AIAA JOURNAL
Vol. 42, No. 8, August 2004

Robust Hybrid Approach to Multiobjective Constrained Optimization in Aerodynamics

Boris Epstein*
Academic College of Tel-Aviv Yaffo, 64044 Tel-Aviv, Israel
and
Sergey Peigin†
Israel Aircraft Industries, 70100 Ben-Gurion Airport, Israel

A new approach to the multiobjective constrained design of aerodynamic shapes is suggested. The approach employs genetic algorithms (GA) as an optimization tool in combination with a reduced-order-models method based on linked local databases obtained by full Navier–Stokes computations. The method was applied to the problem of multiobjective transonic profile optimization with nonlinear constraints. The results demonstrated that the method retains high robustness of conventional GAs while keeping computational-fluid-dynamics computational volume to an acceptable level as a result of a limited use of full Navier–Stokes computations. 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.

Nomenclature

C_D = total drag coeeficient
C_L = total lift coeeficient
$\boldsymbol{F}_f$ = feasible search space
M = Mach number
N = order of Bezier spline
Q = modified objective function
$\boldsymbol{P}_i^k$ = control points of Bezier spline
Re = Reynolds number
S = search string
Δ_i = variation of the Bezier coefficients of the initial profile
Δ_α = variation of the initial angle of attack
δ_i = deviation from the Bezier coefficients of the initial profile
δ_α = deviation from the initial angle of attack

Subscripts and Supercripts

c = coarse grid
f = fine grid
l, u = lower and upper airfoil surfaces
n = optimization step
$*$ = prescribed parameters of the problem

I. Introduction

THERE is a worldwide demand for efficient and robust software that implements multiobjective aerodynamic optimization. This request is explained by pivotal role of advanced aerodynamic design in the process of reducing costs and thus improving competetiveness of aircraft manufacturing.

A conventional ("direct") approach to the constrained design of aerodynamic shapes is oriented to the trial-and-error method. That is, an initial aerodynamic geometry is modified according to the experience of designers and to the aerodynamic data supplied by a previous wind-tunnel experiment and computational-fluid-dynamics (CFD) analysis. A new modification is tested (usually by means of time-consuming CFD methods), and the results are analyzed in order to supply considerations for a new iteration of the design process.

Obviously (as confirmed by the cumulative experience in different aircraft industries) this process does not provide a desirable optimal configuration; at best, it provides a certain improvement of the initial shape. Then (after a number of modification cycles) the configuration is tested in the wind tunnel. It often happens that, because of the limitations of the optimization policy, as well as the insufficient accuracy of CFD analysis, the whole optimization loop is repeated at least twice. The time needed for one optimization loop is measured by months, and the whole process can take even years.

The preceding can occur even if the design treats only a limited part of the whole configuration, for example, a nacelle/fuselage junction or a wing/body fairing. Thus the introduction of an automatic robust aerodynamic shape optimizer based on an accurate flow analysis can dramatically reduce the time devoted to testing and overall cost of design by reducing the number of optimization cycles to a single loop. At the same time, the problem of multiobjective constrained optimization still remains stiff and open (especially in 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 that 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 can be characterized by a mix of continuous, discrete, and integer design variables, and the resulting design space can be nonconvex or even disjointed. For all of these reasons, optimizations methods that do not rely on the computation of gradients, in particular evolutionary programming and GAs, received a considerable growth of interest.[7]

The main objective of the present research is to create a powerful tool of aerodynamic design that will allow to reduce overall cost of design, development, and analysis of aircraft.

With the preceding goals in view, an efficient and robust optimization algorithm was created. The important features of the

Received 13 March 2003; presented as Paper 2003-4092 at the AIAA 16th Computational Fluid Dynamics Conference, Orlando, FL, 23–26 June 2003; revision received 28 October 2003; accepted for publication 9 January 2004.

*Professor, Computer Science Department. Member AIAA.
†Senior Research Scientist, Engineering Division. Member AIAA.

method include 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 reduced-order-model (ROM) method, and a multilevel parallelization of the whole computational framework that efficiently makes use of computational power supplied by massively parallel processors. The proposed tool incorporates 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.

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

II. Statement of the Problem

In this section a transonic flow multipoint optimization problem is considered. In the case of the single point optimization problem, 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, shape "freeze" of certain portions of airfoil [such as lower or upper surfaces, trailing (leading)-edge region, etc.]:

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

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

As a gas-dynamic model for calculating C_D and C_L values, the full Navier–Stokes equations are used. Numerical solution of the full Navier–Stokes equations was based on the code NES,[8–10] which employs the essentially nonoscillatory (ENO) concept[11] with a flux interpolation technique,[12] which allows accurate estimation of sensitive aerodynamic characteristic such as lift, pressure drag, friction drag, and pitching moment.

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. Nonlinear stability is maintained via approximation of inviscid fluxes on a variable template according to local characteristics and smoothness of the fluxes; viscous fluxes are approximated in a straightforward way. The resulting multigrid method[10] retained the high accuracy of ENO approach with a comparatively small number of multigrid cycles needed to reduce the error below the level of truncation errors.

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.

III. Genetic Algorithms for Constrained Optimization Problem

Genetic algorithms (GA) became highly popular as optimization methods in the past two decades.[13,14] The basic idea behind genetic algorithms is to mathematically imitate the evolution process of nature. They are semistochastic 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. The main point is that the probability of survival of new individuals depends on their fitness: the best are kept with a high probability, and the worst are rapidly discarded.

As a basic algorithm, a variant of the floating-point GA is employed. 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 nonuniform mutation defined by Michalewicz.[14] To resolve one of the main problems that arises in GAs, a premature convergence, we used distance-dependent mutation.[15] To improve the convergence of the algorithm, we also used the elitism principle.

A. Treatment of Nonlinear Constraints

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.

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 $\boldsymbol{F}_f$, defined by a nonlinear constraint $g(x) \leq 0$. 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 the conventional search strategy by employing search paths that pass through both feasible and infeasible points (instead of the traditional approach where only feasible points may be included in a path). Because in our case a topology of feasible and infeasible subdomains 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 subdomains 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 nonsimply connected optimization space).

2) To implement the new strategy, suggested extending 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 can be easily implemented by means of GAs because of 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.

In fact, this allows to find an optimal point located exactly on the constraints boundary. Contrary to the conventional penalty function method, the suggested approach does not modify the value of objective function in any feasible point. In this regard, the method implements a zero-penalty approach.

Based on the results of the preceding 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 subdomain. It can be done using the preliminary information on the behavior 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: $f(x) \approx A(x \in \boldsymbol{F}_f)$. Based on this estimation, we define the following first-step approximation for the modified objective function $f^*(x)$:

$$f^*(x) = \begin{cases} f(x) & \text{if} \quad g(x) \leq 0 \\ B & \text{if} \quad g(x) > 0 \end{cases} \tag{2}$$

where $B \gg A$.

The next step in the approach is to run the just-described GA for the solution of the optimization problem (in fact a nonconstrained

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 $\boldsymbol{F}$ is finally defined as follows:

$$f^{**}(x) = \begin{cases} f(x) & \text{if} \quad g(x) \le 0 \\ \alpha_1 C + \alpha_2 g(x) B & \text{if} \quad g(x) > 0 \end{cases} \tag{3}$$

where the constants α_1 (which is close to 1) and $\alpha_2 > 0$ are problem-dependent real numbers.

B. Implementation of the 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 employed. Note, that a number of alternative geometric representations are used in engineering practice, and Bezier splines were chosen for their clear geometrical interpretation.

A Bezier curve of order N is defined by the Bernstein polynomials $B_{N,i}$ (C_N^i, binomial coefficients)

$$\boldsymbol{G}^k(t) = \sum_{i=0}^{N} B_{N,i} \boldsymbol{P}_i^k, \qquad B_{N,i} = C_N^i t^i (1-t)^{N-i}$$

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

where t denotes the parameter of the curve taking values in $[0, 1]$ and superscript $k = u, l$.

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

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:

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

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 preceding approach to the handling of constraints, the modified objective function Q for the solution of drag minimization problem was defined as follows:

$$Q = \begin{cases} 0.1 + [(t/c)^* - (t/c)] & \text{if} \quad (t/c) < (t/c)^* \\ 0.2 + \left[R_L^* - R_L\right] & \text{if} \quad R_L < R_L^* \\ 0.3 + \left[\theta_T^* - \theta_T\right] & \text{if} \quad \theta_T < \theta_T^* \\ 0.5 & \text{if} \quad y^u(t) < y^l(t) \\ C_D & \text{otherwise} \end{cases} \tag{5}$$

The choice of constants in Eq. (5) was based on the following simple considerations. It is necessary to guarantee that, for any feasible point, the value of the objective function will be low in comparison with that of any infeasible point. It is ensured that, for any infeasible point, the value of the objective function will be at least several times greater than the upper bound of C_D (about 0.0500 for the considered class of problems). 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.

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.

IV. Approximation of Objective Function by ROM-Local-Approximation Method

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 $\boldsymbol{M} = 100$ 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, which is practically unacceptable.

One of the popular ways of overcoming this difficulty and reducing the volume of CFD computations is to use a reduced-order models approach in the broad sense of the word. In this work we use a 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 database. The database is obtained by solving the full Navier–Stokes equations in a discrete neighborhood of a basic point positioned in the search space. 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 next in the case of single-point drag minimization.

Denote $x = (a_1^n, a_2^n, \ldots, a_{2N-5}^n, \alpha^n)$ point in the search space, where 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 prescibed 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 (subscript n is omitted and $F = C_L, C_D$):

$$F(a_1 + \delta_1, \ldots, a_{2N-5} + \delta_{2N-5}, \alpha + \delta_\alpha) = F^\circ + \sum_{j=1}^{2N-5} \Delta F_j + \Delta F_\alpha \tag{6}$$

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

$$\Delta F_j^+ = \frac{\delta_j}{\Delta_j}\left(F_j^+ - F^\circ\right), \qquad \Delta F_j^- = -\frac{\delta_j}{\Delta_j}\left(F_j^- - F^\circ\right)$$

$$\Delta F_\alpha = \frac{\delta_\alpha}{\Delta_\alpha}\left(F_\alpha - F^\circ\right)$$

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

In the approximation (6) the following notation is used:

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

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

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

Here the local database values F°, F_j^+, F_j^- and F_α are obtained by solving the full Navier–Stokes equations at the corresponding neighboring points of the basic point in the search space. These neighboring points are determined by positive variations $\{\Delta_j\}$ corresponding to the Bezier coefficients $\{a_j\}$ and by the variation Δ_α of the angle of attack α.

The choice of Δ_j and Δ_α is determined by the following principle. On one hand their value cannot be too low, in order to suppress the influence of numerical noise in the calculation of ΔF_j, whereas, on the other hand, it cannot be too high, in order to ensure a reasonable order of approximation.

In fact, relation (6) represents a mixed linear-quadratic approximation in the neighborhood of the basic point x°, which employs the local database. One dimensionally, we use either a one-sided linear approximation (in the case of monotonic behavior of the functional F) or a quadratic approximation (otherwise).

Such a reduced second-order approximation A_r^2 neglects mixed second derivatives compared to the full second-order approximation of objective function A_f^2 in the vicinity of the basic point x°. However, the reduced approximation retains a number of properties of the full approximation, which are crucial with optimization in view.

In particular, a simple algebraic analysis shows that if a full quadratic form A_f^2 possesses a local extremum, then the reduced form A_r^2 also possesses the corresponding extremum at some point x_e. Moreover, it is ensured that the value of A_f^2 at the point x_e is nearer to the extremum of the full form than the value F° of the objective function at the basic point. Note that as the extremum is stronger, the improvement is higher.

Values of C_L and C_D calculated via relations (6) 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 (6) possesses acceptable accuracy at least in the following range of deviations δ_j and δ_α:

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

Using Eq. (6), we can obtain the following formula for cost function C_D at prescribed lift coefficient C_L^* [for current point at the search space $(a_1 + \delta_1, \ldots, a_{2N-5} + \delta_{2N-5})$], which is based on local database $C_D^\circ, C_{D_\alpha}, C_{D_j}^+, C_{D_j}^-, C_L^\circ, C_{L_\alpha}, C_{L_j}^+, C_{L_j}^-$:

$$C_D = C_D^\circ + \sum_{j=1}^{2N-5} \Delta C_{D_j} + \frac{\delta_\alpha}{\Delta_\alpha}\left(C_{D_\alpha} - C_D^\circ\right)$$

$$\delta_\alpha = \frac{\Delta_\alpha}{C_{L_\alpha} - C_L^\circ}\left[C_L^* - C_L^\circ - \sum_{j=1}^{2N-5} \Delta C_{L_j}\right] \tag{7}$$

V. Description of the Optimization Algorithm

The optimization method involved the following algorithmic steps:

1) Bezier coefficients of the initial profile (the initial basic point in the search space) are determined.

2) For given values of variations Δ_j and Δ_α, the CFD local database for C_L and C_D is obtained by solving the full Navier–Stokes equations at the neighboring points of the basic point in the search space, corresponding to these variations.

3) The local CFD database is included in the global CFD database.

4) Using local approximation of cost function (7), a genetic algorithm is applied for various search domains D_k (corresponding to different search scales), and optimal points O_k for each domain are obtained ($k = 1, \ldots, N_D$; N_D is the number of the search domains).

5) A full Navier–Stokes solver is applied to each optimal point O_k, and the corresponding data are added to the global CFD database.

6) A new basic point is determined as the best point taken from the global CFD database.

7) If the prescribed convergence accuracy is achieved, then stop; otherwise, the optimization process is repeated from step 2.

The general sketch of the optimization algorithm can be presented by the following pseudo-code:

```
opt_step = 0
Determine_Initial_Basic_Point   /starting basic point—initial profile/
while not converged do
Calc_Local_Data_Base            /CFD computations in a discrete neighborhood of the basic point/
Search_Optim_Candidates         /Hybrid GA/ROM-LAM search of optimal points for various search domains/
Verification_Optim_Cand         /CFD computations for optimal points/
Choose_New_Basic_Point          /Choose a new basic point—the best one among all of the points in the global CFD database/
opt_step := opt_step + 1
enddo
```

Note that steps *Search_Optim_Candidates* and *Verification_Optim_Cand* implement, in fact, a multidomain prediction-verification principle that 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.

In each iteration step, the search domains represent a set of embedded multidimensional cubes, which have a maximum linear size of the same order of magnitude as the maximum Δ_j. Search domains related to the successive optimization steps overlap.

VI. Computational Efficiency of the Algorithm

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.

A. Computational Grids

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

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

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:

$$Q_c(x_1) \ge Q_c(x_2) \tag{8}$$

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

$$Q_f(x_1) \geq Q_f(x_2) \tag{9}$$

B. Multilevel 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; and level 3, parallelization of the optimization framework.

The first level parallelization approach[9] 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.

Finally, a large body of computational data demonstrated that the preceding approach for parallel implementation of the multiblock full Navier–Stokes solver enables one to achieve a high level of parallel efficiency while retaining high accuracy of calculations and thus to reduce significantly the execution time for large-scale CFD computations.[9]

The second level of parallelization is needed in order to organize a parallel CFD scanning of the optimization search space. It is applied when executing the steps *Calc_Local_Data_Base* and *Verification_Optim_Cand* in the pseudocode of the optimization algorithm. On this level of parallelization, all of 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 scanned 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 databases. 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%.

The third level of parallelization was used at the step *Search_Optim_Candidates* in the pseudocode of the optimization algorithm (GA search of optimal points O_k corresponding to different search domains D_k). The idea is to improve the convergence rate and accuracy of the GA evolution process by exchanging information related to the best individuals of subpopulations.[16]

On this level of parallelization, all of the processors were divided into N_D groups with N_P processors in each group. (N_D is equal to the number of search domains, and N_P is equal to the number of random initial populations in the GA search.) The goal of each group is to find the optimal point for a corresponding search domain.

The three-level parallelization approach allowed us to sustain a high level of parallel efficiency on massively parallel machines and by this way to dramatically improve the computational efficiency of the suggested optimization algorithm.

VII. Analysis of Results

We present here applications of the preceding optimization algorithm to one-point and multipoint airfoil design. The design problem consists of the minimization of total drag starting from the RAE2822 airfoil at a variety of transonic design points.

The verification of the CFD solver NES was performed by running a NASA test case of the ARA M100 wing-body configuration over the wide range of subsonic and transonic Mach numbers. The results were compared with both experiment[17] and available Navier–Stokes solutions.[18]

The set of computational grids contained three multigrid levels. Each level included 24 blocks. The total number of points in the fine level was close to one million.

Good agreement with experimental data[17] was achieved in the whole range of flight conditions. Figure 1 presents the computed surface-pressure distribution compared with experiment at high transonic Mach number $M = 0.8027$. It is important to emphasize that the NES computation not only favorably compares with experiment but also indicates a good grid convergence. The last property is essential because the present optimization algorithm is driven by NES.

As it was demonstrated in Ref. 8, the NES solver supplies accurate asymptotically converged lift and drag coefficients values for transonic two-dimensional 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 Sec. VI). The ability of the code to perform qualitatively reliable drag computations for RAE2822 airfoil is illustrated in Fig. 2, where Mach drag rise at fixed angle of attack is presented for different

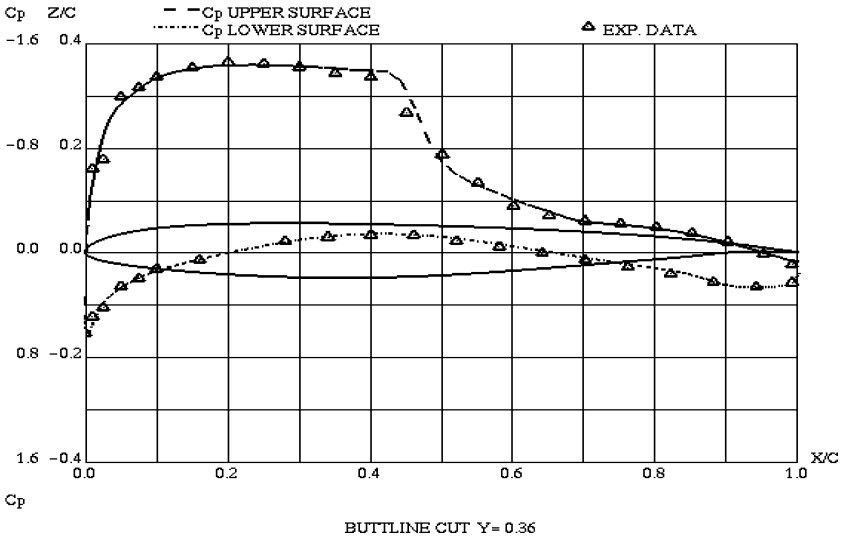

Fig. 1 ARA M100 wing-body configuration. Chordwise pressure distribution for wing span station $Y = 0.36$, $\alpha = 2.873$ deg, $M = 0.8027$, $Re = 13.1 \times 10^6$: – – –, · · · ·, present computation and △, experimental data.[17]

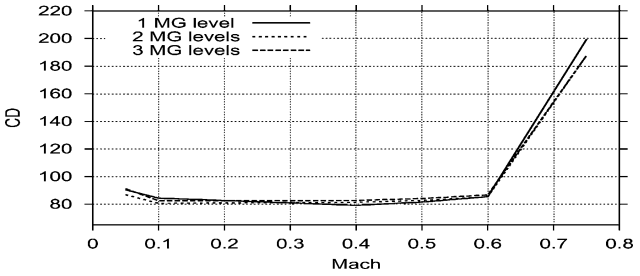

Fig. 2 RAE2822 airfoil. Mach drag rise at $\alpha = 2.0$ deg, computed on 1, 2, and 3 MG levels.

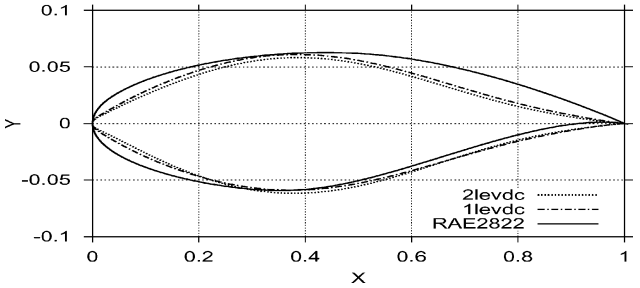

Fig. 3 Airfoil shapes optimized on the coarse (1levdc) and medium (2levdc) grids vs original RAE2822 profile.

levels of grid resolution. It also appeared that the four times coarser in each direction (81×25) grids satisfy the invariancy conditions (8) and (9). This allowed us to use meshes with such a resolution for optimization purposes.

This was verified by the following test case: to find an optimal 12% thickness two-dimensional airfoil at the design point $C_L = 0.0$, $M = 0.6$ in the fully turbulent flow regime.

It is aerodynamically expectable that the resulting optimal shape (which corresponds to zero lift) should be symmetric, and the verification of this property is a good test to check the consistency of results. In this connection, in order to make the problem more challenging, a highly asymmetric supercritical RAE2822 airfoil was chosen as an initial point of the iterative optimization algorithm.

The multigrid set of computational grids contained three levels. The fine, medium, and coarse meshes comprised 320×96, 160×48, and 80×24 points in the streamwise and normal to the surface directions, respectively. The optimization problem was solved twice, based on CFD computations employing the coarse and medium grids, respectively.

The results are presented in Fig. 3, where the optimized aerodynamic shapes vs original RAE2822 airfoil are shown. The resulting optimal profiles are fairly symmetrical in the both considered cases, especially taking into account that the computational meshes originating from the initial asymmetrical airfoil are far from being symmetric.

The optimal shapes corresponding to the coarse and medium CFD computations are rather close one to another with values of the total drag coefficient C_D equal to 74.1 and 73.5 drag counts, respectively.

The fast transformation of grids, mentioned in Sec. VI, was implemented in the following way. Assume that $\Delta \boldsymbol{r}_{i,0}$ is a change in the airfoil geometry at the grid point with indices $(i, 0)$ in the streamwise and normal to surface directions, respectively. Then the coordinates $\boldsymbol{r}_{i,j}^{new}$ of the new grid are obtained by propagation of the shift $\Delta \boldsymbol{r}_{i,0}$ along the grid line $i = const$:

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

Multilevel parallelization strategy based on the Parallel Virtual Machine software package was implemented on a cluster of MIMD multiprocessors consisting of 72 HP NetServer LP1000R nodes. Each node has two processors, 2-GB RAM memory, 512-kB level 2 cache memory, and full duplex 100-Mbps ETHERNET interface. Totally this cluster contained 144 processors with 144-GB RAM and 36-MB level 2 cache memory.

To verify the method, the design point $C_L = 0.65$, $M = 0.75$ was employed. The case served for verification studies in a number of publications, most recently in Ref. 19.

At this point the initial solution (corresponding to the RAE2822 airfoil) gives the total of 149 drag counts (94 counts as a result of pressure drag and 55 counts as a result of viscous forces), while the corresponding drag values in Ref. 19 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 Ref. 19 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.

To further verify the optimization method, the following multipoint optimization of RAE2822 airfoil was employed. The main design point was $M = 0.734$, $C_L = 0.8$, $Re = 6.5 \times 10^6$, and the secondary design points were $M = 0.754$, $C_L = 0.74$, $Re = 6.2 \times 10^6$ and $M = 0.680$, $C_L = 0.56$, $Re = 5.7 \times 10^6$. The target was to minimize a weighted combination of total drag values at these points with the following weight coefficients: $w_1 = 0.5$, $w_2 = 0.25$, and $w_3 = 0.25$. The constraints were imposed on airfoil thickness and leading-edge radius, which cannot decrease. The case served for verification purposes within the AEROSHAPE project.[20]

At the main design point the drag reduction by the current method was equal to 59 drag counts compared to 40 counts reported in Ref. 20. At the second design point ($M = 0.754$) a reduction of 103 counts was achieved vs 34 counts in Ref. 20. At the lowest Mach design point ($M = 0.680$) both methods produced a slight increase in drag (about two counts) in comparison with the original RAE2822 airfoil.

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 12 test cases was studied. Varying design conditions and constraints are summarized in Table 1.

In all of 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 ($R_L \geq R_L^* = 0.0029$) and on the profile shape (in order to avoid "fishtails"). The corresponding optimal profiles are designated by *Case*_1 to *Case*_12.

From the optimization viewpoint the test cases can be divided into one-point (*Case*_1 to *Case*_8) and multipoint (*Case*_9 to *Case*_12) optimization problems. Aerodynamically, the cases covered different parts of aerodynamic polars in the neighborhood 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

Table 1 Optimization conditions and constraints for different test cases

Case no.	Design C_L	Design M	Weight	θ_T^*, deg
1	0.300	0.75	1.0	0.0
2	0.500	0.75	1.0	0.0
3	0.650	0.75	1.0	0.0
4	0.675	0.75	1.0	0.0
5	0.745	0.75	1.0	0.0
6	0.745	0.75	1.0	3.4
7	0.745	0.74	1.0	3.4
8	0.745	0.76	1.0	3.4
9	0.745	0.75	0.5	3.4
	0.745	0.74	0.25	——
	0.745	0.76	0.25	——
10	0.745	0.75	0.66	3.4
	0.720	0.75	0.17	——
	0.770	0.75	0.17	——
11	0.745	0.75	0.5	3.4
	0.500	0.75	0.25	——
	0.300	0.75	0.25	——
12	0.745	0.75	0.8	3.4
	0.500	0.75	0.1	——
	0.300	0.75	0.1	——

profile. *Case*_3 to *Case*_10 correspond to flight conditions with a strong shock-boundary-layer interaction. *Case*_9 deals with multipoint optimization combining different Mach numbers, whereas *Case*_11 and *Case*_12 represent the multipoint optimization, which combines high and low lift conditions.

A. One-Point Optimization

Extensive numerical experiment showed that the suggested optimization algorithm is stable, robust, and possesses a good computational efficiency. Typically, 8–10 optimization steps were required to reach a reasonable convergence. This is illustrated in Fig. 4, where convergence history for *Case*_2, *Case*_3, and *Case*_5 is shown. On the preceding multiprocessor cluster, an average of six hours was needed to get a converged solution of one-point optimization problem.

In Fig. 5 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 $C_L = 0.745$ (*Case*_5), a reduction of 95 drag counts was achieved (out of initial 201 counts). A considerable drag reduction is noticed not only pointwise but in the whole neighborhood 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 $C_{D_\circ}$ of 83.8 counts). Even at this point (*Case*_1), located close to $C_{D_\circ}$, a reduction of 2.6 counts was achieved, with the total drag of the optimized profile equal to 83.4 counts (less than $C_{D_\circ}$ of RAE2822).

Shapes of the optimized profiles for *Case*_1 and *Case*_5 are compared with RAE2822 airfoil in Fig. 6. The analysis demonstrates that a significant drag reduction at $C_L = 0.745$ was obtained by means of decrease of leading-edge radius, diminuation of curvature on both upper and lower surfaces up to 30% of the profile chord, and decrease of thickness in the rear part of airfoil which leads to a more cusped trailing edge.

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. 7. The optimization leads to the destruction of a strong shock, present in the original pressure distribution. A detailed analysis of results showed that in all of the cases a favorable pressure distribution is retained in a vast neighborhood of the design points.

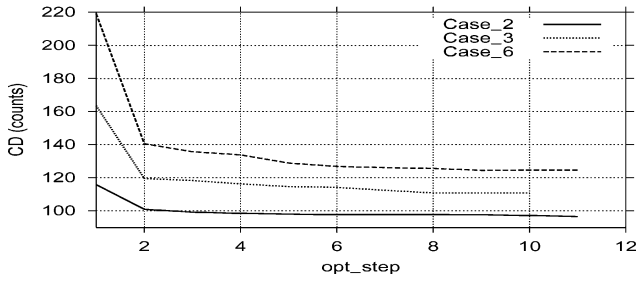

Fig. 4 One-point optimizations. Convergence history for *Case*_2, *Case*_3, and *Case*_5.

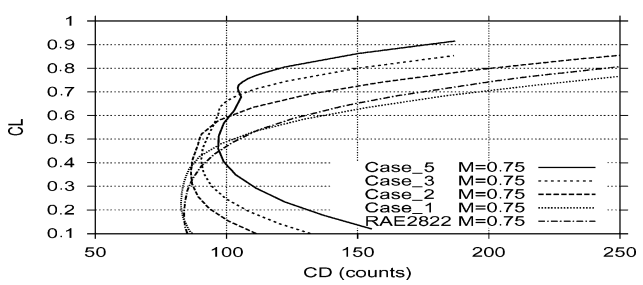

Fig. 5 Lift/drag polars for airfoils, optimized at different target C_L values. *Case*_5, *Case*_3, *Case*_2, and *Case*_1 correspond to design C_L = 0.745, 0.65, 0.5, and 0.3, respectively.

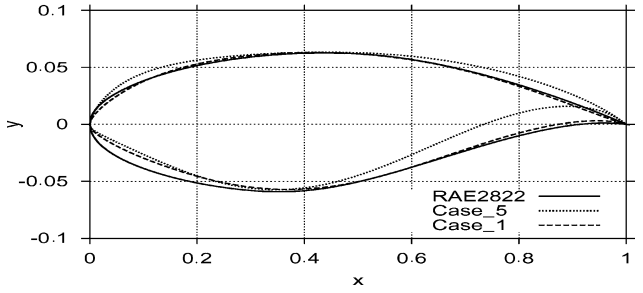

Fig. 6 Airfoils optimized at target C_L = 0.745 (*Case*_5) and C_L = 0.3 (*Case*_1). Comparison with initial RAE2822 profile.

Figure 8 illustrates the influence of design Mach number on the form of drag polars.

At all of 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 $C_L = 0.5$. Note that at $M = 0.74$ and 0.75 the aerodynamic shapes are rather close, whereas 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 $C_L = 0.745$.

Drag polars at $M = 0.76$ and 0.74 are compared with the polar at design Mach number $M = 0.75$ (*Case*_6) in Fig. 9. Comparisons between Figs. 8 and 9 show that the one-point optimization provided a reasonable off-design performance with respect to Mach number.

The form of the trailing edge might 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.

The influence of θ_T^* (minimum allowed value of θ_T) on the results of optimization is shown in Figs. 10 and 11. It is seen that the optimization with the value of $\theta_T^* = 3.4$ deg leads to a slightly worse performance at the design lift point and at higher lift values (compared with $\theta_T^* = 0.0$ deg), whereas it highly favors 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.

B. Multipoint Optimization

The pointwise design airfoil must be analyzed over range of Mach numbers and lift coefficients to ensure the adequacy of the off-design performance. A more extended Mach and C_L sweep might be necessary to verify that the airfoil is performing satisfactorily at other than the design Mach numbers and lift coeffitients. To reach this goal, the multipoint optimization is needed where the objective function is a weighted combination of one-point cost functions. The results of multipoint optimization include a sweep over design Mach numbers at fixed design C_L (*Case*_9) and sweeps over design lift coefficients at fixed design Mach number (*Case*_10, *Case*_11, and *Case*_12).

In *Case*_9, which represents multiple Mach optimization, the main design point was $M = 0.75$, $C_L = 0.745$, and secondary design points were $M = 0.76$, $C_L = 0.745$ and $M = 0.74$, $C_L = 0.745$. In Fig. 12, drag polars are shown for a sequence of Mach numbers in the neighborhood of $M = 0.75$. Based on comparison with the results of one-point optimization (see Figs. 8 and 9), it can be assessed that the multipoint optimization almost has no impact on drag at the main design point, whereas it slightly improves the performance for higher and lower lift coefficients.

Convergence history for *Case*_9 is given in Fig. 13, where airfoil shapes at different optimization steps (iterations) are depicted. A total of 10 optimization steps was needed for full convergence. Starting from iteration 6, aerodynamic profiles practically coincide, indicating a good convergence of the optimization process.

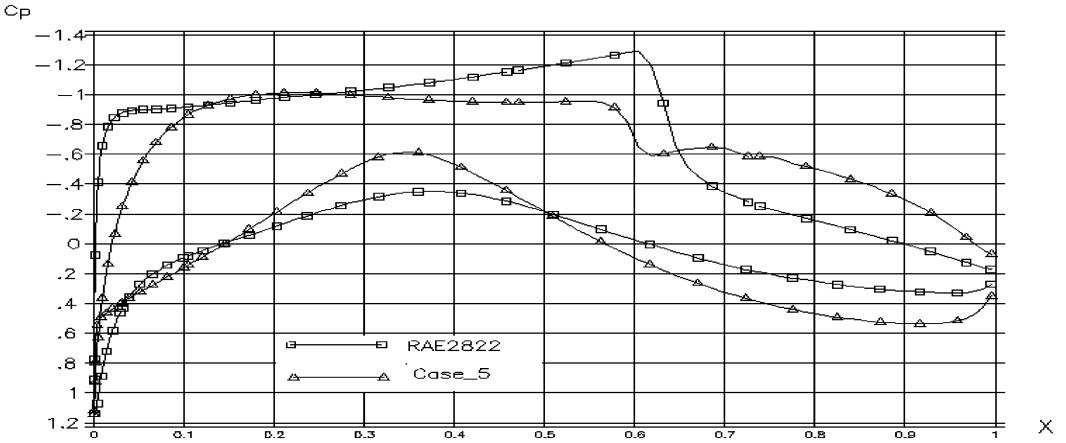

Fig. 7 Comparison of pressure coefficients C_p of the optimized profile (*Case*_5) and the original RAE2822 airfoil at $C_L = 0.745$.

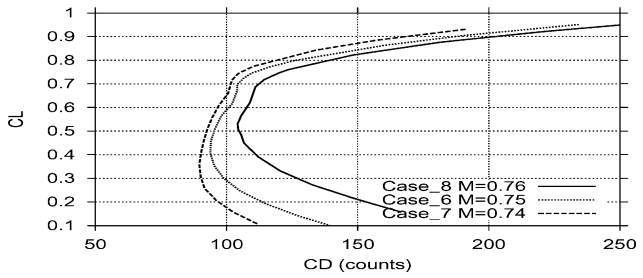

Fig. 8 Lift/drag polars for airfoils, optimized at different design Mach numbers.

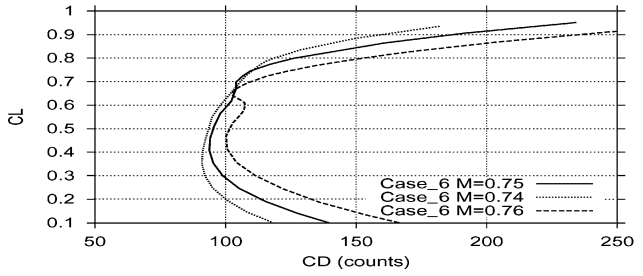

Fig. 9 One-point optimization at design $M = 0.75$, $C_L = 0.745$ (*Case*_6). Drag polars in the neighborhood of design Mach number.

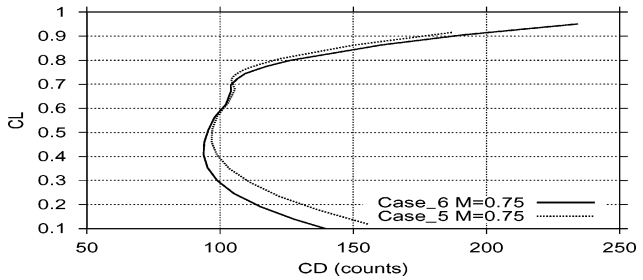

Fig. 10 Influence of design value of the trailing-edge angle on drag polar. *Case*_6: $\theta_T^* = 3.4$ deg. *Case*_5: $\theta_T^* = 0.0$ deg.

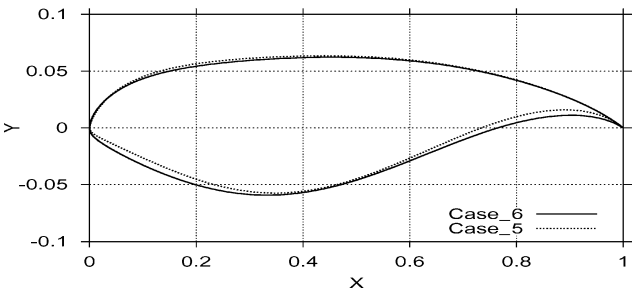

Fig. 11 Influence of design value of the trailing-edge angle on aerodynamic shape. *Case*_6: $\theta_T^* = 3.4$ deg. *Case*_5: $\theta_T^* = 0.0$ deg.

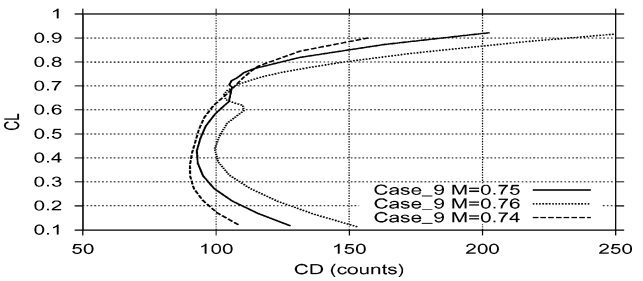

Fig. 12 *Case*_9. Three-point optimization: $M = 0.75$, $C_L = 0.745$ (main design point); $M = 0.76$, $C_L = 0.745$; $M = 0.74$, $C_L = 0.745$ (secondary design points). Drag polars at different Mach numbers.

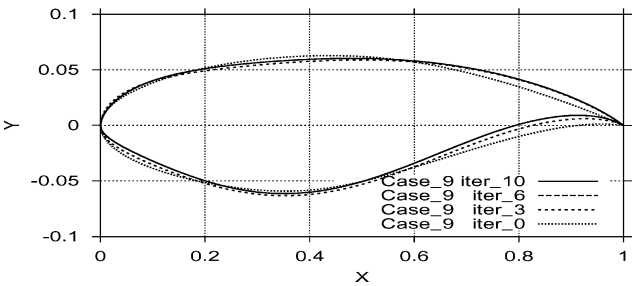

Fig. 13 *Case*_9. Three-point optimization: $M = 0.75$, $C_L = 0.745$ (main design point); $M = 0.76$, $C_L = 0.745$; $M = 0.74$, $C_L = 0.745$ (secondary design points). Convergence history of aerodynamic shape.

In general, a relatively low number of optimization steps (typically from 5 to 12) are needed in order to get a reasonably converged solution in terms of aerodynamic shape as well as of cost function value. This holds for both one-point and multipoint optimizations. The preceding property of fast and robust convergence is especially important in engineering environment.

Figures 14 and 15 present results of comparison between one-point (*Case*_6) and different types of multipoint optimizations (*Case*_9, multiple Mach optimization; *Case*_10 and *Case*_11, multiple C_L optimizations). It can be seen from Fig. 14 that, at the main design point ($M = 0.75$, $C_L = 0.745$), the values of total drag coefficient are very close one to another in all of the cases considered. At the same time, a wide-range multiple C_L optimization (*Case*_11), which includes the influence of secondary design points ($M = 0.75$, $C_L = 0.5$ and $M = 0.75$, $C_L = 0.3$), achieves (as expected) a better performance for lift coefficient values lower than $C_L = 0.4$. On the other hand, a close-range multiple C_L optimization (*Case*_10),

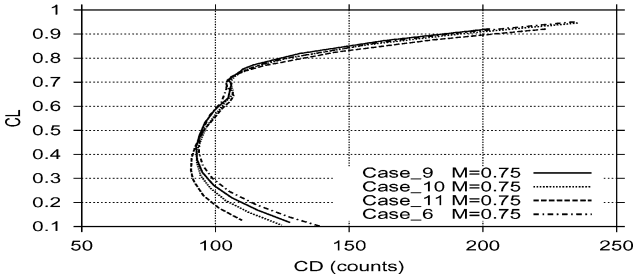

Fig. 14 Drag polars at $M = 0.75$. One-point optimization (*Case*_6) vs multiple Mach (*Case*_9) and two multiple C_L optimizations (*Case*_10, *Case*_11).

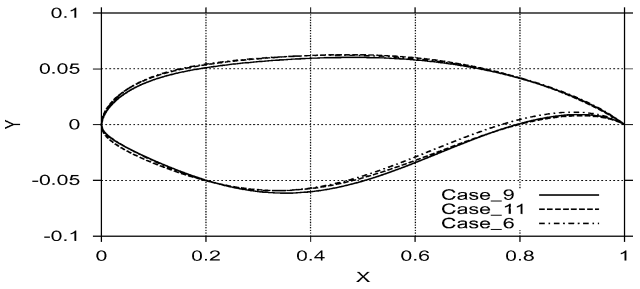

Fig. 15 Designed airfoils. One-point optimization (*Case*_6) vs multiple Mach (*Case*_9) and a multiple C_L optimization (*Case*_11).

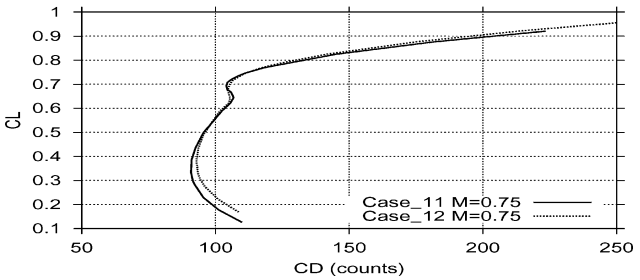

Fig. 16 Drag polars at $M = 0.75$. Two multiple C_L optimizations with different weights (*Case*_11, *Case*_12).

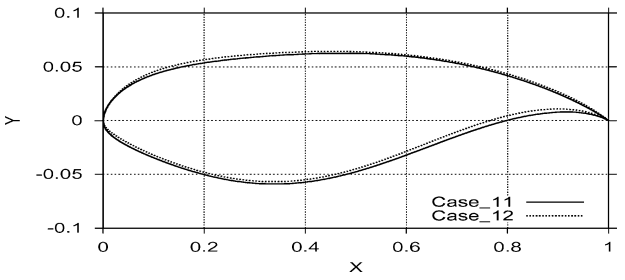

Fig. 17 Designed airfoils. Two multiple C_L optimizations with different weights (*Case*_11, *Case*_12).

with secondary design points $M = 0.75$, $C_L = 0.77$ and $M = 0.75$, $C_L = 0.72$, leads to an improved performance at lift coefficients higher than $C_L = 0.745$ (compared to the wide-range *Case*_11).

In Fig. 16, drag polars for two multiple wide-range C_L optimizations at $M = 0.75$ are shown. In both cases, the main design point was $C_L = 0.745$, $M = 0.75$ and secondary design points were $C_L = 0.5$, $M = 0.75$ and $C_L = 0.3$, $M = 0.75$. The weight coefficients corresponding to the preceding design points are 0.5, 0.25, 0.25 and 0.8, 0.1, 0.1, respectively. The corresponding airfoil shapes are depicted in Fig. 17. The two polars are relatively close one to another. They practically coincide at the main design point while slightly differing at both lower and higher lift coefficients. Note

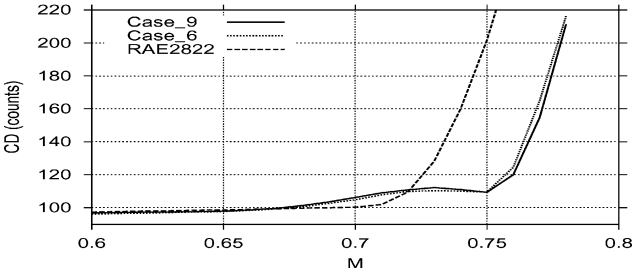

Fig. 18 Mach drag rise at fixed $C_L = 0.745$. Original RAE2822 airfoil vs optimized profiles. *Case*_6, one-point optimization at $C_L = 0.745$, $M = 0.75$; *Case*_9, three-point optimization at $C_L = 0.745$ and $M = 0.75$, 0.74, 0.76.

that the behavior of the polars outside the main design point is well correlated with values of weight coefficients.

In aerodynamic practice, the global rather than pointwise behavior of airfoils is essential. In this connection, in order to estimate the off-design performance of the optimized airfoils, it is important to compare (along with the drag polars) Mach drag rise behavior of optimized airfoils with that of the original RAE2822 profile. In Fig. 18, drag-divergence curves are shown, for the original RAE2822 airfoil alongside those for two optimized airfoils (*Case*_9 and *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. Both the multipoint and one-point optimizations result in essential extension of the low drag zone up to at least $M = 0.75$. Even at lower off-design Mach numbers, the optimized profiles possess a slightly lower drag than the original airfoil.

VIII. Conclusions

A robust hybrid genetic-algorithm/reduced-order-model (GA/ROM) approach to the multiobjective constrained optimization of aerodynamic configurations is suggested. Novel features of the algorithm include an efficient treatment of nonlinear constraints in the framework of genetic algorithms and scanning of the optimization search space by a combination of full Navier–Stokes computations with the reduced-order-models method, along with efficient multilevel parallelization strategy, which makes use of computational power supplied by massively parallel processors.

The method was applied to the one-point and multipoint optimization of transonic airfoils with a variety of nonlinear constraints. The results demonstrated that the method retains high robustness of conventional GAs while keeping computational-fluid-dynamic computational volume at an acceptable level because of limited use of full Navier–Stokes computations. This allowed employment of the method in a demanding engineering environment.

References

[1]Lighthill, M. J., "A New Method of Twodimensional Aerodynamic Design," Aeronautical Research Council, M&R 1111, 1945.

[2]Bauer, F., Garabedian, P., Korn, D., and Jameson, A., *Supercritical Wing Sections II*, Springer-Verlag, New York, 1975.

[3]Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 4, 1978, pp. 407–412.

[4]*Optimum Design Methods for Aerodynamics*, AGARD-R-803, Nov. 1994.

[5]Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 2, 1988, pp. 233–260.

[6]Jameson, A., "Optimum Aerodynamic Design Using Control Theory," *CFD Review*, Wiley, New York, 1995, pp. 495–528.

[7]Hajela, P., "Nongradient Methods in Multidisciplinary Design Optimization—Status and Potential," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 255–265.

[8]Epstein, B., Averbuch, A., and Yavneh, I., "An Accurate ENO Driven Multigrid Method Applied to 3D Turbulent Transonic Flows," *Journal of Computational Physics*, Vol. 168, No. 2, 2001, pp. 316–338.

[9]Peigin, S., Epstein, B., Rubin, T., and Seror, S., "Parallel High Accuracy CFD Code for Complete Aircraft Viscous Flow Simulations," *Parallel Computational Fluid Dynamics. New Frontiers and Multi-Disciplinary Applications*, Elsevier, Amsterdam, 2003, pp. 507–514.

[10]Epstein, B., Rubin, T., and Seror, S., "Accurate Multiblock Navier–Stokes Solver for Complex Aerodynamic Configurations," *AIAA Journal*, Vol. 41, No.4, 2003, pp. 582–594.

[11]Harten, A., Engquist, B., Osher, S., and Chakravarthy, S., "Uniformly High Order Accurate Non-Oscillatory Schemes. 1," *Journal of Computational Physics*, Vol. 71, No. 2, 1987, pp. 231–303.

[12]Shu, C.-W., and Osher, S., "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes," *Journal of Computational Physics*, Vol. 83, No. 1, 1989, pp. 32–78.

[13]Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Westey Longman, Reading, MA, 1989.

[14]Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, New York, 1996.

[15]Sefrioui, M., Periaux, J., and Ganascia, J.-G., "Fast Convergence Thanks to Diversity," *Proceedings of the 5-th Annual Conference on Evolutionary Programming*, MIT Press, Cambridge, MA, 1996, pp. 313–321.

[16]Peigin, S., and Desideri, J.-A., "Parallel Implementation of Genetic Algorithms to the Solution for the Space Vehicle Reentry Trajectory Problem," *Parallel Computational Fluid Dynamics. Trends and Applications*, Elsevier, Amsterdam, pp. 357–364.

[17]Carr, M. P., and Palister, K. C., "Pressure Distributions Measured on Research Wing M100 Mounted on an Axisymmetric Body," AGARD, AR-138, 1984, Addendum.

[18]Marconi, F., Siclari, M., Carpenter, G., and Chow, R., "Comparison of TLNS3D Computations with Test Data for a Transport Wing/Simple Body Configuration," AIAA Paper 94-2237, 1994.

[19]Nadarajah, S. K., and Jameson, A., "Studies of the Continuous and Discrete Adjoint Approaches to Viscous Automatic Aerodynamic Shape Optimization," AIAA Paper 2001-2530, 2001.

[20]Quagliarella, D., "Airfoil Design Using Navier-Stokes Equations and an Asymmetric Multi-Objective Genetic Algorithm," *Proceedings of International Congress on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, edited by G. Bugeda, J.-A. Desideri, J. Periaux, M. Schoenauer, and G. Winter, CIMNE, Barcelona, 2003.

E. Livne
Associate Editor