Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers & Fluids 33 (2004) 1175–1200

computers & fluids

www.elsevier.com/locate/compfluid

Robust optimization of 2D airfoils driven by full Navier–Stokes computations

Sergey Peigin [a,*], Boris Epstein [b]

[a] *Engineering Division, Department 4473, Israel Aircraft Industries, Ben-Gurion International Airport, 70100, Israel*

[b] *Computer Science Department, The Academic College of Tel-Aviv-Yaffo, 4 Antokolsky St., Tel-Aviv 64044, Israel*

Received 26 February 2003; received in revised form 22 September 2003; accepted 25 November 2003
Available online 20 February 2004

Abstract

A new approach to the constrained design of aerodynamic shapes is suggested. 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 important features of the approach include: (1) a new strategy for efficient handling of non-linear constraints in the framework of GAs (2) scanning of the optimization search space by a combination of full Navier–Stokes computations with the ROM method (3) multilevel parallelization of the whole computational framework.

The method was applied to the problem of one-point transonic profile optimization with non-linear constraints. The results demonstrated that the approach combines high accuracy of optimization (based on full Navier–Stokes computations) and efficient handling of various non-linear constraints with high computational efficiency and robustness. A significant computational time-saving (in comparison with optimization tools fully based on Navier–Stokes computations) allowed the method to be used in a demanding engineering environment.

1. Introduction

The introduction of an automatic robust aerodynamic shape optimizer based on an accurate flow analysis may essentially reduce the time devoted to wind-tunnel testing and overall cost of aerodynamic design by reducing the number of optimization cycles to a single loop. On the other

[*] Corresponding author. Tel.: +972-3-935-83-35; fax: +972-3-935-86-75.
E-mail addresses: speigin@iai.co.il (S. Peigin), epstein@mta.ac.il (B. Epstein).

0045-7930/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2003.11.001

hand, the problem of constrained optimization still remains stiff and open (especially in demanding engineering environment). Thus a robust and efficient solution of this problem is highly challenging.

The first optimization method in aerodynamics was that of Lighthill [1] which proposed to employ the conformal mapping for the design of two-dimensional airfoils. An alternate method which solves the potential equations in the hodograph plane was established by Bauer et al. [2]. Constrained optimization was considered by Hicks and Henne [3] where the sensitivity derivatives were evaluated by finite differences.

Numerous optimization tools are based on gradient approach [4]. Methods, developed by Jameson [5,6], compute the gradients from the solutions to the flow equations and its adjoint equations.

The design problem may be characterized by a mix of continuous, discrete and integer design variables, and the resulting design space can be non-convex or even disjointed. For all these reasons, optimization methods which do not rely on the computation of gradients, in particular evolutionary programming and GAs, received a considerable growth of interest [7].

The present research was focused on creation of a robust optimization algorithm for practical design of aerodynamic airfoils.

The important features of the method include a new strategy for efficient handling of non-linear constraints in the framework of 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. New efficient approaches were developed in the following three crucial directions:

- High accuracy CFD computations.
- Robust handling of non-linear constraints.
- High computational efficiency of the whole method.

A CFD solver which drives the optimization process must meet the following requirements: (a) high accuracy of the Navier–Stokes computations on relatively coarse grids, (b) high robustness for a wide range of flows and geometrical configurations and (c) fast computational feedback.

The multiblock parallel full Navier–Stokes code NES [8–11] satisfies the first two requirements. In order to satisfy the requirement of fast computational feedback, we suggest to scan the optimization search space by using the ROM approach based on linked local data bases obtained by full Navier–Stokes computations.

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 which can not be resolved by conventional methods.

As an optimization tool, 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 through both feasible and infeasible points (contrary to the traditional approach where only feasible points may be included in a path).

A weakness of GAs lies in their poor computational efficiency, which prevents their practical use when the evaluation of the cost function is computationally heavy. As mentioned above, in order to overcome this obstacle, we introduced an intermediate computational tool which is based on a ROM approach.

This tool approximates the objective function using a very limited number of exact CFD computations, while providing a fast and reasonably accurate computational feedback in the framework of GAs search. In order to ensure the accuracy and robustness of the method, a multidomain prediction–verification principle is used. On the prediction stage, the genetic optimum search is concurrently performed on a number of search domains, after which, the whole set of corresponding optima is verified through full Navier–Stokes computations. Additionally, in order to ensure the global character of the search, the algorithm is iterated.

Optimization of aerodynamic shapes 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 an 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 and parallel evaluation of objective function.

The method was applied to the problem of transonic profile optimization with non-linear constraints. The results demonstrated that the approach combines high accuracy of optimization (based on full Navier–Stokes computations) and efficient handling of various non-linear constraints, with high computational efficiency and robustness.

A significant computational time-saving (in comparison with optimization tools fully based on Navier–Stokes computations) allowed the use of the method in a demanding engineering environment. The method retains high robustness of conventional GAs while keeping CFD computational volume at an acceptable level due to a limited use of full Navier–Stokes computations.

The problem statement is given in Section 2. The full Navier–Stokes CFD solver NES is presented in Section 3. In Section 4 a new approach to handling of non-linear constraints imposed on the optimization problem in the framework of GAs, is suggested. The application of the ROM method to the approximation of objective functions based on full Navier–Stokes computations is described in Section 5. The method of improving the overall computational efficiency of the optimization algorithm through the multilevel parallelization is presented in Section 6. Implementation of the optimization algorithm is outlined in Section 7. Numerical results of the optimization are given and analyzed in Section 8.

2. Problem definition

In this section a transonic flow one-point optimization problem is considered. The objective is to minimize the cost function (total drag coefficient C_D) of a two-dimensional airfoil subject to the following classes of constraints:

(1) Aerodynamic constraints such as prescribed constant total lift coefficient C_L^*, maximum local Mach number at given C_L.

(2) Geometrical constraints on the shape of the airfoil surface: relative thickness of the profile t/c, radius of leading edge R_L, trailing edge angle θ_T, shape "freeze" of certain portions of airfoil (such as lower or upper surfaces, trailing (leading) edge region, etc.):

$$t/c \geqslant (t/c)^*, \qquad R_L \geqslant R_L^*, \qquad \theta_T \geqslant \theta_T^* \tag{1}$$

where values $(t/c)^*$, θ_T^* and R_L^* are prescribed parameters of the problem.

As a gas-dynamic model for calculating C_D and C_L values, the full Navier–Stokes equations are used. The two-dimensional Navier–Stokes equations in Cartesian coordinate system (x, y) may be written in the form

$$\frac{\partial \mathbf{q}}{\partial t} + \operatorname{div} \mathbf{C} = \operatorname{div} \mathbf{V} \tag{2}$$

where the tensor $\mathbf{C} = (\mathbf{f}, \mathbf{g})$ represents the convection terms, the tensor $\mathbf{V} = (\mathbf{r}, \mathbf{s})$ represents the viscous terms, $\mathbf{q} = (\rho, \rho u, \rho v, E)$, ρ is the density, (u, v) is the velocity vector, E is the energy, t is the time, $\mathbf{f}, \mathbf{g}$ are the inviscid (convection) fluxes and $\mathbf{r}, \mathbf{s}$ are the viscous fluxes which depend in a non-linear mode on $\mathbf{q}$.

Numerical solution of the full Navier–Stokes equations (2) was based on the code NES [8–11], which possesses 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. The optimization technique employed Genetic Algorithms (GAs) in combination with a ROM method based on local data bases obtained by full Navier–Stokes computations. The novel features of the present method include a new strategy for efficient handling of non-linear 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 multilevel parallelization of the whole computational framework.

3. Full Navier–Stokes solver

The NES code is based on the essentially non-oscillatory (ENO) concept [12] with a flux interpolation technique [14] which allows accurate estimation of sensitive aerodynamic characteristics such as lift, pressure drag, friction drag and pitching moment [8,9].

To accelerate the convergence to the steady-state, a defect correction multigrid approach is used which employs a first-order-accurate driver and a high-order ENO defect correction [15]. The defect correction is applied only on the locally finest multigrid level; on other levels the relaxation process is simply driven by the first-order upwind biased operator.

The resulting multigrid method [11] retained the high accuracy of ENO approach with comparatively small number of multigrid cycles needed to reduce the error below the level of truncation errors, thus enabling the attainment of accurate flow characteristics on rather coarse grids.

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 three 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.

Natural finite differences are combined to approximate the viscous fluxes. The effects of turbulence are modeled through an eddy-viscosity hypothesis, with the Baldwin–Lomax turbulence model [13] used for turbulence closure.

A three-stage Runge–Kutta scheme is used in a total variation diminishing (TVD) form [14]. Specially constructed algorithms are used to diminish considerably the amount of computational work associated with changeability of template.

NES is a multiblock–multiface-structured code. This means that the overall computational domain is divided into smaller sub-domains (blocks) which can be treated independently of each other with data exchange among the blocks responsible for the validity of boundary conditions on the block interfaces.

The important advantage of the solver NES as a driver of optimization process is its ability to supply reliable and sufficiently accurate results already on relatively coarse meshes and thus to reduce dramatically the volume of CFD computations.

4. Genetic Algorithm for constrained optimization problem

Evolutionary algorithms (in particular Genetic Algorithms) have drawn much attention as optimization methods in the last two decades [16–19]. They are semi-stochastic optimization methods that are conveniently presented using the metaphor of natural evolution: a randomly initialized population of individuals (set of points of the search space at hand) evolves following a crude parody of the Darwinian principle of the survival of the fittest. Thus, the basic idea behind Genetic Algorithms is to mathematically imitate the evolution process of nature. The GA algorithms are based on the evaluation of a set of solutions, called population. The population is treated with genetic operators.

At the iteration i the population $\mathbf{X}^i$ consists of a number of $\mathbf{M}$ individuals $\{x_j\}$, that is, solutions, where $\mathbf{M}$ is the population size. The characteristics of the individuals can be encoded using either binary or real numbers. The corresponding methods are called binary-coded or real-coded Genetic Algorithms, respectively. Thus each individual x_j is denoted by a vector of variables. The suitability of an individual is determined by the value of objective function, to be called a fitness function.

A new population of individuals is generated using variations of "genetic" operators: selection, crossover and mutation. Parents are chosen by selection, and new offsprings are produced with crossover and mutation. All these operations include randomness. The success of the optimization process is improved by the principle of elitism, where the best individuals of the current generation are included into the next generation. The main point is that the probability of survival of new individuals depends on their fitness: the best are kept with a high probability, the worst are rapidly discarded.

4.1. Real-coded Genetic Algorithm

As a basic algorithm, a variant of the floating-point GA is used. The floating-point GAs realize a compromise between binary-coded GAs and Evolution Strategies, since they use most of the classical GAs mechanisms, whereas, they work directly at the phenotypic level like Evolution Strategies [20].

We used the tournament selection, which enables us to increase the diversity of the parents. Three types of the crossover operator have been employed: single point, uniform and arithmetical crossover. As the mutation operator we applied the non-uniform mutation defined by Michalewicz [19]. To resolve one of the main problems that arises in GAs—a premature convergence—we used distance-dependent mutation [21]. This approach consists of using the distance between two mates in order to compute the mutation rate. This means that the mutation rate is no longer a constant parameter; it is dynamically computed for both children and depends on the parents. Every time a couple of individuals is chosen for mating, the mutation rate is computed and will be applied to their children after crossover occurs. If the parents are quite close (that is, their mating is likely to be considered as "incest"), that would lead to a high mutation rate for their children, whereas mutation rate will be smaller if they are different. To improve the convergence of the algorithm we also use the elitism principle.

4.2. Handling of non-linear constraints

In the considered optimization problem, the presence of constraints is of major importance. The reason for this lies in the fact that, like in most challenging problems, the optimal solution does not represent a local minimum in the conventional sense of the word. Instead, it is located on an intersection of hypersurfaces of different dimensions, generated by linear and non-linear constraints. The problem of finding such an extremum is essentially complicated by the fact that these hypersurfaces, which define a feasible search space, are not known in advance and possess irregular topology.

For example, in the problem defined in Section 2, it is aerodynamically expected that an optimal airfoil will possess a minimum allowed thickness. This implies that the optimal point will reside exactly on the corresponding hypersurface.

Unfortunately, in their basic form, Genetic Algorithms are not capable of handling constraint functions limiting the set of feasible solutions. Therefore, additional methods are needed to keep the solutions in the feasible region [25].

To explain the new constraint-handling approach let us assume that we have to find the minimum of the objective function $f(x)$ in the feasible region $\mathbf{F_f}$, defined by a non-linear constraint $g(x) \leqslant 0$. Let us also consider the most challenging case when the optimal point x_{opt} is located close to the constraint boundary of the feasible region $g(x) = 0$, and, that this boundary is not exactly known in advance, before the evaluation of the objective function.

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). Since in our case a topology of feasible and infeasible sub-domains is rather complex, the realization of such a strategy should significantly improve the accuracy and

efficiency of optimization. The idea is that the information from infeasible sub-domains can be very important for the optimization as a path to the optimal point via infeasible points can be essentially shorter (or even the only possible, in the case of a non-simply-connected optimization space).

(2) To implement the new strategy, it is suggested 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 by means of GAs due to their basic property: contrary to classical optimization methods, GAs are not confined to only smooth extensions.

In fact, this extension should satisfy only the condition that follows immediately from the main idea to increase a diversity of current population: the objective function in infeasible regions should be defined in such a way that it keeps in the current population a certain number of infeasible individuals, which are located rather close to the constraint boundary. In such a case we can expect, with rather high probability, that the crossover between feasible and infeasible individuals will produce high-fitness children.

Based on the results of the previous analysis, the following two-step approach to the extension of the objective function into infeasible region is proposed. A starting point at the first step is to estimate the order of the objective function value over the feasible sub-domain. It can be done using the preliminary information on the behaviour of the objective function, or it can be based on testing a number of feasible points.

Let us assume that the estimation has the form: $f(x) \approx A$ $(x \in \mathbf{F_f})$. Based on this estimation we define the following first-step approximation for the modified objective function $f^*(x)$:

$$f^*(x) = \begin{cases} f(x) & \text{if } g(x) \leqslant 0 \\ B & \text{if } g(x) > 0 \end{cases} \tag{3}$$

where $B \gg A$.

The next step in the approach is to run the above described GA for the solution of the optimization problem (in fact a non-constrained problem) with the first-step modified objective function $f^*(x)$ and to estimate the value of the objective function for feasible individuals in the neighbourhood of the constraint boundary. Using the results of these calculations the first-step estimation is corrected: $f(x) \approx C$. Then the modified objective function $f^{**}(x)$ for the total search space $\mathbf{F}$ is finally defined as follows:

$$f^{**}(x) = \begin{cases} f(x) & \text{if } g(x) \leqslant 0 \\ \alpha_1 C + \alpha_2 g(x) B & \text{if } g(x) > 0 \end{cases} \tag{4}$$

where $\alpha_1 \geqslant 1.0$ and $\alpha_2 > 0$ are problem-dependent real numbers.

4.3. Implementation of the Genetic Algorithm

It was assumed that in the Cartesian coordinate system (x, y) 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)

$$\vec{G}^k(t) = \sum_{i=0}^{N} B_{N,i}\vec{P}_i^k, \quad B_{N,i} = C_N^i t^i (1-t)^{N-i}, \quad C_N^i = \frac{N!}{i!(N-i)!}, \tag{5}$$

where t denotes the parameter of the curve 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 (5) the Bezier curve is completely determined by the Cartesian coordinates of the control points.

For our optimization problem the first $\vec{P}_0^k = (0,0)$ and the last $\vec{P}_N^k = (1,0)$ $(k = u, l)$ points are fixed just fixing the position of leading and trailing edges. We also fix all the abscisses x_i^k of the control points $\vec{P}_1^k, \ldots, \vec{P}_{N-1}^k$. 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 $y_1^{\mathrm{u}} = -y_1^{\mathrm{l}}$.

This means that a search string $S = (a_1, a_2, \ldots, a_{N-1}, a_N, \ldots, a_{2N-5})$ has the following form:

$$S = \begin{cases} a_i = y_i^{\mathrm{u}}, & 1 \leqslant i \leqslant (N-1) \\ a_i = y_{i-N+2}^{\mathrm{l}}, & N \leqslant i \leqslant (2N-5) \end{cases}$$

Thus a string S contains $2N-5$ values (ordinates of control points). These values are varied within the search domain D. The domain D is determined by Min_i and Max_i values, which are the lower and upper bounds of the variable a_i.

Based on the above approach to the handling of constraints, the modified objective function Q for the solution of the single point drag minimization problem was defined as follows:

$$Q = \begin{cases} 0.1 + [(t/c)^* - (t/c)] & \text{if } (t/c) < (t/c)^* \\ 0.2 + [R_{\mathrm{L}}^* - R_{\mathrm{L}}] & \text{if } R_{\mathrm{L}} < R_{\mathrm{L}}^* \\ 0.3 + [\theta_{\mathrm{T}}^* - \theta_{\mathrm{T}}] & \text{if } \theta_{\mathrm{T}} < \theta_{\mathrm{T}}^* \\ 0.5 & \text{if } y^{\mathrm{u}}(t) < y^{\mathrm{l}}(t) \\ C_{\mathrm{D}} & \text{otherwise} \end{cases} \tag{6}$$

The choice of constants in Eq. (6) was fairly straightforward, based on the following principle. On the one hand, their values should be at least several times greater than the upper bound of C_{D} (about 0.0500 for the considered class of problems), which ensures that, for any feasible point, the value of the objective function will be low in comparison with that of any infeasible point. On the other hand, these constants should not be too high, in order to ensure that a sufficient number of infeasible points will be present in the population (as it was explained above in Section 4.3).

The above formulation may be extended for multipoint design problems. In this case the value of C_{D} should represent a weighted combination of total drag values at the flight points participating in the optimization.

5. ROM–LAM approximation of fitness function

Genetic Algorithms which are less susceptible to pitfalls of convergence to local optima, suggest a good alternative to conventional optimization techniques. The major weakness of GAs lies in their poor computational efficiency, which prevents the practical use when the evaluation of the cost function is expensive as happens in the framework of the full Navier–Stokes model even in the two-dimensional case.

For example, an algorithm with the population size $\mathbf{M} = 100$ requires (for the case of 200 generations) at least 20,000 evaluations of the cost function (CFD solutions). A fast full Navier–Stokes evaluation takes at least a couple of minutes of CPU time. That means that one step of such an algorithm takes about 650 h, what is practically unacceptable.

One of the popular ways of overcoming this difficulty is to use ROM approach in the broad sense of the word. Among these we may mention the use of simpler gas-dynamic models (see e.g. [22]), representation of the solution of gas-dynamic problem in terms of its eigenmodes [23] or representation of the aerodynamic system using the Volterra theory of non-linear systems [24].

In this work we use ROM approach in the form of Local Approximation Method (LAM). On every optimization step, the solution functionals which determine a cost function (such as lift and drag coefficients in the case of drag minimization), 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. This space is formed by Bezier coefficients of an airfoil, and the basic point corresponds to the initial profile at a current optimization step.

The specific algorithm is described below in the case of single point drag minimization.

Denote a_i^n and α^n the Bezier coefficients of an initial profile at nth optimization step and the angle of attack, corresponding to the prescribed C_L^*, respectively. Then each airfoil can be determined by deviations δ_i^n from the coefficients of the initial profile. At fixed values of other flow parameters, the solution functionals depend on the values of δ_i^n and δ_α^n (a deviation from the initial angle of attack). In the optimization process the following local approximation of a functional F^n is used (superscript n is omitted and $F = C_L, C_D$):

$$F(a_1 + \delta_1, \dots, a_{2N-5} + \delta_{2N-5}, \alpha + \delta_\alpha) = F^\circ + \sum_{j=1}^{2N-5} \Delta F_j + \Delta F_\alpha \tag{7}$$

$$\Delta F_j = \begin{cases} \Delta F_j^+ & \text{if } (F_j^+ - F^\circ)(F_j^- - F^\circ) \leqslant 0, \quad \delta_j \geqslant 0 \\ \Delta F_j^- & \text{if } (F_j^+ - F^\circ)(F_j^- - F^\circ) \leqslant 0, \quad \delta_j < 0 \\ \Delta F_j^{+-} & \text{if } (F_j^+ - F^\circ)(F_j^- - F^\circ) > 0 \end{cases}$$

$$\Delta F_j^+ = \frac{\delta_j}{\Delta_j}(F_j^+ - F^\circ), \qquad \Delta F_j^- = -\frac{\delta_j}{\Delta_j}(F_j^- - F^\circ), \qquad \Delta F_\alpha = \frac{\delta_\alpha}{\Delta_\alpha}(F_\alpha - F^\circ)$$

$$\Delta F_j^{+-} = \frac{\delta_j(\delta_j + \Delta_j)}{2\Delta_j^2} F_j^+ - \frac{\delta_j^2}{\Delta_j^2} F^\circ + \frac{\delta_j(\delta_j - \Delta_j)}{2\Delta_j^2} F_j^-$$

In the approximation (7) the following notation is used:

$$F^\circ = F(a_1, \dots, a_{2N-5}, \alpha), \qquad F_\alpha = F(a_1, \dots, a_{2N-5}, \alpha + \Delta_\alpha),$$

$$F_j^+ = F(a_1, \dots, a_{j-1}, a_j + \Delta_j, a_{j+1}, \dots, a_{2N-5}, \alpha) \quad (j = 1, \dots, 2N-5)$$

$$F_j^- = F(a_1, \dots, a_{j-1}, a_j - \Delta_j, a_{j+1}, \dots, a_{2N-5}, \alpha) \quad (j = 1, \dots, 2N-5)$$

Here the local data base values F°, F_j^+, F_j^- and F_α 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 positive variations $\{\Delta_j\}$ corresponding to the Bezier coefficients $\{a_j\}$ and by the variation Δ_α of the angle of attack α.

In fact, relation (7) represents a mixed linear-quadratic approximation in the neighbourhood of the basic point F° which employs the local data base. One-dimensionally, we use either a one-sided linear approximation (in the case of monotonic behaviour of the functional F) or a quadratic approximation (otherwise).

Values of C_L and C_D calculated via relations (7) were systematically compared with results of numerical solution of the full Navier–Stokes equations for a wide range of feasible values of Δ_j. It was concluded that the formula (7) possesses acceptable accuracy at least in the following range of deviations δ_j and δ_α:

$$-2\Delta_\alpha \leqslant \delta_\alpha \leqslant 2\Delta_\alpha, \qquad -2\Delta_j \leqslant \delta_j \leqslant 2\Delta_j \quad (j = 1, \dots, 2N-5)$$

Using (7) we can obtain the following formula for cost function C_D at prescribed lift coefficient C_L^* (for current point at the search space $(a_1 + \delta_1, \dots, a_{2N-5} + \delta_{2N-5})$) which is based on the local data base C_D°, C_{D_α}, $C_{\mathrm{D}_j}^+$, $C_{\mathrm{D}_j}^-$, C_L°, C_{L_α}, $C_{\mathrm{L}_j}^+$, $C_{\mathrm{L}_j}^-$:

$$C_\mathrm{D} = C_\mathrm{D}^\circ + \sum_{j=1}^{2N-5} \Delta C_{\mathrm{D}_j} + \frac{\delta_\alpha}{\Delta_\alpha}(C_{\mathrm{D}_\alpha} - C_\mathrm{D}^\circ) \tag{8}$$

$$\delta_\alpha = \frac{\Delta_\alpha}{C_{\mathrm{L}_\alpha} - C_\mathrm{L}^\circ}\left[C_\mathrm{L}^* - C_\mathrm{L}^\circ - \sum_{j=1}^{2N-5} \Delta C_{\mathrm{L}_j}\right]$$

6. Improvement in computational efficiency of the method

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 large number of heavy CFD runs, and a number of such steps is needed to reach the optimum. Though the use of the suggested ROM–LAM approach essentially reduces the volume of computations, the total number of CFD runs is still measured by hundreds.

Thus the construction of a computationally efficient algorithm is vital for the success of the method in engineering environment.

6.1. Computational grids

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 way 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.

An additional source of decreasing the volume of computational work is the use of computational grids coarser than those needed for exact estimations of the objective function. It is feasible if the grid coarsening preserves the hierarchy of fitness function values on the search space (that is, the relation of order is invariant with respect to grid coarsening). This means that the objective function Q_c defined on a coarse grid can be used for solution of the optimization problem, if for every pair of points x_1, x_2 belonging to the search space, the following relation between the values of an objective function Q_c on a coarse grid:

$$Q_c(x_1) \geqslant Q_c(x_2) \tag{9}$$

implies the same order relation for the objective function Q_f defined on a fine grid:

$$Q_f(x_1) \geqslant Q_f(x_2) \tag{10}$$

6.2. Two-level parallelization strategy

An additional way to improve the computational efficiency of the algorithm is to use the following multilevel parallelization strategy:

- Level 1—Parallelization of full Navier–Stokes solver.
- Level 2—Parallel evaluation of objective function.

The first level parallelization approach [10] was based on the geometrical decomposition principle. All processors were divided into two groups: one master-processor and P_s slave-processors. An overall computational domain being block-structured, the groups of blocks were mapped to the slave-processors. The main goal of each slave was to carry out calculations associated with the solution of the Navier–Stokes equations, at the cell points in the blocks mapped to the slave-processor. The aim of the master-processor was to form output files containing global information for the whole configuration by receiving the necessary data from the slave-processors.

In this case the message-passing can be divided into two classes: data transfer from the slaves to the master-processor, and between the slaves (exchange of boundary data between those neighbouring blocks which are located in different slave-processors).

The important advantage of the approach is that the suggested structure of message-passing ensures a very high scalability of the algorithm from a network point of view, because, on the average, the communication work per processor is not increased if the number of processors is increased [10].

An additional means of increasing the parallel efficiency, which was also used in the parallel NES code, is based on the overlapped communication and computation concept. This means that the "send" procedures are independently executed in each block as soon as the data to be transferred have been calculated. In fact, such an asynchronous approach enables us to reduce the losses due to communication, because the data are transferred during the load balancing waiting time.

Finally, 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 reduce significantly the execution time for large-scale CFD computations.

The second level of parallelization is needed in order to organize a parallel CFD scanning of the optimization search space. On this level of parallelization all the processors were 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 geometries).

The aim of the main-processor was to distribute the geometries among master-processors, to receive from these master-processors the results of CFD computations and, finally, to create local CFD data bases. The goal of each master-processor was to organize the first level parallelization of the full Navier–Stokes solver corresponding to its own point in the search space (its own airfoil).

Because the volume of data transfer between main-processor and master-processors was negligible, and master-processors execute their own computations independently, the parallel efficiency of the second level parallelization was very close to 100%.

Finally we can conclude that the two-level parallelization approach allowed us to sustain a high level of parallel efficiency on massively parallel machines, and by this way to essentially improve the computational efficiency of the suggested optimization algorithm.

7. Implementation of the optimization algorithm

The optimization method involved the following algorithmic steps:

1. The initial basic point in the search space is determined.
2. The CFD local data base for C_L and C_D is obtained by solving the full Navier–Stokes equations at the neighbouring points of the basic point in the search space.
3. The local CFD data base is included in the global CFD data base.
4. Using a local approximation of the cost function (by means of ROM–LAM method), the Genetic Algorithm is applied for various embedded search domains D_k (corresponding to different search scales), and optimal points O_k for each domain are obtained ($k = 1, \dots, N_D$; N_D is the number of search domains). The full Navier–Stokes solver NES is then applied to each optimal point O_k, and the corresponding data are added to the global CFD data base.
5. A new basic point is determined as the best point taken from the global CFD data base.
6. If the prescribed convergence accuracy is achieved, then stop. Otherwise, the optimization process is repeated from step 2.

Note that step 4 implements, in fact, a multidomain prediction–verification principle which allows the method to overcome the local nature of the ROM–LAM approximation, and thus to ensure the accuracy and robustness of the optimization method.

On the prediction stage, the genetic optimum search is concurrently performed on a number of search domains, after which, the whole set of corresponding optima is verified through full Navier–Stokes computations. Additionally, in order to ensure the global character of the search, the algorithm is iterated.

8. Analysis of results

We present here applications of the the above optimization algorithm to airfoil design. The design problem consists of the minimization of total drag starting from the RAE2822 airfoil at a variety of transonic design points. In all the computations a fully turbulent flow model is assumed.

In order to verify the Navier–Stokes solver NES, the test-case of flow over ARA M100 wing-body at subsonic and transonic conditions, was selected. The case has been the subject of numerous studies in the past (see e.g. [26]), with reliable experimental data available [27].

The verification runs employed the multigrid set containing three levels, each multigrid level contains 24 blocks united in a multiblock structure. The finest computational mesh comprised slightly less than one million points.

The analysis of results showed good agreement with experiment [27], both in terms of surface pressure coefficients and in terms of the prediction of the total drag coefficient in transonic regime starting from $M = 0.50$. Two examples of comparison with the above experiment and previous Navier–Stokes computations [26] are presented in Figs. 1 and 2. Computed chordwise pressure distribution is compared with experiment in Fig. 1 at high transonic flight conditions. Lift/drag polars at $M = 0.8027$ are shown in Fig. 2. Note, that the present computation not only favourably compares with experiment but also indicates a good grid convergence.

As it was demonstrated in [8,9], the NES solver supplies accurate asymptotically converged lift and drag coefficients values for transonic 2D airfoils with grids containing on fine level about 321×97 computational points in the streamwise and normal to surface directions, respectively. 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 invariancy of the hierarchy of objective function values on coarse and fine grids (see Section 6). The ability of the code to produce qualitatively

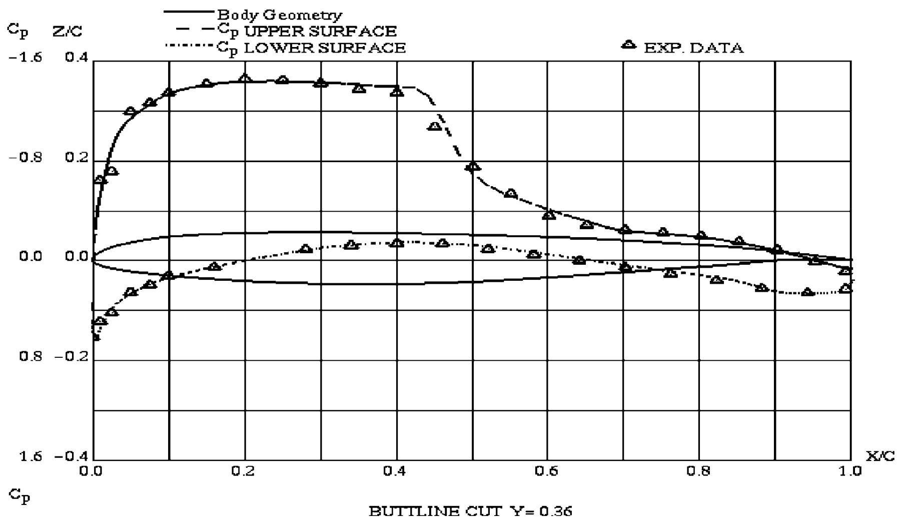

Fig. 1. ARA M100 wing-body configuration. Chordwise pressure distribution for wing span station $Y = 0.36$, $\alpha = 2.873°$. Dashed and dotted lines—present computation; triangles—experimental data [27].

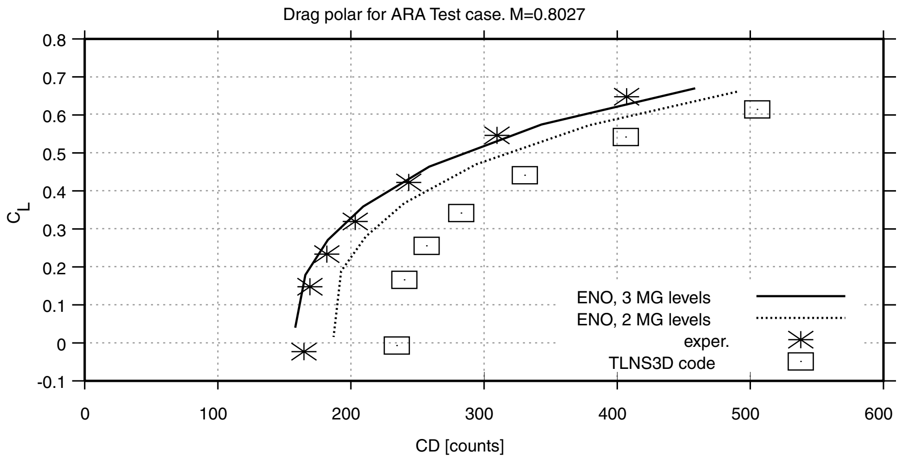

Fig. 2. ARA M100 wing-body configuration. Drag polar at $M = 0.8027$. Solid and dotted lines—present computations; asterisks—experimental data [27]; squares—computation [26].

reliable results on coarse grids in transonic regime in the presence of shock waves, is demonstrated in Fig. 3. The location and amplitude of shock are predicted correctly already on the coarse level, outside the shock region, the two pressure distributions are almost identical.

It appeared that the four times coarser in each direction (81×25) grids satisfy the invariancy conditions (9) and (10). This allowed us to use meshes with such a resolution for optimization purposes.

The fast transformation of grids, mentioned above in Section 6, was implemented in the following way. Assume that $\Delta\mathbf{r}_{i,0}$ is a change in the airfoil geometry at the grid point with indices

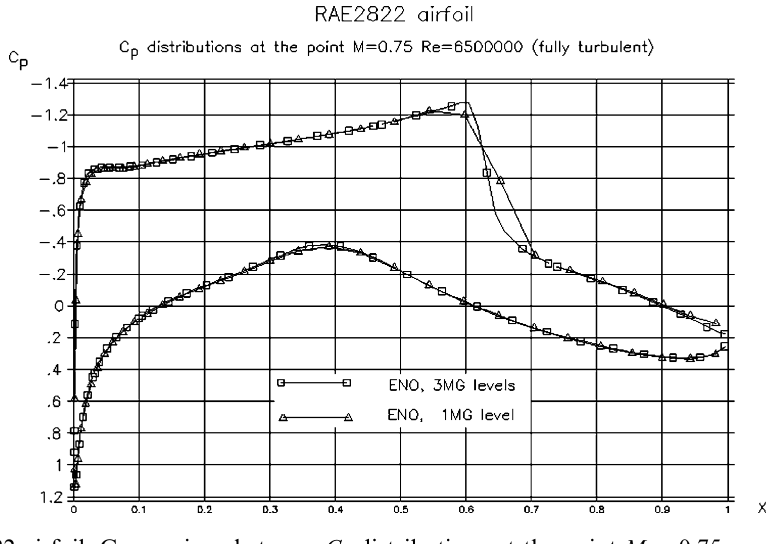

Fig. 3. RAE2822 airfoil. Comparison between C_p distributions at the point $M = 0.75$, $\alpha = 2.0°$, $C_L = 0.72$.

$(i, 0)$ in the streamwise and normal to surface directions, respectively. Then the coordinates $\mathbf{r}_{i,j}^{\text{new}}$ of the new grid are obtained by propagation of the shift $\Delta\mathbf{r}_{i,0}$ along the grid line $i = \text{const}$:

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

Two-level parallelization strategy based on the PVM software package was implemented on a cluster of MIMD multiprocessors consisting of 72 HP NetServer LP1000R nodes. Each node has two processors, 2GB RAM memory, 512 KB Level 2 Cache memory and full duplex 100 Mbps ETHERNET interface. Totally this cluster contained 144 processors with 144 GB RAM and 36 MB Level 2 Cache memory managed by the MOSIX software package [28] that enhances the LINUX kernel with cluster computing capabilities.

In the implementation of the Genetic Algorithm the population size was equal to 200, while the number of search domains was equal to 8. In order to ensure the robustness of the GA search, 10 randomly generated initial populations (for each search domain) were employed. In addition to the serial version of the search, a parallel variant of the GA was also implemented.

Beyond the reduction in computer time (needed for the optimization search), the parallelization allowed to improve the convergence rate of the GA by means of data exchange between concurrent search processes. An example of the GA convergence is given in Fig. 4. On the whole, the algorithm exhibited a good convergence: on average about 200 generations were needed in order to achieve a reasonable accuracy.

Thus the number of fitness evaluations by means of the approximated relation (8) was equal to 3,200,000 (population size × number of generations × number of initial populations × number of search domains) per optimization step. The corresponding number of "exact" Navier–Stokes computations was equal to 63 (36 data base computations + 27 runs for the verification of optimal points). An average number of optimization steps was about 10.

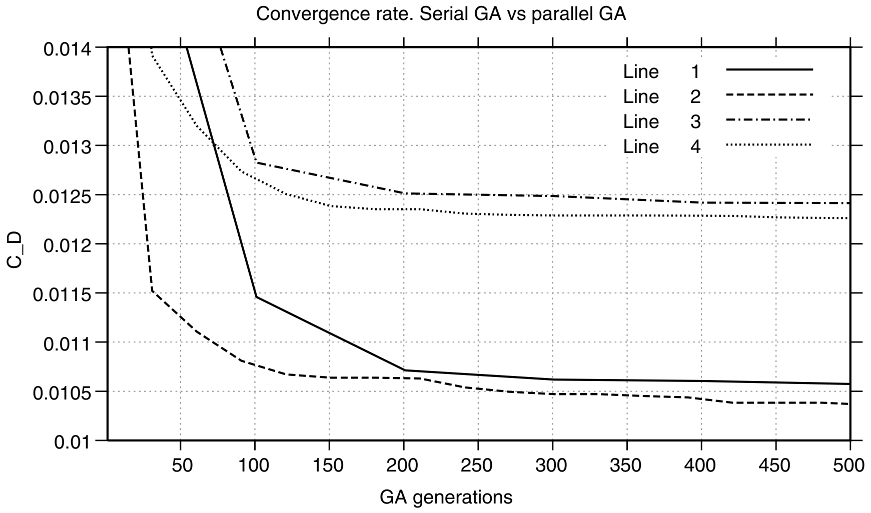

Fig. 4. Convergence rate of GAs optimum search. Serial GA (lines 1,3) vs. parallel GA (lines 2,4). Lines 1,2 and lines 3,4 correspond to different search domains.

Table 1
Optimization conditions and constraints for different test cases

Case no.	Design C_L	Design M	θ_T^*
1	0.300	0.75	0.0°
2	0.500	0.75	0.0°
3	0.650	0.75	0.0°
4	0.675	0.75	0.0°
5	0.745	0.75	0.0°
6	0.745	0.75	3.4°
7	0.745	0.74	3.4°
8	0.745	0.76	3.4°

In the following, we present the results of drag minimization of RAE2822 airfoil at $Re = 6.5 \times 10^6$ and different values of design C_L and Mach numbers for a wide range of flight conditions. A total of eight test cases was studied. Varying design conditions and constraints are summarized in Table 1.

As already mentioned above, the use of high accuracy Navier–Stokes computations makes the problem very time-consuming. Due to the suggested ROM–LAM approach and the multilevel parallelization strategy, the time needed for a optimization is reduced to a level acceptable in engineering environment. The code possesses a high efficiency of parallelization (about 96%). A sample optimization run takes 4–5 h on the preceding cluster with 144 processors, instead of 24–25 days on a serial processor.

In all the test cases the profile maximum thickness was kept on the level of $(t/c)^* = 12\%$. Additional constraints were imposed on the minimum of leading edge radius value ($R_L \geqslant R_L^* = 0.0029$) and on the profile shape (in order to avoid "fishtails"). The corresponding optimal profiles are designated by *Case_1–Case_8*.

Aerodynamically, the cases covered different parts of aerodynamic polars in the neighbourhood of $M = 0.75$. The first test case (*Case_1*) corresponds to a design point lying in the vicinity of the minimum drag point of the original RAE2822 profile. *Case_4–Case_8* correspond to flight conditions with a strong shock-boundary layer interaction.

In order to verify the method, the design point $C_L = 0.65$, $M = 0.75$ (a moderate shock case—*Case_3*) was employed. The case served for verification studies in a number of publications, most recently in [29].

At this point the initial solution (corresponding to the RAE2822 airfoil) gives the total of 149 drag counts (94 counts due to pressure drag and 55 counts due to viscous forces), while the corresponding drag values in [29] amount to 148, 92 and 56 counts, respectively. The two initial solutions reasonably agree, thus giving a fair basis for comparison of the optimization results.

In [29] a reduction of 50 drag counts was achieved when the total drag C_D was used as the objective function. In the present work, a converged (after nine optimization steps) optimal solution gave the same reduction in the total drag coefficient value.

Additional verification is obtained by comparison with the results obtained by Li and Padula [30]. The design point was $C_L = 0.733$, $M = 0.74$. In both optimizations, the initial profile was the RAE2822 airfoil. The drag reduction achieved by the present method with respect to the original RAE2822 airfoil is compared in Fig. 5 with that of [30], for a transonic range of free-stream Mach

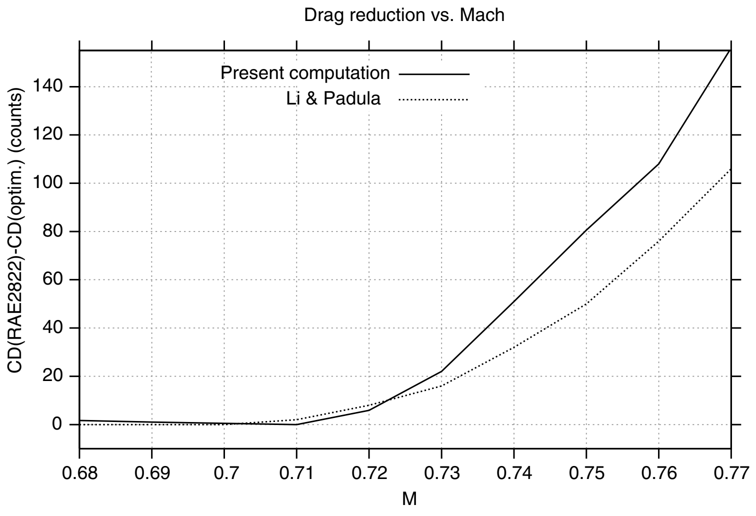

Fig. 5. Mach drag rise reduction due to optimization of RAE2822 airfoil at $C_L = 0.733$. Present results vs. optimization by Li and Padula [30].

numbers ($0.68 \leqslant M \leqslant 0.77$). Here the solid line corresponds to the present optimization, while the dotted line represents the computation of [30]. It can be concluded that both optimizations achieve a similar drag reduction in the range $0.680 \leqslant M \leqslant 0.725$, while for $M > 0.73$, the present optimization technique demonstrates a better performance.

In Fig. 6 the lift/drag polars for airfoils, optimized at different target C_L values are compared with the polar, corresponding to the initial RAE2822 profile. For a high transonic target

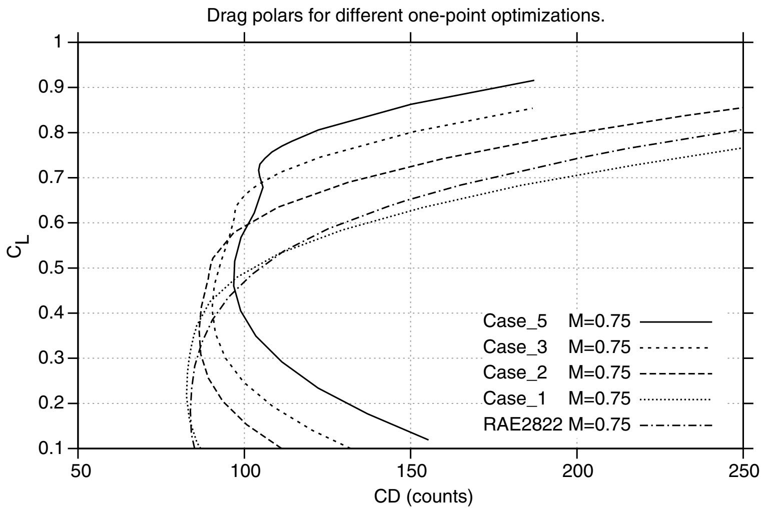

Fig. 6. Lift/drag polars for airfoils, optimized at different target C_L values. *Case_5*, *Case_3*, *Case_2* and *Case_1* correspond to design $C_L = 0.745$, 0.65, 0.5 and 0.3, respectively.

$C_L = 0.745$ (*Case_5*), a reduction of 95 drag counts was achieved (out of initial 201 counts). It is worthwhile to mention that, in this case, a considerable drag reduction is noticed not only pointwise but in the whole neighbourhood of the design point from $C_L = 0.6$ up to $C_L = 0.76$. At an intermediate design point $C_L = 0.5$ (*Case_2*), the drag reduction was equal to 17 counts (out of total 106.5 counts). Similar to $C_L = 0.745$ the range of significant drag reduction extends beyond the target lift coefficient up to $C_L = 0.55$.

For the original RAE2822 airfoil, the drag coefficient value at $C_L = 0.3$ was equal to 86.1 counts (compared to the minimum value C_{D_o} of 83.8 counts). Even at this point (*Case_1*), located close to C_{D_o}, a reduction of 2.6 counts was achieved, with the total drag of the optimized profile equal to 83.4 counts (less than C_{D_o} of RAE2822).

Shapes of the optimized profiles for *Case_1* and *Case_5* are compared with RAE2822 airfoil in Fig. 7. The analysis demonstrates that a significant drag reduction at $C_L = 0.745$ was obtained by means of (1) decrease of leading edge radius (2) diminution of curvature on both upper and lower surfaces up to 30% of the profile chord (3) decrease of thickness in the rear part of airfoil which leads to a more cusped trailing edge.

Fig. 8 shows comparison between lift/drag polars for the optimized airfoils at close target lift coefficients $C_L = 0.650$ (*Case_3*) and $C_L = 0.675$ (*Case_4*).

Pressure coefficients of the optimized profile (*Case_5*) are compared with those of RAE2822 airfoil at the corresponding design point $C_L = 0.745$ in Fig. 9. It is clearly seen that the optimization leads to the destruction of a strong shock, present in the original pressure distribution.

Figs. 10–12 show the behaviour of pressure coefficients distribution of optimized profiles (*Case_5*, *Case_2* and *Case_1*, respectively), in the vicinity of the corresponding design points. It is seen that in all the cases a favourable pressure distribution is retained in a vast neighbourhood of the design points.

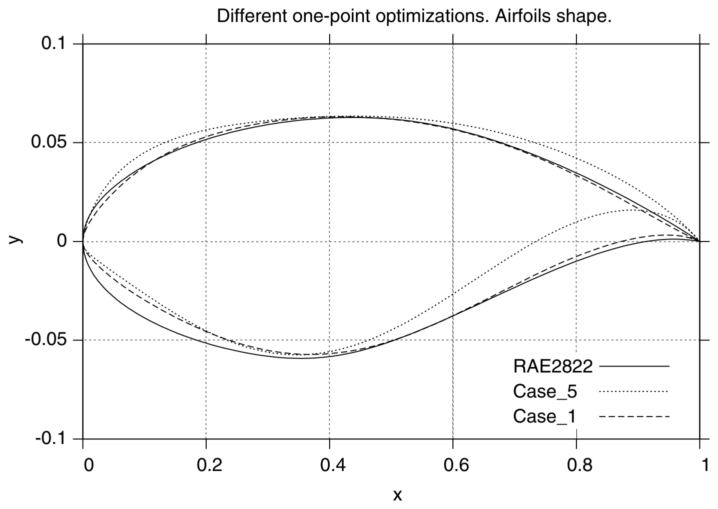

Fig. 7. Airfoils optimized at target $C_L = 0.745$ (*Case_5*) and $C_L = 0.3$ (*Case_1*). Comparison with initial RAE2822 profile.

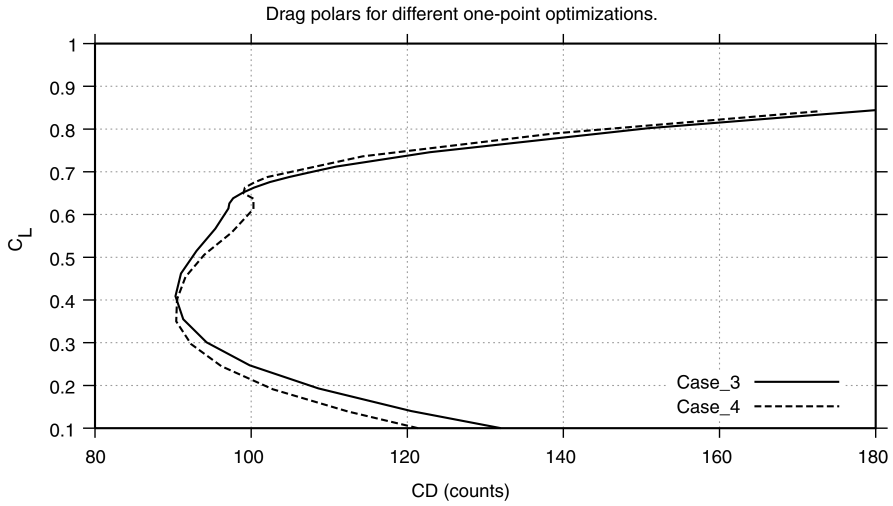

Fig. 8. Comparison between lift/drag polars for airfoils optimized at target $C_L = 0.650$ (*Case_3*) and $C_L = 0.675$ (*Case_4*).

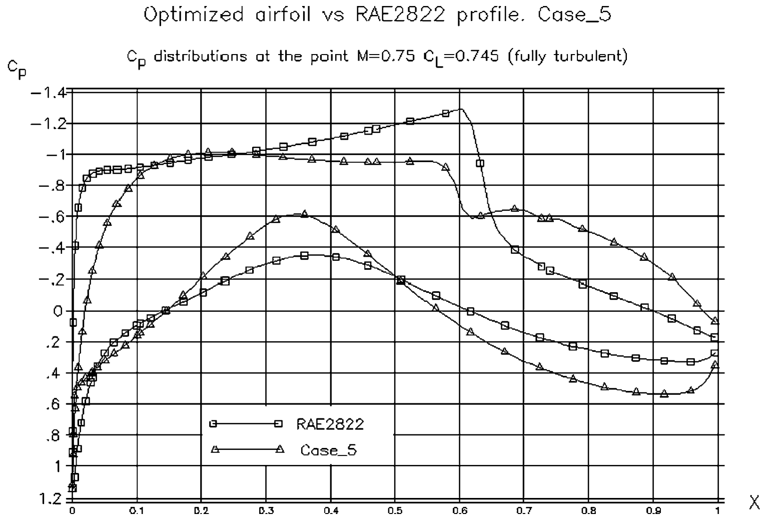

Fig. 9. Comparison of pressure coefficients C_p of the optimized profile (*Case_5*) and the original RAE2822 airfoil at $C_L = 0.745$.

Figs. 13 and 16 illustrate the influence of design Mach number on the form of drag polars and on the shape of optimized profiles, respectively. The corresponding drag polars for the original RAE2822 airfoil are presented in Fig. 14.

At all the Mach numbers, the optimization essentially improved the performance of the original profile at a design lift value of $C_L = 0.745$. The optimized airfoils also possess a relatively low total drag well beyond the design point, starting from $C_L = 0.5$. It may be assessed from Fig. 16 that at

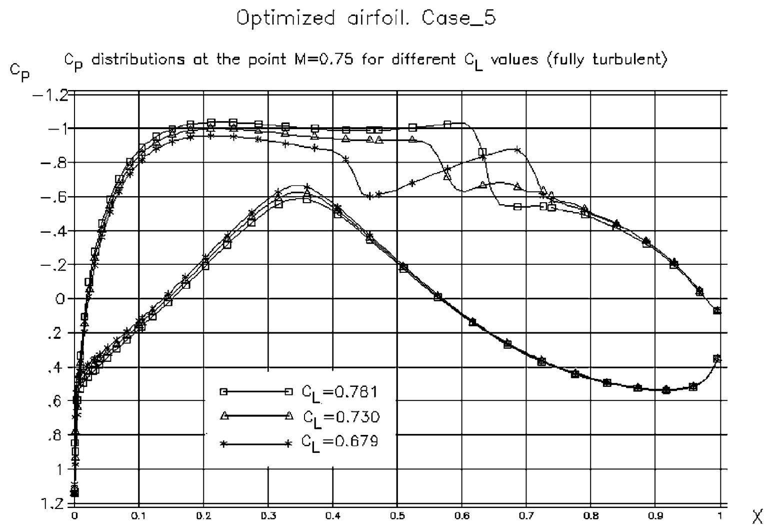

Fig. 10. *Case_5*. Behaviour of pressure coefficients C_p at different C_L values (close to the target $C_L = 0.745$). Lines with squares, triangles and asterisks correspond to $C_L = 0.781$, 0.730 and 0.679, respectively.

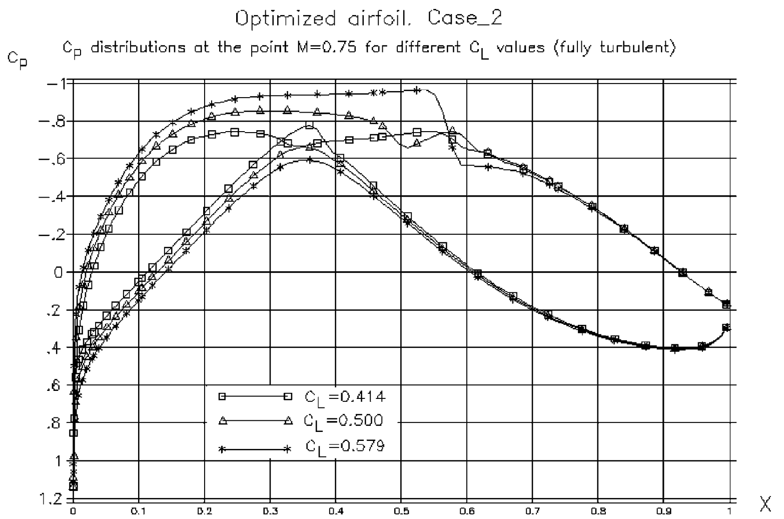

Fig. 11. *Case_2*. Behaviour of pressure coefficients C_p at different C_L values (close to the target $C_L = 0.500$). Lines with squares, triangles and asterisks correspond to $C_L = 0.414$, 0.500 and 0.579, respectively.

$M = 0.74$ and 0.75 the aerodynamic shapes are rather close, while the optimization at $M = 0.76$ leads to an essential amplification of design trends. This possibly indicates on the proximity of the design point at $M = 0.76$ to drag divergence at $C_L = 0.745$.

The off-design behaviour of a one-point optimized profile is presented in Fig. 15, where drag polars at $M = 0.76$ and 0.74 are compared with the polar at design Mach number

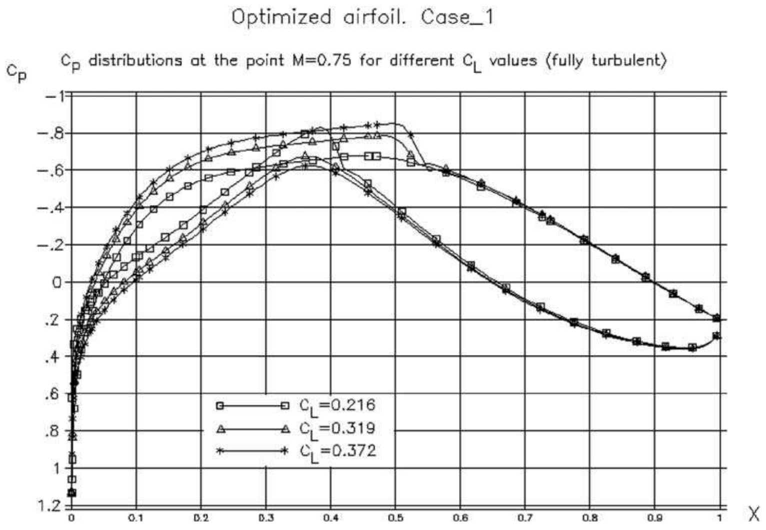

Fig. 12. *Case_1*. Behaviour of pressure coefficients C_p at different C_L values (close to the target $C_L = 0.300$). Lines with squares, triangles and asterisks correspond to $C_L = 0.216$, 0.319 and 0.372, respectively.

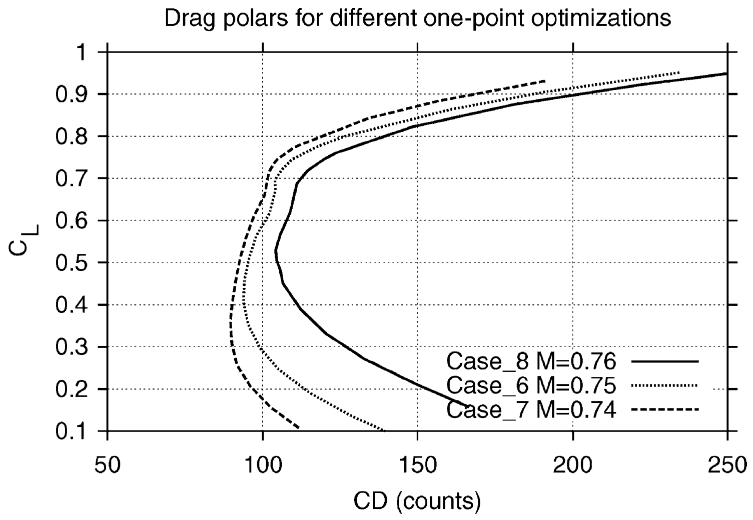

Fig. 13. Lift/drag polars for airfoils, optimized at different design Mach numbers.

$M = 0.75$ (*Case_6*). Based on comparisons between Figs. 13 and 15, it may be assessed that the one-point optimization provided a reasonable off-design performance with respect to Mach number.

The form of the trailing edge may be crucial for aerodynamic design of profiles as it highly influences the circulation. On the other hand, constraints must be imposed on value of the trailing edge angle θ_T, in order to make optimized profiles feasible.

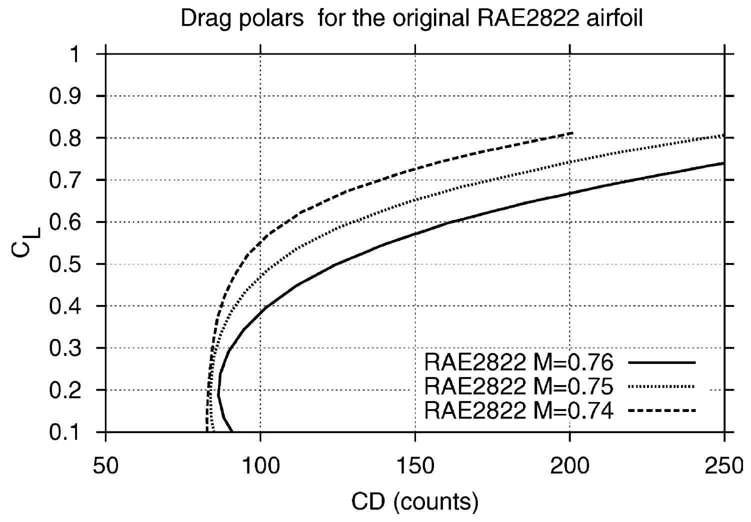

Fig. 14. Lift/drag polars for RAE2822 airfoil at different Mach numbers.

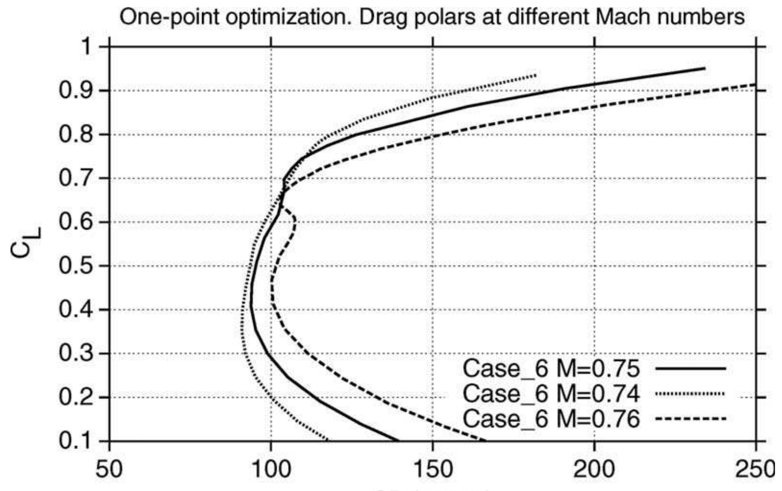

Fig. 15. One-point optimization at design $M = 0.75$, $C_L = 0.745$ (*Case_6*). Drag polars in the neighbourhood of design Mach number.

The influence of θ_T^* (minimum allowed value of θ_T) on the results of optimization is shown in Figs. 17 and 18. It is seen that the optimization with the value of $\theta_T^* = 3.4°$ leads to a slightly worse performance at the design lift point and at higher lift values (compared with $\theta_T^* = 0.0°$), while it highly favours the lift coefficients below $C_L = 0.45$. In terms of aerodynamic shape, a more constrained trailing edge optimization produces a more moderate curvature distribution on the lower surface, especially near the leading and trailing edges of the profile.

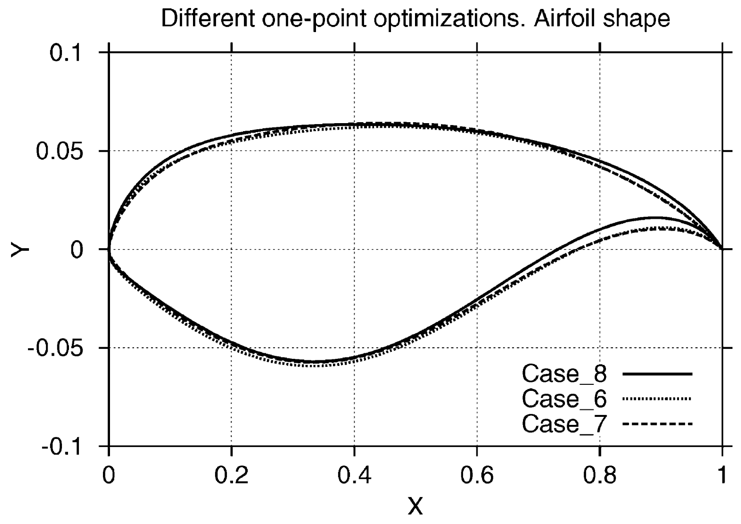

Fig. 16. Airfoil shapes, optimized for different design Mach numbers.

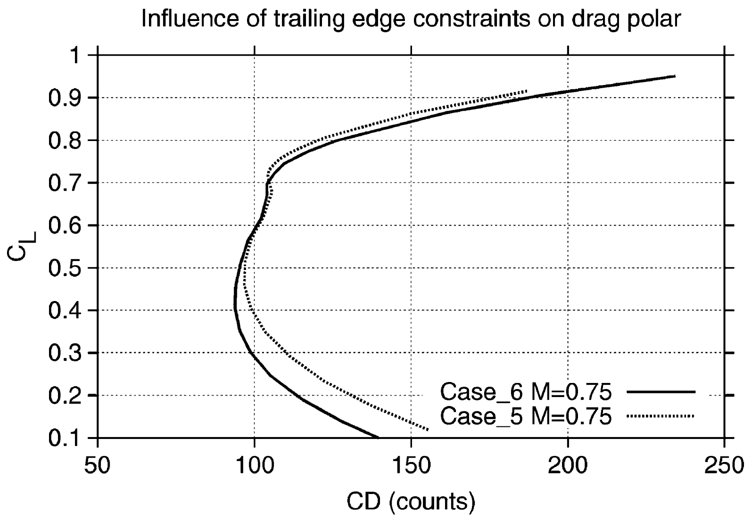

Fig. 17. Influence of design value of the trailing edge angle on drag polar. *Case_6*: $\theta^*_T = 3.4°$. *Case_5*: $\theta^*_T = 0.0°$.

In aerodynamic practice, the global rather than pointwise behaviour of airfoils, is essential. In this connection, it is important to compare, along with the drag polars, Mach drag rise behaviour of optimized airfoil with that of the original RAE2822 profile. In Fig. 19, drag divergence curves are shown, for the original RAE2822 airfoil alongside those for the optimized airfoil (*Case_6*). It is seen that, for the original airfoil, the drag divergence occurs immediately after $M = 0.71$. This means that, for RAE2822, the main design point ($C_L = 0.745$, $M = 0.75$) lies far inside the domain of drag divergence. It is seen that the one-point optimization results in essential extension of the low drag zone up to at least $M = 0.75$. It is important to note that even at lower Mach numbers, the optimized profile possesses a slightly lower drag than the original airfoil.

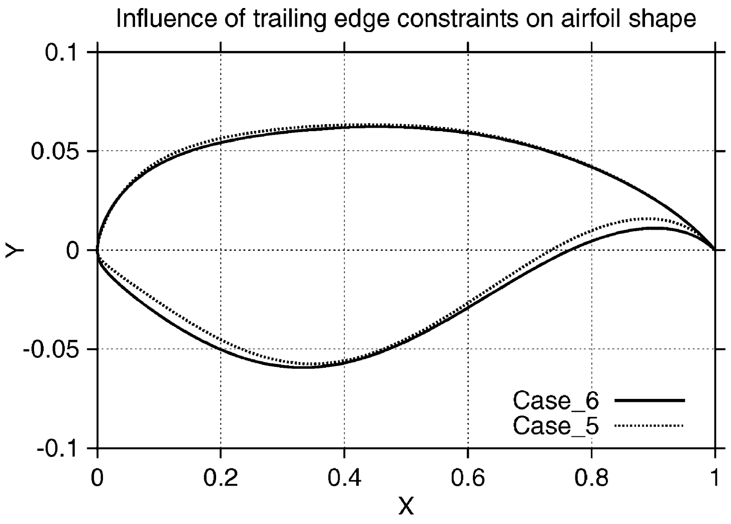

Fig. 18. Influence of design value of the trailing edge angle on aerodynamic shape. *Case_6*: $\theta_T^* = 3.4°$. *Case_5*: $\theta_T^* = 0.0°$.

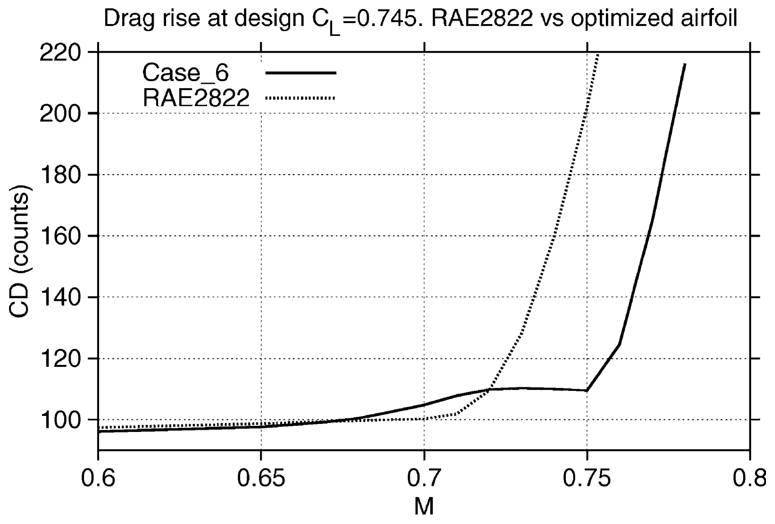

Fig. 19. Mach drag rise at design point $C_L = 0.745$. Original RAE2822 airfoil vs. optimized profile.

9. Conclusions

A new approach to the constrained design of aerodynamic shapes was suggested. The algorithm features a new strategy of efficient handling of non-linear 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 multilevel parallelization of the whole computational framework.

The method was applied to the optimization of transonic airfoils with a variety of non-linear constraints. The results demonstrated that the method combines high accuracy with computational efficiency. All the results presented in the paper are "first shot" automatic runs what indicates robustness of the proposed method. The above qualities of the method allowed to employ it in engineering environment.

References

[1] Lighthill MJ. A new method of twodimensional aerodynamic design. ARC 1945; Rend M 2112.
[2] Bauer F, Garabedian P, Korn D, Jameson A. Supercritical wing sections II. New York: Springer Verlag; 1975.
[3] Hicks RM, Henne PA. Wing design by numerical optimization. J Aircraft 1978;15(4):407–12.
[4] Optimum design methods for aerodynamics. AGARD 1994; R-803.
[5] Jameson A. Aerodynamic design via control theory. J Sci Comput 1988;3(2):233–60.
[6] Jameson A. Optimum aerodynamic design using control theory. CFD review. Wiley; 1995.
[7] Hajela P. Nongradient methods in multidisciplinary design optimization—status and potential. J Aircraft 1999;36(1):255–65.
[8] Epstein B, Jacobs A, Nachshon A. Aerodynamically accurate three-dimensional Navier–Stokes method. AIAA J 1997;35(6):1089–90.
[9] Epstein B, Averbuch A, Yavneh I. An accurate ENO driven multigrid method applied to 3D turbulent transonic flows. J Comput Phys 2001;168:316–38.
[10] Peigin S, Epstein B, Rubin T, Seror S. Parallel large scale high accuracy Navier–Stokes computations on distributed memory clusters. J Supercomput 2004;27:49–68.
[11] Epstein B, Rubin T, Seror S. Accurate multiblock Navier–Stokes solver for complex aerodynamic configurations. AIAA J 2003;41:582–94.
[12] Harten A, Engquist B, Osher S, Chakravarthy S. Uniformly high order accurate non-oscillatory schemes. 1. J Comput Phys 1987;71:231–303.
[13] Baldwin BS, Lomax H. Thin layer approximation and algebraic model for separated turbulent flows. AIAA Paper 78-257, 1978.
[14] Shu CW, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. J Comput Phys 1989;83:32–78.
[15] Koren B. Multigrid and defect correction for the steady Navier–Stokes equations. J Comput Phys 1990;87:1–25.
[16] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley; 1989.
[17] Back T. Evolutionary algorithms in theory and practice. New York: Oxford University Press; 1995.
[18] Fogel DB. Evolutionary computation. Toward a new philosophy of machine intelligence. IEEE Press; 1995.
[19] Michalewicz Z. Genetic algorithms + data structures = evolution programs. New York: Springer Verlag; 1996.
[20] Hoffmeister F, Back T. Genetic algorithms and evolution strategies: similarities and differences. In: Proceedings of the 5th Conference on Parallel Problems Solving from Nature. Springer Verlag; 1991. p. 455–69.
[21] Sefrioui M, Periaux J, Ganascia JG. Fast convergence thanks to diversity. In: Proceedings of the 5th Annual Conference on Evolutionary Programming. MIT Press; 1996. p. 313–21.
[22] Vicini A, Quagliarella D. Airfoil and wing design through hybrid optimization strategies. AIAA J 1997;35(5).
[23] Dowell EH. Eigen-mode analysis in unsteady aerodynamics: reduced-order models. In: Proceedings of the 36th Structures, Structural Dynamics and Materials Conference. Reston, VA: AIAA Press; 1995. p. 2545–57.
[24] Raveh DE. Reduced-order models for nonlinear unsteady aerodynamics. AIAA J 2001;39(8):1417–29.
[25] Peigin S, Desideri JA. Parallel implementation of genetic algorithms to the solution for the space vehicle reentry trajectory problem. In: Parallel computational fluid dynamics Trends and applications. Amsterdam: Elsevier Science B.V.; 2001. p. 357–64.
[26] Marconi F, Siclary N, Carpenter G, Chow R. Comparison of TLNS3D computations with test data for a transport wing/simple body configuration. AIAA Paper 94-2237, 1994.

[27] Carr MP, Palister KC. Pressure distributions measured on research wing M100 mounted on an axisymmetric body. AGARD 1994; AR-138.
[28] Barak A, Guday S, Wheeler R. The MOSIX distributed operating system, load balancing for UNIX. In: Lecture notes in computer science, 672. Springer Verlag; 1993.
[29] Nadarajah SK, Jameson A. Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization. AIAA Paper 2001-2530, 2001.
[30] Li W, Padula S. Performance trades study for robust airfoil shape optimization. AIAA Paper 2003-3790, 2003.