

# Constrained Aerodynamic Optimization of Three-Dimensional Wings Driven by Navier–Stokes Computations

Boris Epstein\*

*Academic College of Tel-Aviv Yaffo, 64044 Tel-Aviv, Israel*

and

Sergey Peigin†

*Israel Aircraft Industries, 70100 Ben-Gurion Airport, Israel*

A robust and efficient approach to the multiobjective constrained design, previously developed by the authors, is extended to optimization of three-dimensional aerodynamic wings. The objective is to minimize the total drag at fixed lift subject to various geometrical and aerodynamical constraints. The approach employs genetic algorithms (GAs) as an optimization tool in combination with a reduced-order-models (ROM) method, based on linked local databases obtained by full Navier–Stokes computations. The work focuses on the following issues: geometrical representation of three-dimensional shapes, handling of sensitive nonlinear constraints such as pitching moment, and the influence of flight conditions on the results of optimization. The method, implemented in the computer code OPTIMAS (Optimization of Aerodynamic Shapes), was applied to the problem of multipoint transonic three-dimensional wing optimization with nonlinear constraints. The results include a variety of optimization cases for two wings: a classical test case of ONERA M6 wing and a generic cranked transport-type wing. For the investigated class of problems, significant aerodynamic gains have been obtained.

## Nomenclature

$C_D$	=	total drag coefficient
$C_L$	=	total lift coefficient
$C_M$	=	total pitching-moment coefficient
$C_p$	=	pressure coefficient
$M$	=	freestream Mach number
$N_D$	=	dimension of the search space
$N_{ws}$	=	number of sectional airfoils
$Q$	=	objective function
$R/c$	=	relative radius of the airfoil leading edge
$Re$	=	freestream Reynolds number
$t/c$	=	relative thickness of airfoil
$\alpha$	=	angle of attack
$\theta$	=	trailing-edge angle of airfoil

## I. Introduction

THE pivotal role of advanced aerodynamic design in the process of reducing costs of aircraft manufacturing prompts a demand for efficient and robust aerodynamic optimization.

A traditional process of aerodynamic design has been carried out by trial and error, which rely on the intuition and experience of designers. It is not at all likely that such interactive analysis procedures will lead to a truly optimal design. To efficiently examine a large design space, the numerical simulations are to be combined with automatic search tools. This can lead to computer-aided design methods, which will exploit a remarkable gain in simulation capabilities achieved by computational fluid dynamics (CFD) and thus will fully realize the potential improvements in aerodynamic efficiency.<sup>1–3</sup>

In Refs. 4 and 5 an efficient and robust algorithm for optimization of two-dimensional aerodynamic shapes was proposed. The

important features of the method included a new strategy for efficient handling of nonlinear constraints in the framework of genetic algorithms (GAs), scanning of the optimization search space by a combination of full Navier–Stokes computations with the reduced-order-models (ROM) method and a multilevel parallelization of the whole computational framework, which efficiently makes use of computational power supplied by massively parallel processors.

The main objective of the present research is to extend the method to optimization of three-dimensional configurations and to create an efficient tool of industrial aerodynamic design, which will allow the reduction of overall cost of the aircraft design and analysis.

The work focuses on the following issues: geometrical representation of three-dimensional shapes, handling of sensitive nonlinear constraints such as pitching moment, and the influence of flight conditions on the results of optimization.

It must be underlined that compared to two-dimensional optimization the requirements for the three-dimensional computational tools involved in the optimization process are much higher.

First, the problem of global geometrical representation of three-dimensional aerodynamic shapes is much more tricky (even for three-dimensional wings) and, in general, still remains open. Second, the construction of a reliable three-dimensional CFD solver, suitable for optimization in terms of accuracy and robustness, is much more complicated. Third, the three-dimensional optimization necessitates high-dimensional search spaces, which make the optimal search essentially more difficult, especially in the presence of nonlinear constraints. Finally, in view of the huge overall computational volume needed for optimization, stringent requirements should be placed upon the computational efficiency of the whole method.

The developed three-dimensional tool [code OPTIMAS (Optimization of Aerodynamic Shapes)] incorporates the state-of-the-art CFD software and innovative optimization algorithms into the core of the aerodynamic design and can be used for practical design of aerodynamic shapes.

A CFD solver, which drives the optimization process, must possess high accuracy of the Navier–Stokes computations on relatively coarse grids, high robustness for a wide range of flows and geometrical configurations, and fast computational feedback.

The full Navier–Stokes code NES<sup>6,7</sup> satisfies the first two requirements. This was recently confirmed by computations performed in the framework of the 2nd Drag Prediction Workshop.<sup>8</sup>

Received 22 April 2004; revision received 17 March 2005; accepted for publication 4 April 2005. Copyright © 2005 by Boris Epstein and Sergey Peigin. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/05 \$10.00 in correspondence with the CCC.

\*Professor, Computer Science Department; epstein@mta.ac.il. Member AIAA.

†Professor, Engineering Division; speigin@iai.co.il. Member AIAA.

To satisfy the requirement of fast computational feedback, the optimization search space is scanned by using the ROM approach in the form of local approximation method (LAM), based on local databases obtained by full Navier–Stokes computations (which dramatically reduce the overall volume of computational work).

Optimization problems in aeronautics necessarily include constraints in their formulation. Unfortunately the presence of constraints significantly decreases the performance and the computational efficiency of classical optimization methods. The reason for this lies in the fact that the calculation of derivatives of the objective function in the vicinity of the constraints boundary is an ill-posed problem that cannot be resolved by conventional methods.

The situation is especially troublesome in the case of constraints imposed on aerodynamic characteristics (such as pitching moment). The point is that the feasibility of the current geometry (in the context of the above constraints) can be tested only a posteriori, that is, only through the full CFD run. This means that, in the case of a negative answer (infeasible geometry), the corresponding CFD run is wasted, and the overall computational efficiency of the optimization algorithm is essentially decreased.

To create a robust and computationally efficient method for solution of the considered optimization problem, genetic algorithms were employed. A specific feature of the new approach consists of the change in the conventional search strategy by employing search paths that pass via both feasible and infeasible points (contrary to the traditional approach, where only feasible points can be included in a path).

The problem of optimization of aerodynamic shapes is very time consuming as it requires a huge amount of computational work. Each optimization step requires a number of heavy CFD runs, and a large number of such steps is needed to reach the optimum. Thus the construction of a computationally efficient algorithm is vital for the success of the method in engineering environment.<sup>9</sup>

To achieve this goal, a multilevel parallelization strategy was used. It includes parallelization of the multiblock full Navier–Stokes solver, parallel evaluation of an objective function, and, finally, parallelization of the optimization framework.

The method was applied to the problem of multipoint transonic three-dimensional wing optimization with nonlinear constraints. The results include a variety of optimization cases for two wings: a classical test case of ONERA M6 wing and a cranked transport-type wing.

## II. Statement of the Problem

In this section the transonic flow multipoint drag minimization problem is considered. In the case of the single-point optimization problem, the objective is to minimize the cost function  $Q$  (total drag coefficient  $C_D$ ) of a three-dimensional wing subject to the following classes of constraints:

1) Aerodynamic constraints such as prescribed constant total lift coefficient  $C_L^*$  and maximum allowed pitching moment  $C_M^*$ :

$$C_L = C_L^*, \quad C_M \geq C_M^* \quad (1)$$

2) Geometrical constraints on the shape of the wing surface in terms of properties of sectional airfoils—relative thickness  $(t/c)_i$ , relative radius of leading edge  $(R/c)_i$ , trailing-edge angle  $\theta_i$ :

$$(t/c)_i \geq (t/c)_i^*, \quad (R/c)_i \geq (R/c)_i^*, \quad \theta_i \geq \theta_i^* \quad (2)$$

where  $i = 1, \dots, N_{ws}$  and values  $(t/c)_i^*$ ,  $\theta_i^*$ ,  $(R/c)_i^*$ ,  $C_L^*$ , and  $C_M^*$  are prescribed parameters of the problem.

The single-point design must be analyzed over a range of Mach numbers and lift coefficients. To ensure the adequacy of the off-design performance, the multipoint optimization is needed where the objective function is a weighted combination of single-point cost functions:

$$Q = \sum_{j=1}^{j=N_p} w_j C_D^j, \quad \sum_{j=1}^{j=N_p} w_j = 1 \quad (3)$$

where  $w_j$  are nonnegative weight coefficients and  $N_p$  is the number of design points.

The aerodynamic coefficients  $C_D$ ,  $C_L$ , and  $C_M$  are estimated through full Navier–Stokes solutions by means of a multiblock code NES.<sup>6,7</sup>

The code that employs structured point-to-point matched grids is based on the essentially nonoscillatory concept<sup>10</sup> with a flux interpolation technique.<sup>11</sup> Nonlinear stability is maintained via approximation of inviscid fluxes on a variable template according to local characteristics and smoothness of the fluxes; viscous fluxes are approximated in a straightforward way.

The optimization technique employed GAs in combination with a ROM method based on local databases obtained by full Navier–Stokes computations. The novel features of the present multipoint optimization method include a new strategy for efficient handling of nonlinear constraints in the framework of GAs, a method of scanning of the optimization search space by a combination of full Navier–Stokes computations with the ROM method and multilevel parallelization of the whole computational framework.

## III. Description of the Optimization Algorithm

### A. Genetic Algorithms for Constrained Optimization Problems

The GAs are semistochastic semideterministic optimization methods based on an analogy to the theory of evolution. The problem to be optimized is parameterized into a set of decision variables (or genes). Each set of variables (a point in the search space) that fully defines one design is called an individual. A set of individuals is called a population (or a generation). Each individual is evaluated using a fitness (objective) function that determines survivability of that individual. In aerospace applications the variables can be a series of geometrical parameters associated with an aerodynamic configuration.

During solution advance (or evolution of generations), each individual is ranked according to its fitness. The population is treated with genetic operators: selection, crossover, and mutation. All of these operations include randomness. The probability of survival of new individuals depends on their fitness: the best are kept with a high probability; the worst are rapidly discarded.

As a basic algorithm, a variant of the floating-point GA is used. Unfortunately, in their basic form, genetic algorithms are not capable of handling constraint functions limiting the set of feasible solutions. To resolve this, a new approach was proposed. For the sake of completeness, it is briefly outlined next. More details can be found in Ref. 4.

Basically the new approach can be outlined as follows:

1) Change the conventional search strategy by employing search paths that pass through both feasible and infeasible points.

2) To implement the new strategy, it is proposed to extend the search space. A needed extension of an objective function can be easily implemented in the framework of GAs because of their basic property: contrary to classical optimization methods, GAs are not confined to only smooth extensions.

3) The extension should comply with two basic requirements: a) each feasible point is certain to be better (in terms of fitness) than any infeasible point, and b) the objective function in infeasible regions should be defined in such a way that it keeps in the current population a sufficient number of infeasible individuals, located close to the constraints boundary. In such a case we can expect, with a rather high probability, that the crossover between feasible and infeasible individuals will produce high-fitness children.

### B. Global Geometrical Representation of Three-Dimensional Wing

An optimization process can be described as a path in the search space, the points of which represent different geometries. Thus the choice of an appropriate search space is of crucial importance.

The main difficulty of such a choice is that the requirements for the search space are completely contradictory.

On the one hand, the search space should include a wide spectrum of shapes in order to be sufficiently representative. This means that any geometrical shape feasible from the engineering viewpoint must be represented (with sufficient accuracy) by a point in the preceding search space.

On the other hand, the complexity of optimal search grows exponentially concurrently with the search space dimensions. Thus, to ensure a successful and efficient search, a total number of parameters should not be too high.

In engineering practice, an aerodynamic surface is described in a local way by a set of discrete points. A total amount of the points varies from tens of thousand (for a simple wing) to millions (for a wing-body-nacelle configuration). As just explained, the search space of such dimensions is totally unacceptable.

Hence, a global representation of aerodynamic surfaces (which is based on a limited number of parameters and ensures sufficient representativeness) is needed. Note that in the general case the preceding problem remains open.

Nevertheless, for a specific class of aerodynamic surfaces such as three-dimensional wings, the global geometrical representation of the wing shape from the root section up to the tip region, is available.

In this work the following is assumed:

1) The geometry is described by the absolute Cartesian coordinate system  $(x, y, z)$ , where the axes  $x$ ,  $y$ , and  $z$  are directed along the streamwise, normal to wing surface and span directions, respectively.

2) Wing planform is fixed.

3) Wing surface is generated by a linear interpolation (in the span direction) between sectional two-dimensional airfoils.

4) The number of sectional airfoils  $N_{ws}$  is fixed.

5) The shape of sectional airfoils is determined by Bezier splines.

In the absolute coordinate system, the location of the preceding profiles is defined by the corresponding span positions of the trailing edge on the wing planform, twist angles  $\{\alpha_i^{tw}\}$ , and dihedral values  $\{\gamma_i^{dh}\}$  (relatively to the root section).

The wing planform is defined by the following parameters: the chord length at the root section  $c_1$ , span location of the wing sections  $\{z_i\}$ , and the corresponding leading and trailing edge sweep angles  $\{\lambda_i^{le}\}$  and  $\{\lambda_i^{tr}\}$ .

For each wing section, the nondimensional shape of the airfoil (scaled by the corresponding chord) is defined in a local Cartesian coordinate system  $(\bar{x}, \bar{y})$  in the following way. The coordinates of the leading edge and trailing edge of the profile were respectively  $(0, 0)$  and  $(1, 0)$ . For approximation of the upper and lower airfoil surface, Bezier spline representation was used. A Bezier curve of order  $N$  is defined by the Bernstein polynomials  $B_{N,i}$  ( $C_N^i$ , binomial coefficients)

$$G^k(t) = \sum_{i=0}^N B_{N,i} P_i^k, \quad B_{N,i} = C_N^i t^i (1-t)^{N-i}$$

$$C_N^i = \frac{N!}{i!(N-i)!} \quad (4)$$

where  $t$  denotes the curve parameter taking values in  $[0, 1]$ ,  $P_i^k$  are the control points and superscript  $k = u$ , and  $l$  corresponds to upper and lower surfaces of profile. So, as it is seen from Eq. (4), the Bezier curve is completely determined by the Cartesian coordinates of the control points.

For the considered optimization problem the first  $P_0^k = (0, 0)$  and the last  $P_N^k = (1, 0)$  ( $k = u, l$ ) points are set just fixing the position of leading and trailing edges. We also fix all the abscisses  $x_i^k$  of the control points  $P_1^k, \dots, P_{N-1}^k$ . We set  $x_1^k = 0$  in order to ensure that the upper and lower surfaces of the profile are tangent to the  $y$  axes at the leading edge. Finally, assuming the continuity of the airfoil curvature at the leading edge we obtain the additional relation  $y_1^u = -y_1^l$ .

Thus, the shape of a sectional profile is completely determined by a total of  $2N - 5$  parameters ( $a_1, a_2, \dots, a_{N-1}, a_N, \dots, a_{2N-5}$ ):

$$a_j = y_j^u \quad 1 \leq j \leq (N-1)$$

$$a_j = y_{j-N+2}^l \quad N \leq j \leq (2N-5)$$

To fully specify the wing shape, it is necessary to set locations of the two-dimensional sectional airfoils, in addition to their shapes.

Assuming that the chord value and trailing-edge location are defined by the wing planform, the sectional locations are specified by means of two additional parameters per section: twist angle  $\{\alpha_i^{tw}\}$  and dihedral value  $\{\gamma_i^{dh}\}$ . Note that for the root section these values are set to zero.

Thus, the dimensions  $N_D$  of the search space are equal to

$$N_D = N_{ws} \cdot (2N - 5) + 2 \cdot (N_{ws} - 1)$$

and a search string  $S$  contains  $N_D$  floating point variables  $a_j$  ( $j = 1, \dots, N_D$ ). The string components are varied within the search domain  $D$ . The domain  $D$  is determined by values  $Min_j$  and  $Max_j$ , which are the lower and upper bounds of the variable  $a_j$ .

Based on the just described approach to the constraints handling, the modified objective function  $Q$  for the solution of drag minimization problem was defined as follows:

$$Q = \begin{cases} 0.1 + [(t/c)_i^* - (t/c)_i] & \text{if } (t/c)_i < (t/c)_i^* \\ 0.15 + [C_M^* - C_M] & \text{if } C_M < C_M^* \\ 0.2 + [R_i^* - R_i] & \text{if } R_i < R_i^* \\ 0.3 + [\theta_i^* - \theta_i] & \text{if } \theta_i < \theta_i^* \\ 0.5 & \text{if } y_i^u(t) < y_i^l(t) \\ C_D & \text{otherwise} \end{cases} \quad (5)$$

where each condition is tested independently for all sectional airfoils ( $i = 1, \dots, N_{ws}$ ). In the case of multipoint optimization, the value of  $C_D$  represents a weighted combination of total drag values at the flight points participating in optimization.

One of key difficulties in the implementation of optimization algorithms is because, roughly speaking, each CFD run requires a different geometry and, therefore, the construction of a new computational grid.<sup>12</sup> For novel complex geometries, meshes are generally constructed manually, which is very time consuming.

To overcome this obstacle and to maintain the continuity of optimization stream, we suggest making use of topological similarity of geometrical configurations (involved in the optimization process) and building the grids by means of a fast automatic transformation of the initial grid, which corresponds to the starting basic geometry.

The preceding fast transformation of grids was implemented in the following way. First of all, the wing surface was divided into two parts: the wing proper ( $z_{root} \leq z \leq z_{tip}$ ) and the tip region ( $z > z_{tip}$ ), where  $z_{root}$  and  $z_{tip}$  are the span locations of the root and tip wing sections, respectively.

The transformation of the wing surface includes three stages. Assume that the grid possesses  $i, j, k$  structure with the coordinate plane  $j = 0$  representing the grid points lying on the wing surface. Denote  $\Delta r_{i,0,k}$  the change in the geometry of the wing surface at a grid point with indices  $(i, 0, k)$ . For the inner part of the wing (the first part),  $\Delta r_{i,0,k}$  represent the differences between the current geometry and the basic grid. For the tip region (the second part),  $\Delta r_{i,0,k}$  are determined in a way that ensures a smooth conjugation of both parts.

At the first stage, the coordinates  $r_{i,j,k}^{new}$  of the new grid are obtained by propagation of the shift  $\Delta r_{i,0,k}$  along the grid line  $i = \text{const}, k = \text{const}$ :

$$r_{i,j,k}^{new} = r_{i,j,k}^{initial} + \Delta r_{i,0,k}$$

At the second stage, the twist transformation is performed. For each grid point, the  $z$  coordinate is fixed, whereas the  $x$  and  $y$  coordinates are modified in accordance with the value of twist angle obtained by the linear interpolation from the nearby wing sections.

Finally, the dihedral transformation is applied. Here  $x$  and  $z$  coordinates are fixed, while the  $y$  coordinate is shifted in accordance with the interpolated value of dihedral.

### C. Approximation of Objective Function by ROM-LAM Method

One of the main weaknesses of GAs lies in their poor computational efficiency. This prevents their practical use where the evaluation of the cost function is computationally expensive as is the case for the full Navier-Stokes model. To overcome this, we introduce

an intermediate “computational agent,” a computational tool, which, on the one hand, is based on a very limited number of exact evaluations of objective function and, on the other hand, provides a fast and reasonably accurate computational feedback in the framework of GAs’ search.

We construct the computational agent by means of a ROM approach in the form of LAM. With the ROM–LAM method, the solution functionals that determine a cost function and aerodynamic constraints (such as pitching moment, lift and drag coefficients) are approximated by a local database. The database is obtained by solving the full Navier–Stokes equations in a discrete neighborhood of a basic point positioned in the search space.

So on the one hand, the number of exact estimations of the objective function (full Navier–Stokes solutions) is proportional to the dimensions of the search space. On the other hand, the computational volume required to provide approximate estimates of the objective function in the framework of GAs optimum search is negligible.

Thus the requirements just mentioned to the computational agent, related to its computational efficiency, are fulfilled. However, because of the approximate nature of the approach an additional effort should be made in order to ensure the accuracy and robustness of the method.

To reach this goal a multidomain prediction-verification principle is employed. That is, on the prediction stage the genetic optimum search is concurrently performed on a number of search domains. Each domain produces an optimal point, and the whole set of these points is verified (through full Navier–Stokes computations) on the verification stage of the method, and thus the final optimal point is determined.

Besides, to ensure the global character of the search, it is necessary to overcome the local nature of the preceding approximation. For this purpose, iterations are performed in such a way that, in each iteration, the result of the optimization serves as an initial point for the next iteration step (further referred to as optimization step).

The specific algorithm is described here.

Denote  $x = (a_1^n, a_2^n, \dots, a_{N_D}^n, \alpha^n)$  a point in the search space, where  $a_j^n$  specify an initial wing shape at  $n$ th optimization step and  $\alpha^n$  is the angle of attack, corresponding to the prescribed  $C_L^*$ , respectively. Then each wing shape can be determined by deviations  $\delta_j^n$  from the coefficients of the initial wing. At fixed values of other flow parameters, the solution functionals depend on the values of  $\delta_j^n$  and  $\delta_\alpha^n$  (the deviation from the initial angle of attack). In the optimization process the following local approximation of a functional  $F^n$  is used (subscript  $n$  is omitted and  $F = C_L, C_D$ , and  $C_M$ ):

$$F(a_1 + \delta_1, \dots, a_{N_D} + \delta_{N_D}, \alpha + \delta_\alpha) = F^\circ + \sum_{j=1}^{N_D} \Delta F_j + \Delta F_\alpha \quad (6)$$

Here  $F^\circ$  is the functional value at the basic point, and the values  $\Delta F_j$  ( $j = 1, \dots, N_D$ ) and  $\Delta F_\alpha$  are determined by means of a mixed linear-quadratic approximation, which employs the local database. One dimensionally, we use either the one-sided linear approximation (in the case of monotonic behavior of the solution functionals) or the quadratic approximation (otherwise).

The needed local database values are obtained by solving the full Navier–Stokes equations at the corresponding neighboring points of the basic point in the search space. These neighboring points are determined by variations  $\{\Delta_j\}$  corresponding to the coefficients  $\{a_j\}$  and by the variation  $\Delta_\alpha$  of the angle of attack  $\alpha$ .

#### D. General Sketch of the Algorithm

Finally, the optimization algorithm can be presented by the following pseudocode:

```

opt_step = 0
  Determine_Initial_Basic_Point
while not converged do
  Calc_Local_Data_Base
  Search_Optim_Candidates
  Verification_Optim_Cand
  Choose_New_Basic_Point
opt_step := opt_step + 1
enddo

```

At the step Determine\_Initial\_Basic\_Point, geometrical parameters that specify an initial wing shape (the initial basic point in the search space) are determined.

At the step Calc\_Local\_Data\_Base, the CFD local database for  $C_L, C_M$ , and  $C_D$  is obtained by solving the full Navier–Stokes equations at the neighboring points of the basic point in the search space. The local CFD database is included in the global CFD database.

Genetic algorithm is applied to various search domains  $D_k$  (corresponding to different search scales) at the step Search\_Optim\_Candidates. Thus, the optimal points  $O_k$  for each domain are obtained ( $k = 1, \dots, N_S$ , where  $N_S$  is the number of the search domains).

The full Navier–Stokes solver is applied to each optimal point  $O_k$  at the step Verification\_Optim\_Cand, and the corresponding data are added to the global CFD database.

At the final step of the loop (Choose\_New\_Basic\_Point), a new basic point is determined as the best point in the global CFD database.

As it is seen from the preceding pseudocode, the optimization algorithm includes a number of markedly different subalgorithms. In particular, the subalgorithms dealing with CFD computations and with genetic optimization search can be mentioned. As it usually happens in practice, such subalgorithms are not created from scratch, but, instead, are based on already existing computational core software (which is much less expensive). Moreover, the basic core codes can be written in different programming languages. For example, in our case, the CFD subalgorithms (employing the core code NES<sup>7,13</sup>) were written in the C language, whereas the GAs subalgorithms employ FORTRAN-77.

To resolve the difficulties caused by this heterogeneity and to ensure the correct interaction between different parts of the pseudocode, we drive the overall optimization algorithm by means of a control code, which monitors the algorithmic flow stream. The control code was written in the C language, which facilitates the interconnection of computational and system software and thus increases the ability of managing different executable codes and system calls. Note, that the modular approach also ensures a flexible upgrade of the objects included into the algorithm.

An additional important issue is the fault tolerance of the whole computational procedure. It is mainly related to the fact that the optimization process necessitates massive CFD runs (numbered in hundreds), which statistically increases the probability that one of the CFD processes fails. The failure probability is additionally increased if the computations are performed on massively parallel processors (also numbered in hundreds).

To overcome this, the control code monitors all of the CFD runs and, in the case of a system fail, excludes the corresponding point from the global database and restores the continuity of the optimization stream. On the whole, because of the robustness of the algorithm, this happens extremely infrequently.

#### IV. Computational Efficiency of the Algorithm

Aerodynamic optimization of three-dimensional shapes is an example of a highly challenging integral problem. To solve it, we need to resolve a number of nontrivial partial problems: 1) to create a robust, accurate, and efficient full Navier–Stokes solver; 2) to find an appropriate global geometrical representation of the optimized shape; and 3) to develop an efficient optimal search able to handle various nonlinear constraints.

Nevertheless, even a successful solution of all three partial problems is not sufficient for the success of the method as a whole. The reason is that the overall computational time needed for optimization is prohibitively high because of a significant computational cost of full Navier–Stokes CFD runs and the huge number of the runs.

This means that to make the optimization practically feasible it was necessary to significantly improve the computational efficiency of the algorithm. In fact, this is vital for the success of the method in engineering environment.

This was partially done by decreasing the total number of heavy CFD runs in the framework of the ROM–LAM approach, which allowed the reduction of the computational volume by at least 1–2 orders of magnitude. However, the total number of CFD runs

remained high, which is hardly acceptable even at the research level.

Extensive parallelization is particularly advantageous for achieving a further decrease of the computational volume because a highly scalable parallel implementation allows the dramatic reduction of the overall computation time.

To reach this goal, it was proposed to employ an embedded multilevel parallelization strategy that includes the following: level 1, parallelization of full Navier–Stokes solver; level 2, parallel CFD scanning of the search space; level 3, parallelization of the GAs' optimization process; level 4, parallel optimal search on multiple search domains; and level 5, parallel grid generation.

The first parallelization level (for a detailed description, see Ref. 13) is based on the geometrical decomposition principle. All processors are divided into two groups: one master processor and  $N_s$  slave processors. A large body of computational data demonstrated that the preceding approach for parallel implementation of the multiblock full Navier–Stokes solver enables one to achieve a high level of parallel efficiency while retaining high accuracy of calculations and thus to significantly reduce the execution time for large-scale CFD computations.

The first level of parallelization is embedded with the second level, which performs parallel scanning of the search space and thus provides parallel CFD estimation of fitness function on multiple geometries. It is applied when executing steps Calc\_Local\_Data\_Base and Verification\_Optim\_Cand of the optimization pseudocode. At this level of parallelization, all of the processors are divided into three groups: one main processor,  $N_m$  master processors, and  $N_m \cdot N_s$  of slave processors (where  $N_m$  is equal to the number of geometries).

The third level parallelizes the GAs optimization work unit. At this level of parallelization, all of the processors are divided into one master processor and  $P_s$  slave processors. The goal of the master processor is to distribute the initial random populations among the slaves and to get back the results of optimal search ( $P_s$  is the number of initial random populations).

The third level of parallelization is embedded with the fourth level, which performs parallel optimal search on multiple search domains. It is applied when executing step Search\_Optim\_Candidates of the pseudocode. At this level of parallelization, all of the processors are divided into three groups: one main processor,  $P_m$  master processors, and  $P_m \cdot P_s$  of slave processors (where  $P_m$  is equal to the number of domains).

The fifth parallelization level handles the grid-generation process. At this level, one master processor and  $G_s$  slave processors are employed ( $G_s$  is the number of evaluated geometries). This is performed prior to steps Calc\_Local\_Data\_Base and Verification\_Optim\_Cand of the pseudocode.

Finally we can conclude that the five-level parallelization approach allowed us to sustain a high level of parallel efficiency on massively parallel machines and thus to dramatically improve the computational efficiency of the optimization algorithm.

## V. Analysis of Results

The method was applied to the problem of the multipoint transonic three-dimensional wing drag minimization with nonlinear constraints. The results include a variety of optimization cases for two wings: a classical test case of ONERA M6 wing and a generic cranked transport-type wing.

Multilevel parallelization strategy based on the PVM software package was implemented on a cluster of MIMD multiprocessors consisting of 108 (72 HP NetServer LP1000R and 36 IBM Blade Server) nodes. Each node has two processors, 2 GB RAM memory, 512 KB level 2 cache memory, and full duplex 100 Mbps ETHERNET interface. Totally this cluster contained 216 processors with 216 GB RAM and 54 MB level 2 cache memory. On this cluster a sample three-dimensional wing optimization takes about 9–15 h CPU time (depending on the number of wing sections subject to design).

### A. Verification Studies

The CFD solver NES (used as a driver of the optimization process) ensures high accuracy of the Navier–Stokes computations on

relatively coarse grids as well as high robustness, for a wide range of flows and geometrical configurations. High performance of NES was systematically demonstrated by testing it a wide range of aerodynamic configurations of different complexity: from one-element two-dimensional airfoils (such as NACA0012, RAE2822) through ONERA M6 wing, transport-type supercritical wings up to ARA M100 wing-body.<sup>6,7</sup>

This was also recently confirmed by the computations performed in the framework of the 2nd Drag Prediction Workshop.<sup>8</sup> In these computations two complex three-dimensional cases were tested: DLR F6 wing-body configuration and DLR F6 wing-body-nacelle-pylon configuration. The results by the code NES demonstrated a high accuracy of drag prediction (within four–five counts) in the whole range of flight conditions. Note that the prediction accuracy of component drag increments (with nacelle on and off) was even higher. This is indicative of the NES suitability as a CFD driver of optimization process.

For transonic three-dimensional wings, NES provides accurate asymptotically converged estimates of aerodynamic coefficients with grids containing about  $193 \times 33 \times 33$  computational points on the fine level. Unfortunately, such computations, though feasible for a single optimization, are too heavy to be used in the industrial framework.

To overcome this limitation, we used the invariance of the hierarchy of objective function values on the medium and fine grids.<sup>4</sup> It also appeared that the two times coarser in each direction ( $97 \times 17 \times 17$ ) grids satisfy the invariance conditions. This allowed us to use meshes with such a resolution for optimization purposes.

To further verify the optimization method, the following multipoint optimization of RAE2822 airfoil was performed. The main design point was  $M = 0.734$ ,  $C_L = 0.8$ ,  $Re = 6.5 \times 10^6$ , whereas the secondary design points were  $M = 0.754$ ,  $C_L = 0.74$ ,  $Re = 6.2 \times 10^6$ , and  $M = 0.680$ ,  $C_L = 0.56$ ,  $Re = 5.7 \times 10^6$ . The target was to minimize a weighted combination of total drag values at these points with the following weight coefficients:  $w_1 = 0.5$ ,  $w_2 = 0.25$ , and  $w_3 = 0.25$ . The constraints were imposed on airfoil thickness and leading-edge radius, which cannot decrease. The case served for verification purposes in a number of studies, most recently performed within the European AEROSHAPE project and presented in Ref. 14.

First, the applicability of the hierarchy principle was verified. With this end in view, an optimization based on three CFD drivers (labeled 1levdc, 2levdc, 3levdc) with different grid resolution (one, two, and three multigrid levels, respectively) was performed. The study was done for a one-point and three-point optimization problems. In the former case the design point was one of the secondary points of the AEROSHAPE test case ( $M = 0.754$ ,  $C_L = 0.74$ ,  $Re = 6.2 \times 10^6$ ), whereas the latter case was that of the full AEROSHAPE test case. The shapes as a result of the one-point optimization are given in Fig. 1. The optimal solutions are very close to one another in terms of both shape and aerodynamic performance of optimized airfoils.

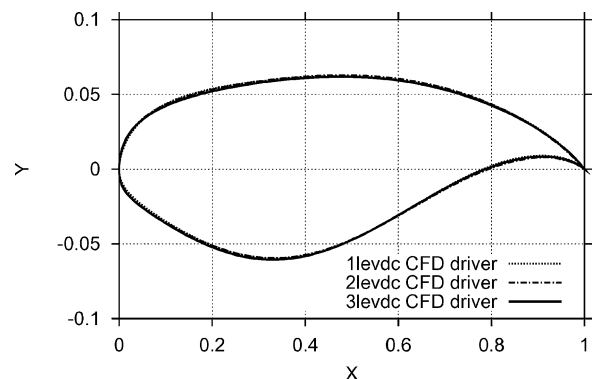


Fig. 1 Influence of the CFD grid resolution on the shape of optimized airfoils. One-point optimization.

Finally, the comparison of drag reduction achieved by the current optimization tool OPTIMAS with the corresponding AEROSHAPE results is summarized in Table 1.

It can be observed that OPTIMAS achieves an essentially higher drag reduction, especially at the high transonic flight conditions. A detailed analysis shows that this is attributed to a successful shock destruction, which allowed the elimination of most of the wave drag.

### B. Optimization of ONERA M6 Wing

In this section, we present the results of one- and multipoint drag minimization of ONERA M6 wing at  $Re = 11.72 \times 10^6$  and different values of design  $C_L$  and Mach numbers representing a wide range of flight conditions. A total of 10 test cases was studied. Design conditions and constraints are summarized in Table 2. The corresponding optimal wing shapes are designated by Case.1 to Case.10.

Geometrical constraints on relative thickness, relative leading-edge radius, and trailing-edge angle were kept on a constant level

**Table 1 Drag reduction (counts) for multipoint transonic test case [comparison between current optimization (OPTIMAS) and the results by Quagliarella<sup>14</sup>; one aerodynamic count = 0.0001]**

Design point	OPTIMAS	Quagliarella <sup>14</sup>
$M = 0.734, C_L = 0.80$	-59.0	-40.0
$M = 0.754, C_L = 0.74$	-103.0	-34.0
$M = 0.680, C_L = 0.56$	+2.0	+3.0

**Table 2 ONERA M6 wing (optimization conditions and constraints for different test cases)**

Case no.	$C_L^*$	Design $M$	Weight	$C_M^*$	$N_{ws}$
1	0.265	0.84	1.0	$-\infty$	2
2	0.500	0.84	1.0	$-\infty$	2
3	0.500	0.86	1.0	$-\infty$	2
4	0.500	0.87	1.0	$-\infty$	2
5	0.500	0.87	1.0	-0.100	2
6	0.500	0.87	1.0	-0.075	2
7	0.500	0.87	0.6	$-\infty$	2
	0.400	0.85	0.4	$-\infty$	2
8	0.500	0.87	1.0	$-\infty$	3
9	0.500	0.87	1.0	-0.100	3
10	0.500	0.87	1.0	-0.100	4

in all of the optimization cases:

$$(t/c)_i^* = 0.097, \quad (R/c)_i^* = 0.0029, \quad \theta_i^* = 3.6 \text{ deg}$$

$$i = 1, \dots, N_{ws}$$

Note that the value of the relative thickness was not allowed to be lower than that of the original ONERA M6 wing, whereas the value of the relative leading-edge radius was allowed to be lower than the original one.

The considered design points lie in the high transonic Mach range with lift coefficient values varying from moderate to high. At the preceding flight conditions, the flow over the original ONERA M6 wing develops a strong lambda shock with intensive shock-boundary-layer interaction.

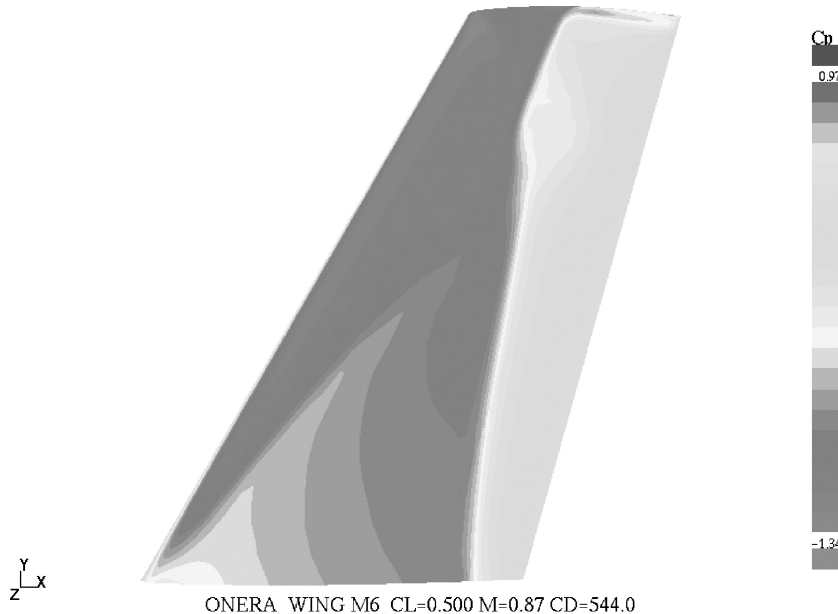
Now let us analyze the results of optimization at a demanding design point characterized by the combination of a high target lift coefficient and a high freestream Mach number ( $C_L = 0.5, M = 0.87$ , Case.4–Case.10).

At these flight conditions, the original ONERA M6 geometry generates a very strong shock, which results in a high total drag value ( $C_D = 544$  counts). The optimization allowed essentially the decrease of the total drag down to  $C_D = 300$  counts (Case.4). At this point, the theoretical induced drag for the ONERA M6 at  $C_L = 0.5$  is equal to 209 counts, whereas the minimum drag value is equal to about 87 counts for the original wing, which indicates a very low level of wave drag for the optimized wing.

The corresponding results are presented in Figs. 2–5, where the pressure distribution on the upper surface of the original wing and the chordwise pressure distributions at a midsection of the wing are compared with those of the optimized one.

The off-design behavior of the optimized wing (Case.4) is shown in Figs. 6 and 7. In Fig. 6 drag polars are presented at different freestream Mach numbers close to those of the design. Drag rise curves of the wings optimized at  $C_L = 0.5$  for different design freestream Mach numbers are compared to that of the ONERA M6 wing in Fig. 7. The optimization allowed the significant shifting of the drag-divergence point in the direction of higher Mach numbers and the radical extension of the low drag zone. The shift is greater for greater design Mach numbers with a small payoff for  $0.77 \leq M \leq 0.85$ .

As just mentioned, in the case of three-dimensional optimization there exists an additional class of constraints to be taken into account: the aerodynamic constraints such as the constraint on the pitching moment  $C_M$ . This class of constraints is difficult to handle. The point



**Fig. 2 Original ONERA M6 wing. Pressure distribution on the upper surface of the wing at  $M = 0.87, C_L = 0.5$ .**

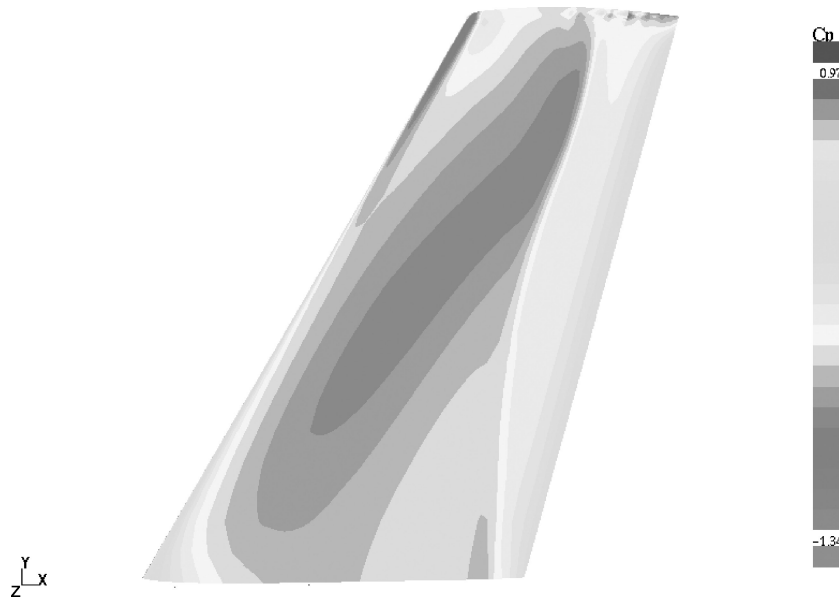


Fig. 3 One-point optimization: Case\_4. Pressure distribution on the upper surface of the wing at  $M = 0.87$ ,  $C_L = 0.5$ .

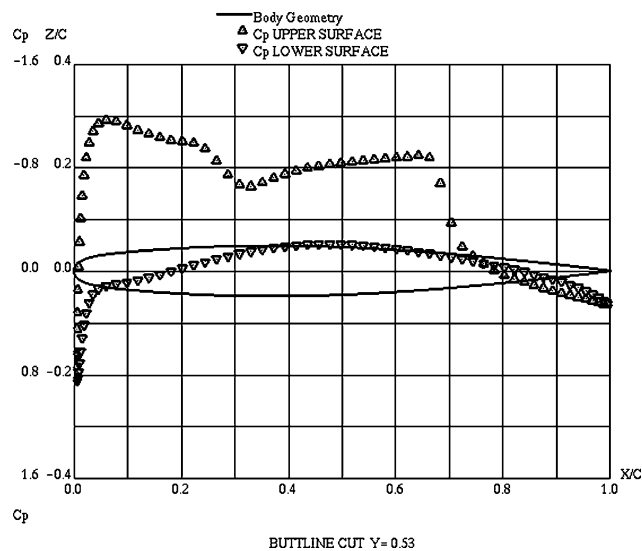


Fig. 4 Original ONERA M6 wing. Chordwise pressure distribution at the midsection of the wing at  $M = 0.87$ ,  $C_L = 0.5$ .

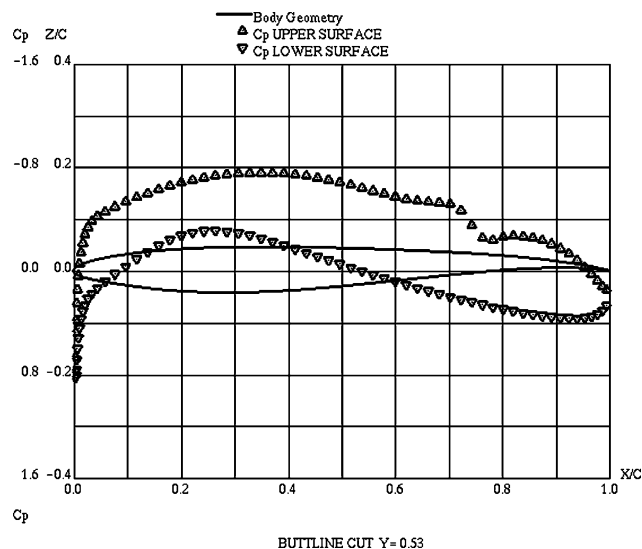


Fig. 5 One-point optimization: Case\_4. Chordwise pressure distribution at the midsection of the wing at  $M = 0.87$ ,  $C_L = 0.5$ .

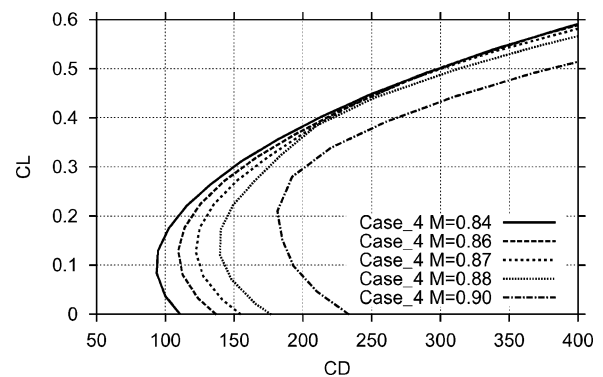


Fig. 6 Mach off-design behavior of the optimized wing (Case\_4). Drag polars at different Mach numbers:  $C_L$  vs  $C_D$ , target  $C_L = 0.500$   $M = 0.87$ .

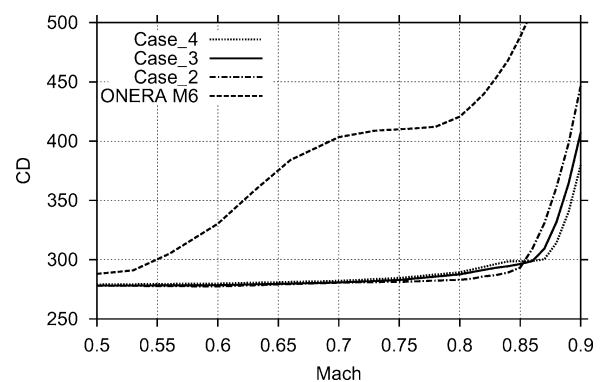


Fig. 7 Mach drag divergence of the optimized wings vs the original ONERA M6 wing. Case\_2, Case\_3, and Case\_4 correspond to design  $M = 0.84$ ,  $0.86$ , and  $0.87$ , respectively.

is that the position of testing point (testing aerodynamic shape) in the search space with respect to the constraints boundary is not known in advance (contrary to the geometrical constraints) and requires a computationally heavy CFD run.

The results of optimization indicate that the present approach is also able to efficiently handle this class of constraints. Several optimization cases, with different values of  $C_M^*$  (maximum allowed value of the pitching moment), were considered.

The unconstrained optimum wing (Case\_4) possesses  $C_M = -0.15$  and  $C_D = 300.0$  counts. A constrained optimization with

$C_M^* = -0.10$  (Case\_5) achieved a similar drag reduction at the design point  $M = 0.87$ ,  $C_L = 0.5$  ( $C_D = 300.5$  counts), whereas for  $C_M^* = -0.075$  (Case\_6) the optimized wing possesses a slightly higher  $C_D = 305.0$  counts at the same design point. It is important to underline that up to  $C_M^* = -0.10$  the total drag of optimized wings is weakly influenced by  $C_M^*$  not only at the design point but also in the off-design zone  $C_L > 0.3$ .

Thus the following two conclusions can be drawn. First, the performance of unconstrained pitching-moment optimization can be also achieved by a constrained optimization even with a rather significant increase in the maximum allowed value of the pitching moment. Second, the same optimal total drag value  $C_D$  can be obtained by markedly different aerodynamic shapes. In other words, the considered optimization problem is ill posed.

In this connection, it can be also assessed that the incorporation of constraints into the optimization problem is twofold. On the one hand, the presence of constraints (as it was explained above) makes the solution of the optimization problem much more complicated. But at the same time, the constrained problem is more well posed, which facilitates its solution.

Another important issue is the influence of the parameter  $N_{ws}$  (number of sectional airfoils) on the optimal solution. Increasing the value of this parameter leads to a more detailed geometrical representation of the wing surface and thus improves the design flexibility. However, this also increases the search space dimensions, which augments the complexity of the problem.

The comparison of the original ONERA M6 root and tip shapes with those of wings optimized for different  $N_{ws}$  is given in Figs. 8 and 9. The analysis of results shows that, in the middle part of the wing, the optimal shapes tend to possess low curvature immediately outside the leading-edge region. Note that to support this trend the optimal design with  $N_{ws} = 2$  produced the tip section with a concave lower surface in the preceding region.

For all of the considered cases, the chordwise pressure distributions are very close one to another exhibiting a virtually shockless behavior. At the same time the increase in the number of sectional airfoils  $N_{ws}$  leads to a more uniform chordwise wing loading.

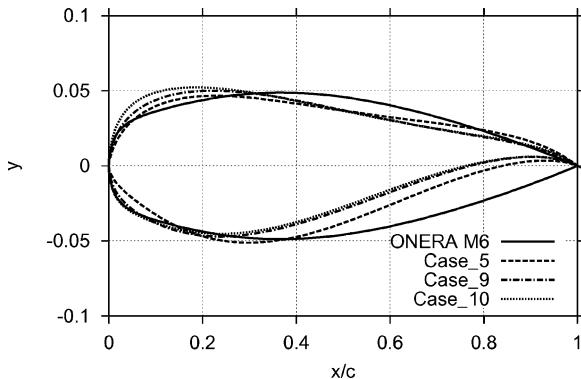


Fig. 8 Shape of optimized wings at root section for different  $N_{ws}$ .

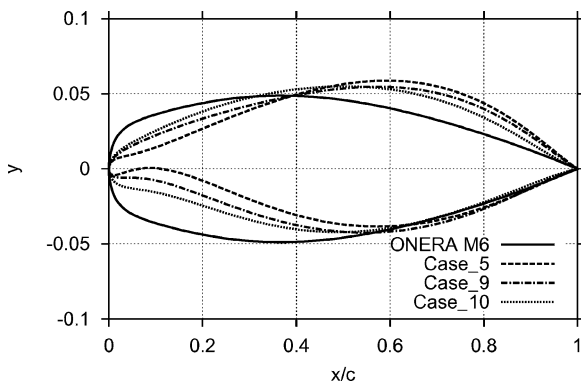


Fig. 9 Shape of optimized wings at tip section for different  $N_{ws}$ .

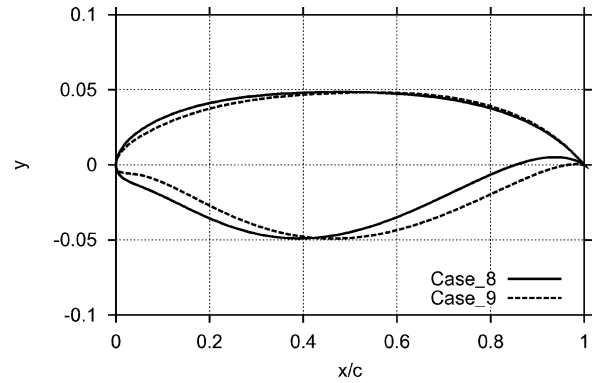


Fig. 10 Shape of the optimized wings at midsection for different values of constraint on pitch moment: Case\_8, no constraint on  $C_M$ ; Case\_9,  $C_M > -0.1$ .

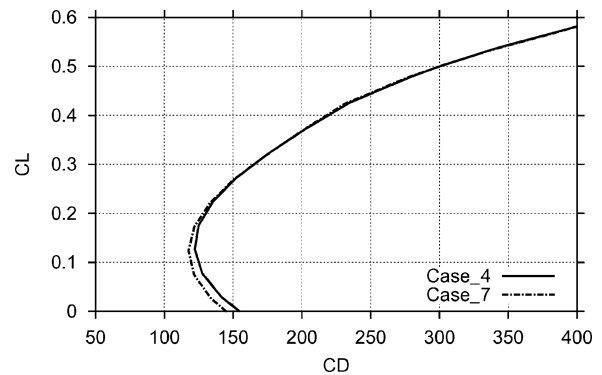


Fig. 11 Drag polars at  $M = 0.87$ . One-point optimization vs two-point optimization.

It is also aerodynamically interesting to analyze the influence of the pitching-moment constraint on the wing loading. Comparing the chordwise pressure distribution of the optimized wings, it can be concluded that a more severe constraint on the pitching-moment value leads to a visible shift of the loading to the leading-edge area.

To implement the preceding phenomenon of reloading, an essential change in the shape of optimized wings might be needed. This is illustrated in Fig. 10, where airfoil shapes at the midsection of the optimized wings for  $N_{ws} = 3$  and different values of constraint on the wing pitching moment are shown.

The proposed optimization algorithm was able to discover the basic optimization tools widely accepted in engineering practice: supercritical trailing edge and drooped leading edge. The unconstrained (with respect to  $C_M^*$ ) optimization makes the most of supercritical resources, which results in a highly rear-loaded airfoil shape with a strongly negative pitching moment. It is aerodynamically expectable that the imposition of constraint on  $C_M^*$  should lead to the pressure redistribution by increasing the loading in the leading-edge area. This means that a constrained optimal profile must be less supercritical than the unconstrained one, and this is to be compensated by a stronger leading-edge droop. This is exactly what can be assessed by observing the corresponding shapes depicted in Fig. 10.

Finally, the comparison of lift/drag polars corresponding to a one-point (Case\_4) and a two-point (Case\_7) optimization is presented in Fig. 11. The multipoint optimization allows the improvement of the wing performance at low  $C_L$  with no penalty at the design  $C_L$  value.

As it was underlined above the present optimization tool is driven by full Navier–Stokes computations. At the same time, many well-known applications are based on the use of simpler gasdynamic models, such as Euler equations. In this connection it was interesting to compare the results of optimization achieved by a Navier–Stokes driver vs an Euler one. Such a comparison was also

performed by OPTIMAS because its CFD driver also allows for Euler computations.

The results of comparison at the design point  $M = 0.87$ ,  $C_L = 0.5$  are given in Fig. 12, where drag polars at the design Mach value are shown. Note that the both polars are computed by means of the full Navier–Stokes computations. It is clearly seen that the Navier–Stokes optimization enabled the achievement of better optimization results. At the design point the optimization driven by Navier–Stokes computations yielded a total drag value of 300 counts compared to 312 counts in the case of the Euler CFD driver (about 4% more). This advantage is preserved in the whole range of  $C_L > 0.45$ , which indicates its nonpointwise nature.

Note that, in aircraft industry, an additional 4% drag reduction is highly significant because even a 1% increase in aircraft drag value can lead (at a fixed flight range) to the 7.6% reduction in the aircraft payload.<sup>15</sup>

### C. Optimization of a Transport-Type Wing

The goal of this study is to estimate the performance of the proposed method by applying it to optimization of a transport-type cranked wing with twisted and cusped profiles. At the transonic flight conditions (including those close to the cruise flight regime), the case is representative of a flow that is highly influenced by shock-boundary-layer interaction.

The geometry of the initial wing, which has a glove-like planform, is given in Ref. 16. In the following, we present the results of one-

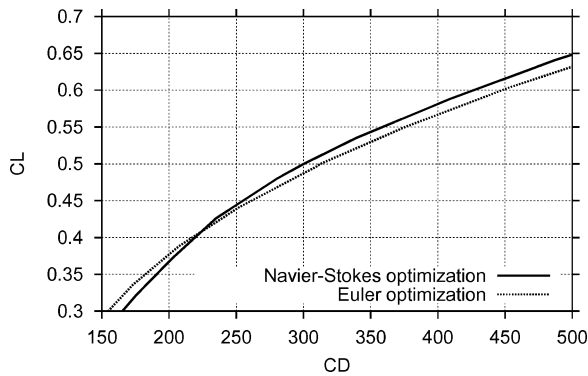


Fig. 12 Drag polars at  $M = 0.87$ . Navier–Stokes-driven optimization vs Euler-driven optimization.

point drag minimization of the wing at  $Re = 12.0 \times 10^6$  at a fixed design  $C_L = 0.4$  and different values of Mach numbers. A total of five test cases was studied. Design conditions and constraints are summarized in Table 3. The corresponding optimal wing shapes are designated by Case\_1.TW to Case\_5.TW. In all of the optimization cases, the geometrical constraints on the relative thickness were those of the initial wing, whereas the relative leading-edge radii and trailing-edge angles were allowed to be lower than the original ones.

At the preceding flight conditions, the flow over the original transport-type wing develops a strong shock characterized by intensive shock-boundary-layer interaction.

At  $M = 0.80$  the original wing yielded the total drag  $C_D = 135.4$  counts. Note that already the original wing possesses good aerodynamic characteristics at the considered flight conditions because its drag value is very close to the sum of  $C_{D_0}$  and the theoretical induced drag. Nevertheless, the optimization for Case\_1.TW resulted in a drag reduction of about 5% ( $C_D = 128.5$  counts).

At  $M = 0.83$  and  $0.85$  the total drag of the original wing was equal to  $C_D = 142.9$  and  $170.5$  counts, respectively. The total drag of the optimized wings (Case\_3.TW and Case\_4.TW) amounts to 133.4 and 143.3 counts (a reduction of 7 and 16%, respectively). Thus the increase in Mach number leads to a higher drag reduction caused by optimization, both in absolute and relative values.

The analysis demonstrated that a significant reduction in drag values was achieved by the destruction of a strong shock, present in the original pressure distribution. This is illustrated by Figs. 13–15, where the pressure distribution on the upper surface of the original wing at  $M = 0.85$ ,  $C_L = 0.40$  is compared with these of the optimized ones (Case\_4.TW and Case\_5.TW). The corresponding chordwise pressure distributions at three different wing sections are shown in Figs. 16–18.

The influence of the design Mach number on the shape of optimized wings can be assessed from Figs. 19 and 20, respectively, where the root and tip sectional airfoils are shown. For the

Table 3 Transport-type wing (optimization conditions and constraints for different test cases)

Case no.	$C_L^*$	Design $M$	$C_M^*$	$N_{ws}$
1.TW	0.40	0.80	$-\infty$	3
2.TW	0.40	0.80	$-0.08$	3
3.TW	0.40	0.83	$-0.08$	3
4.TW	0.40	0.85	$-\infty$	3
5.TW	0.40	0.85	$-0.08$	3

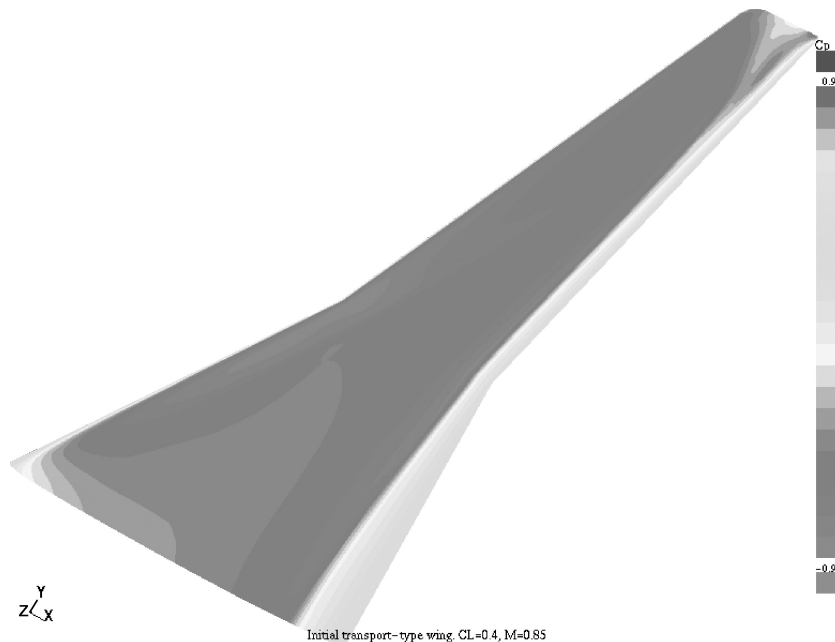


Fig. 13 Original transport-type wing. Pressure distribution on the upper surface of the wing at  $M = 0.85$ ,  $C_L = 0.40$ .

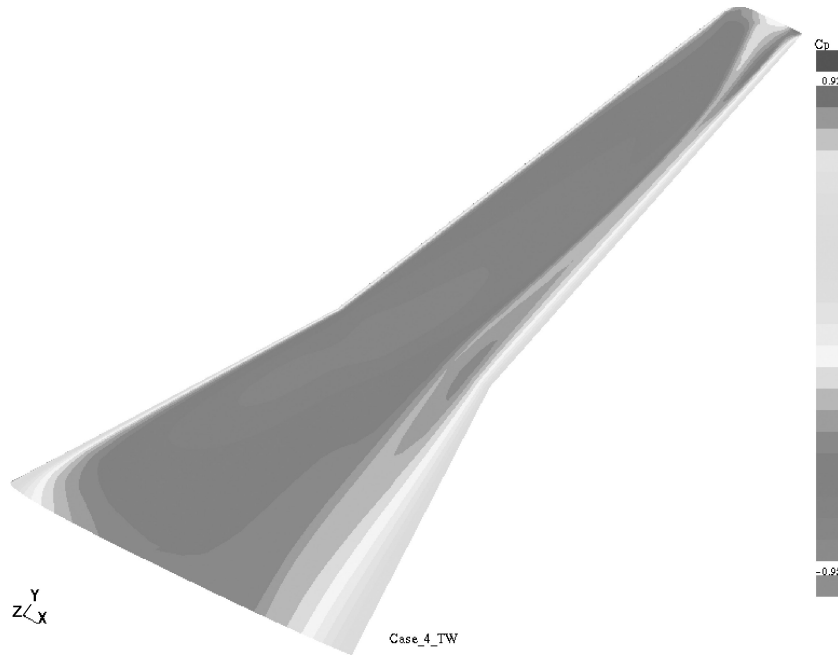


Fig. 14 Optimized transport-type wing: Case\_4\_TW. Pressure distribution on the upper surface of the wing at  $M = 0.85$ ,  $C_L = 0.40$ .

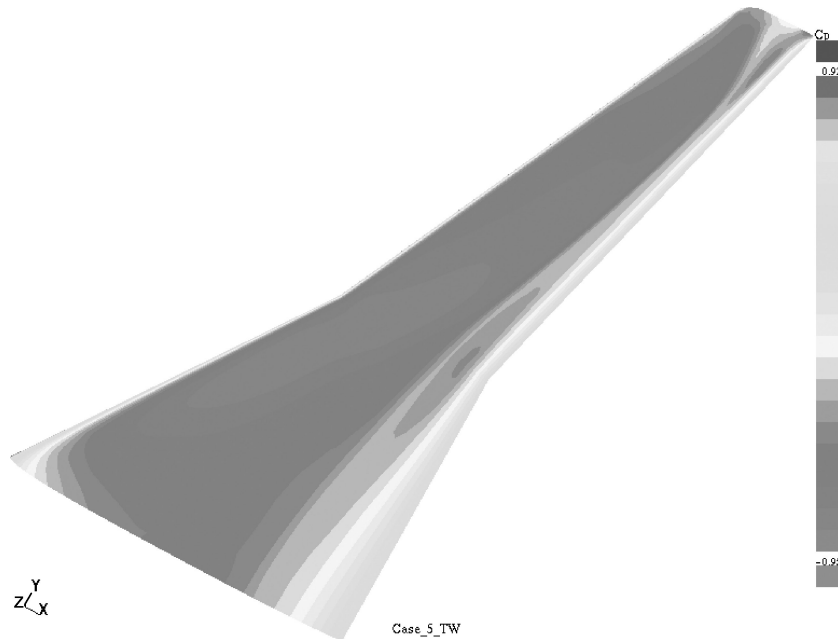


Fig. 15 Optimized transport-type wing: Case\_5\_TW. Pressure distribution on the upper surface of the wing at  $M = 0.85$ ,  $C_L = 0.40$ .

corresponding test cases (Case\_2\_TW, Case\_3\_TW, and Case\_5\_TW), the target lift coefficient was fixed at a level of  $C_L = 0.4$ , whereas the pitching moment was kept at a level very close to this of the original wing.

It is seen that the shock destruction (needed for drag minimization while maintaining constant values of  $C_L$  and  $C_M$ ) is achieved in different ways at different wing sections. At the root section, the increase in target Mach number leads to both the increase in the value of the leading-edge radius and the decrease in thickness of the rear part of the airfoil. By contrast, at the crank and tip sections both the leading-edge radius and the upper-surface curvature of airfoil are diminished.

For practical purposes, a solution of the drag minimization problem must possess a reasonable value of the pitching moment. Thus it is aerodynamically important to estimate the payoff as a result of the imposition of the constraint on pitching moment in terms of

total drag. The comparison between unconstrained and constrained optimizations at  $C_L = 0.4$ ,  $M = 0.80$ , and  $M = 0.85$  (Case\_1\_TW vs Case\_2\_TW and Case\_4\_TW vs Case\_5\_TW, respectively) shows that, in the both cases, the payoff for keeping the pitching-moment value at the original level does not exceed 1.0 count.

The shapes of crank and tip sections corresponding to these optimizations are, respectively, depicted in Figs. 21 and 22. In both cases the imposition of the constraint on the pitching moment resulted in significantly less cusped wings, which, in general, leads to the shift of the wing loading in the leading-edge direction. Note that though the values of total drag corresponding to the constrained and unconstrained optimizations are very close, the optimal shapes are markedly different.

The off-design behavior of the optimized wings is presented in Figs. 23–25 by means of lift-drag polars and Mach drag-divergence curves.

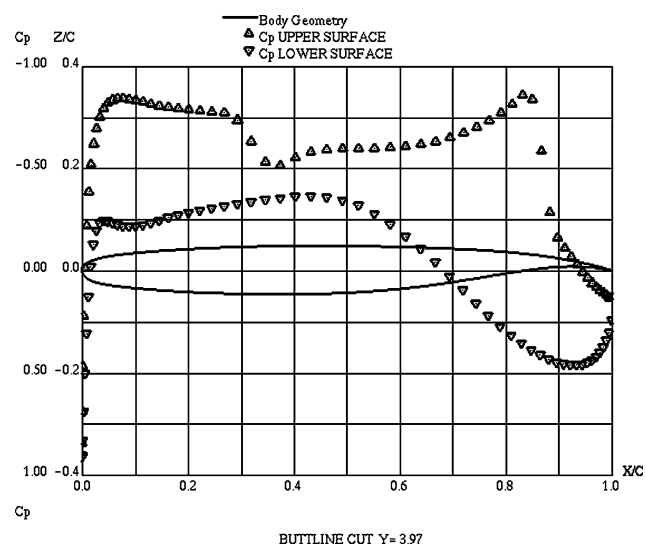


Fig. 16 Original transport-type wing at  $M=0.85$ ,  $C_L=0.40$ . Chordwise pressure distribution at  $2y/b=0.48$ .

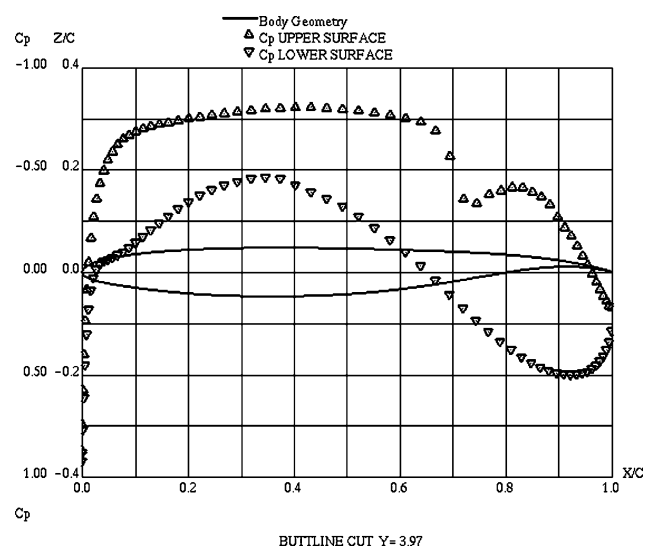


Fig. 17 Optimized transport-type wing (Case\_4\_TW) at  $M=0.85$ ,  $C_L=0.40$ . Chordwise pressure distribution at  $2y/b=0.48$ .

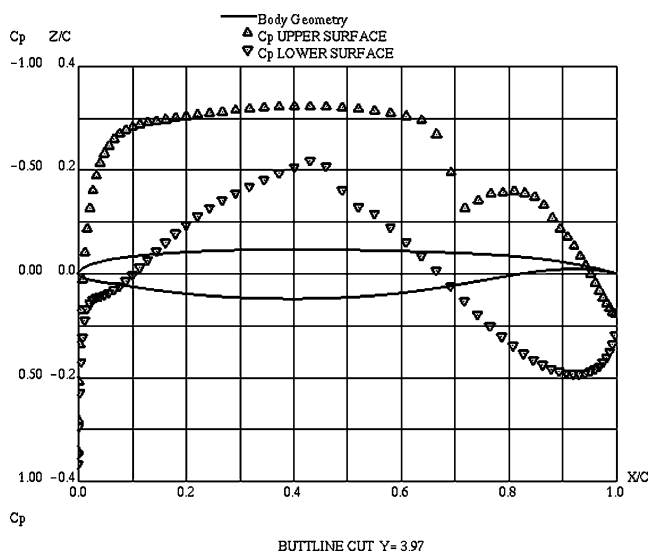


Fig. 18 Optimized transport-type wing (Case\_5\_TW) at  $M=0.85$ ,  $C_L=0.40$ . Chordwise pressure distribution at  $2y/b=0.48$ .

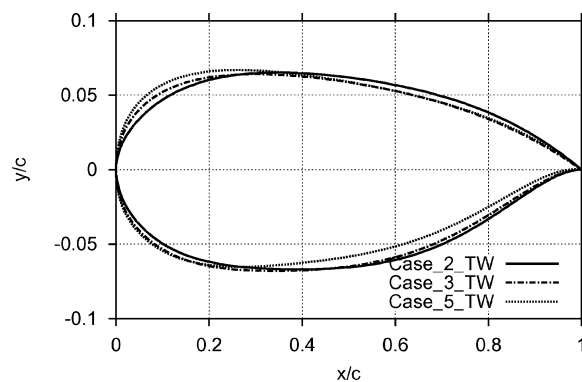


Fig. 19 Transport-type wing. Optimized shape of the root section for different target freestream Mach numbers.

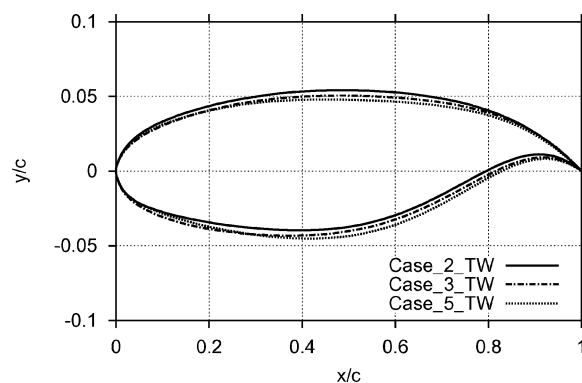


Fig. 20 Transport-type wing. Optimized shape of the tip section for different target freestream Mach numbers.

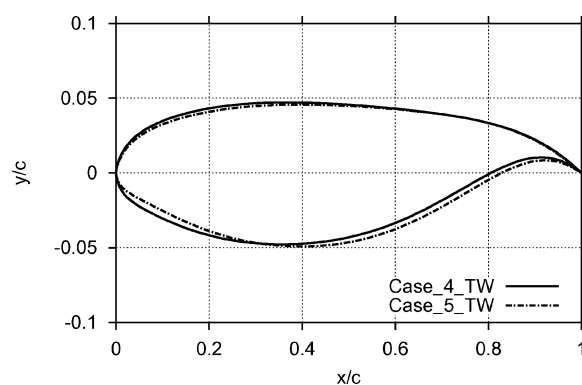


Fig. 21 Transport-type wing. Constrained vs unconstrained pitch optimization. Optimized shape of the crank section.

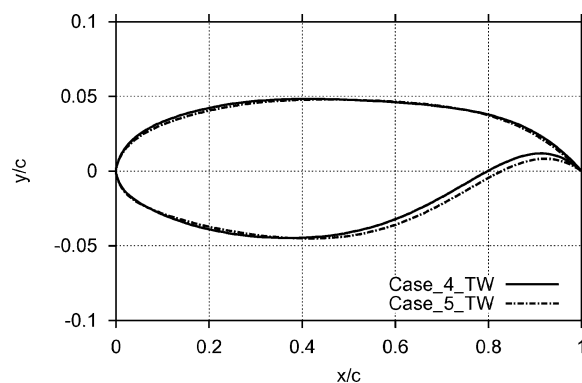


Fig. 22 Transport-type wing. Constrained vs unconstrained pitch optimization. Optimized shape of the tip section.

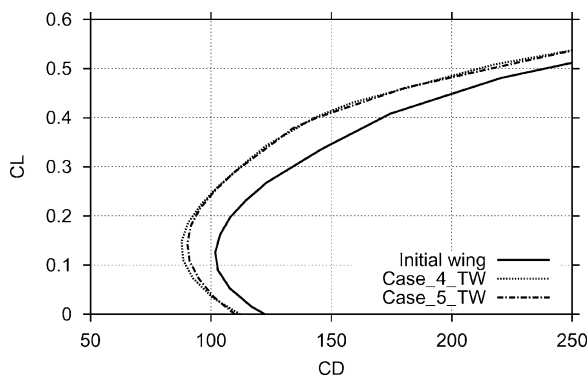


Fig. 23 Transport-type wing. Constrained vs unconstrained pitch optimization. Drag polar at  $M = 0.85$ .

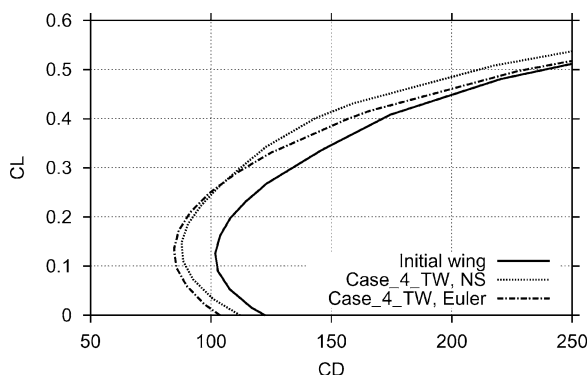


Fig. 24 Transport-type wing. Optimization driven by full Navier–Stokes CFD model vs optimization driven by Euler CFD model. Drag polar at  $M = 0.85$ .

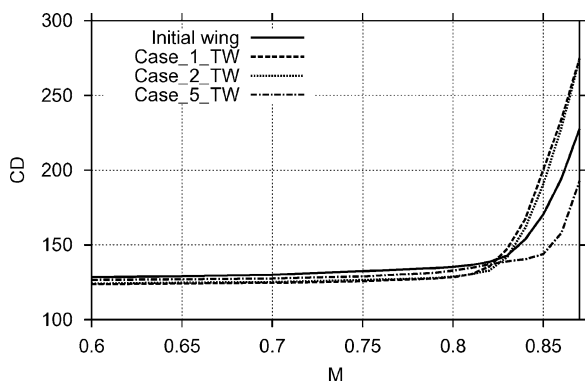


Fig. 25 Transport-type wing. Drag vs Mach at  $C_L = 0.4$ . Optimized shapes vs the original one.

Analyzing the lift-drag curves (Fig. 23), it can be concluded that the optimized wings possess better aerodynamic characteristics not only at the design point ( $C_L = 0.4$ ) but in the whole considered range of lift coefficients. It can be also observed that a payoff for keeping  $C_M$  at a level of the original wing is almost negligible.

Comparing the optimization driven by full Navier–Stokes computations with that driven by Euler computations (Fig. 24), we see that at a high target Mach number of 0.85 the Euler optimization yields the shape with inferior aerodynamic properties. At the design point the drag reduction achieved by the Navier–Stokes optimization was equal to 27.2 counts, whereas the Euler optimization reduced the drag by only 13.6 counts. The analysis of off-design behavior of the shapes shows that the preceding advantage of the Navier–Stokes optimization increases for higher lift coefficients.

Also, in terms of Mach, the optimality of solutions is not pointwise (Fig. 25). Nevertheless, contrary to the off-design behavior with respect to lift coefficient, the aerodynamic advantage of the

optimized wings in terms of Mach drag rise at fixed  $C_L$  can be confined to the neighborhood of a target Mach number.

At the design Mach  $M = 0.80$  (Case\_1\_TW and Case\_2\_TW), the zone of drag reduction extends from subsonic Mach numbers ( $M = 0.6$ ) up to  $M = 0.825$ . The optimization for a higher target Mach number (Case\_5\_TW,  $M = 0.85$ ) allowed the shift of the point of drag divergence from  $M = 0.825$  (the original wing) to  $M = 0.855$  (the optimized shape).

## VI. Conclusions

A robust approach to the multiobjective constrained optimization of three-dimensional aerodynamic wings is proposed. The method features an efficient treatment of nonlinear constraints in the framework of GAs' optimal search, a combination of full Navier–Stokes computations with the reduced-order-models method, and an efficient multilevel parallelization strategy. The results include a variety of optimization cases for two wings: a classical test case of ONERA M6 wing and a cranked transport-type wing. It was demonstrated that the accuracy and efficiency of the method allow for its application in engineering environment.

## References

- Jameson, A., "Optimum Aerodynamic Design Using Control Theory," *CFD Review*, Wiley, New York, 1995, pp. 495–528.
- Hajela, P., "Nongradient Methods in Multidisciplinary Design Optimization—Status and Potential," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 255–265.
- Guruswamy, G. P., and Obayashi, S., "Study on the Use of High-Fidelity Methods in Aeroelastic Optimization," *Journal of Aircraft*, Vol. 41, No. 3, 2004, pp. 616–619.
- Epstein, B., and Peigin, S., "A Robust Hybrid GA/ROM Approach to Multiobjective Constrained Optimization in Aerodynamics," *AIAA Journal*, Vol. 42, No. 8, 2004, pp. 1572–1581; also AIAA Paper 2003-4092, July 2003.
- Peigin, S., and Epstein, B., "Embedded Parallelization Approach for Optimization in Aerodynamic Design," *Journal of Supercomputing*, Vol. 29, No. 3, 2004, pp. 243–263.
- Epstein, B., Averbuch, A., and Yavneh, I., "An Accurate ENO Driven Multigrid Method Applied to 3D Turbulent Transonic Flows," *Journal of Computational Physics*, Vol. 168, No. 2, 2001, pp. 316–338.
- Epstein, B., Rubin, T., and Seror, S., "Accurate Multiblock Navier–Stokes Solver for Complex Aerodynamic Configurations," *AIAA Journal*, Vol. 41, No. 4, 2003, pp. 582–594.
- Seror, S., Rubin, T., Peigin, S., and Epstein, B., "Implementation and Validation of the Spalart–Allmaras Turbulence Model for a Parallel CFD Code," *Journal of Aircraft*, Vol. 42, No. 1, 2005, pp. 179–188.
- Byun, C., Farhangnia, M., and Guruswamy, G. P., "Aerodynamic Influence Coefficient Computations Using Euler/Navier–Stokes Equations on Parallel Computers," *AIAA Journal*, Vol. 37, No. 11, 1999, pp. 1393–1400.
- Harten, A., Engquist, B., Osher, S., and Chakravarthy, S., "Uniformly High Order Accurate Non-Oscillatory Schemes. 1," *Journal of Computational Physics*, Vol. 71, No. 2, 1987, pp. 231–303.
- Shu, C.-W., and Osher, S., "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes," *Journal of Computational Physics*, Vol. 83, No. 1, 1989, pp. 32–78.
- Potsdam, M. A., and Guruswamy, G. P., "A Parallel Multiblock Mesh Movement Scheme for Complex Aeroelastic Applications," AIAA Paper 2001-0716, Jan. 2001.
- Peigin, S., Epstein, B., Rubin, T., and Seror, S., "Parallel High Accuracy CFD Code for Complete Aircraft Viscous Flow Simulations," *Parallel Computational Fluid Dynamics. New Frontiers and Multi-Disciplinary Applications*, Elsevier, Amsterdam, 2003, pp. 507–514.
- Quagliarella, D., "Airfoil Design Using Navier–Stokes Equations and an Asymmetric Multi-Objective Genetic Algorithm," *Proceedings of 5th Evolutionary Methods for Design, Optimization and Control. Applications to Industrial and Social Problems Conference (EUROGN 2003)*, CIMNE, Barcelona, 2003, pp. 43–45.
- Jameson, A., Martinelli, L., and Vassberg, J., "Using Computational Fluid Dynamics for Aerodynamics—A Critical Assessment," *ICAS Proceedings*, Optimage, Ltd., Edinburgh, Paper ICAS 2002-1.10.1, 2002.
- Epstein, B., Jacobs, A., and Nachshon, A., "An ENO Navier–Stokes Method: Towards an Aerodynamically Accurate CFD Tool," *15th International Conference on Numerical Methods in Fluid Dynamics*, June 1996.