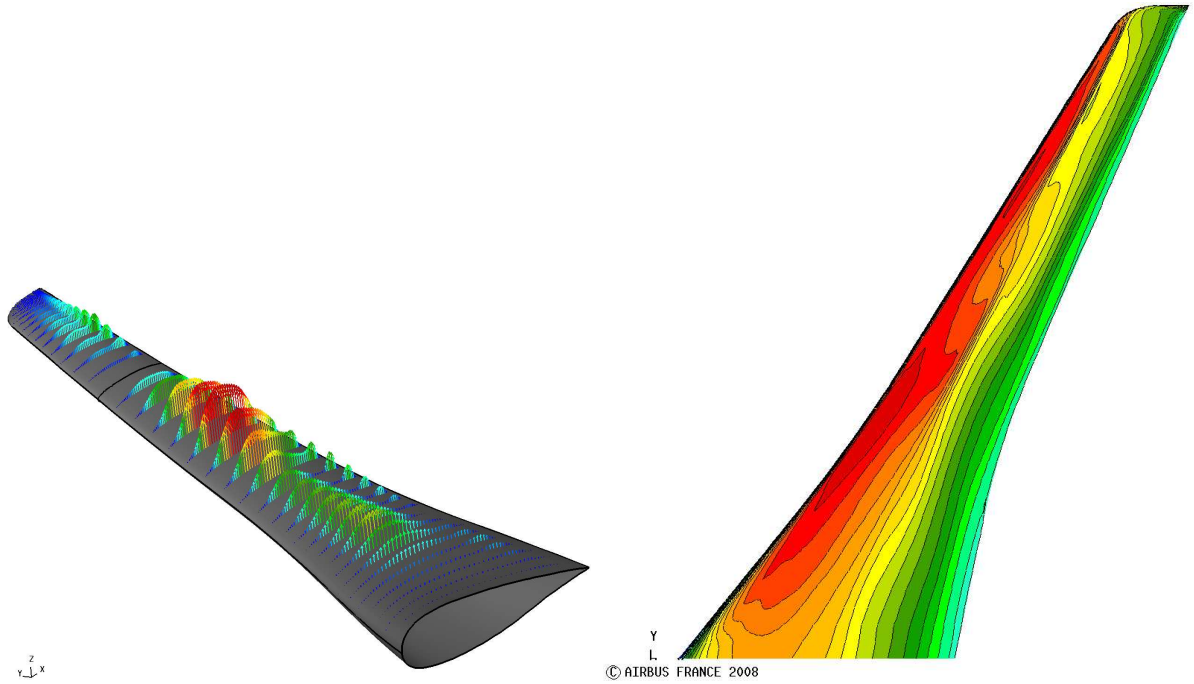

Figure 12. Optimum given by the Kriging based optimizer for the AS28 wing. Left: vectorial field of deformation. Right: pressure coefficient.

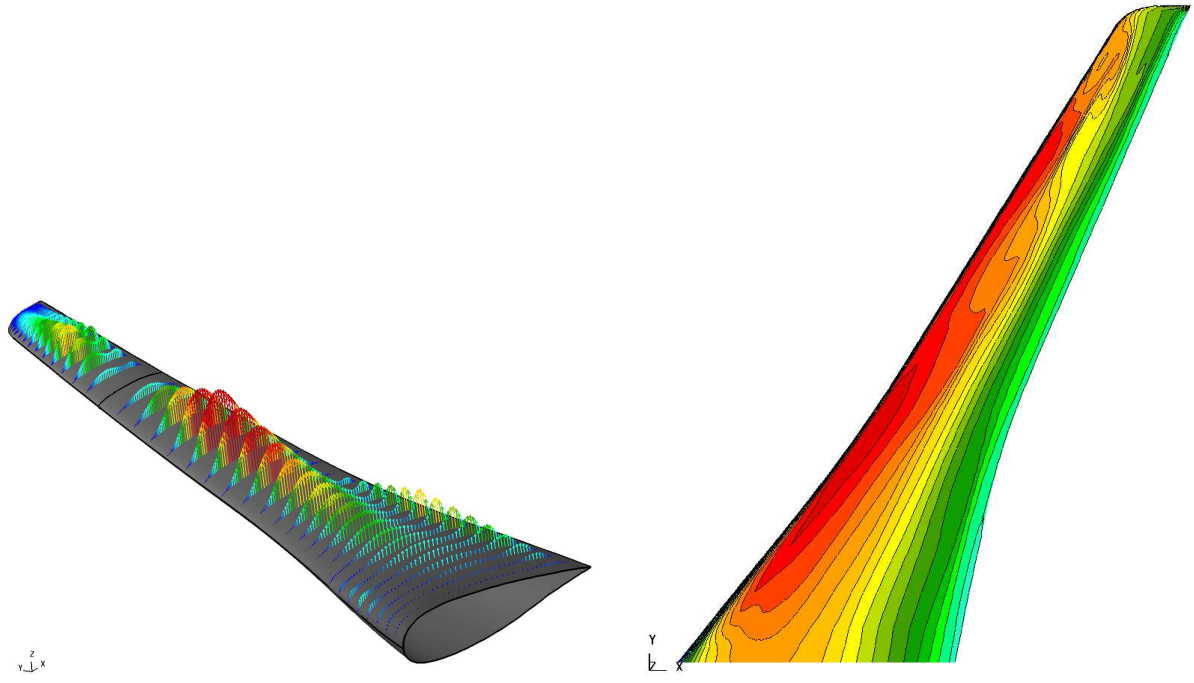

Figure 13. Optimum given by the Cokriging based optimizer for the AS28 wing. Left: vectorial field of deformation. Right: pressure coefficient.

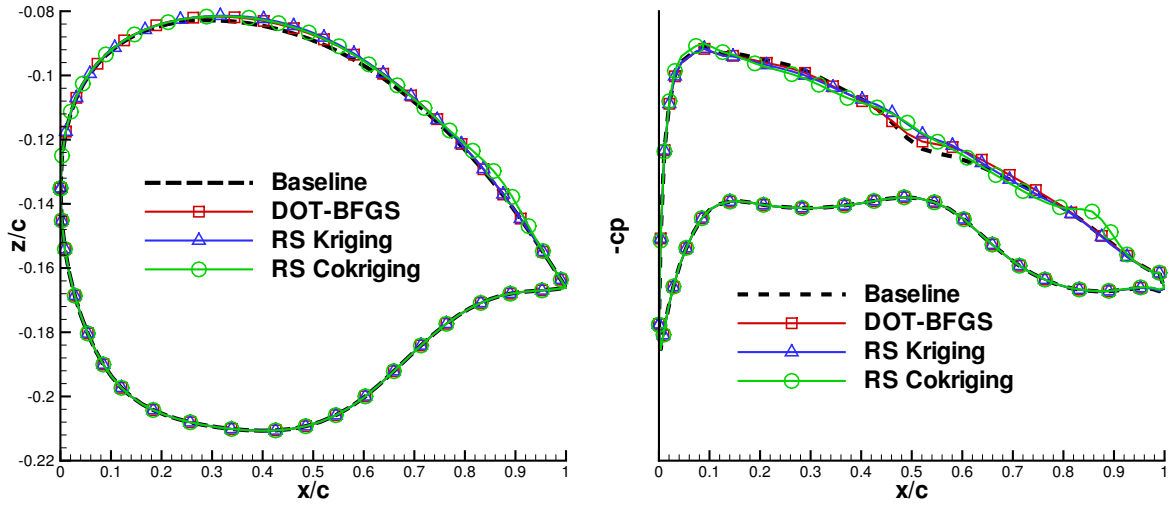

Figure 14. Optimized shape and pressure distribution at 6% of the wing span

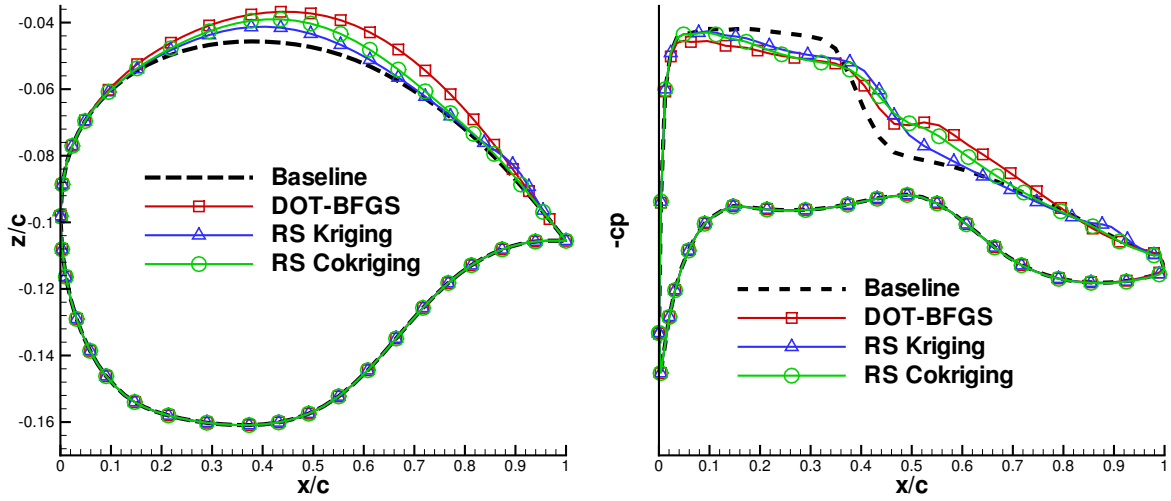

Figure 15. Optimized shape and pressure distribution at 19% of the wing span

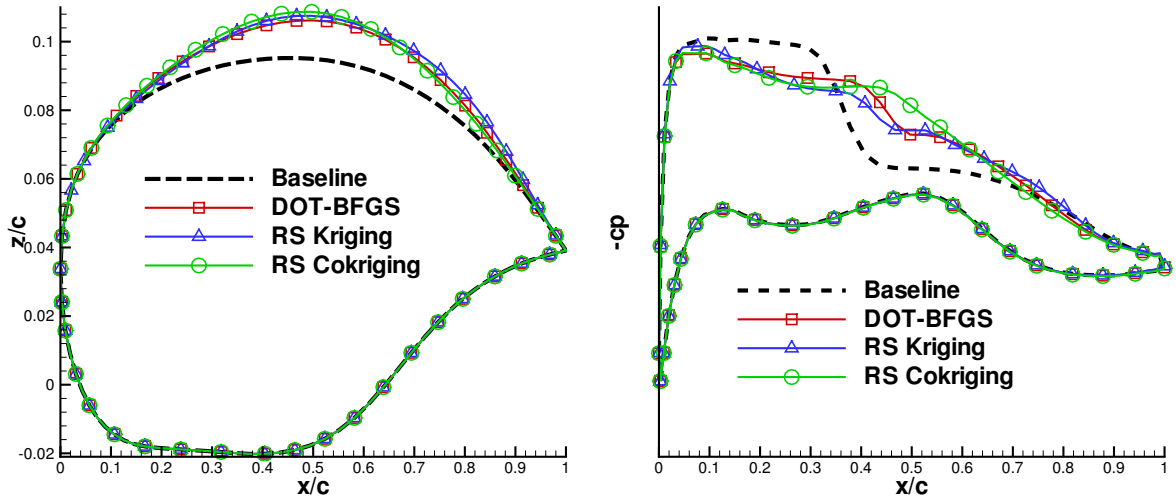

Figure 16. Optimized shape and pressure distribution at 35% of the wing span

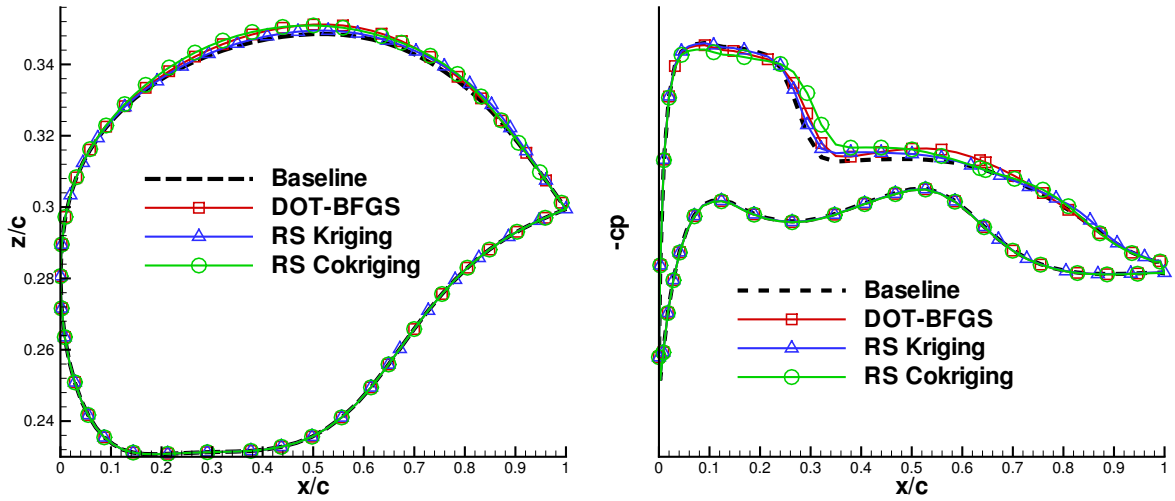

Figure 17. Optimized shape and pressure distribution at 60% of the wing span

in these conditions.

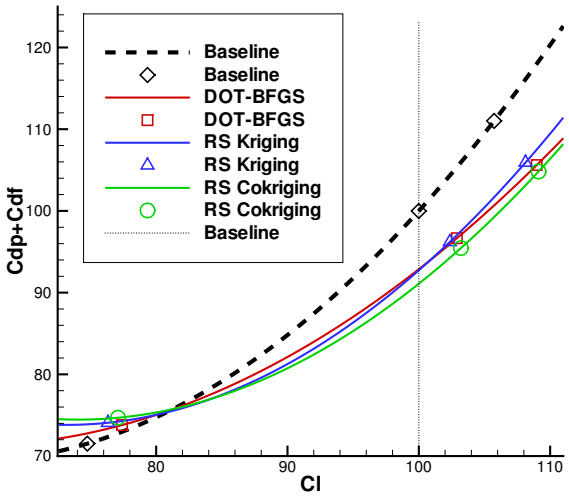

Figure 18. Approximated drag polars for the AS28 wing

V. Conclusion

A general framework for optimization enabling the use of various optimizers was set up. Two algorithms based on response surfaces built with Kriging and Cokriging were then implemented and compared to the quasi-Newton reference algorithm. These optimizers seem very promising as they achieve better function improvement and are complementary to the reference gradient optimizer converging very quickly to the nearest optimum. Best practices can be drawn from the results obtained on the two drag reduction test problems.

For the low dimensional problem, both response surface based optimizers largely outperform the gradient in terms of final function value, but also outperform it at equivalent computational cost and restitution time. Despite the fact that the gradient algorithm converges to a local optimum, it needs fewer functions evaluations to reach its final optimum. The interpolation of gradient information does not significantly improve Cokriging based optimizer performances, that is why in conclusion the gradient free Kriging based optimization algorithm should be preferred for low dimensional problems.

When increasing the complexity of the problem by considering 45 design variables on a wing, the Kriging based optimizer requires twice more iterations after gradient convergence before outmatching it. The lack of accuracy of the Kriging model makes it converge permaturously due to an excessive

exploration of the domain. The sample limited Cokriging based optimizer does not suffer this problem and effectively converges to a better solution than the gradient reference. The additional cost needed is small in terms of restitution time due to the efficiency of the parallel framework, but is large in terms of total function evaluations. The cost of the gradient optimizer being totally independent of the number of design variables, it should be preferred for high dimensional problems when computational cost is the major issue. The Cokriging based optimizer can converge to a more global optimum and should be preferred when performance is the major issue.

These conclusions can be explained by considering the quantity of information, N, contained inside each optimizer. At a given iteration, the gradient algorithm only knows a search direction given by comparing two gradient vectors and a few function evaluations, $\forall n_{eval}\ N_{Gradient} \approx 10 + 2 \cdot n_{dv}$. Response surfaces are a mean to cumulate all known information on the function in a sample database and the sampling refinement process aims at having an accurate model in the vicinity of the global optimum. For the Kriging based algorithm, the quantity of information is independent of the number of variables $\forall n_{eval}\ N_{RSKriging} = n_{eval}$ explaining its domain of applicability. The Cokriging based algorithm interpolates a fixed number (n_{aug}) of gradient vectors $\forall n_{eval}\ N_{RSKriging} = n_{eval} + n_{aug} \cdot n_{dv}$ giving an efficient algorithm even for high dimensional problems.

Lastly, it was assessed that the use of global optimizers imply a careful definition of the optimization problem in order to avoid unrealistic shapes. The gradient based optimizer, departing from an already existing shape and converging to a close local optimum, is less prone to this problem.

Acknowledgments

The authors would like to acknowledge M. Montagnac from Cerfacs for its support and suggestions. We used the computer implementation of J. Burkardt (`http://www.scs.fsu.edu/~burkardt/`) to generate the initial space filling sampling and the genetic algorithm of D. Caroll for optimization on response surfaces and would to thank them for sharing their source codes.

References

[1]Hicks, R. M. and Henne, P., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, 1978, pp. 407–412.

[2]Jameson, A., "Efficient Aerodynamic Shape Optimization," *10th Multidisciplinary Analysis and Optimization Conference*, No. AIAA-04-4369, AIAA/ISSMO, 2004.

[3]Liu, P.-Y. J. and Zingg, D. W., "Comparison of Optimization Algorithms Applied to Aerodynamic Design," *42nd Aerospace Science Meeting & Exhibit*, No. AIAA-04-454, AIAA, 2004.

[4]Moigne, A. L. and Qin, N., "Variable-Fidelity Aerodynamic Optimization for Turbulent Flows Using a Discrete Adjoint Formulation," *AIAA Journal*, Vol. 42, No. 7, 2004, pp. 1281–1292.

[5]Meaux, M., Cormery, M., and Voizard, G., "Viscous Aerodynamic Shape Optimization Based On The Discrete Adjoint State For 3D Industrial Configurations," European Congress on Computational Methods in Applied Sciences and Engineering, Jul. 2004.

[6]Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, 1998, pp. 455–492.

[7]Torczon, V. and Trosset, M. W., "Using approximations to accelerate engineering design optimization," Tech. Rep. TR-98-33, ICASE, Institute for Computer Applications in Science and Engineering, 1998.

[8]Peigin, S. and Epstein, B., "Robust Optimization of 2D Airfoils Driven by Full Navier-Stokes Computations," *Computers & Fluids*, Vol. 33, No. 9, 2003, pp. 1175–1200.

[9]Jeong, S., Murayama, M., and Yamamoto, K., "Efficient Optimization Design Method Using Kriging Model," *Journal of Aircraft*, Vol. 42, No. 2, 2005, pp. 413–419.

[10]Kanazaki, M., Tanaka, K., Jeong, S., and Yamamoto, K., "Multi-Objective Aerodynamic Exploration of Elements' Setting for High-Lift Airfoil Using Kriging Model," *Journal of Aircraft*, Vol. 44, No. 3, 2007, pp. 858–864.

[11]Booker, A. J., Cramer, E. J., Frank, P. D., Gablonsky, J. M., and Dennis, J. E., "MoVars: Multidisciplinary Optimization Via Adaptive Response Surfaces," *48th Structures, Structural Dynamics, and Materials Conference*, AIAA/ASME/ASCE/AHS/ASC, 2007.

[12]Mouton, S., Laurenceau, J., and Carrier, G., "Aerodynamic and Structural Optimisation of Powerplant Integration under the Wing of a Transonic Transport Aircraft," *42éme colloque d'Aérodynamique Appliquée*, AAAF, 2007.

[13]Cambier, L. and Veuillot, J.-P., "Status of the *elsA* CFD Software for Flow Simulation and Multidisciplinary Applications," *46th Aerospace Science Meeting & Exhibit*, No. AIAA-08-664, AIAA, 2008.

[14]Jameson, A., "Aerodynamic Design Via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988.

[15]Peter, J., Pham, C.-T., and Drullion, F., "Contribution to Discrete Implicit Gradient and Discrete Adjoint Method for Aerodynamic Shape Optimisation," *European Conference on Computational Fluid Dynamics*, ECCOMAS, 2004.

[16]Destarac, D. and van der Vooren, J., "Drag/Thrust Analysis of Jet-Propelled Transonic Transport Aircraft; Definition of Pysical Drag Components," *Aerospace Science and Technology*, Vol. 8, 2004, pp. 545–556.

[17]Vanderplaats, *D.O.T. Users Manual*, Vanderplaats Research & Developement, Inc, 1995.

[18]Bellman, R., *Adaptive Control Processes: A Guided Tour*, Princeton University Press, 1961.

[19]Laurenceau, J. and Sagaut, P., "Building Efficient Response Surfaces of Aerodynamic Functions with Kriging and Cokriging," *AIAA Journal*, Vol. 46, No. 2, 2008, pp. 498–507.

[20]Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–435.

[21]Liu, W., *Development of Gradient-Enhanced Kriging Approximations for Multidisciplinary Design Optimization*, Ph.D. thesis, University of Notre Dame, Notre Dame, IN, USA, 2003.

[22]Kim, S. and Chung, H.-S., *Computational Science and Its Applications - ICCSA 2006*, chap. Multiobjective Optimization Using Adjoint Gradient Enhanced Approximation Models for Genetic Algorithms, 2006, pp. 491–502.

[23]Sasena, M. J., *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*, Ph.D. thesis, University of Michigan, Department of Mechanical Engineering, Ann Arbor, Michigan, USA, 2002.

[24]Booker, A. J., Dennis, J. E., Jr., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W., "A Rigorous Framework for Optimization of Expensive Functions by Surrogates," Tech. Rep. TR-98-47, Center for Research on Parallel Computation, 1998.

[25]Jones, D. R., "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, , No. 21, 2001, pp. 345–383.

[26]Meunier, M., "Simulation and Optimization of Flow Control Strategies for Novel High-Lift Configurations," *25th Applied Aerodynamics Conference*, No. AIAA-07-4276, AIAA, 2007.

[27]Cox, D. and John, S., "SDO: A Statistical Method for Global Optimization," *Multidisciplinary design optimization (Hampton, VA, 1995)*, SIAM, Philadelphia, PA, 1997, pp. 315–329.

[28]Wolpert, D. H. and Macready, W. G., "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, April 1997, pp. 67–82.