Multilevel parallelization strategy for optimization of aerodynamic shapes

S. Peigin[1*], B.Epstein[2], S.Gali[3]

[1] Israel Aircraft Industries, Israel
[2] The Academic College of Tel-Aviv Yaffo, Israel
[3] R&D IMoD, Israel

Introduction

There is a worldwide demand for efficient and robust software which implements aerodynamic optimization. This request is explained by the 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 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 the 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, due to 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 1 optimization loop is measured by months and the whole process may take even years.

The above may occur even if the design treats only a limited part of the whole configuration, e.g. 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 may dramatically reduce the time devoted to testing and overall cost of design by reducing the number of optimization cycles to a single loop (one-shot optimization approach). On the other hand, the problem of multiobjective 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 problem of optimization of aerodynamic shapes is very time-consuming as it requires a huge amount of computational work. Each optimization step requires a number of heavy CFD runs, and a large number of such steps is needed to reach 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 it is suggested to use a multilevel parallelization strategy. It includes parallelization of the multiblock full Navier-Stokes solver, parallel evaluation of objective function and, finally, parallelization of the optimization framework.

The method was applied to the problem of 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.

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

* E-mail: peigin@barak-online.net

computational volume at an acceptable level due to a limited use of full Navier-Stokes computations and multilevel parallelization of the whole computational framework which efficiently makes use of computational power supplied by massively parallel processors (MPP).

Statement of the problem

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, shape "freeze" of certain portions of airfoil (such as lower or upper surfaces, trailing (leading) edge region etc.):

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

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

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.

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 of the Bezier coefficients and angle of attack, 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, corresponding to these variations.
3. The local CFD data base is included in the global CFD data base.
4. Using local approximation of cost function, 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,...,N_D; N_D is the number of the search domains).
5. Full Navier-Stokes solver is applied to each optimal point O_k, and the corresponding data are added to the global CFD data base.
6. A new basic point is determined as the best point taken from the global CFD data base.
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 neighbourhood
                                 of the basic point/
Search_Optim_Candidates         /Hybrid GA-ROM 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 the points in the global CFD data base/
```

opt_step := *opt_step* +1
enddo

Multilevel parallelization strategy

In order to improve the computational efficiency of the algorithm the following multi-level parallelization strategy was used:

- Level 1 - Parallelization of full Navier-Stokes solver
- Level 2 - Parallel evalutation of objective function
- Level 3 - Parallelization of the optimization framework

The first level parallelization approach 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 average, the communication work per processor is not increased if the number of processors is increased.

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. It is applied when executing the steps *Calc_Local_Data_Base* and *Verification_Optim_Cand* in the pseudo-code of the optimization algoritm. 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 profiles 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%.

The third level of parallelization was used at the step *Search_Optim_Candidates* in the pseudo-code of the optimization algoritm (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.

On this level of parallelization, all 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, 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 parallel asynchronous GA algorithm involved the following main stages:

a) The initial population (based on random search) is independently obtained on each processor of the group.
b) Non-blocking data receiving from other processors of the group. The received individuals are included in the current population. If the requested message has not arrived this step is skipped.
c) With a preassigned value of the generation step, the information related to the best individual in the generation is broadcast to the other processors of the group.

Finally we can conclude that 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.

Preliminary results

We present here applications of the above optimization algorithm to airfoil design. The design problem consists of the minimization of total drag starting from the RAE2822 airfoil at a transonic design point.

Multilevel parallelization strategy based on the PVM software package was implemented on a cluster of MIMD multiprocessors consisting of 57 HP NetServer LP1000R nodes. Each node has 2 processors, 2GB RAM memory, 512KB Level 2 Cache memory and full duplex 100Mbps ETHERNET interface. Totally this cluster contained 114 processors with 114GB RAM and 29MB Level 2 Cache memory.

The results of computations demonstrated that application of suggested multilivel parallelization strategy allowed to reduce execution time for one airfoil optimization from 20 days to 5-6 hours on 114 processors cluster.

In the following, we present the results of drag minimization of RAE2822 airfoil at M=0.75, Re=6.5·10^6 and fixed lift coefficient C_L = 0.745. Note, that this design point corresponds to flight conditions with a strong shock-boundary layer interaction.

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 trailing edge shape (in order to avoid “fishtails”).

For a high transonic target C_L = 0.745, 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 neighborhood of the design point from C_L = 0.6 up to C_L = 0.76.

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) diminuation 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 than cusped trailing edge.

Pressure coefficients of optimized profile were compared with those of RAE2822 airfoil at the corresponding design point C_L = 0.745. It was clearly seen that the optimization leads to the destruction of a strong shock, present in the original pressure distribution. It is important that a favourable pressure distribution is retained in a vast neighbourhood of the design point.

Conclusions

A new approach to the constrained design of aerodynamic shapes was suggested. The algorithm features a new strategy of efficient handling of nonlinear constraints in the framework of GAs, scanning of the optimization search space by of the combination of full Navier-Stokes computations with the Reduced Order Models method and multilevel parallelization of the whole computational framework. The method was applied to the optimization of transonic airfoil with a variety of nonlinear constraints. The results demonstrated that the method combines high accuracy with computational efficiency and can be used for practical applications on a daily basis needed in industry.