

Kriging-based optimization applied to flow control

R. Duvinéau^{1,*} & P. Chandrashekar²

¹ *INRIA Sophia-Antipolis Mediterranée, OPALE Project-Team, France*

² *Tata Institute of Fundamental Research, Center for Applicable Mathematics, Bangalore, India*

SUMMARY

The automatic optimization of flow control devices is a delicate issue, due to the drastic computational time related to unsteady high-fidelity flow analyses and the possible multimodality of the objective function. Thus, we experiment in this article the use of kriging-based algorithms to optimize flow control parameters, since these methods have shown their efficiency for global optimization at moderate cost. Navier-Stokes simulations, carried out for different control parameters, are used to build iteratively a kriging model. At each step, a statistical analysis is performed to enrich the model with new simulation results by exploring the most promising areas, until optimal flow control parameters are found. This approach is validated and demonstrated on two problems, including comparisons with similar studies: the control of the flow around an oscillatory rotating cylinder and the reduction of the intensity of a shock wave for a transonic airfoil by adding a bump to the airfoil profile. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: Flow control ; Optimization ; Kriging

*Correspondence to: INRIA, 2004 route des Lucioles, BP 06902 Sophia-Antipolis, France

Contract/grant sponsor: Publishing Arts Research Council; contract/grant number: 98–1846389

Received 3 July 1999

Copyright © 2000 John Wiley & Sons, Ltd.

Revised 18 September 2002

1. INTRODUCTION

Flow control strategies based on pulsating jets, suction or locally moving walls, have become of great interest to modify the characteristics of complex flows, without modifying too much a baseline design. As example, such devices can efficiently modify the dynamics of detached flows to control the vortex shedding, or perturbate the interaction between a shock wave and a boundary layer to stabilize fluttering phenomenon. Contrary to classical shape design approaches, flow control strategies can efficiently deal with a large variety of problems, with only small modifications of the shape of the body of interest. Moreover, control laws can be adapted to various flow conditions, yielding more robust solutions. Therefore, flow control strategies are particularly appealing for drag reduction in the automotive industry for instance, or stall control in aerodynamics.

However, the parameters of the control law (e.g. actuation frequency or location) should be set precisely to obtain the flow modification expected. In the past, the determination of these parameters has been achieved experimentally by trial-and-error approaches or parametric studies [1, 2]. Now, numerical studies allow to optimize automatically the control parameters for a given problem, by coupling the simulation procedure with an automated search for optimal control parameters. This has been successfully achieved in some previous works. For instance, it has been shown in [3, 4] that an automatic optimization procedure can determine suitable control parameters for a synthetic jet for stall control purpose. In [5, 6], the optimal control of the flow around a rotating oscillatory cylinder has been demonstrated.

However, this optimization procedure is usually very expensive, from computational point

of view, since several dozens, or even hundreds, of costly flow simulations are required to determine accurately the optimal parameters. Therefore, the use of reduced-order models, such as Proper Orthogonal Decomposition (POD), has emerged to solve optimality conditions at low cost [7, 8]. Nevertheless, this approach has been faced to some difficulties, because the reduced-order model, built for some initial flow conditions, cannot represent accurately the flow if the underlying physical phenomenon changes too much during the optimization phase. Therefore, some update strategies have recently been introduced with success [8].

In this article, an alternate strategy is explored to determine optimal flow control parameters with a reasonable computational cost. During an initial phase, some simulations are carried out for different control parameters, which allows to build a kriging model. This is a statistical model that can predict the objective function value with respect to the control parameters, but also provides an estimate of the model error. These features are used during a second phase to determine the optimal control parameters, by performing a few number of additional simulations in most promising areas, in a statistical sense.

The purpose of this study is to test the proposed algorithm in the particular context of flow control, that is characterized by complex and expensive simulations (turbulent and unsteady), yielding cost functions with possible several minima and tedious gradient computation. In particular, the results obtained with the proposed algorithm are compared with those using other optimization strategies to conclude about its efficiency in this context.

In a first section, the proposed strategy is described and illustrated with problems involving analytical functions. Then, the proposed method is validated and demonstrated on two flow control problems, including comparisons with similar studies: the control of the flow around an oscillatory rotating cylinder and the reduction of the shock wave for a transonic airfoil.

2. KRIGING MODEL OF THE FLOW RESPONSE

Kriging models (also called Gaussian process models) belong to response surface methods, that allow to predict a function value f at a given point x , on the basis of a set of known function values $F_N = \{f_1, f_2, \dots, f_N\}$ at some points $X_N = \{x_1, x_2, \dots, x_N\} \subset \mathbb{R}^d$, that are stored in a database.

Kriging models [9, 10] treat the response of some experiment as if it were a realization of a stochastic process. In physical experiments, there are independent random errors which make the assumption of a stochastic process plausible. But in a computer experiment, where the response is the output of a computer code which is deterministic, its plausibility is less clear. However, this approach has proved its efficiency to model computer experiments in complex fields.

In the following, we adopt the Bayesian viewpoint of Gaussian processes [11]. The vector of known function values F_N is assumed to be one realization of a multivariate Gaussian process with joint probability density:

$$p(F_N) = \frac{\exp\left(-\frac{1}{2}F_N^\top C_N^{-1}F_N\right)}{\sqrt{(2\pi)^N \det(C_N)}}, \quad (1)$$

where C_N is the $N \times N$ covariance matrix. The element C_{mn} of the covariance matrix gives the correlation between the function values f_m and f_n obtained respectively at points x_m and x_n . We assume that these values are correlated, since they correspond to underlying physical phenomena. This is expressed in terms of a correlation function c , i.e., $C_{mn} = c(x_m, x_n)$.

Now, we suppose that we would like to evaluate the function value at a new point x_{N+1} . When adding a new point x_{N+1} , the resulting vector of function values F_{N+1} is assumed to

be a realization of the $(N + 1)$ -variable Gaussian process with joint probability density:

$$p(F_{N+1}) = \frac{\exp\left(-\frac{1}{2}F_{N+1}^\top C_{N+1}^{-1}F_{N+1}\right)}{\sqrt{(2\pi)^{N+1} \det(C_{N+1})}}. \quad (2)$$

Using the rule of conditional probabilities $p(A|B) = p(A, B)/p(B)$, we can write the probability density for the unknown function value f_{N+1} , given the data (F_N) as:

$$p(f_{N+1}|F_N) = \frac{p(F_{N+1})}{p(F_N)}. \quad (3)$$

In order to simplify this expression we notice that the $(N + 1)$ -variable covariance matrix C_{N+1} can be written as

$$C_{N+1} = \begin{bmatrix} C_N & k \\ k^\top & \kappa \end{bmatrix}$$

where

$$k = [c(x_1, x_{N+1}), c(x_2, x_{N+1}), \dots, c(x_N, x_{N+1})]^\top \quad \text{and} \quad \kappa = c(x_{N+1}, x_{N+1}).$$

This allows us to write the inverse of correlation matrix as

$$C_{N+1}^{-1} = \begin{bmatrix} M & m \\ m^\top & \mu \end{bmatrix}$$

where

$$M = C_N^{-1} + \frac{1}{\mu} m m^\top, \quad m = -\mu C_N^{-1} k, \quad \mu = (\kappa - k^\top C_N^{-1} k)^{-1}.$$

Then, we obtain that the probability density for the function value at the new point is

$$p(f_{N+1}|F_N) \propto \exp\left[-\frac{(f_{N+1} - \hat{f}_{N+1})^2}{2\sigma_{f_{N+1}}^2}\right],$$

where

$$\hat{f}_{N+1} = k^\top C_N^{-1} F_N \quad \sigma_{f_{N+1}}^2 = \kappa - k^\top C_N^{-1} k. \quad (4)$$

Thus, the probability density for the function value at the new point x_{N+1} is also Gaussian with mean \hat{f}_{N+1} and standard deviation $\sigma_{f_{N+1}}$. Therefore, the most likely value at the new

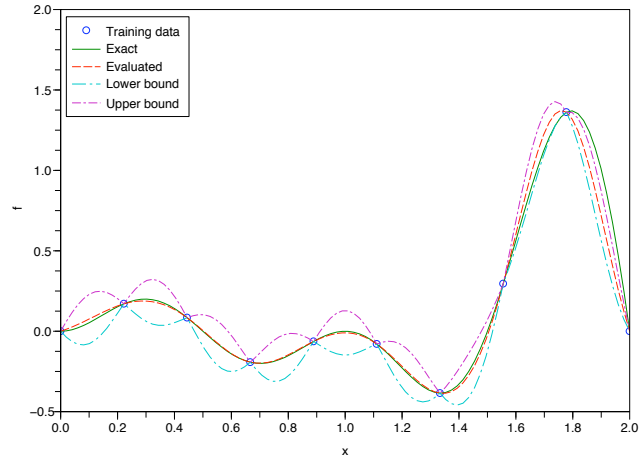


Figure 1. Example of kriging approximation of an analytical function.

point x_{N+1} is \hat{f}_{N+1} . This value will be considered as the prediction of the kriging model. The variance $\sigma_{f_{N+1}}$ can be interpreted as a measure of uncertainty in the value prediction. The function value can be expected to vary in some range like $[\hat{f} - 3\sigma, \hat{f} + 3\sigma]$.

The function approximation using kriging model is illustrated in figure 1. We consider the approximation of the function:

$$f(x) = x(1-x)\sin(2\pi x), \quad x \in [0, 2],$$

using 10 known points. One can see that the kriging approximation is satisfactory. The variance provides an estimate of the uncertainty of the model that allows to pick areas where the model is of poor quality.

The covariance function must reflect the characteristics of the output of the computer code since, it represents the way the different point values are correlated. In the absence of any knowledge regarding the unknown function, the most commonly used correlation function is

an exponential; in this work we take the correlation function of the form:

$$c(x, y) = \theta_1 \exp \left[-\frac{1}{2} \sum_{i=1}^d \frac{(x_i - y_i)^2}{r_i^2} \right] + \theta_2,$$

where $\Theta = (\theta_1, \theta_2, r_1, r_2, \dots, r_d)$ are some parameters to be determined. The first term is a distance-dependent correlation between the function values at two data points; if their distance is small compared to the length scales r_i , the exponential term is close to one while it decays exponentially as their distance increases. Practical studies have shown that the use of a non-isotropic scaling is beneficial for the problems considered here. The parameter θ_1 scales this correlation. In the second term, θ_2 gives an offset of the function values from zero.

It now remains to find the parameters Θ in the correlation function. These parameters are determined by maximizing the joint probability density $p(F_N)$. Indeed, we want the constructed model to be most consistent with the observed data. This is equivalent to minimizing the log-likelihood function given by:

$$\mathcal{L} = F_N^\top C_N^{-1} F_N + \log \det(C_N)$$

This function is known to be multi-modal; hence a PSO (Particle Swarm Optimization) [12] technique is employed for this work, that has the capability to avoid local minima. There are many practical issues that must be taken care of so that all the computations are stable; we follow [13] in this respect. The hyper-parameters must be non-negative; hence it is better to work with the logarithm of the parameters so that they always remain positive. The correlation matrix can be ill-conditioned in which case the computation of its inverse will not be accurate. If the condition number is above a specified tolerance, then the log-likelihood is taken to be a large positive value. If the condition number is within the specified tolerance, an LU decomposition of C_N is first performed; then $C_N^{-1} F_N$ and $C_N^{-1} k$ are computed using the LU decomposition. The logarithm of the determinant of C_N is also computed using the LU decomposition

$$\log \det(C_N) = \sum_{n=1}^N \log L_{nn} + \sum_{n=1}^N \log U_{nn}$$

This avoids the under-flow problem associated with multiplying a large number of small numbers which would be the case if the matrix C_N is badly scaled. An upper limit of $1/\epsilon$ is imposed on the condition number of C_N , where ϵ is the machine precision.

Again following [13] we scale the coordinates and function values. Moreover, the parameters in the covariance function are constrained by upper and lower bounds:

$$\begin{aligned} \theta_1 &\in [10^{-3}, 1] \\ \theta_2 &\in [10^{-3}, 1] \\ r_i &\in [10^{-2}, 10], \quad i = 1, \dots, d \end{aligned}$$

3. OPTIMIZATION STRATEGY BASED ON KRIGING MODELS

The optimization strategy used in this study is based on the iterative construction of a kriging model (see [14] for a review paper on these methods). The use of such a model for optimization must be an iterative process since it is not possible to build a model accurate enough to find the optimal parameters in only one step. The model should be updated with the results from flow simulations until some convergence criterion is fulfilled. Therefore, the algorithm is organized in two phases. During the first one, an initial *a priori* database is constructed, that gathers the flow responses (objective function values) corresponding to different control parameter values. The control parameters are chosen in order to explore uniformly the search space, according to a Design Of Experiments (DOE) method. During the second phase, a kriging model is

constructed on the basis of available data and is used to determine which flow simulations should be carried out and added into the database. This second phase is then repeated until convergence of the algorithm.

This constitutes a baseline algorithm, whose efficiency depends on the way the new flow simulations are chosen at each iteration. Since we intend to minimize a cost function, the simplest algorithm in this family can be described as:

1. Build an *a priori* database
2. Construct a kriging model
3. Find the point x^* that minimizes the model
4. Evaluate the point x^* and add it in the database
5. Return to step 2 until convergence

Using such an algorithm, the database and the model are iteratively enriched and reconstructed. The choice of the optimization algorithm used to solve the minimization problem in step 3 is not critical, since this is an inexpensive task. A PSO method is used in practice since it is robust and not sensitive to local optima.

This simple algorithm can provide satisfactory results for some problems, but one can easily find cases for which this algorithm even does not converge to a stationary point [14]. Actually, the efficiency of such an algorithm depends strongly on the capability of the first *a priori* database to capture the main characteristics of the cost function.

However, one can easily improve the robustness and the efficiency of this first algorithm by using the capability of a kriging model to predict the uncertainty related to value predictions. Indeed, for a given design vector x , a kriging model provides not only a prediction of the function value $\hat{f}(x)$, but also an estimation of the uncertainty of this prediction $\sigma(x)$. Therefore,

one can use this knowledge to determine not only the point that minimizes the model, but also areas for which the model is uncertain, i.e. of poor quality. Evaluating such points promotes the minimization of the cost function as well as the improvement of the kriging model.

Thus, we define a *merit function* that can be interpreted as a statistical lower bound of the model, by aggregating the function value and the standard deviation value [14]:

$$f_M(x) = \hat{f}(x) - \rho\sigma(x), \quad (5)$$

where ρ is a positive parameter that allows to balance the two terms. The previous algorithm is modified as:

1. Build an *a priori* database
2. Construct a kriging model
3. For $i = 1$ to p :

Find the point x_i^* that minimizes the merit function $\hat{f}(x) - \rho_i\sigma(x)$

(with $\rho_i = i - 1$ in general)

4. Evaluate the p points $(x_i^*)_{i=1,\dots,p}$ and add them in the database
5. Return to step 2 until convergence

The use of $\rho_1 = 0$ yields the minimization of the kriging model. Moreover, the use of other values, e.g. $\rho_2 = 1$, $\rho_3 = 2$, $\rho_4 = 3$, promotes the exploration of areas where the standard deviation is high and where possible interesting points can be found (see figure 1). These extra points will be useful since they allow to improve the model quality. This approach has been found more robust than the previous one and applied to various problems [13, 15]. In particular, the sensitivity of the results with respect to the choice of the initial database is significantly reduced [16]. Moreover, it is not more expensive, provided that the p function evaluations performed in step 3 are achieved using parallel computing.

The optimization process is stopped according to three possible criteria : the user can set a maximum number of simulations to perform or a target for the cost function value. Moreover, there is a third criterion, related to the iterative construction of the kriging model: the process is stopped when the minimization of the merit functions yields to points already in the database (for a given accuracy).

Nevertheless, this algorithm exhibits some limitations: first, the kriging model can be sensitive to the error originating from the evaluation process of the cost function. Typically, if the cost function is evaluated through a numerical simulation, a low convergence of the solver can lead the search to spurious local optima. Secondly, this approach is limited to problems of rather low dimension. The number of parameters should be typically lower than twenty in order to have an efficient search, according to the literature results. However, this is usually the case for flow control applications. In a previous work [16], this algorithm was used to solve aerodynamic shape optimization problems on the basis of inviscid flow models, with an increasing number of design parameters. In this context, it has been found that the proposed algorithm provides satisfactory results up to 32 variables.

4. APPLICATION TO FLOW CONTROL

The optimization strategy presented above is applied to two flow control problems to determine optimal control parameters. We consider first the incompressible flow around a cylinder at Reynolds number $Re = 200$. The vortex shedding can be controlled by applying an oscillatory rotation to the cylinder, for some frequency and amplitude values. We apply the proposed method to determine optimal frequency and amplitude values and compare the results with similar studies based on other methods.

The second test-case deals with the turbulent compressible flow around an airfoil at transonic regime. To reduce the shock wave that appears at suction side, a cubic bump is added to the airfoil. The position, amplitude and extent of the bump are optimized with the proposed strategy and compared again with similar studies found in literature.

4.1. Oscillatory rotating cylinder

4.1.1. Initial configuration The flow analysis is performed using the NUM3SIS 3D Navier-Stokes flow solver developed at INRIA. It is based on a classical vertex-centered finite-volume formulation using tetrahedral grids. Second-order space accuracy is obtained using MUSCL extrapolation, while a second-order time integration is performed using a classical three-step backward scheme, including a pseudo-time integration. Details can be found in [17].

The grid used to simulate the flow around the cylinder contains about 70,000 nodes (see figure 2). Computations are validated by performing a time-step refinement study; figure 3 shows a comparison of the drag coefficient obtained using two different time steps. Obviously, the coarsest one is fine enough to have an accurate drag history. Results, in terms of Strouhal number St and time-averaged drag coefficient C_d , are compared with other numerical studies [5, 6, 8, 18, 19] (see table I) and a satisfactory agreement is obtained. The flow is obviously characterized by a massive vortex shedding (see vorticity field in figure 4).

4.1.2. Controlled flow Then, we apply a flow control process by imposing an oscillatory rotation, with frequency N and amplitude A . As first exercise, the control parameters (A, N) are chosen such as $A = 3U_\infty$ and $N = 0.75U_\infty/D$, where U_∞ represents the free-stream velocity and D the cylinder diameter, according to the study of He *et al.* [5]. We denote $A^* = A/U_\infty$ and $N^* = ND/U_\infty$ the non-dimensional control parameters. Again, we perform

Ref.	St	C_d
Bergmann <i>et al.</i> (2005)	0.195	1.382
Braza <i>et al.</i> (1986)	0.200	1.400
Henderson (1997)	0.197	1.341
Homescu <i>et al.</i> (2002)	-	1.440
Present study	0.198	1.370

Table I. Comparison of Strouhal number and drag coefficient.

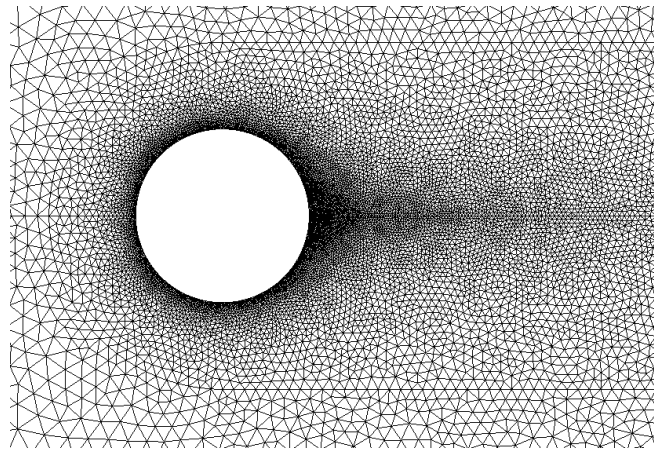


Figure 2. Mesh around the cylinder.

a systematic time-step refinement study to assess the results (see figure 5). A small gap exists between the results obtained using a coarse and a fine time step. However, the drag history using a coarse time step appears to be accurate enough for optimization, provided that only the time-averaged drag is taken into account for optimization.

For these control parameters, the drag coefficient reduction is about 25%. This decrease compares favorably with the results of Bergmann *et al.* [8] and He *et al.* [5], who obtained

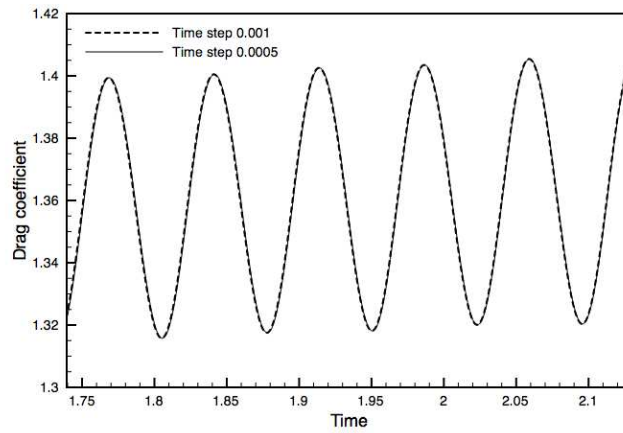


Figure 3. Time step refinement study (no control).

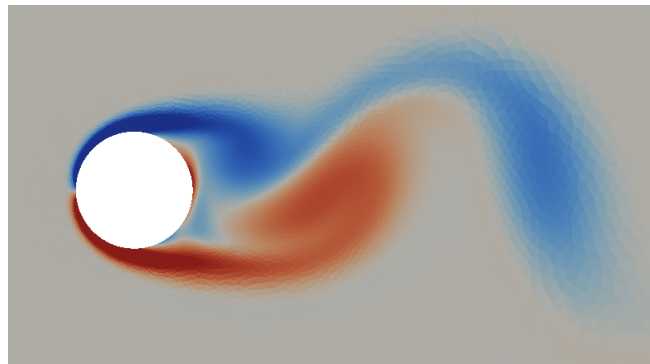


Figure 4. Vorticity field without control.

respectively reductions of 25% using a POD model and 30% using a 2D incompressible Navier-Stokes model. Figure 6 shows that the actuation has significantly modified the flow, by generating small-size high frequency vortices. The flow tends to be symmetric.

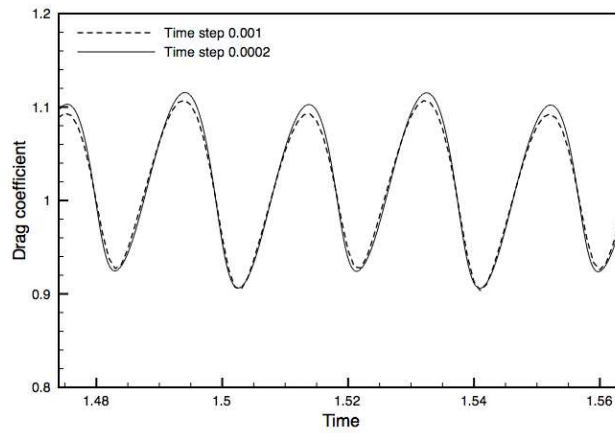


Figure 5. Time step refinement study (controlled flow with parameters from He *et al.* $A^* = 3$ $N^* = 0.75$).

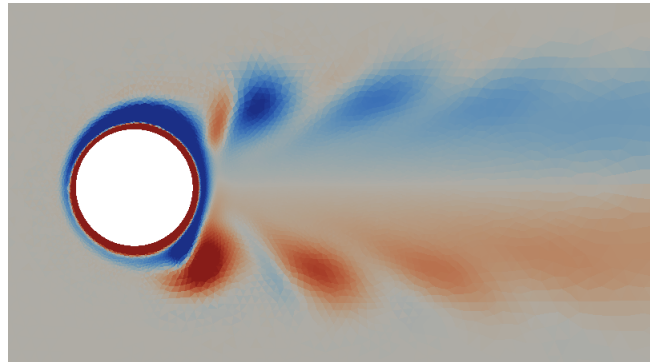


Figure 6. Vorticity field with control (parameters from He *et al.* $A^* = 3$ $N^* = 0.75$).

4.1.3. *Optimal flow control* The optimal control problem reads:

$$\text{Find}(A, N) \text{ to minimise } \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} C_d(t; A, N) dt. \quad (6)$$

t_0 and t_1 are chosen in such a way that i) transient effects have vanished, ii) integration is performed for a sufficient number of periods. In practice, the solution of this problem is found

by coupling the flow solver to the automatic optimization procedure presented above.

The search domain is limited to the intervals $A^* \in [0; 5]$ $N^* \in [0; 1]$. We first construct a database of size 16 using a Latin Hypercube Sampling (LHS) [20] method. At each optimization iteration, four different control parameter sets are tested, that correspond to $\rho_1 = 0$, $\rho_2 = 1$, $\rho_3 = 2$, $\rho_4 = 5$ in the above algorithm. About 20 optimization iterations are performed. The cost function history is plotted in figure 7. As can be seen, the optimization algorithm finds efficient control parameters with a few iterations. If the search is stopped at the fifth iteration, only 32 Navier-Stokes simulations are necessary to obtain an accurate approximation of optimal control parameters. Although this cost is moderate, the proposed approach is more expensive than the TR-POD method used in [8], for which about 10 Navier-Stokes simulations were necessary to obtain optimal control parameters.

We compare the optimal parameters found with those determined by other authors [5, 6, 8] in table II. As can be seen, the optimal parameters are close to those found by He *et al.* One can notice a gap with those found using POD models. These differences can originate from the fact that 3D flow computations are performed in the present study.

Ref.	Method	A^*	N^*	ΔC_d
Bergmann <i>et al.</i> (2005)	POD	2.2	0.53	25%
Bergmann <i>et al.</i> (2008)	TR-POD	4.25	0.74	30%
He <i>et al.</i> (2002)	NS 2D	3.00	0.75	30%
Present study	NS 3D	3.20	0.80	25%

Table II. Comparison of optimal parameters with the literature.

The use of kriging models allows to reconstruct a global model of the cost function with

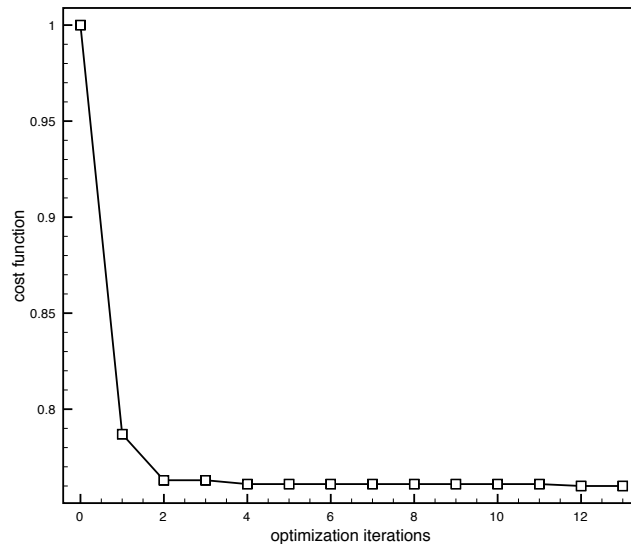


Figure 7. Cost function history.

respect to control parameters, on the basis of the CFD evaluations performed during the optimization phase. Thus, figure 8 shows the evolution of the drag coefficient with respect to the amplitude and frequency actuation, as well as CFD evaluations used to construct the kriging model. One can observe a high-drag area that corresponds to an actuation with low frequency and high amplitude. On the contrary, a quite large low-drag area can be found for high frequency actuation. One can notice that the optimization algorithm has focused CFD simulations in low value areas, but has also performed exploration in other regions.

The drag evolution is clearly multi-modal and counts several local minima. These fluctuations can originate from the physical phenomena or from some noise due to CFD simulations or kriging model. Nevertheless, the overall drag behavior is close to that from other similar studies based on POD model or parametric study using Navier-Stokes simulations [5, 8].

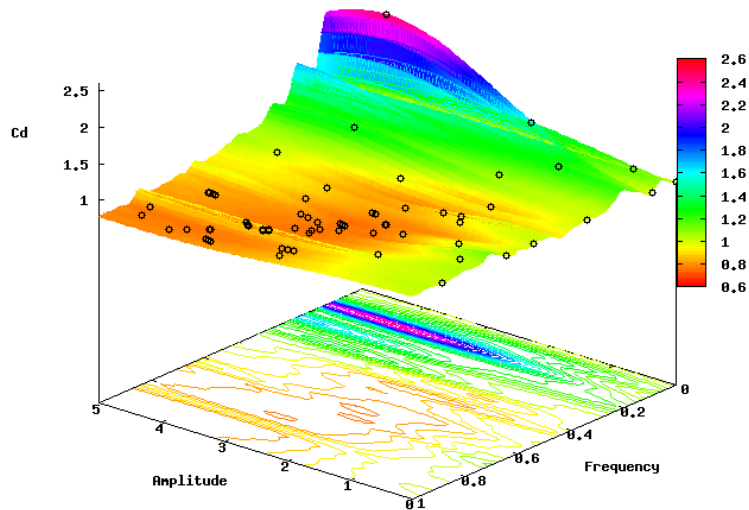


Figure 8. Kriging model surface (markers represent CFD evaluations).

4.2. Shock control bump optimization

Many flows involve the presence of discontinuities or shocks, for example, in high subsonic and transonic flows encountered in aeronautics. The presence of shocks on moving bodies like airfoils and wings leads to considerable wave drag which is undesirable from a performance point of view. In the case of natural laminar flow airfoils, the presence of a shock can cause very adverse pressure gradients leading to separation. Moreover, new concerns about the environment and global warming has made it necessary to improve the fuel efficiency of aircrafts and thus reduce carbon emissions. For existing aircrafts, the idea of adding bumps to the airfoil/wing surface to mitigate the shock strength and hence reduce wave drag is an

attractive option [21]. The addition of a bump of suitable height at an appropriate location, usually near the shock, can lead to the splitting of the shock into a number of weaker oblique compression waves.

The size, shape and location of the bump can be determined in an optimal way so as to achieve maximum reduction in drag. In order to study this problem, we choose a test case of a laminar flow airfoil RAE5243, proposed recently for a multidisciplinary design optimization workshop [22], and solve the problem of minimizing the drag under a lift constraint

$$\min C_d \quad \text{subject to} \quad C_l = 0.82$$

The Reynolds number and Mach number for the test case are 19 million and 0.68 respectively, and we consider the case of fully turbulent flow.

4.2.1. Flow model and its validation The turbulent flow is computed using a RANS code called NUWTUN [23] which is based on finite volume method on block structured grids. The inviscid fluxes are computed using Roe scheme together with MUSCL approach for higher order accuracy, while the turbulence is modeled with $k - \omega$ model. Initially, we validate the grid and numerical scheme by computing the flow at $M_\infty = 0.68$ and $\alpha = 0.77$ degrees, for which experimental results are available. On a grid of 353×97 points (270 on airfoil), the pressure coefficient is shown in figure 9 which indicates that the grid used is sufficiently fine to resolve the flow. The optimization problem involves a lift constraint of $C_l = 0.82$ which is closely satisfied at an angle of attack of 2.5 degrees for the initial RAE5243 airfoil; the forces for the RAE5243 airfoil at this lift coefficient are given in table III which compares favorably with the results of Qin et al. [24]. We notice that the pressure drag C_{d_p} accounts for about 65% of the total drag.

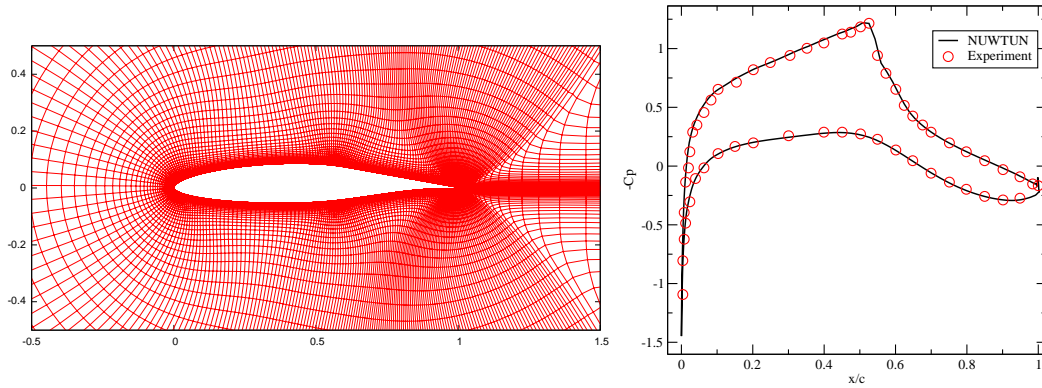


Figure 9. Grid used for CFD and comparison of pressure for RAE5243 airfoil with experiments at

$$Re = 19 \text{ million}, M_\infty = 0.68, \alpha = 0.77 \text{ deg.}$$

Result	α deg.	C_l	C_d	C_{d_p}	C_{d_v}
Present	2.5	0.8244	0.01627	0.01052	0.005757
Qin et al.	-	0.82	0.01622	0.01063	0.005586

Table III. Forces on RAE5243 airfoil at design lift coefficient $C_l = 0.82$.

4.2.2. Bump shape parameterization and grid deformation The bump is assumed to be a cubic curve which is added to the upper surface of the airfoil in such a way that continuity of the first derivative of the airfoil surface is maintained. The bump shape and location is parameterized using four variables as shown in figure 10. The angle of attack α is also taken as a design variable since it is useful to change α to enforce the lift constraint $C_l = 0.82$. Hence

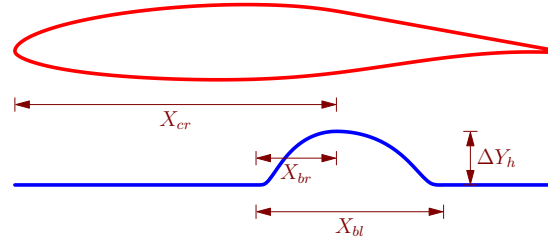


Figure 10. RAE5243 airfoil and parameterization of cubic bump.

we have five design variables and their bounds are as follows

$$0.4 < X_{cr} < 0.8$$

$$0.05 < X_{br} < X_{bl}$$

$$0.1 < X_{bl} < 0.4$$

$$0 < \Delta Y_h < 0.01$$

$$2 < \alpha < 3$$

where α is given in degrees and the airfoil chord is normalized to unity. The upper bound on the variable X_{br} depends on the variable X_{bl} . Hence the variables are transformed so as to make the bounds independent of the variables.

The addition of a bump to the RAE5243 airfoil changes its shape; it is then necessary to modify the CFD grid to conform to the new shape. We use radial basis function (RBF) interpolation [25] to perturb the initial grid for RAE5243 airfoil to conform to the new shape. The displacement of the grid points on the airfoil are known by the addition of the bump, while the points on the outer boundary are held fixed. Using these displacements, an RBF interpolation for the displacement in the x and y coordinates is constructed using thin plate spline functions. The displacement for any interior grid point is then obtained using the RBF

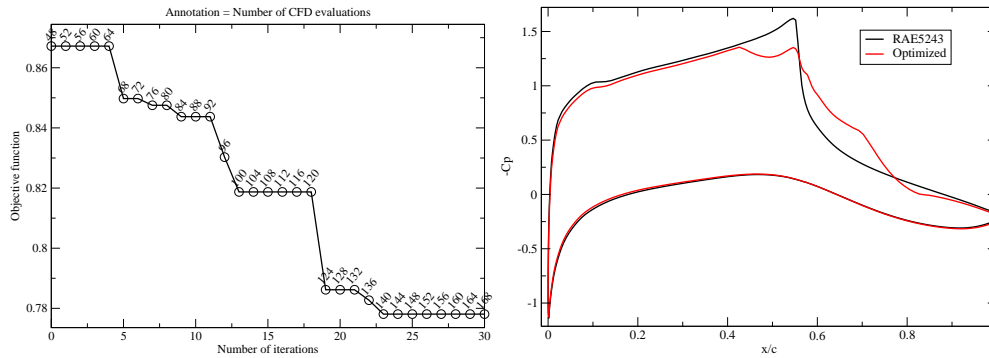


Figure 11. Convergence of objective function for shock control bump optimization, and the initial and final pressure distribution.

interpolation and leads to a smooth deformation of the grid.

4.2.3. Optimal shape control The drag force is minimized under lift constraint using the four shape parameters of the cubic bump and the angle of attack (five design variables). The constraint is enforced using penalty function approach; the function to be minimized is

$$\min \frac{C_d}{C_{d_0}} + 10^4 \max \left(0, 1 - \frac{C_l}{C_{l_0}} \right)$$

where $C_{l_0} = 0.82$ and C_{d_0} is the corresponding drag coefficient for the RAE5243 airfoil, see table III. We first construct a database of 48 design variables using LHS. At each optimization iteration, four different control parameters sets are tested, that correspond to $\rho_1 = 0$, $\rho_2 = 1$, $\rho_3 = 2$, $\rho_4 = 3$, with a total of 30 iterations. The convergence of the objective function and the initial and final pressure distribution on the airfoil are shown in figure 11. The objective function is seen to converge in about 25 iterations and requiring about 140 RANS computations. The pressure distribution shows the weakening of the strong shock on the upper surface by the introduction of a weak compression zone, which is indicated by a small dip in the optimized pressure distribution. This is also shown in figure 12 where the strong shock

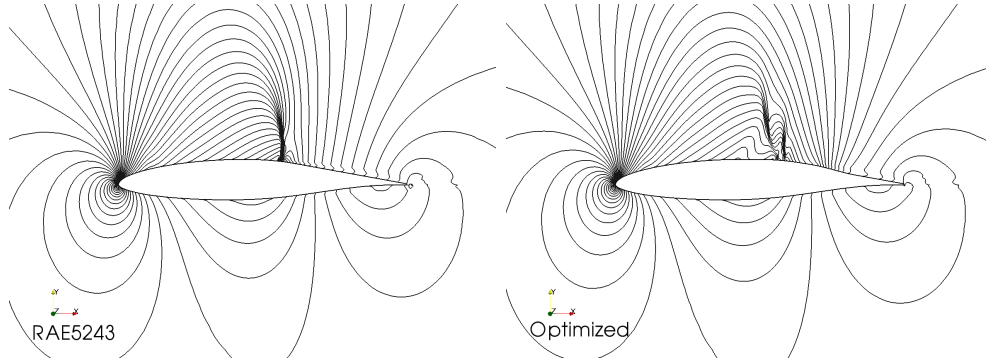


Figure 12. Pressure contours for RAE5243 airfoil and for optimized airfoil with bump.

Case	C_d	ΔC_d	C_{d_p}	C_{d_v}	C_l	AOA
Present	0.01266	-22.2%	0.00680	0.00586	0.8204	2.19
Qin et al.	0.01326	-18.2%	0.00756	0.00570	0.82	-

Table IV. Forces for the optimized airfoil.

is seen to be replaced by a compression wave at the foot of the shock, with the shock being broken into two weaker shocks.

Table IV shows the forces for the optimized airfoil and these are also compared with the results from Qin et al. [24]. We note that the lift constraint is well satisfied and the drag coefficient is reduced by about 22%. This can be compared with the reduction of 18% obtained by Qin et al. [24] using a gradient-based method. The larger reduction in the drag obtained in the present work might be attributed to the use of a global optimization method. The drag is decomposed into pressure drag C_{d_p} and viscous drag C_{d_v} . Since the shock is weakened by the addition of a bump, we see that the pressure drag is reduced by a large extent while the viscous drag increases slightly. This is consistent with the results of Qin et al. [24]. The bump

Case	X_{cr}	X_{bl}	X_{br}	$\Delta Y_h \times 10^{-3}$
Present	0.688	0.399	0.257	8.578
Qin et al.	0.597	0.313	0.206	5.900

Table V. Optimized bump parameters.

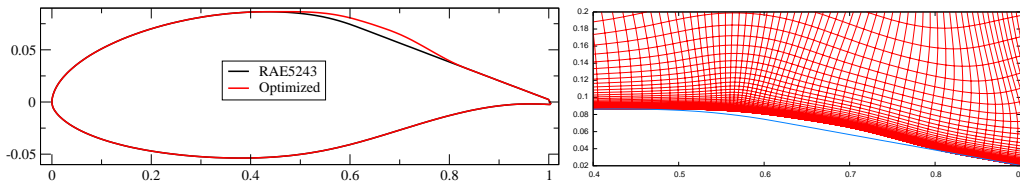


Figure 13. Initial and optimized shapes for shock control bump and grid corresponding to optimal shape.

shape parameters are shown in table V together with those from Qin et al. [24]. The two sets of parameters are different due to the different drag reductions; the higher drag reduction of 22% in our case corresponds to a taller bump which is expected. Figure 13 shows the initial and optimized shapes, and also shows a closeup view of the grid corresponding to the optimal shape.

5. CONCLUSION

A numerical optimization method based on kriging models is developed and tested in the context of flow control problems. Due to the complexity of the physical model considered (unsteady flow, turbulence, etc), the evaluation process is computationally expensive and often generates cost functions with several minima. The proposed algorithm is interesting in this context, since it offers a global optimization strategy for a moderate cost, and is well suited

to problems that include a low number of parameters.

This algorithm is tested for two typical flow control problems and the results obtained are compared with those found in the literature based on other optimization strategies. As shown, the determination of optimal control parameters can be achieved using $O(10)$ - $O(100)$ Navier-Stokes computations, which makes the proposed approach realistic for complex configurations. In terms of accuracy, the results obtained for the oscillatory rotating cylinder are similar with those based on other methods. Moreover, in the case of shock control, it is shown that the use of a global optimization approach yields better solutions instead of being restricted to a local optimum. This feature makes the algorithm appealing for the targeted applications.

Future work will concern the introduction of uncertainty or error estimation in the cost function values stored in the database and used to build the kriging model. Then, the database points will not be considered as exact values anymore. The objective will be to accelerate the search by using low converged simulations and low fidelity models.

REFERENCES

1. Gilarranz J, Traub L, Rediniotis O. Characterization of a compact, high power synthetic jet actuator for flow separation control. AIAA Paper 2002-0127, 2002.
2. Seifert A, Bashar T, Koss D, Shepshelovich M, Wygnanski I. Oscillatory blowing : a tool to do delay boundary layer separation. *AIAA Journal*, 1993; **31**:2052-2060.
3. Duvigneau R, Visonneau M. Optimization of a synthetic jet actuator for aerodynamic stall control. *Computers and Fluids*, 2006; **35**:624-638.
4. Duvigneau R, Hay A, Visonneau M. Optimal location of a synthetic jet on an airfoil for stall control. *Journal of Fluid Engineering*, 2007; **129**:825-833.
5. He JW, Glowinski R, Metcalfe R, Nordlander A, Periaux J. Active control and drag optimization for flow

- past a circular cylinder: I. oscillatory cylinder rotation. *Journal of Computational Physics*, 2000; **163**:83–117.
6. Homescu C, Navon IM, Li Z. Suppression of vortex shedding for flow around a circular cylinder using optimal control. *International Journal for Numerical Methods in Fluids*, 2002; **38**.
 7. Bergmann M, Cordier L, Brancher JP. Optimal rotary control of the cylinder wake using POD reduced order model. *Phys. Fluids*, 2005; **17**.
 8. Bergmann M, Cordier L. Optimal control of the cylinder wake in the laminar regime by trust-region methods and POD reduced-order models. *J. Comput. Physics*, 2008; **227**.
 9. MacKay D J C. Introduction to Gaussian processes. *Neural Networks and Machine Learning*, 1998; **168**:133-165.
 10. Sacks J, Welch W, Mitchell T, Wynn H. Design and analysis of computer experiments. *Statistical Science*, 1989; **4**:409–435.
 11. Williams CKI. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. Technical Report NCRG/97/012, Neural Computing and Research Group, Dept. of Comp. Sci. and App. Math., Aston University, Birmingham, 1997.
 12. Venter G, Sobieszczanski-Sobieski J. Particle swarm optimization. *AIAA Journal*, 2003; **41**:1583–1589.
 13. Büche D, Schraudolph N N, Koumoutsakos P. Accelerating evolutionary algorithms with Gaussian process fitness function models. *IEEE Tran. on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 2005; **35**.
 14. Jones D. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 2001; **21**:345–383.
 15. Duchaine F, Morel T, Gicquel L. Kriging based optimization tool for the design of aeronautic gas turbine combustion chambers by use of a 3D reacting CFD solver. *AIAA Journal*, 2009; **47**:631–645.
 16. Chandrashekar P and Duvigneau R. Study of some strategies for global optimization using Gaussian process models with application to aerodynamic design. INRIA Research Report RR-6964, June 2009.
 17. Kloczko, T. Concept, architecture and performance study for a parallel code in CFD. In Proc. of Parallel CFD Conference, May 2008.
 18. Braza M, Chassaing P and Minh H. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *J. Fluid Mechanics*, 1986; **165**:79–130.
 19. Henderson R D. Nonlinear dynamics and pattern formation in turbulent wake transition. *J. Fluid Mechanics*, 1997; **352**.

20. Keane A, Nair P. *Computational Approaches for Aerospace Design: The Pursuit of Excellence*. John-Wiley and Sons, 2005.
21. Ashill, P R, Fulker, J L and Shires, A. A novel technique for controlling shock strength of laminar flow aerofoil sections. In the First European Symposium on Laminar Flow, Hamburg, March 1992.
22. <http://jucri.jyu.fi/?q=node/25>
23. <http://nuwtun.berlios.de>
24. Qin, N, Wong, W S and Le Moigne, A. Three-dimensional contour bumps for transonic wing drag reduction. In Proc. of IMechE Vol. 222, Part G: J. Aerospace Engineering, 2008.
25. Kumar, K and Nair, M T. A mesh deformation strategy for multiblock structured grids. Project Document CF 0708, National Aerospace Laboratories, Bangalore, 2007.