OPTIMIZATION LOOP BASED ON A CFD RANS CODE.

F. Duchaine*, L. Y. M. Gicquel*, T. Poinsot**
***CERFACS - Toulouse France, ****IMFT - Toulouse France**

Keywords: *CFD optimization platform - Metamodel - Coupling device*

Abstract

The design of gas turbine engines is a complex and time consuming engineering task involving numerous iterative processes. Among all the devices composing an engine, the combustion chamber benefits from recent progress in High Performance Computing for Computational Fluid Dynamics (CFD) which is intensively used. Although CFD offers substantial improvements in flexibility and costs when compared to experimental approaches, the number of CFD computations still remains large and engineering intensive. A strategy to alleviate the CFD specialist involvement in reaching an optimal solution is desired to fully capitalize on the method. One solution aims at automating the optimization process performed with CFD computations.

From the optimization point of view, the expensive step is the CFD run necessary to evaluate the objective function value representing the performance of a set of control parameters. The promising method developed in this work consists in performing optimization with classical techniques based on an inexpensive metamodeling [1] of the objective function. The idea is to reduce the number of CFD computations while proposing a rigorous framework to converge robustly to an optimal design.

Implementation issues such as efficiency, numerical stability of the model, techniques to accelerate the optimization process, exploration / exploitation conflict and the different levels of parallelism involved in CFD optimization are addressed in this paper. Practicality of the optimization tool is here inherited from the powerful coupling device PALM [2].

Illustration of the approach when applied to a simple engine cooling system is presented: Validations steps and comparisons of the proposed surrogate approach with the very well known optimization method Simplex [3] are shown.

1 Introduction

Due to drastic norms on pollution, aeronautical engine manufacturers need to propose new technological solutions for the next generation of aero-engine devices. Indeed, ambitious NOx reduction targets of 80% are set for 2020. These strict objectives yield many research projects aiming at the definitions and studies of innovative combustion systems. Among all the initiatives from the European research community, the project No. FP6-502961 INTEgrated Lean Low Emission CombusTor Design Methodology (INTELLECT D.M.) investigates new design rules and methodologies for the definition and validation of low emission combustors. The specificity of the retained technology for NOx reduction is that the combustion chambers will more likely operate with an excess of air; the primary reason being to reduce significantly the flame temperature and thereof the NOx production. With this approach, up to 70% of the total air flow through the chamber is premixed with the fuel before entering the reaction zone. As a consequence, the design of the pre-diffuser and cooling systems become crucial as they are key elements to provide good air distribution throughout the chamber and to avoid high temperature peaks. Another characteristic of this technology is the narrowing of the operating range of the new burners. In-

deed, combustion instabilities arise rapidly even with small deviations from the initially defined operating point.

This work, initiated by TURBOMECA (SAFRAN Group), focuses on the first issue and analyzes the suitability of optimization techniques for the design of an areo-engine cooling system. The main idea is to assess and construct an automatic optimizer based on a Computational Fluid Dynamics (CFD) solver so as to obtain design solutions satisfying given chamber outlet temperature profiles. Nowadays, optimization takes a very large place in the scientific community and real world offers numerous examples where one can use optimization. However, optimization methods and performances depend on the type of problem to be treated. The coupling of an optimizer with a CFD solver imposes obvious restrictions such as computer power, time and memory... Among existing methods, the most performant optimization processes are the gradient-based techniques. They are nonetheless not accessible with standard CFD codes and adjoint solvers [4] are often mandatory to obtain good evaluations of the gradients. More recent methods, such as evolutionary algorithms (*i.e.* genetic algorithms [5]), are very often used in CFD optimization. They give good results and coincide with the philosophy of our problematic, that is they allow several computations of differents design points at the same time. The number of objective function computations (CFD runs) remains important with this approach and the overall response-time of the process is large. An original approach to avoid such problems and adopted in this work, consists in using a low order model, or surrogate model [6] [7], to substitute high fidelity CFD runs.

The optimization tool developed at CERFACS is based on the powerful coupling device PALM. This software is written by *Climate Modelling and Global Change* team - CERFACS - for the operational oceanography project MERCATOR [1].

The paper is organized as follow: First, the optimization tool is presented, giving its main characteristics, insisting on the optimization strategy and on the chosen surrogate model. At this occasion, issues of shape parameterization, automatic mesh generation / deformation are discussed. Then, a simple optimization problem is given and numerical results are shown. Finally, we propose some perspectives to our work.

2 MIPTO: Management of an Integrated Plateform for auTomatic Optimization

2.1 Overview of MIPTO

An overview of the optimization tool is presented on Fig. 1. Each rectangular box with thick lines runs independently on different processors. They coincide with the first level of parallelism taken into account by PALM. The second level corresponds to the capacity of handling parallel codes as underlined in the description of PALM. MPI communications necessary for the exchange of data between processes (*i.e.* meshes and fields) are depicted by arrows.

The surrogate assisted optimization algorithm is briefly detailed on the flowchart of Fig. 2. The corresponding unit, named “Optimization algorithm" exchanges the optimization parameters and the associated objective function value with the rest of the application. This unit automatically asks for the real objective function value or its estimation when necessary.

An interface manages and launches the CFD computations in a “multi-run" way: It aims at automatically distribute the tasks depending on the available ressources.

Dealing exclusively with the CFD computations, the optimization parameters are first transformed by the so-called “Pre-processing" unit: It handles the mesh as far as design optimization is concerned as well as the boundary conditions when controlling the operating point. Note that automatic mesh generation usually infers automatic quality assessment of the grid directly parameterized by the optimization parameters. Such tasks remain a central problem for proper operation of the optimization device. Two meth-

[1] http://www.mercator-ocean.fr/

ods are conceivable:

- Given an initial shape and mesh, a new mesh is produced based on the new design parameters and by deforming the previous grid. Several techniques exit and one can cite Laplacian, Spring and explicit methods. These approaches have the advantage of conserving the total number of nodes and mesh connectivity garrantying homogeneous MPI communications between units during the optimization process. Mesh deformation is however limited to small geometrical changes to avoid negative cell volumes or poor quality meshes,

- The second method consists in implementing a fully automatic mesh generator in the loop. The main difficulties concern the interface between the shape modeller and the mesher as well as the prohibitive computational time issued by the generation of a new mesh.

In the context of this study, both methods are used to construct $2D$ meshes. In particular, the remeshing approach includes *Ipol* and *Delaundo*, the $1D$-shape and the $2D$-domain meshers developed by Müller [8]. It is important to note that moving nodes and remeshing techniques often need smoothing steps to enhance cell quality.

The CFD solver used in the study is N3S-Natur, a parallel Reynolds Average Navier Stokes (RANS) code dealing with two phase reactive flows on unstructured meshes. N3S-Natur is distributed by INCKA.

Finally, at the end of each CFD computations, flow solutions are post-treated in order to evaluate the performance of the optimization variables through an objective function. At the moment, only single objective function problems are possible but extension to multi-objective calculations can be done by summing the corresponding fitness or by implementing a Pareto approach in the optimization strategy.

In this paper, the surrogate assisted algorithm is compared to the Simplex optimization approach. The flexibility of PALM allows quick modifications in the application, as the use of another optimization algorithm or CFD code. In the case of the Simplex PALM application, the Simplex unit exchanges optimization parameters and objective function values directly with the CFD branch.

2.2 PALM software, a powerful interface for MIPTO

CERFACS has an important experience in code coupling and is in charge of the PALM project [2]. The universality of this software and its capacities have lead us to use it in development of MIPTO.

The PALM project aims at implementing a general tool to easily integrate high performance computing applications in a flexible and evolutive way. It is originally designed for oceanographic data assimilation algorithms, but its domain of application extends to every kind of scientific application. In the framework of PALM, applications are split into independant elementary components, *ie.* the optimization algorithm, the CFD code, the pre- and post-treatement of the CFD results, that can exchange data. Its main features are:

- The dynamic launching of the coupled components due to MPI2,

- The full independence of the components from the application algorithm,

- The parallel data exchanges with redistribution,

- The separation of the physics from the algebraic manipulations (performed by the PALM algebra toolbox).

To generate a PALM application, the first necessary step requires the existing codes to be compatible with PALM. Modifications concerning parallelism management are rapidly achieved and some PALM instructions such as *Put* and *Get* which will allow to exchange data from one code to another must be inserted in the source codes. An interface or ID card of the code describing

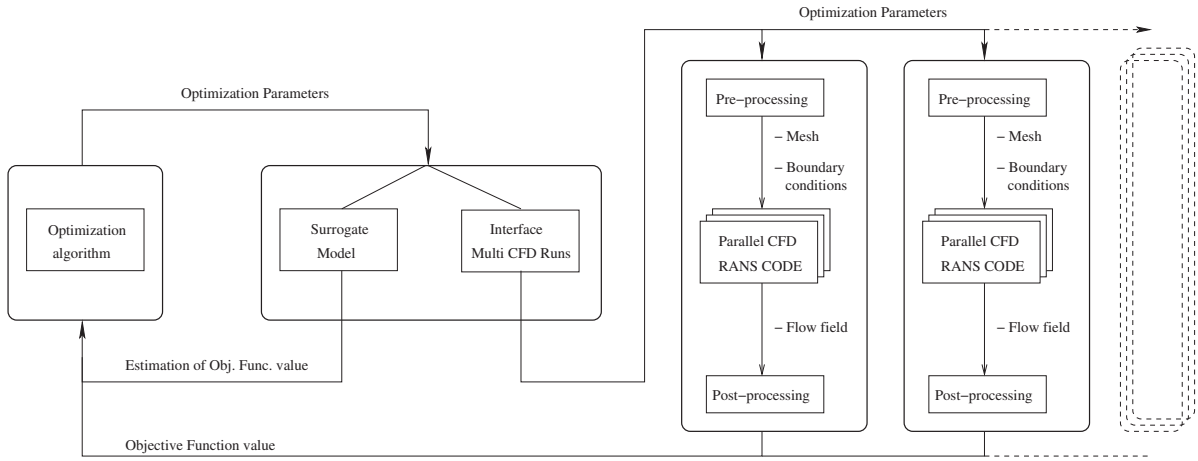

Fig. 1 Overview of the optimization tool MIPTO

the exchanged data have to be written. Once all the components of the application are available, PrePALM, the PALM graphical pre-processor, is used to develop the structure of the algorithm: Order the launch of the components either concurrently or successively, in loops or conditionally.

PALM is able to handle many parallel codes as well as several instances of the same code. This particularity is very usefull in the context of optimization with CFD codes. Indeed, most of the codes we use are parallel, and the possibility to make different computations at the same time will compress the whole clock time needed to complete the set of simulations prescribed by an optimization application. Finally, PALM automatically manages the processors distribution for an application between the different units and based on a user-defined priority fashion.

2.3 Optimization strategy

The algorithm to be used is detailed on Fig. 2. Prior to the optimization loop, one requires the use of the Design Of Experiments (DOE) approach to initialize the surrogate database D. To extract as much information as possible from a limited set of computer experiments on the whole decision space, the user can choose between three methods for space filling: Latin Hypercube Sampling [9], Halton sequence and Hammersley sequence [10]. The database D can also be the result of previous optimization processes preserving the computing effort through a hot start of the method. In this context, the surrogate model stands for a global approximation of the fitness function over the entire domain of optimization parameters. Thus, its construction requires all the sample points.

The optimization loop itself consists in finding new interesting samples to enhance the surrogate database hence improving the response function approximation. The loop continues for a pre-defined number of iterations or when the maximum allowed CFD computations is reached. At a given iteration, the new sample points are determined from the found potential optimums on the metamodel. This optimization of the model is obtained from a "Low memory Broyden Fletcher Goldfard Shanno Bounded" (LBFGS-B) which corresponds to the quasi-Newton algorithm proposed by Byrd *et al.* [11]. One iteration of the surrogate-based method performs several LBFGS-B runs with different starting points spread over the decison domain. The resulting optimums are then evaluated with the CFD code if there are far enougth from existing points in the surrogate database or in the so call “avoid database". The “avoid database" contains the operating conditions that have lead to ill-posed CFD solutions. The radius of the hypersphere surrounding the database points where no new sam-

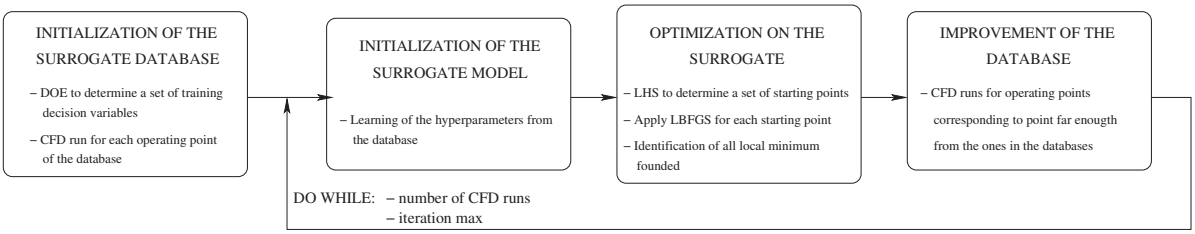

Fig. 2 Flowchart of the surrogate assisted optimization algorithm

ple point is authorized is automatically adjusted during the process.

As the total number of processors of the application is constant during the optimization procedure, if the number of operating points provided by the LBFGS-B step is not proportional to the number of simultaneously allowed CFD runs, other points are added by “crossing" the best sample with other ones. Such a crossing technique is inspired by evolutionnary methods [12].

Using a gradient-based optimization algorithm coupled with a surrogate approach yields good exploitation of the available data which is essential for fast convergence. The multi-start aspect of the gradient optimizer enforced by the use of a merite function instead of the approximate fitness function allows a great exploration of the decision domain. The crossing operation and the notion of hypersphere also take part in the exploration. The notion of exploration is needed in the case of multimodal function in order to find a global extremum. The merite function is a weighted sum of the fitness function approximation and the estimation of the density of points ρ at a given place: $f_M = f_{obj} - \alpha\rho$, $\alpha > 0$. Gaussian processes naturally include the notion of density through the variance of the estimation $\sigma^2_{\hat{t}}$ which is high where the prediction $\hat{t}$ is probably not accurate. So the optimization is performed on $f_M = \hat{t} - \alpha\sigma_{\hat{t}}$.

It is important to underline that a surrogate-based optimization algorithm not only gives a candidate for an optimum but also a general trend of the fitness through the parameters over the decision space. Hence, the user can extract interresting regions and perform local searches to enhance the optimal solution.

2.4 Gaussian processes to approach an expensive objective function

A wide variety of surrogate models are used in the literature to approximate expensive fitness functions in the context of optimization. The most prominent methods among all approaches are polynomial models [13], artificial neural networks [14], radial basis function networks [15] and Gaussian processes (GPs) [7]. Among the previous empirical models, Gaussian processes appear to be the most promising for fitness function approximation. Indeed, a GP combines the following decisive properties:

- The implentation of GPs is independent of the number of decision variables,
- GPs can approximate accurately arbitrary functions including multimodality and discontinuities,
- GP contains meaningful hyper-parameters that can be obtained theoretically with an optimization procedure,
- GP yields an uncertainty measure of the predicted value in the form of a standard deviation.

To briefly expose the theorical framework of GP, the notations of MacKay [1] is adopted. Given some noisy data set $D = \{\mathbf{x}_n, t_n\}_{n=1}^N$ consisting of N pairs of L-dimensional input vectors $\mathbf{x}_n$ and scalar outputs t_n, the aim of the process is to find a prediction t_{N+1} at a new point $\mathbf{x}_{N+1} \notin D$. Let’s denote the set of input vectors by $\mathbf{X}_N = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ and the set of corresponding function values by the vector $\mathbf{t}_N = \{t_1, t_2, ..., t_N\}$.

The GP theory gives a univariate Gaussian density for t_{N+1} through the following expression:

$$p(t_{N+1}|\mathbf{X}_{N+1},\mathbf{t}_N) = \frac{1}{Z} exp\left(-\frac{1}{2}\frac{(t_{N+1}-\hat{t}_{N+1})^2}{\sigma^2_{\hat{t}_{N+1}}}\right). \quad (1)$$

where the mean $\hat{t}_{N+1}$ and variance $\sigma^2_{\hat{t}_{N+1}}$ can be expressed as:

$$\hat{t}_{N+1} = \mathbf{k}^T C_N^{-1}\mathbf{t}_N, \quad (2)$$

$$\sigma^2_{\hat{t}_{N+1}} = \kappa - \mathbf{k}^T C_N^{-1}\mathbf{k}. \quad (3)$$

where $\mathbf{k}$ is the covariance vector, C_N the covariance matrix and κ the variance of the GP. There are many ways to impose the covariance matrix C_N with the only constraint that it must generate a none-negative definite covariance matrix for any set of points $\mathbf{X}_N$. In this study, we choose the stationary covariance function of the Gaussian distribution with zero mean. It reads for the two data points $\mathbf{x}_p$ and $\mathbf{x}_q$:

$$C(\mathbf{x}_p,\mathbf{x}_q,\Theta) = \theta_1 exp\left(-\frac{1}{2}\sum_{l=1}^{L}\frac{(x_{p,l}-x_{q,l})^2}{r_l^2}\right) + \theta_2 + \delta_{pq}\theta_3 \quad (4)$$

l represents the l^{th} component of the L-dimensional vectors $\mathbf{x}_p$ while $\mathbf{x}_q$ and $\Theta = (\theta_1,\theta_2,\theta_3,r_l)$ are the hyper-parameters of the covariance function. r_l corresponds to a length scale characterising the direction l. A large length scale means that the ouput value $\hat{t}_{N+1}$ is expected to be essentially a constant function of that input (smoothing effect). The ratio θ_2/θ_1 allows to take into account the mean value of the sample points as a constant regression function. If this ratio is set to zero, then the regression function will be zero, otherwise it tends to the mean value of the sample points as the ratio increases. Finally, θ_3 represents a white input-independant noise applied only on to the diagonal terms of the covariance matrix. θ_3 allows the interpolation not to reach the sample points considering a certain level of noise in the function evaluations $\mathbf{t}_N$.

The covariance matrix C_N, the covariance vector $\mathbf{k}$ and the variance κ of Eq. 2 and 3 are expressed in terms of the covariance function as:

$$C_{N_{ij}} = C(\mathbf{x}_i,\mathbf{x}_j,\Theta),\ (\mathbf{x}_i,\mathbf{x}_j) \in D^2 \quad (5)$$

$$k_i = C(\mathbf{x}_i,\mathbf{x}_{N+1},\Theta),\ \mathbf{x}_i \in D \quad (6)$$

$$\kappa = C(\mathbf{x}_{N+1},\mathbf{x}_{N+1},\Theta) = \theta_1+\theta_2+\theta_3 \quad (7)$$

The hyper-parameters can either be set by the user or retrieved from the data. As the values of the hyper-parameters have a large influence on the result of the GP, an optimal GP is obtained such that the log-likelihood of the given function values $\mathbf{t}_N$ under multivariate Gaussian with zero mean and covariance $C_N = C(\mathbf{X}_N,\Theta)$ is maximal. The expression of the log-likelihood is:

$$\lambda = -\frac{1}{2}\left(log\ detC_N + \mathbf{t}_N^T C_N^{-1}\mathbf{t}_N + N\ log(2\pi)\right) \quad (8)$$

Even if it is straightforward to find the derivatives $\partial\lambda/\partial\Theta$, MacKay [1] shows that λ is often multimodal. For this reason, the evolutionary algorithm proposed by Michalewicz [16] is used to find the optimal hyper-parameters.

As GP is an analytical model, it is possible to access the gradient of $\hat{t}_{N+1}$ and $\sigma^2_{\hat{t}_{N+1}}$. One can note that the vector $\mathbf{k}$ is the only part of the expressions which is function of $\mathbf{x}_{N+1}$. The gradient matrix of $\mathbf{k}$ is:

$$(Grad(\mathbf{k}))_{pl} = \frac{\partial k_p}{\partial x_l} = \theta_1 \frac{x_{p,l}-x_{N+1,l}}{r_l^2} exp\left(-\frac{1}{2}\sum_{l=1}^{L}\frac{(x_{p,l}-x_{N+1,l})^2}{r_l^2}\right) \quad (9)$$

Giving $(Grad(\mathbf{k}))$, it is possible to compute the gradient of $\hat{t}_{N+1}$:

$$Grad(\hat{t}_{N+1}) = (Grad(\mathbf{k}))^T C_N^{-1}\mathbf{t_N} \quad (10)$$

the gradient of $\sigma^2_{\hat{t}_{N+1}}$:

$$Grad(\sigma^2_{\hat{t}_{N+1}}) = -(Grad(\mathbf{k}))^T C_N^{-1}\mathbf{k} - \mathbf{k}^T C_N^{-1}(Grad(\mathbf{k})) \quad (11)$$

and finally, the gradient of $\sigma_{\hat{t}_{N+1}}$:

$$Grad(\sigma_{\hat{t}_{N+1}}) = \frac{Grad(\sigma^2_{\hat{t}_{N+1}})}{2\sigma_{\hat{t}_{N+1}}} \quad (12)$$

To conclude on GP, most of the CPU time requested by the method is due to the inversion of the covariance matrix C_N. To reduce the potential impact of this drawback, numerical tricks as LU decomposition can be used.

3 Numerical results

3.1 Presentation of the test case

The studied configuration consists in a 2D channel ($0.03m \times 0.25m$) in which hot gases flow ($T_h = 1500K$) while two dilution injectors aim at cooling ($T_h = 300K$) the hot stream before it exits the pipe (Fig. 3). This configuration has the particularity of being representative of a cooling process found in the dilution region of a combustion chamber. The attempt of the presented computations is to find the optimal locations of the two cooling injectors to reach a given output temperature profile. From the test case, it is expected that for large values of the dilution injector positions, noted L_{cu} and L_{cl}, the cooling will be less efficient due to poor mixing of the hot gas with the cold one. Note that $L_{cu} \neq L_{cl}$ for the upper and lower injector positions should lead to assymetric exit temperature profile if the inflows are the same. Likewise if the flow rate of the upper and lower injectors are not the same, a none symmetric repartition of the cooling flow should be observed in the main pipe.

The objective function value associated to a set of injector positions is extracted from CFD computations by comparing the obtained exit temperature profiles with a target one through the following expression:

$$f_{obj} = \frac{1}{D_h} \int \left(\frac{T^t(y) - T^c(y)}{T^t(y)} \right)^2 dy \qquad (13)$$

where D_h is the diameter of the channel, T^t is the target temperature and T^c is the computed exit temperature. The expression of the objective function is a root mean square between target and computed temperature profiles. It is non-dimensionalized by the target temperature to yield relative importance to the standard RMS (indeed, the temperature goes from approximatively $300K$ to $1500K$ in the channel). Tests prove this objective function to yield better results than standard RMS, in terms of solution quality and convergence rapidity for this particular configuration. The target temperature profile is presented on figure 5-c.

Since the shape deformations are important in this optimization study, remeshing techniques are adopted. The generated meshes have a mean of 9000 nodes and 17500 cells. Each CFD calculation takes about 15 minutes on 5 processors of a DEC ALPHA.

3.2 Optimization results

The initial database D is designed to contain 20 objective function evaluations using the CFD code. Then, the improvement of the model is done thougth 6 iterations requiring a total of 71 additional CFD computations.

Figure 4 illustrates the impact of improving the model by comparing $\hat{t}$ and $f_M = \hat{t} - \rho\sigma_{\hat{t}}$ after initialization and 6 iterations of the algorithm.

We first note the important reduction of $-\rho\sigma_{\hat{t}}$ within the search domain. Indeed, from the intialization to the end of the process, $\hat{t}$ tends to f_M.

Then, the choice of ρ has lead to a quite homogeneous repartition of the sample points across the domain with higher densities in the regions where optimums are detected. With a lower value of ρ, exploration would have been less privileged giving a more heterogeneous repartition with less sample points. Many points are computed on the bounds of the variables. This is due to the fact that the model often gives poor predictions at these locations because their exists no information about the function outside the search domain

As expected, the method is able to find several attraction regions linked to local optimums. Temperature fields of the global minimum ($L_{cu} = 0.111m, L_{cl} = 0.079m$) and a local minimum ($L_{cu} = 0.117m, L_{cl} = 0.055m$) are presented on figure 5-a and figure 5-b respectively. For comparison, the obtained and desired exit temperature profile are shown on figure 5-c.

Finally, the surrogate assited optimizer gives important informations for designer concerning the tendency of the objective function over the decision space. For the case considered, we see a general orientation of the major attraction region along the line $L_{cu} - L_{cl} = k$, meaning that the ob-

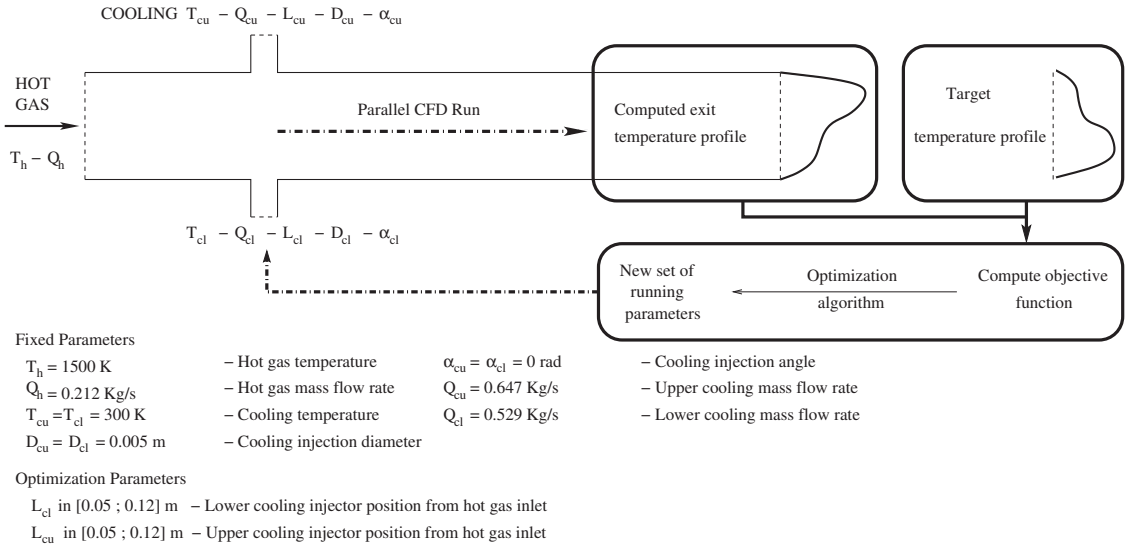

Fig. 3 Optimization configuration

jective corresponds to a fixed separation distance between injectors. A direct consequence of this observation is the possible reduction of the search space of the optimization by replacing the inital parameters by the constraint $L_{cu} - L_{cl} = k$ while keeping for example L_{cu} as a variable.

To conclude on numerical results, we compare the surrogate assisted method with the Simplex method. Figure 6 shows the obtained metamodel after 6 iterations on which 5 Simplex convergence histories are plotted. A mean of 35 CFD runs were necessary to reach a satisfatory convergence of the Simplex algorithm. Depending on the initial guess, Simplex optimizations tend to converge to several distinct optimums that have been found by the surroagte approach. Different drawbacks issued by the Simplex method are clearly evidenced by this application: the local convergence, the importance of the initializations (two very nearby initial guesses may not yield the same result), the chaotic behaviour of the approach (when fronted with a drastic valley, the method leads to well known convergence diseases). The exit temperature profile of the best solution found by the Simplex runs is shown on figure 5-c ($L_{cu} = 0.106m, L_{cl} = 0.086m$).

Finally, the surrogate method gives much more informations and a better optimum than the

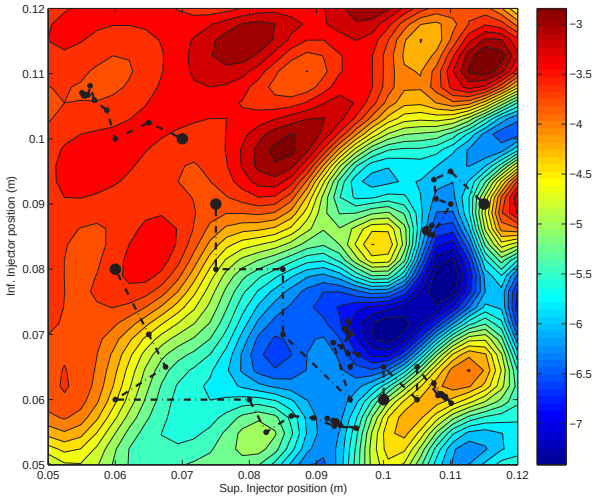

Fig. 6 Comparison between the surrogate assisted algorithm and the Simplex method: dotted lines with black small circles represent convergence histories of 5 Simplex runs initialized on larger black circles

Simplex with less CFD computations and clock time.

4 Conclusion and perspectives

We have presented an effective global optimization method dealing with expensive fitness functions. The proposed algorithm allows to reduce

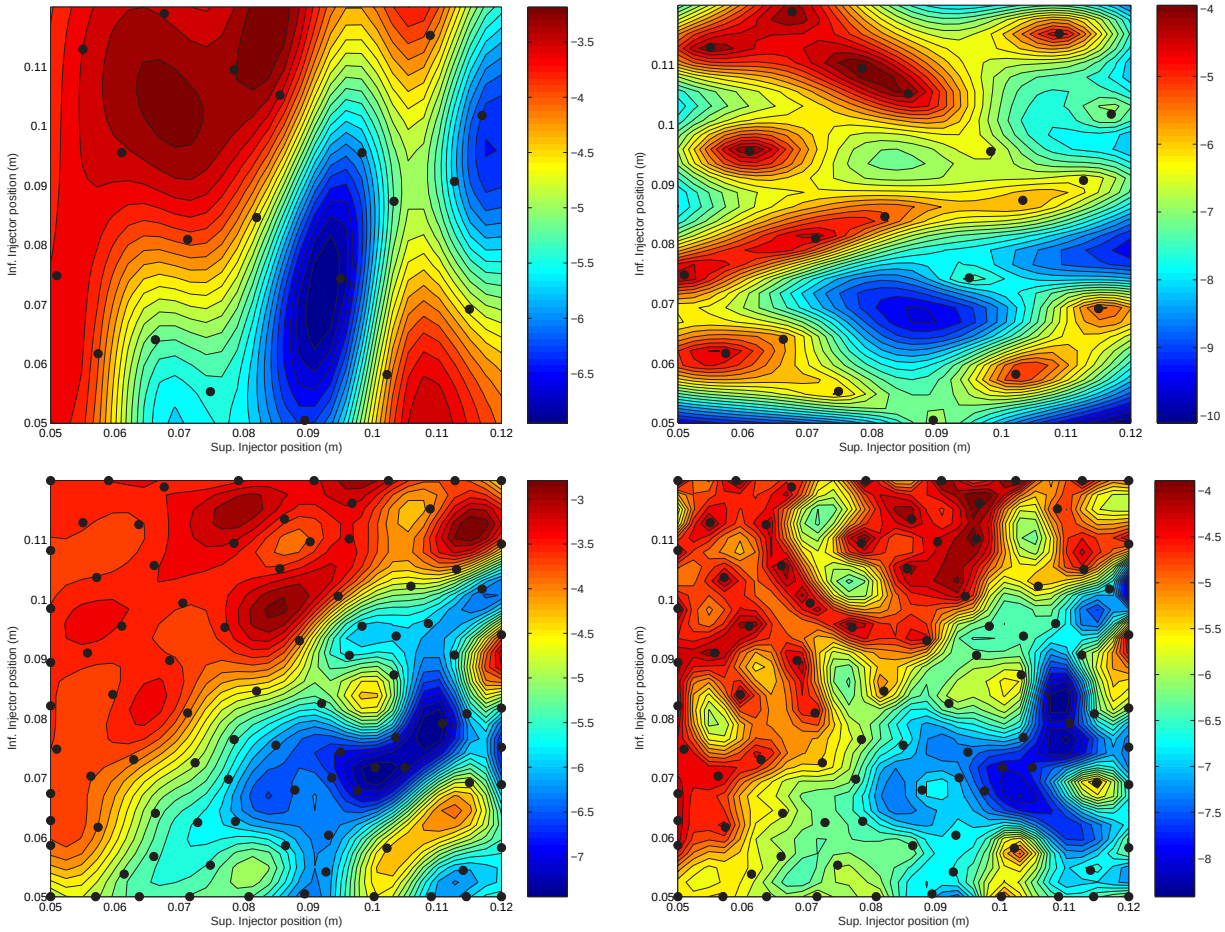

Fig. 4 Logarithm of $\hat{t}$ (left) and f_M (right) after initialization (top) and 6 iterations (bottom) - The black circles represent the sample points

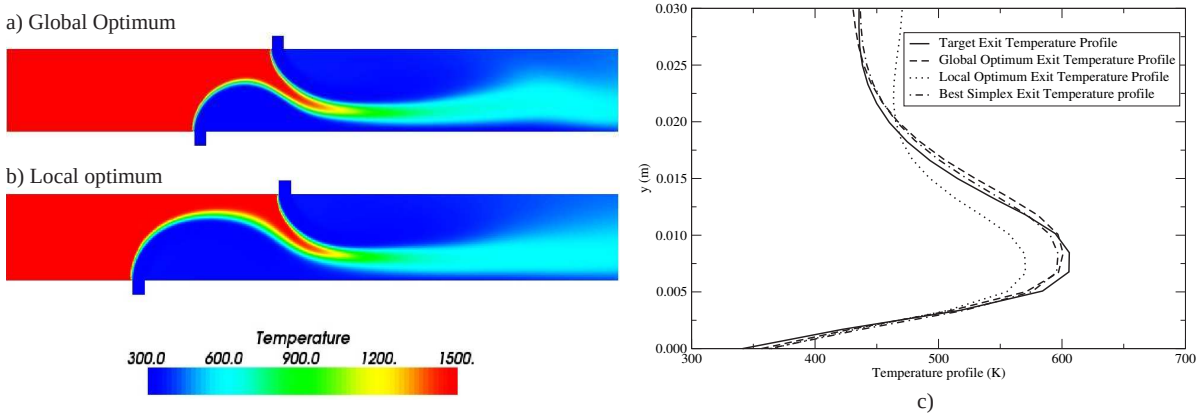

Fig. 5 Temperature fields and exit temperature profiles for the global and a local optimum

the total CPU time of an optimization process by using an adaptive approximation of the real objective function issued from a Gaussian process. Comparison between this method and the very popular Simplex algorithm leads to encouraging results and validates the tool prior to further de-

velopments and uses in an industrial context.

Actual developments concern the adaptation of MIPTO to undertake complex 3*D* reactive flows. The main difficulties are the control of 3*D* shapes and the corresponding meshes either by using moving mesh techniques or re-meshing strategies. A key aspect when dealing with design optimization is the way to manage the shape of the geometry. CAD based approaches seem promising but quite difficult to implement in a fully automatic loop.

5 Acknowledgements

The authors express their gratitude to N. Savary, D. Bissières and C. Bérat (TURBOMECA, Bordes - France) for the support and the industrial context of the study, as well as the valued support from T. Morel and S. Buis (CERFACS, Toulouse - France) for their help concerning PALM. The contribution from the INCKA team is acknowledged when dealing with the N3S-Natur grid management tools. This work is sponsored by the E.C. under grant No. FP6-502961 INTELLECT D.M.

References

[1] D.J.C. MacKay. Gaussian processes - a replacement for supervised neural networks? *Lecture notes for a tutorial at NIPS*, 1997.

[2] S. Buis, A. Piacentini, and D. Déclat. Palm: A computational framework for assembling high performance computing applications. *CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE*, 2005.

[3] J.A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308–313, 1965.

[4] A. Jameson, Sriram, L. Matinelli, and B. Haimes. Aerodynamic shape optimization of complete aircraft configurations using unstructured grids. In *AIAA Paper 2004-0533*, Reno, Nevada, 5-8 January, 2004. 42nd AIAA Aerospace Science Meeting and Exhibit.

[5] A. Nelson, M. Nemec, M.J. Aftosmis, and T.H. Pulliam. Aerodynamic optimization of rocket control surfaces using cartesian methods and cad geometry. In *AIAA Paper 2005-4836*, Toronto, Ontario, 6-9 Jun, 2005. 23rd AIAA Applied Aerodynamics Conference.

[6] A.L. Marsden, M. Wang, Jr. Dennis, and P. Moin. Optimal aeroacoustic shape design using the surrogate management framework. *Optimization and Engineering*, 5(2), 235-262 2004.

[7] D. Büche, N.N. Schraudolph, and P. Koumoutsakos. Accelerating evolutionary algorithms with gaussian process fitness function models. *IEEE Transactions on Systems, Man and Cybernetics*, 35:183–194, 2005.

[8] J.D. Müller. *On Triangles and Flow*. PhD thesis, The University of Michigan, 1996.

[9] M. McKay, R. Beckman, and W. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[10] L Kocis and W Whiten. Computational investigations of low-discrepancy sequences. *ACM Transactions on Mathematical Software*, 23(2):266–294, 1997.

[11] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Scientific Computing*, 5(16):1190–1208, 1995.

[12] D.E. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Professional, 1989.

[13] T.W. Simpson, T.M. Mauery, J.J. Korte, and F. Misree. Comparison of response surface and kriging models for multidisciplinary optimization. *AIAA*, 98-4755, 1998.

[14] Y. Jin, M. Olhofer, and B. Sendhoff. On evolutionary optimisation with approximate fitness functions. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO*, pages 786–793, July 2000.

[15] Y.S. Ong, P.B. Nair, and A.J. Keane. Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA*, 41(4):687–696, 2003.

[16] Z. Michalewicz. *Genetic Algorithms + Data Structure = Evolution Programs*. Springer, 1996.