THREE-DIMENSIONAL MULTIOBECTIVE CONSTRAINED OPTIMIZATION OF AERODYNAMIC SHAPES

Sergey Peigin, Boris Epstein
Israel Aircraft Industries, The Academic College of Tel-Aviv Yaffo

Keywords: *Constarined optimization, Geneteic Algorithm, Navier-Stokes driver, parallel computations*

Abstract

In this work, a robust and efficient approach to the multiobjective constrained design, previously developed by the authors, is extended to optimization of 3D 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 data bases obtained by full Navier-Stokes computations. The work focuses on the following issues: geometrical representation of three-dimensional shapes, handling of sensitive non-linear 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. It was demonstrated that the method retains high robustness of conventional GAs while keeping CFD computational volume to an acceptable level, which allowed the algorithm to be used in a demanding engineering environment.

1 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 interective analysis procedures will lead to a truly optimal design. In order 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 CFD and thus will fully realize the potential improvements in aerodynamic efficiency.

In [1]-[2] an efficient and robust algorithm for optimization of 2D aerodynamic shapes was suggested. The important features of the method included a new strategy for efficient handling of nonlinear constraints in the framework of GAs, scanning of the optimization search space by a combination of full Navier-Stokes computations with the ROM method, and a multilevel parallelization of the whole computational framework which efficiently makes use of computational power supplied by massively parallel processors (MPP).

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 to reduce overall cost of the aircraft design and analysis.

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

It must be underlined that compared to 2D 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 3D aerodynamic shapes is much more tricky (even for 3D wings) and, in general, still remains open. Second, the construction of a reliable 3D 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 makes the optimal search essentially more difficult, especially in the presence of non-linear 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 3D tool (code OPTIMAS) 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 [3]-[4] satisfies the first two requirements. This was recently confirmed by computations performed in the framework of the 2nd Drag Prediction Workshop [5]. In these computations two complex 3D geometries were tested: DLR F6 wing-body configuration and DLR F6 wing-body-nacelle-pylon configuration. The results by the code NES demonstrated high accuracy of drag prediction (within 4-5 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 code suitability as a CFD driver of optimization process.

In order to satisfy the requirement of fast computational feedback, the optimization search space is scanned by using the Reduced-Order Models (ROM) approach in the form of Local Approximation Method (LAM), based on local data bases obtained by full Navier-Stokes computations (which dramatically reduces the overall volume of computational work).

Optimization problems in aeronautics necessaraly 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 which can not 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.

In order to create a robust and computationally efficient method for solution of the considered optimization problem, Genetic Algorithms (optimization methods based on coupling deterministic and probabilistic strategies in search of optimum) were employed. A specific feature of the new approach consists of the change in the conventional search strategy by employing search paths which pass via both feasible and infeasible points (contrary to the traditional approach, where only feasible points may 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.

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. For the investigated class of problems, significant aerodynamic gains have been obtained. It was demonstrated that the method retains high robustness of conventional GAs while keeping CFD computational volume to an acceptable level. A significant computational time-saving (in comparison with optimization tools fully based on Navier-Stokes computations) allowed the algorithm to be used in a demanding engineering environment.

The paper has the following structure. The problem statement is given in Section 2. In Section 3 a general description of the optmization algorithm is outlined. A method of improving the overall computational efficiency of the optimization algorithm through the multilevel parallelization is presented in Section 4. Results of optimization are given and analyzed in Section 5.

2 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 3D wing subject to the following classes of constraints:

1) Aerodynamic constraints such as prescribed constant total lift coefficient C_L^* and maximum allowed pithching moment C_M^*.

$$C_L = C_L^*, \quad C_M \geq C_M^* \tag{1}$$

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 θ_i:

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

where $i = 1,...,N_{ws}$ (N_{ws} - the number of sectional airfolis) and values $(t/c)_i^*$, θ_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. In order 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:

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

where w_j are non-negative weight coefficients, 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 [3]-[4].

The code which employs structured point-to-point matched grids, is based on the Essentially Non-Oscillatory (ENO) concept [6] with a flux interpolation technique [7]. The algorithm is implemented in the physical space by the finite volume method on grids which are defined as a set of vertices, and it is applicable to reasonably smooth computational meshes which are not necessarily defined by mapping functions.

Non-linear 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. An ENO interpolation template (typically consisting of 3 points on the

finest multigrid level) is determined separately for each characteristic field, primarily according to the sign of the corresponding eigenvalue, and then according to the smoothness of fluxes.

The code allows to attain high accuracy Navier-Stokes solutions on relatively coarse grids, and it is highly robust for a wide range of flows and geometrical configurations. The high performance of NES was systematically demonstrated by testing it in a wide range of aerodynamic configurations of different complexity: from one-element 2D airfoils (such as NACA0012, RAE2822) through ONERA M6 wing, transport-type supercritical wings, ARA M100 wing-body up to DLR F6 wing-body-nacelle-pylon configuration [3],[5].

The optimization technique employed Genetic Algorithms (GAs) in combination with a Reduced-Order Models (ROM) method based on local data bases 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.

3 Description of the Optimization Algorithm

3.1 Genetic Algorithms for Constrained Optimization Problems

The Genetic Algorithms (GAs) are semi-stochastic semi-deterministic 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 may 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 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 [1]. The optimization method resulted in the following pseudo-code:

$t = 0$
initpopulation $P(t)$
while not converged do
$P^*(t)$: = *selectparents* $P(t)$
recombine $P^*(t)$
mutate $P^*(t)$
evaluate $P^*(t) : P(t+1) := P^*(t) + best(P)$
$t := t+1$
enddo

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 in [1]. For the sake of completeness, it is briefly outlined below. More details may be found in [1].

Basically the new approach can be outlined as follows:

1. Change of the conventional search strategy by employing search paths which pass through both feasible and infeasible points (instead of the traditional approach where only feasible points may be included in a path).

2. To implement the new strategy, it is proposed to extend the search space. This requires the evaluation (in terms of fitness) of the points, which do not satisfy the constraints imposed by the optimization problem. A needed extension of an objective function may be easily implemented in the framework of GAs due to 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 in-

feasible 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.

3.2 Global Geometrical Representation of 3D 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 above search space.

On the other hand, the complexity of optimal search grows exponentially concurrently with the search space dimensions. Thus, in order 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 above problem remains open.

Nevertheless, for a specific class of aerodynamic surfaces such as 3D wings, the global geometrical representation of the wing shape from the root section up to the tip region, is available.

In this work it is assumed that:

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 2D airfoils.

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

5) Shape of sectional airfoils is determined by Bezier Splines. In the absolute coordinate system, the location of the above 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 non-dimensional 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)

$$\vec{G}^k(t)=\sum_{i=0}^{N} B_{N,i}\vec{P}_i^k, \quad B_{N,i}=C_N^i t^i (1-t)^{N-i}, \qquad C_N^i=\frac{N!}{i!(N-i)!} \tag{4}$$

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

For the considered optimization problem the first $\vec{P}_0^k=(0,0)$ and the last $\vec{P}_N^k=(1,0)$ $(k=u,l)$

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 $\vec{P}_1^k$, ..., $\vec{P}_{N-1}^k$. We set $x_1^k = 0$ in order to ensure that the upper and lower surfaces of the profile to be 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 $(a_1, a_2, ..., a_{N-1}, a_N, ..., a_{2N-5})$:

$$\begin{array}{ll} a_j = y_j^u, & 1 \le j \le (N-1) \\ a_j = y_{j-N+2}^l, & N \le j \le (2N-5) \end{array}$$

In order to fully specify the wing shape it is necessary to set locations of the 2D 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 $(j = 1, ..., N_D)$. 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 above 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 $(i = 1, ..., N_{ws})$. 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 due to the fact that, roughly speaking, each CFD run requires a different geometry and, therefore, the construction of a new computational grid. For novel complex geometries, meshes are generally constructed manually which is very time-consuming.

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

The above fast transformation of grids was implemented in the following way. First of all, the wing surface was divided into two parts: the wing proper ($z_{root} \le z \le z_{tip}$) 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 $\Delta\mathbf{r}_{i,0,k}$ 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), $\Delta\mathbf{r}_{i,0,k}$ represent the differences between the current geometry and the basic grid. For the tip region (the second part), $\Delta\mathbf{r}_{i,0,k}$ are determined in a way which ensures a smooth conjugation of both parts.

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

$$\mathbf{r}_{i,j,k}^{new} = \mathbf{r}_{i,j,k}^{initial} + \Delta\mathbf{r}_{i,0,k}$$

At the second stage, the twist transformation is performed. For each grid point, z coordinate is fixed, while x and y coordinates are modified in accordance with the value of twist angle obtained

by the linear interpolation from the near-by wing sections.

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

3.3 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 in the case where the evaluation of the cost function is computationally expensive as is the case for the full Navier-Stokes model.

For example, an algorithm with the population size **M**= 100 requires (for the case of 200 generations) at least 20000 evaluations of the cost function (CFD solutions). A fast full Navier-Stokes evaluation over a 3D wing takes at least a 10-15 minutes of CPU time. That means that one step of such an algorithm takes about 3500-4000 hours, which is practically unacceptable.

In order 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 Reduced-Order Models (ROM) approach. Among others, this approach includes the use of simpler gas-dynamic models [14], representation of the solution of gas-dynamic problem in terms of its eigenmodes [15] and aerodynamic analysis based on the Volterra theory of nonlinear systems [16].

In this work we use Reduced-Order Models approach in the form of Local Approximation Method (LAM). With the ROM-LAM method, the solution functionals which determine a cost function and aerodynamic constraints (such as pitching moment, lift and drag coefficients), are approximated by a local data base. The data base is obtained by solving the full Navier-Stokes equations in a discrete neighbourhood 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 above mentioned requirements to the computational agent, related to its computational efficiency, are fulfilled. However, due to 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, in order to ensure the global character of the search, it is necessary to overcome the local nature of the above 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 below.

Denote $x = (a_1^n, a_2^n, \ldots, a_{N_D}^n, \alpha^n)$ a point in the search space, where a_j^n specify an initial wing shape at n^{th} optimization step, and α^n is the angle of attack, corresponding to the prescribed C_L^*, respectively. Then each wing shape can be determined by deviations δ_j^n from the coefficients of the initial wing. At fixed values of other flow parameters, the solution functionals depend on the values of δ_j^n and δ_α^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):

$$F(a_1+\delta_1,\ldots,a_{N_D}+\delta_{N_D},\alpha+\delta_\alpha) =$$

$$= F^\circ + \sum_{j=1}^{N_D} \Delta F_j + \Delta F_\alpha \qquad (6)$$

Here F° is the functional value at the basic point, and the values ΔF_j ($j = 1,\ldots,N_D$) and ΔF_α are determined by means of a mixed linear-quadratic approximation which employs the local data-base. One-dimensionally, we use either the one-sided linear approximation (in the case of monotonic behaviour of the solution functionals) or the quadratic approximation (otherwise).

The needed local data base values are obtained by solving the full Navier-Stokes equations at the corresponding neighbouring points of the basic point in the search space. These neighbouring points are determined by variations $\{\Delta_j\}$ corresponding to the coefficients $\{a_j\}$ and by the variation Δ_α of the angle of attack α.

3.4 General Sketch of the Algorithm

Finally, the optimization algorithm can be presented by the following pseudo-code:

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 which 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 data base for C_L, C_M and C_D is obtained by solving the full Navier-Stokes equations at the neighbouring points of the basic point in the search space. The local CFD data base is included in the global CFD data base.

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 ($k = 1,\ldots,N_S$, 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 data base.

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 data base.

As it is seen from the above pseudo-code, the optimization algorithm includes a number of markedly different sub-algorithms. In particular, the sub-algorithms dealing with CFD computations and with genetic optimization search, may be mentioned. As it usually happens in practice, such sub-algorithms 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 may be written in different programming languages. For example, in our case, the CFD sub-algorithms (employing the core code NES [4],[8]) were written in the C language, while the GAs sub-algorithms employ FORTRAN-77.

In order to resolve the difficulties due to this heterogeneity and to ensure the correct interaction between different parts of the pseudo-code, 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 neccessitates massive CFD runs (numbered in hundreds) which statistically increases the probability that one of the CFD pro-

cesses 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 the CFD runs and, in the case of a system fail, excludes the corresponding point from the global data base and restores the continuity of the optimization stream. On the whole, due to the robustness of the algorithm, this happens extremely seldom.

4 Computational Efficiency of the Algorithm

Aerodynamic optimization of 3D shapes is an example of a highly challenging integral problem. To solve it we need to resolve a number of non-trivial 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 non-linear 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 due to a significant computational cost of full Navier-Stokes CFD runs and the huge number of the runs.

This means that in order 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 to reduce 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.

4.1 Multilevel Embedded Parallelization

Extensive parallelization is particularly advantageous for achieving a further decrease of the computational volume, since a highly scalable parallel implementation allows to dramatically reduce the overall computation time.

To reach this goal it was proposed to employ an embedded multilevel parallelization strategy which includes:

- 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
- Level 5 - Parallel grid generation

The first two levels are intended to improve the computational efficiency of the CFD part of the algorithm, while the next two levels handle the optimization part of the method.

The first parallelization level (for a detailed decription see [8]) 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 above approach for parallel implementation of the multiblock full Navier-Stokes solver, enables one to achieve 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 pseudo-code. It must be emphasized, that

the above two steps can not be executed concurrently, since the input data needed for step *Calc_Local_Data_Base* may be assessed only upon the full completion of the preceeding step *Verification_Optim_Cand*.

At this level of parallelization all 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). Since the volume of data transfer between the main-processor and master-processors is negligible, and the master-processors execute their own jobs independently, the parallel efficiency of the second level parallelization is very close to 100%.

The third level parallelizes the GAs optimization work unit. At this level of parallelization, all 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 pseudo-code. At this level of parallelization all 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). The goal of the master processor is to distribute geometries among the slave-processors, while each slave-processor creates a grid, corresponding to its own geometry. This is performed prior to steps *Calc_Local_Data_Base* and *Verification_Optim_Cand* of the pseudo-code. It is clear that the parallel efficiency at this level is also almost 100%.

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.

5 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 2 processors, 2GB RAM memory, 512KB Level 2 Cache memory and full duplex 100Mbps ETHERNET interface. Totally this cluster contained 216 processors with 216GB RAM and 54MB Level 2 Cache memory.

5.1 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 2D airfoils (such as NACA0012, RAE2822) through ONERA M6 wing, transport-type supercritical wings up to ARA M100 wing-body [3],[4].

This was also recently confirmed by the computations performed in the framework of the 2nd Drag Prediction Workshop [5]. In these computations two complex 3D 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 4-5 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.

Below are given the results of verification of the NES code for a popular ONERA-M6-Wing test case. The set of computational grids contained three multigrid levels. Each level included 8 blocks. The total number of points in the fine level was close to 200 000.

Fig. 1 presents computed surface pressure distribution compared with experiment [9] at high transonic Mach number $M = 0.84$, $\alpha = 3.06^\circ$, $Re = 11.72 \cdot 10^6$ at a midsection of the wing. The agreement is reasonably close which indicates that the computational grid is sufficiently resolved. It is important to emphasize that the NES computation not only favourably compares with experiment but also indicates a good grid convergence.

For transonic 3D 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 [1]. 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 $M = 0.734$, $C_L = 0.8$, $Re = 6.5 \cdot 10^6$ while the secondary design points were: $M = 0.754$, $C_L = 0.74$, $Re = 6.2 \cdot 10^6$ and $M = 0.680$, $C_L = 0.56$, $Re = 5.7 \cdot 10^6$. The target was to minimize a weighted combination of total drag values at these points with the following weight coefficients: $w_1 = 0.5$, $w_2 = 0.25$, $w_3 = 0.25$. The constraints were imposed on airfoil thickness and leading edge radius which can not decr ease. The

Design Point	OPTIMAS	Ref.[17]
$M = 0.734$, $C_L = 0.80$	-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 1 Drag reduction (counts) for multipoint transonic test case. Comparison between current optimization (OPTIMAS) and the results by Quagliarella [17]. 1 aerodynamic count = 0.0001.

case served for verification purposes in a number of studies, most recently performed within the European AEROSHAPE project and presented in [17].

First, the applicability of the hierarchy principle was verified. With this end in view, an optimization based on three CFD drivers 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 ($M = 0.754$, $C_L = 0.74$, $Re = 6.2 \cdot 10^6$), while the latter case was that of the full AEROSHAPE test-case. The shapes due to the one-point optimization are given in fig. 2. The optimal solutions are very close one to another both in terms of shape and aerodynamic performance of optimized airfoils.

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 to eliminate most of the wave drag.

5.2 Optimization of ONERA M6 Wing

In this section, we present the results of one- and multi-point drag minimization of ONERA M6 wing at $Re = 11.72 \cdot 10^6$ and different values of design C_L and Mach numbers representing a wide

Case No.	C_L^*	Design M	W_i	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

Table 2 ONERA M6 wing. Optimization conditions and constraints for different test cases.

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 in all the optimization cases:

$$(t/c)_i^* = 0.097, \quad (R/c)_i^* = 0.0029,$$

$$\theta_i^* = 3.6^\circ, \quad i = 1, \ldots, N_{ws}$$

Note, that the value of the relative thickness was not allowed to be lower than that of the original ONERA M6 wing, while 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 above flight conditions, the flow over the original ONERA M6 wing developes 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 free-stream 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 to essentially decrease 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, while 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 fig.3-6 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 behaviour of the optimized wing (*Case*_4) is shown in fig.7-8. In fig.7 drag polars are presented at different free-stream Mach numbers close to that of the design. Drag rise curves of the wings optimized at $C_L = 0.5$ for different design free-stream Mach numbers are compared to that of the ONERA M6 wing in fig.8. The optimization allowed to significantly shift the drag divergence point in the direction of higher Mach numbers and to radically extend the low drag zone. The shift is greater for greater design Mach numbers with a small pay-off for $0.77 \leq M \leq 0.85$.

As mentioned above, in the case of 3D 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 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) while 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 may 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 may be obtained by markedly different aerodynamic shapes. In other words, the considered optimization problem is ill-posed.

In this connection, it may 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 fig. 9-10. 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 in order to support this trend, the optimal design with $N_{ws} = 2$ produced the tip section with a concave lower surface in the above region.

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

It is also aerodynamically interesting to analyse 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.

In order to implement the above phenomenon of reloading, an essential change in the shape of optimized wings may be needed. This is illustrated in fig. 11 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.

It is important to note that 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 may be assessed by observing the corresponding shapes depicted in fig. 11.

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. 12. It is interesting to note that the multipoint optimization allows to improve 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, since 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. 13, 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 to achieve 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 non-pointwise nature.

Note, that, in aircraft industry, an additional 4% drag reduction is highly significant, since even a 1% increase in aircraft drag value may lead (at a fixed flight range) to the 7.6% reduction in the aircraft payload [18].

5.3 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 which is highly influenced by shock-boundary layer interaction.

The geometry of the initial wing which has a glove-like planform is given in [19], [13]. In the following, we present the results of one-point drag minimization of the wing at $Re = 12.0 \cdot 10^6$ at a fixed design $C_L = 0.4$ and different values of Mach numbers. A total of 5 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 the optimization cases the geometrical constraints on the relative thickness were those of the initial wing, while the relative leading edge radii and trailing edge angles were allowed to be lower than the original ones.

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

Table 3 Transport-type wing. Optimization conditions and constraints for different test cases.

At the above flight conditions, the flow over the original transport-type wing developes 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 since its drag value is very close to the sum of $C_{D_\circ}$ 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 $M = 0.85$ the total drag of the original wing was equal to $C_D = 142.9$ and $C_D = 170.5$ 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 due to 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 fig.14-16 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 fig.17-19.

The influence of the design Mach number on the shape of optimized wings can be assessed from fig. 20-22, where the root, crank and tip sectional airfoils are respectively shown. For the corresponding test cases (*Case*_2_*TW*, *Case*_3_*TW* and *Case*_5_*TW*), the target lift coefficient was fixed at a level of $C_L = 0.4$, while 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 pay-off due to 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 pay-off for keeping the pitching moment value at the original level does not exceed 1.0 count.

The shapes of root, crank and tip sections corresponding to these optimizations are respectively depicted in fig. 23-25. 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 behaviour of the optimized wings is presented in fig. 26-28 by means of lift-drag polars and Mach drag divergence curves.

Analyzing the lift-drag curves (fig. 26), 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 may be also observed that a pay-off 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. 27) 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 while the Euler optimization reduced the drag by only 13.6 counts. The analysis of off-design behaviour of the shapes shows that the above advantage of the Navier-Stokes optimization increases for higher lift coefficients.

Also, in terms of Mach, the optimality of solutions is not pointwise (see fig. 28). Nevertheless, contrary to the off-design behaviour with respect to lift coefficient, the aerodynamic advantage of the optimized wings in terms of Mach drag rise at fixed C_L may be confined to the neighbourhood 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 to shift the point of drag divergence from $M = 0.825$ (the original wing) to $M = 0.855$ (the optimized shape).

6 Conclusions

A robust approach to the multiobjective constrained optimization of 3D 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 Re-

duced Order Models method, and an efficient multilevel parallelization strategy.

The method (implemented in the computer code OPTIMAS) was applied to the solution of drag minimization problem at fixed lift subject to various geometrical and aerodynamical 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. Significant aerodynamic gains have been obtained and it was demonstrated that the accuracy and efficiency of the method allow for its application in engineering environment.

References

[1] Epstein, B. and Peigin, S., *A robust hybrid GA/ROM approach to multiobjective constrained optimization in eerodynamics*, AIAA Paper 2003-4092, also accepted for publication to AIAA Journal.

[2] Peigin, S. and Epstein, B., *Embedded parallelization approach for optimization in aerodynamic aesign*, The Journal of Supercomputing, Vol. 29, 2004.

[3] Epstein, B., Averbuch, A., and Yavneh, I., *An accurate ENO driven multigrid method applied to 3D turbulent transonic flows*, J. Comput. Phys., Vol. 168, 2001.

[4] Epstein, B., Rubin, T., and Seror, S., *Accurate multiblock Navier-Stokes solver for complex aerodynamic configurations*, AIAA Journal, Vol. 41, No. 4, 2003.

[5] Seror, S., Rubin, T., Peigin, S. and Epstein, B., *Implementation and validation of the Spalart-Allmaras turbulence model for apParallel CFD code*, accepted for publication to Journal of Aircraft.

[6] Harten, A., Engquist, B., Osher, S., and Chakravarthy, S., *Uniformly high order accurate non-oscillatory schemes. 1*, J. of Comput. Phys., Vol. 71, 1987.

[7] Shu, C.-W., and Osher, S., *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. of Comput. Phys., Vol. 83, 1989.

[8] Peigin, S., Epstein, B., Rubin, T., and Seror, S., *Parallel high accuracy CFD code for complete aircraft viscous flow simulations*, Parallel Computational Fluid Dynamics. Trends and Applications. Elsevier. 2002.

[9] Schmitt, V. and Charpin, F., *Pressure distributions on the ONERA-M6-wing at transonic mach numbers*, AGARD AR 138, May 1979.

[10] Jameson, A. *Aerodynamic design via control theory*, Journal of Scientific Computing, Vol. 3, No. 2, 1988.

[11] Jameson, A. *Optimum aerodynamic design using control theory*, CFD Review, Wiley, 1995, pp. 495–528.

[12] Hajela, P. *Nongradient methods in multidisciplinary design optimization - status and potential* Journal of Aircraft, Vol.36, No.1, 1999.

[13] Epstein, B., Jacobs, A., and Nachshon, A., *Aerodynamically accurate three-dimensional Navier-Stokes method*, AIAA Journal, Vol. 35, No. 6, 1997.

[14] Vicini, A., and Quagliarella, D., *Airfoil and wing design through hybrid optimization strategies*, AIAA Paper, 98-2729, 1998.

[15] Dowell, E.H., *Eigen-mode analysis in unsteady aerodynamics: reduced-order models*, Proc. of the 36th Structures, Structural Dynamics and Materials Conference, AIAA, Reston, VA, 1995.

[16] Raveh, D.E., *Reduced-order models for nonlinear unsteady aerodynamics*, AIAA Journal, Vol. 39, No. 8, 2001.

[17] Quagliarella, D., *Airfoil design using Navier-Stokes equations and an asymmetric multiobjective Genetic Algorithm*, EUROGEN 2003.

[18] Jameson, A., Martinelli, L. and Vassberg, J., *Using computational fluid dynamics for aerodynamics - a aritical assessment*, ICAS Proceed., Toronto, Canada, September 2002.

[19] Epstein, B., Jacobs, A., and Nachshon, A., *An ENO Navier-Stokes method: towards an aerodynamically accurate CFD tool*, presented at The 15th International Conference on Numerical Methods in Fluid Dynamics, Monterey, California, June 1996.

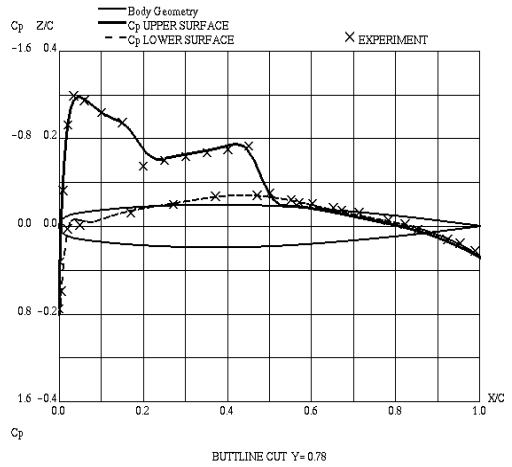

Fig. 1 ONERA-M6-Wing. Chordwise pressure distribution for wing span station $Y = 0.78$, $\alpha = 3.06°$, M=0.84, $Re = 11.72 \cdot 10^6$. Solid and dashed lines - present computation; crosses - experimental data [9].

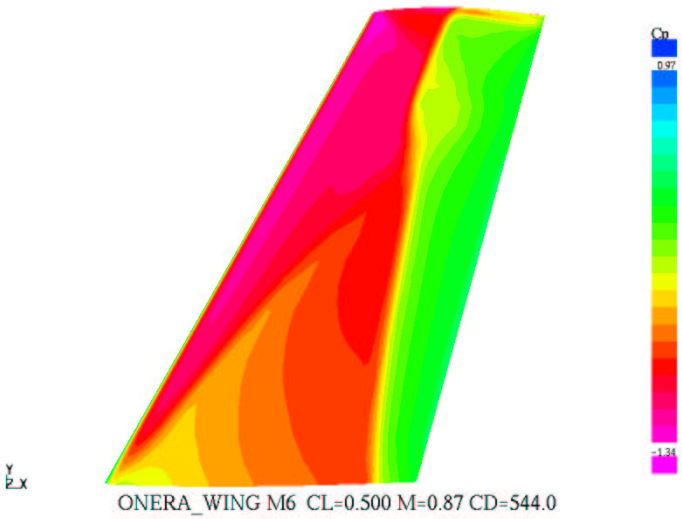

Fig. 3 Original ONERA-M6-Wing. Pressure distribution on the upper surface of the wing at M=0.87, $C_L = 0.5$.

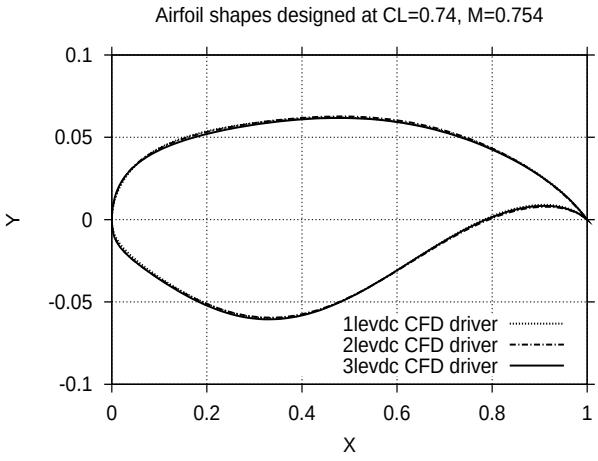

Fig. 2 Influence of the CFD grid resolution on the shape of optimized airfoils. One-point optimization.

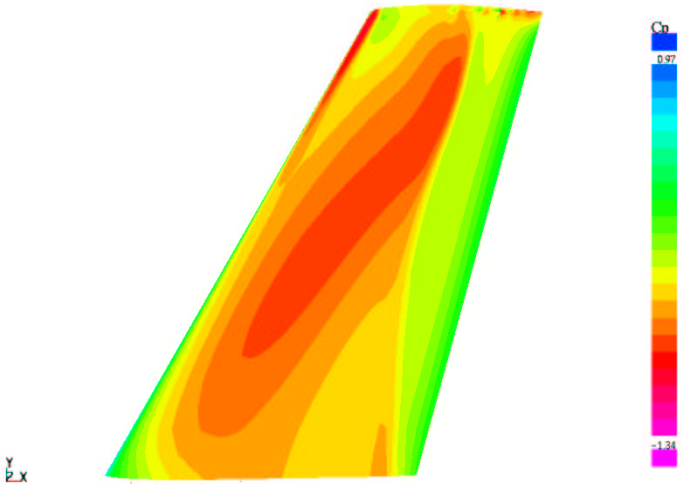

Fig. 4 One-point optimization - *Case*_4. Pressure distribution on the upper surface of the wing at M=0.87, $C_L = 0.5$.

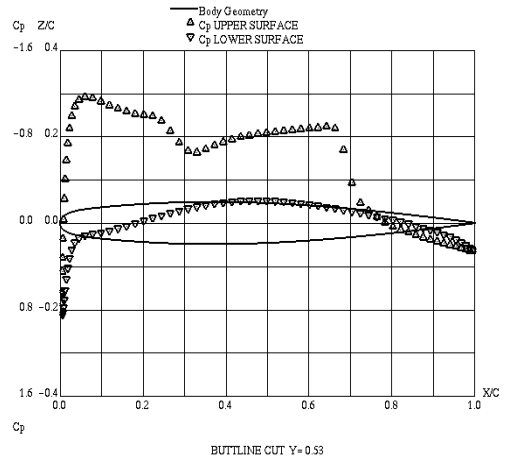

Fig. 5 Original ONERA-M6-Wing. Chordwise pressure distribution at the midsection of the wing at M=0.87, $C_L = 0.5$.

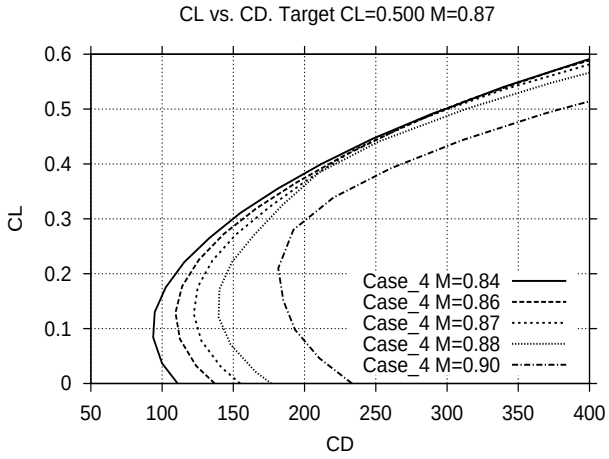

Fig. 7 Mach off-design behaviour of the optimized wing (*Case*_4). Drag polars at different Mach numbers.

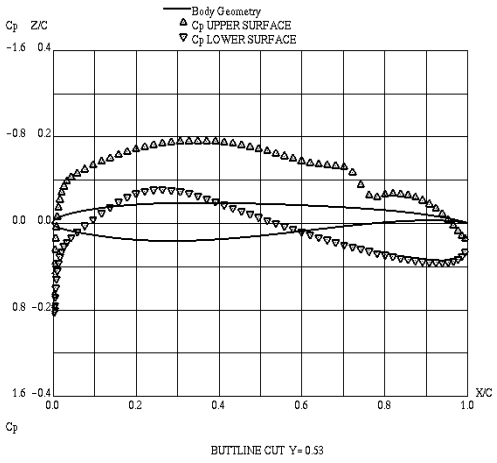

Fig. 6 One-point optimization - *Case*_4. Chordwise pressure distribution at the midsection of the wing at M=0.87, $C_L = 0.5$.

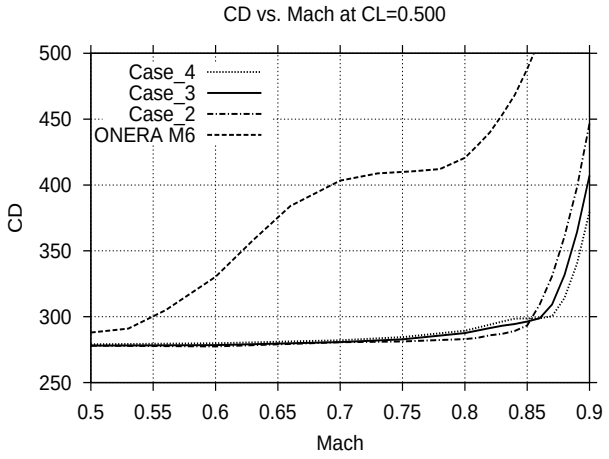

Fig. 8 Mach drag divergence of the optimized wings versus the original ONERA M6 wing. *Case*_2, *Case*_3 and *Case*_4 correspond to design *M*=0.84, 0.86 and 0.87 respectively.

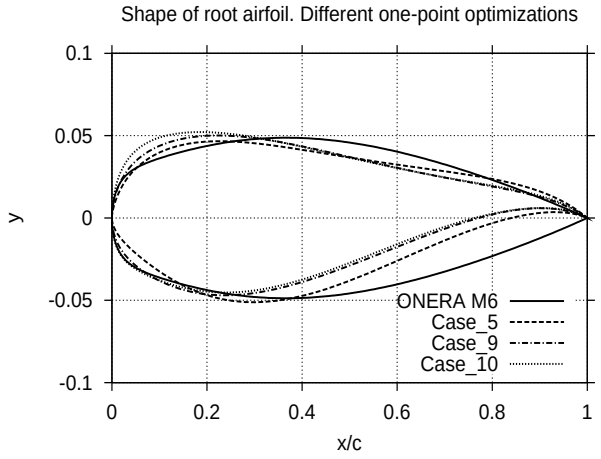

Fig. 9 Shape of optimized wings at root section for different N_{ws}.

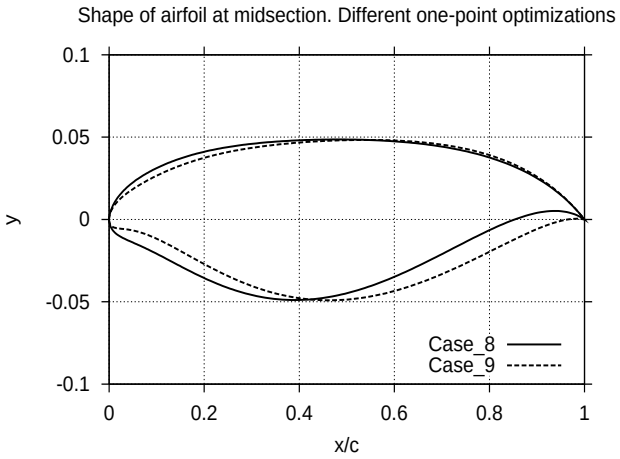

Fig. 11 Shape of the optimized wings at mid-section for different values of constraint on pitch moment. *Case*_8 - no constraint on C_M; *Case*_9 - $C_M > -0.1$.

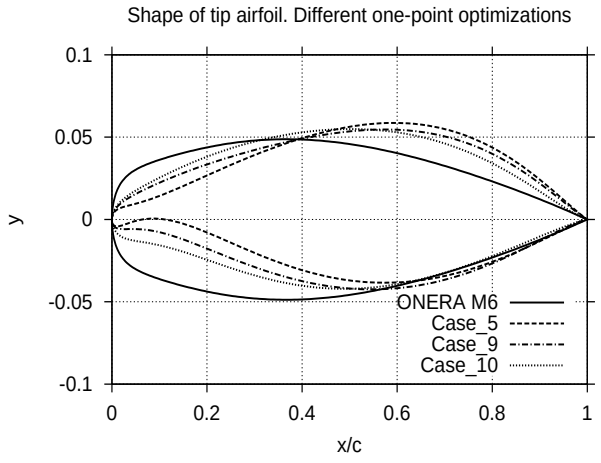

Fig. 10 Shape of optimized wings at tip section for different N_{ws}.

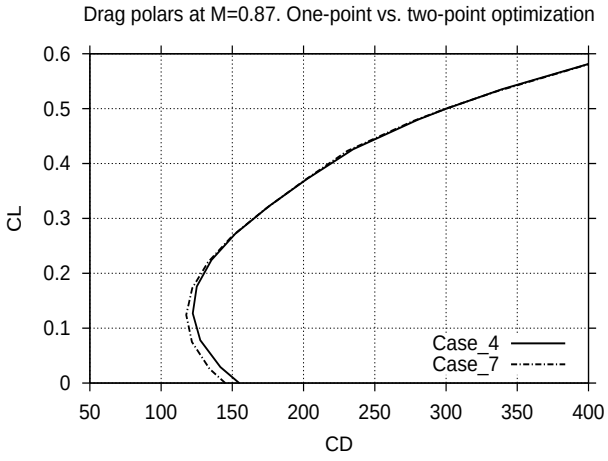

Fig. 12 Drag polars at $M = 0.87$. One-point optimization vs. two-point optimization.

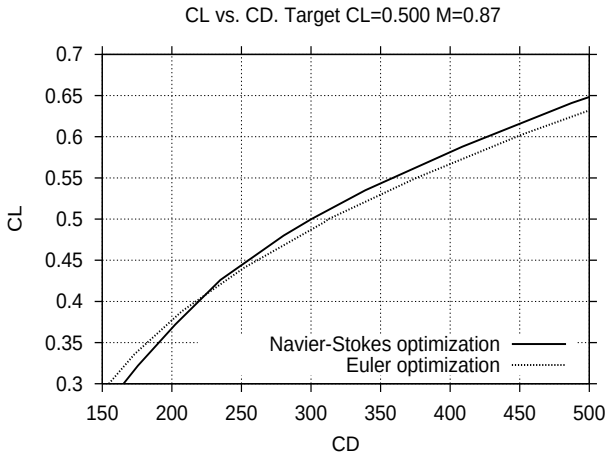

Fig. 13 Drag polars at $M = 0.87$. Navier-Stokes driven optimization vs. Euler driven optimization.

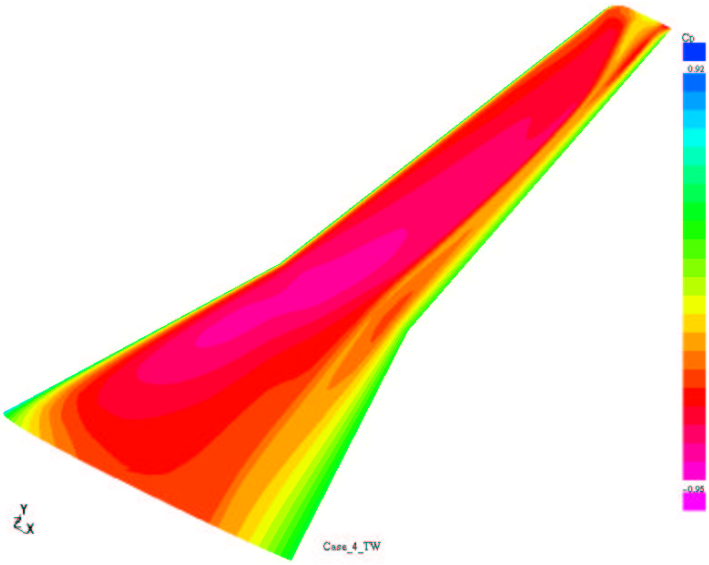

Fig. 15 Optimized transport-type wing - *Case_4_TW*. Pressure distribution on the upper surface of the wing at M=0.85, $C_L = 0.40$.

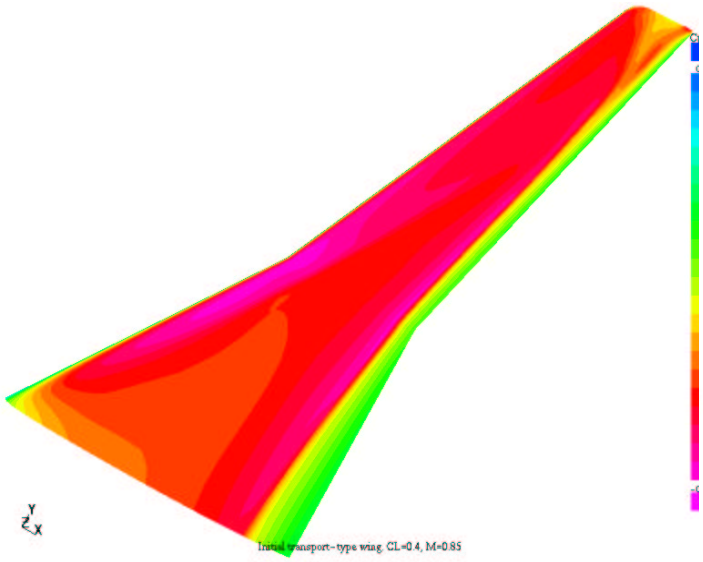

Fig. 14 Original transport-type wing. Pressure distribution on the upper surface of the wing at M=0.85, $C_L = 0.40$.

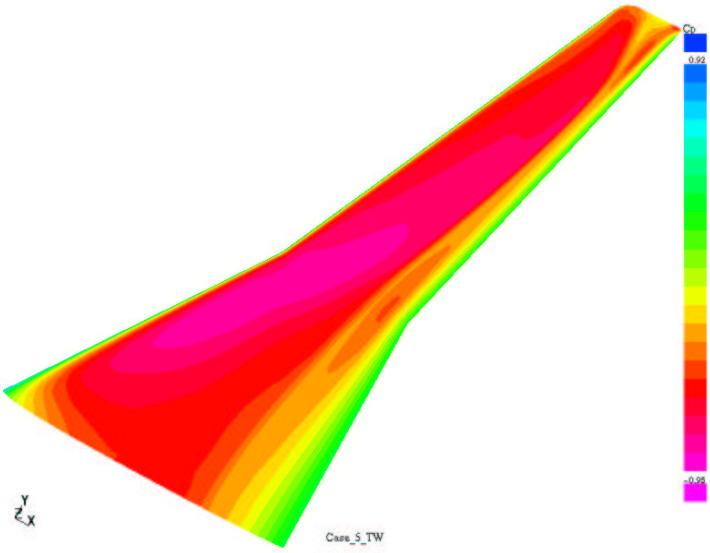

Fig. 16 Optimized transport-type wing - *Case_5_TW*. Pressure distribution on the upper surface of the wing at M=0.85, $C_L = 0.40$.

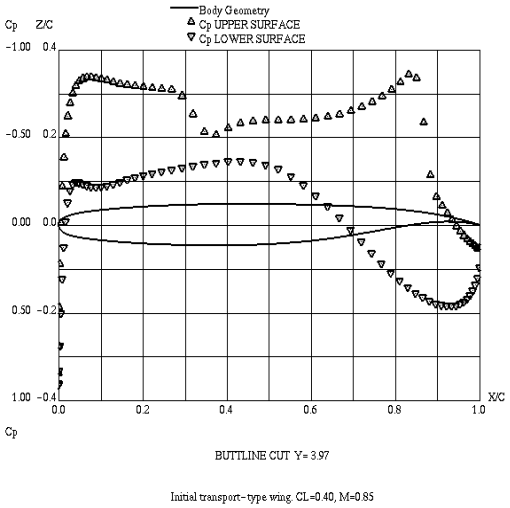

Fig. 17 Original transport-type wing at M=0.85, $C_L = 0.40$. Chordwise pressure distribution at 2y/b=0.48.

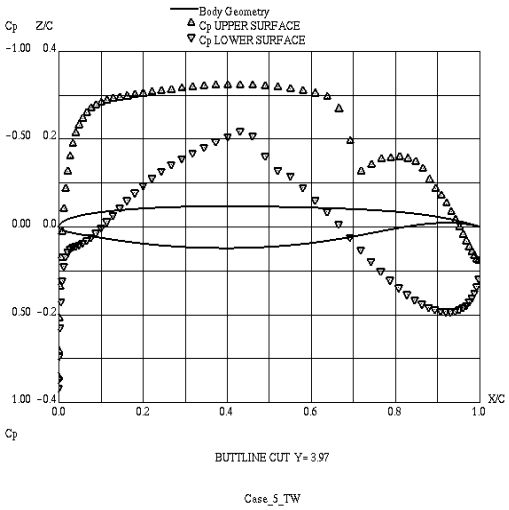

Fig. 19 Optimized transport-type wing (*Case*_5_*TW*) at M=0.85, $C_L = 0.40$. Chordwise pressure distribution at 2y/b=0.48.

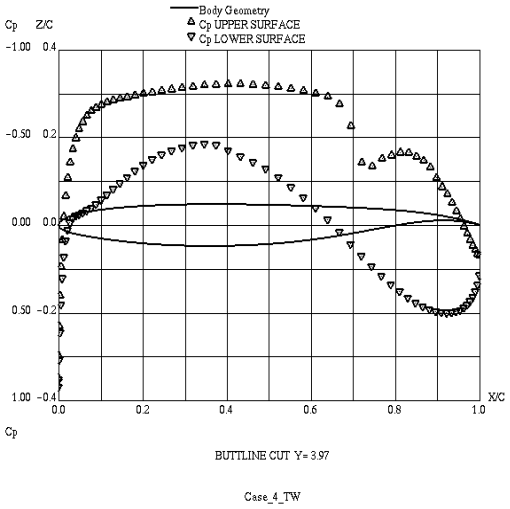

Fig. 18 Optimized transport-type wing (*Case*_4_*TW*) at M=0.85, $C_L = 0.40$. Chordwise pressure distribution at 2y/b=0.48.

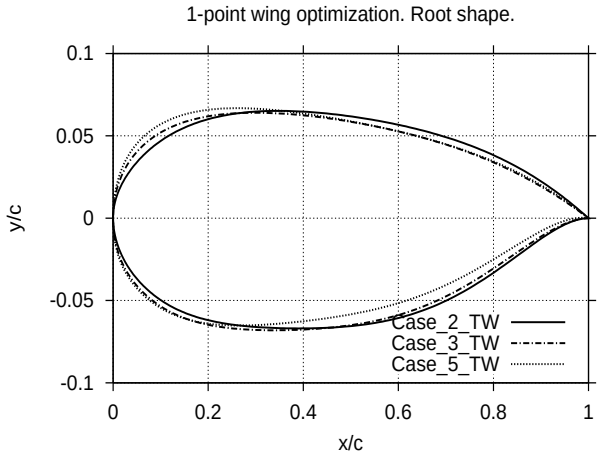

Fig. 20 Transport-type wing. Optimized shape of the root section for different target free-stream Mach numbers.

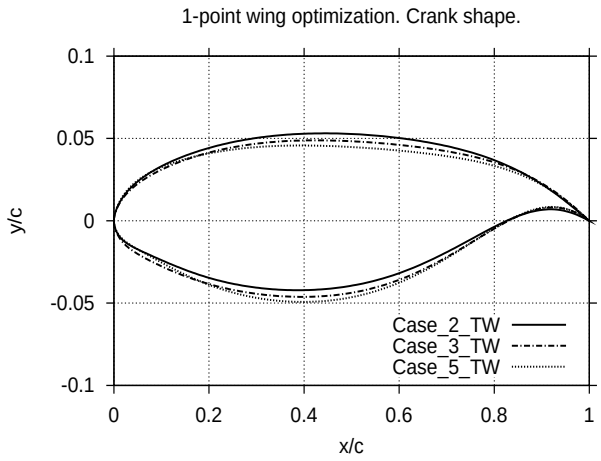

Fig. 21 Transport-type wing. Optimized shape of the crank section for different target free-stream Mach numbers.

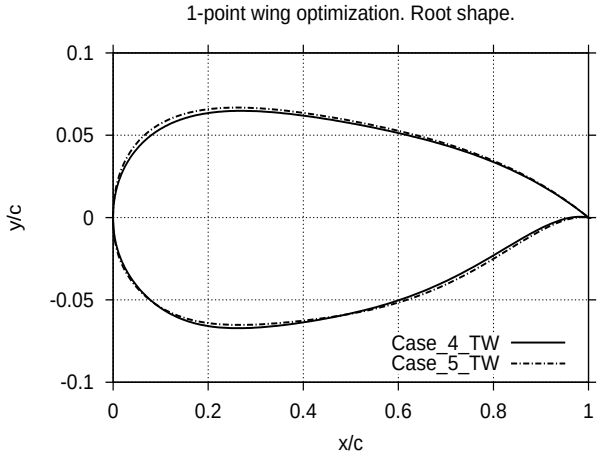

Fig. 23 Transport-type wing. Constrained vs. unconstrained pitch optimization. Optimized shape of the root section.

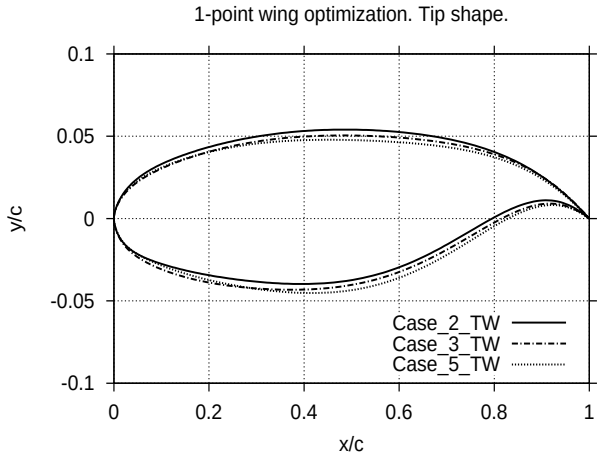

Fig. 22 Transport-type wing. Optimized shape of the tip section for different target free-stream Mach numbers.

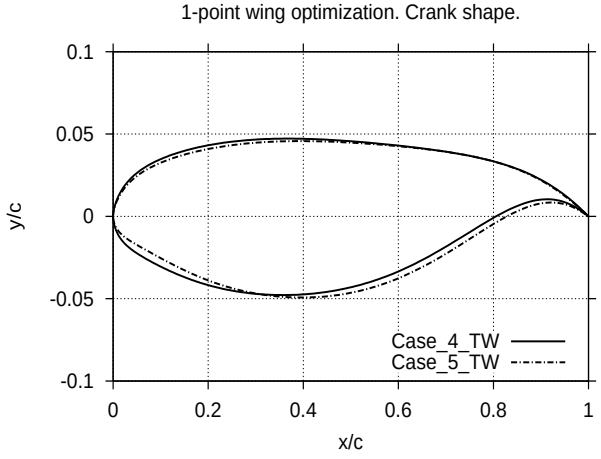

Fig. 24 Transport-type wing. Constrained vs. unconstrained pitch optimization. Optimized shape of the crank section.

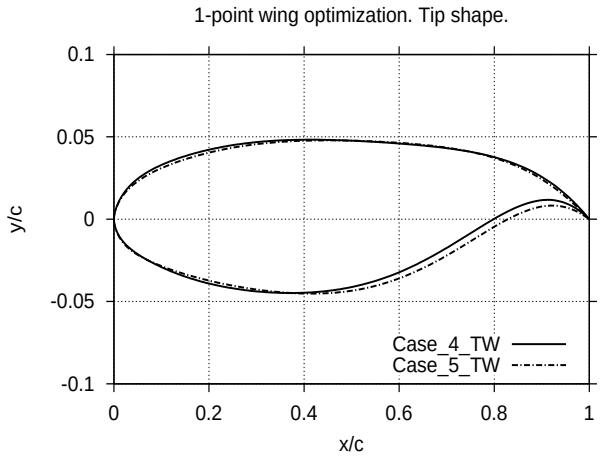

Fig. 25 Transport-type wing. Constrained vs. unconstrained pitch optimization. Optimized shape of the tip section.

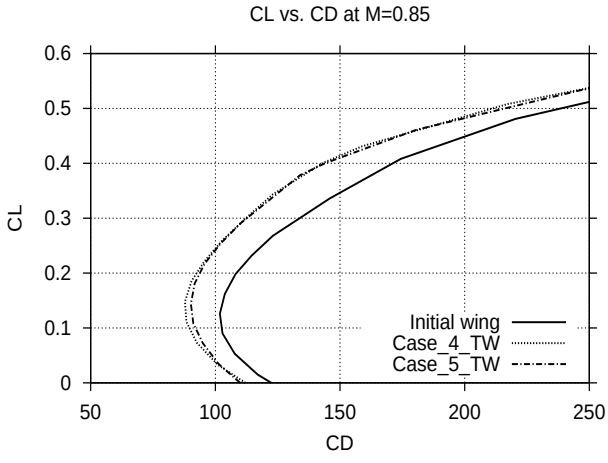

Fig. 26 Transport-type wing. Constrained vs. unconstrained pitch optimization. Drag polar at $M = 0.85$.

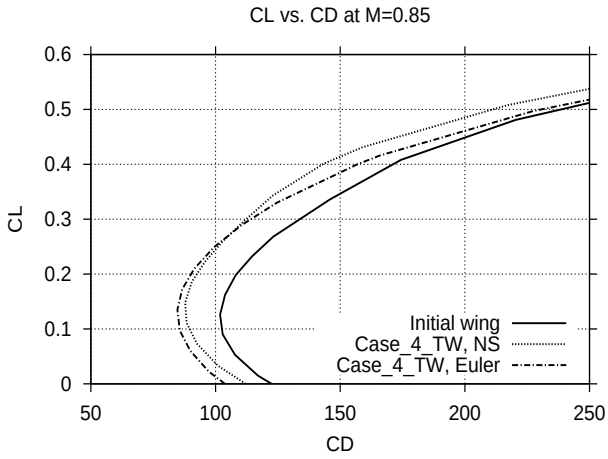

Fig. 27 Transport-type wing. Optimization driven by full Navier-Stokes CFD model vs. optimization driven by Euler CFD model. Drag polar at $M = 0.85$.

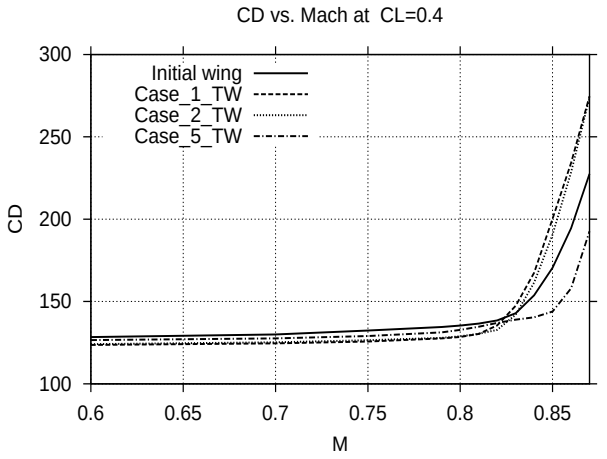

Fig. 28 Transport-type wing. Drag vs. Mach at $C_L = 0.4$. Optimized shapes vs. the original one.