AN AERODYNAMIC OPTIMIZATION FRAMEWORK BASED ON GENETIC ALGORITHMS

Alexandre P. Antunes*, Ricardo Galdino da Silva*, João Luiz F. Azevedo**

***Universidade de São Paulo, USP**
e-mail: ap-antunes@uol.com.br
e-mail: ricardo.galdino@poli.usp.br

**** Instituto de Aeronáutica e Espaço, CTA/IAE/ALA**
e-mail: azevedo@iae.cta.br

Keywords: *Aerodynamic Design, Optimization Framework, Genetic Algorithm, MDO*

Abstract

The present paper describes the efforts on the construction of a computational framework for 2-D and 3-D aerodynamic optimizations. The creation of the framework is an attempt to generate a design environment capable of coupling various tools from different levels of complexity and with diverse functionalities. The conceptual framework is developed to be inserted into daily activities of an aerodynamic CFD group. The framework is implemented for both Windows and Linux-running platforms, and it is augmented by a user-friendly graphical interface. Usage of the framework is illustrated in the paper by 2-D aerodynamic optimization of a cruise configuration for different flight conditions. The aspects investigated include the influence of the number of individuals on the aerodynamic coefficients, the effect of boundary layer transition location on the final optimized shape, and the benefits of the solver fidelity level as compared to the computational cost.

1 Introduction

The aircraft is one of those few products in which a large amount of interdisciplinary engineering technologies are coupled. For this reason, aircraft design consists of a multidisciplinary task, which must be accomplished in a reliable and efficient way. The process of design is carried out through diverse phases, in which the level of complexity and component discretization increase at every stage. In order to sustain an efficient design process, it is mandatory to minimize the need for rework at more advanced phases. Usually, a design exercise, in which all disciplines can be taken into account in the earlier phases of the process, avoids unpleasant surprises in the latter stages. Thus, it is quite clear the importance of coupling as many disciplines as possible during the earlier stages of design. *Multi Disciplinary Optimization* (MDO) procedures [1] comprise approaches to obtain reliable and efficient designs. Computational cost associated with this multidisciplinarity is the main subject that must be evaluated in order to decide on the feasibility of full or partial integration of the disciplines.

In this context, the paper describes current efforts on the development of a computational framework to provide 2-D and 3-D external aerodynamic optimizations. The creation of the framework is an attempt to generate a design environment capable of coupling various tools from different levels of complexity and with diverse functionalities. The framework is built to be inserted into daily activities of a typical aerodynamic CFD group. On such groups, the main activities are related to three processes: mesh generation, numerical simulation and post-processing. In order to allow integration and communication among these different applications, a set of subroutines, written in

Fortran-90, and templates are developed. Furthermore, in order to confer a certain level of flexibility in the use of the framework, it is implement for both Windows and Linux-running platforms. Moreover, a GUI is also developed to serve as a user-friendly interface between the user and the coupled tools.

At the present stage of the framework development, only the aerodynamic activities are integrated. However, in a second moment, other disciplines will be coupled to it in order to create an MDO environment. The framework provides an improvement in the aerodynamic design efficiency, through the integration of various design tools with an optimization algorithm. Such integration intends to eliminate most of the time spent during the preparation of the simulation. The saved time can be used to increase the number of simulations or to perform more thorough and critical analyses of the results.

The optimization process might be simply defined as finding the most suitable solution for a specific problem being analyzed. Many times the problem at hand might appear simple at a first glance. However, the complexity increases with the amount of the problem variables and the number of objectives to be achieved. A large amount of numerical and mathematical methods can be used to perform the optimization of a problem,. One of the reasons for the existence of a vast number of techniques is the fact that many of them have been developed to deal with specific types of optimization problems.

The available optimization methodologies can be classiefied as:

- classical methods;
- evolutionary methods.

Examples of classical methods are geometric programming [2], nonlinear programming [3], linear programming [4], and conjugate gradient methods [5], among others. On the other hand, genetic algorithms [6], artificial life [7], and ant colony [8] techniques can be cited as examples of evolutionary methods.

There are advantages and disadvantages on both approaches. Usually, the classical methods are blamed for getting stuck on suboptimal solutions and for having optimal solution which depend on the chosen initial condition. On the other hand, the evolutionary methods are more expensive due to the fact that they work with a large number of solutions.

In the present case, the decision was to use genetic algorithms (GA's) [9,10] as the main optimization methodology. GA's are search optimization methods that use principles of natural genetics and natural selection. In such methods, the possible solutions for a certain problem are represented by some form of biological population, which evolve over generations to adapt to an environment by selection, crossover and mutation. Instead of working with a single solution at each iteration of the process, a GA works with a number of solutions, known as a population.

The optimization process can be classified, with regard to the objective function, as single-objective optimizations and multi-objective optimizations. For the first optimization type, a single objective drives the search for the best, or most feasible, solution, and the optimization is expected to obtain a single solution. On the other hand, for multi-objective optimizations, more than one objective drives the process. In the real world, most of the optimization problems are multi-objective and, usually, the objectives are conflicting with each other. Such conflicting characteristics do not allow a single solution, but one obtains a set of solutions, which are commonly known as a Pareto front.

The two important goals in multi-objective optimization are:

- to find solutions as close as possible to the Pareto-optimal solutions;
- to find soultions as diverse as possible in the resulting non-dominated front.

Results for single-objective and multi-objective optimizations are presented in this work.

2. Framework Structure

As already discussed, in the present initial stage, the conceptual framework is developed to be inserted into the daily activities of a CFD group. Therefore, the main activities performed are related to three processes: mesh generation, numerical simulation and post-processing. In order to incorporate the three main processes of a CFD analysis, the framework is built with the

capability of automatically generating the geometry and the mesh and, also, of running the simulation. These three steps are accomplished in conjunction with a specific GA. The following GA's are currently available: SGA (Single Genetic Algorithm), VEGA (Vector Evaluated Genetic Algorithm), NSGA (Non-Sorting Genetic Algorithm), and NSGA-II (Non-Sorting Genetic Algorithm-II). The capability of selecting a binary codification or a real-coded representation of the design variables is also available to the user. Moreover, the ability of restarting the optimization from the last generation creates an important capability, since it allows the use of non-dedicated machines during the night. In Fig. 1, one can see a flowchart of the routines of the implemented GA algorithms.

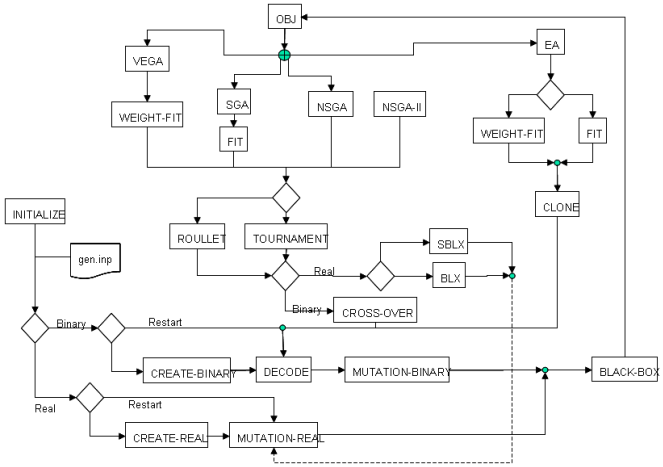

Fig. 1. Flowchart of the GA subroutines implemented in the framework.

One can observe another flowchart in Fig. 2 containing the main concept behind the framework structure. The inter-connections between the different procedures or tasks are handled by a Fortran program, which is the core of the framework. One can notice that geometry and mesh generations are treated as separate modules. This modularity concept allows the development of each procedure independently, improving the efficiency of the development.

In order to start the optimization process, a set of input files containing all the information about geometry, mesh, flight conditions and selected solver is necessary. Since a large amount of information is required by the input files, a debug module was created to manage conflicting or senseless setups. Such a module avoids that an unphysical or unrealistic optimization might be attempted.

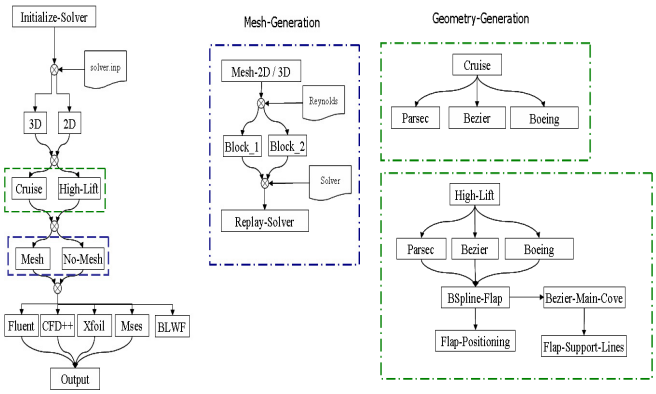

Fig. 2. Framework main structure.

3 Modules

3.1 Geometry Module

The geometry module is responsible for the automatic creation of the aerodynamics shapes. These geometric shapes can be constructed by three different forms of parametrizations, namely Bezier curves [11], Boeing curves [12] and Parsec airfoils [13]. For a 2-D optimization, cruise and high-lift configurations are already implemented. However, for a 3-D optimization, only the cruise configuration is available. The geometric input files contain the setup of the configuration to be optimized. For the cruise configuration, the geometric shape generation is straightforward. On the other hand, for the high-lift configuration, a more elaborate geometric treatment is necessary due to the intrinsic complexities of the configuration.

The shape generation always begins with a cruise configuration for the 2-D optimization. If a high-lift optimization is selected in the geometry input file, the geometry module automatically generates the flap geometry from the cruise profile. The creation of the flap geometry is performed using information on the spar, the spoiler and four control points. The spar position determines the flap chord. The spoiler data define the point of tangency between the flap leading edge and the cruise

configuration upper surface. This point of tangency, situated on the upper surface of the main airfoil, is projected to the lower surface. This procedure generates the two starting control points, one for the upper surface and the other for the lower surface. From these initial points, small displacements might be performed in order to introduce changes in the flap leading edge. The last two control points, necessary to create the flap leading edge, are obtained performing a projection of the initial control point into the spar plane. The direction of the projection is obtained taking the tangent vector to the surface at the respective control points.

In Fig. 3, one can observe a sketch of the airfoil geometry, defining all the important parameters that characterize the flap geometric shape. In Fig. 4, one can also observe how different geometric flap shapes can be created as a consequence of different parametrization changes in the spar, spoiler and control point setup.

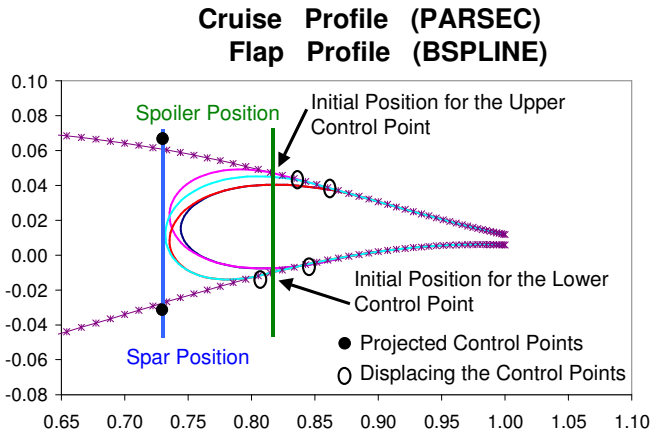

Fig. 3. Geometrical parameters used to define the flap shape.

After the flap has been generated, it is displaced and rotated according to the data from the geometry input files. The first element of the high-lift configuration is obtained introducing the cove shape in the cruise profile and, afterwards, eliminating those geometric parts that, now, belong to the flap upper and lower surfaces. In Fig. 5, one can see the deflection of the flap and the modified cruise profile which was used to create the main element of the high-lift configuration. Figure 5 also shows the supporting lines for two flap deflections. Such lines are created to provide some guiding in the mesh generation process.

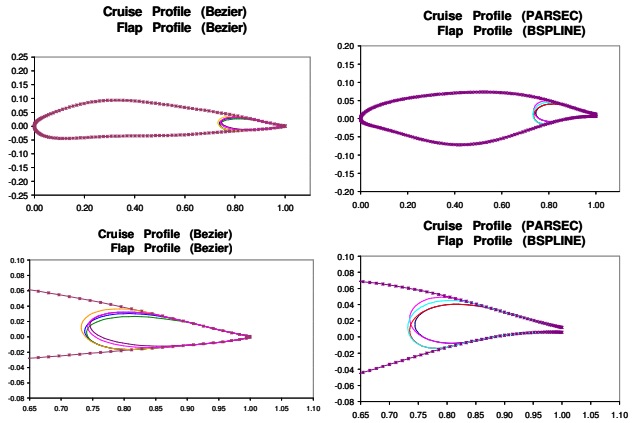

Fig. 4. Generation of the cruise configuration and creation of alternative flap geometries for a high-lift configuration.

For the 3-D cases, the geometry input file contains the information on the number of sections and on the design variables that define the wing planform for the optimization process. Such geometric information is used in a ICEM_CFD [14] script file to automatically generate the mesh. At the present development stage, the wing thus generated is attached to the symmetry plane. However, as the framework development progresses, the wing will be attached to the parameterized fairing and fuselage geometries.

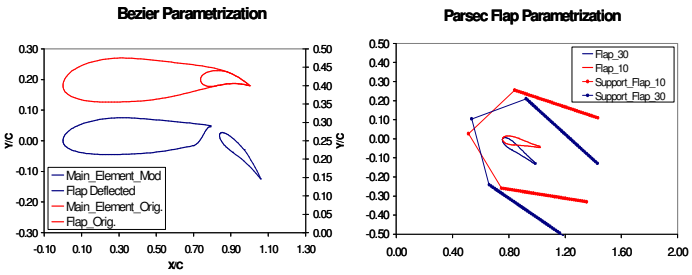

Fig. 5. High-lift configuration and supporting lines for the mesh generation process.

In Fig. 6, one can see the wing surface and the supporting lines created for the mesh generation process. These lines are important to support the hexahedral mesh, avoiding the existence of crossing lines between the blocks. Such crossing lines would lead to negative volumetric elements. The framework can also generate hybrid prismatic-tetrahedral meshes, besides the hexahedral meshes.

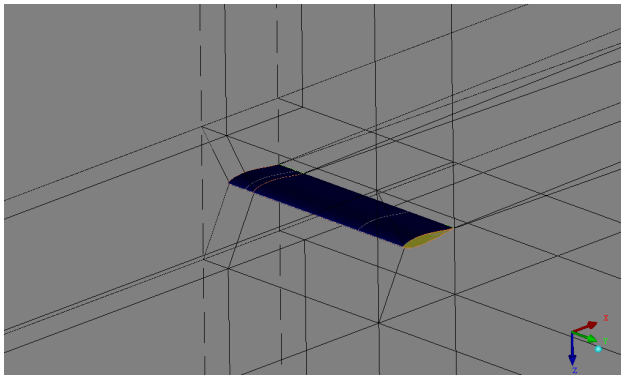

Fig. 6. Supporting lines for the hexahedral mesh.

3.2 Mesh Module

This module is responsible for the 2-D and 3-D mesh generation. The process of generating the mesh is accomplished using the ICEM-CFD script language. General aspects of the mesh, such as farfield position, number of points, distribution of points over the surfaces and off the surfaces, among others, are defined based on the experience of the authors' [15]. The distance of the first mesh point outside the wall surface is specified as function of the Reynolds number of the simulation. In order to have a reasonable y+ value for a solution to the wall approach, an increase in the Reynolds number is accompanied by a decreases in the distance of the first mesh point off the wall. This procedure yields good results for the 2-D optimization. However, for 3-D simulations, wall functions are implemented due to the prohibitive computational costs of an integration to the wall approach in 3-D optimization processes.

For cruise configurations, grid generation over the profile uses a C-mesh construction structure. The high-lift configuration proceeds with the same methodology for mesh generation over the main and the flap elements. Such an approach is possible for high-lift configurations, in the present case, because the work considers solvers that can handle overset grids. This decision has considerably facilitated the mesh generation process.

Figure 7 shows meshes generated over different airfoil shapes. Mesh generation is accomplished with the aid of the support points and curves. These auxiliary entities are obtained from the proper mesh module. The mesh blocks are parameterized in such a way that a good quality mesh is created even for the unusual profiles that may be generated during the optimization process.

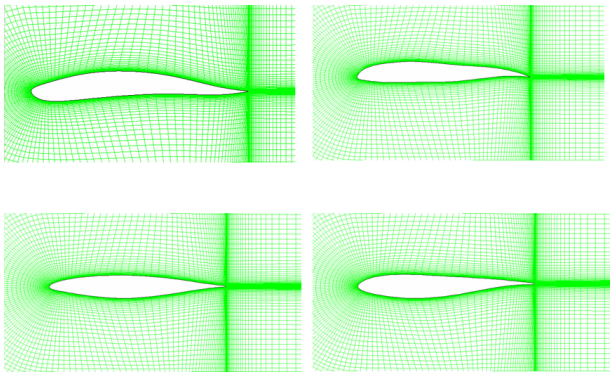

Fig. 7. Meshes generated over the parameterized airfoils.

Finally, Fig. 8 shows a station-cut in the 3-D mesh generated over a wing surface. A prism layer is generated over wing surface to allow for a better capture of the boundary layer. The height of the prism layer is determined as a function of the boundary layer thickness. The number of the prism layers depends on the total height and on the prism growth ratio, which is fixed at 1.15 in this example.

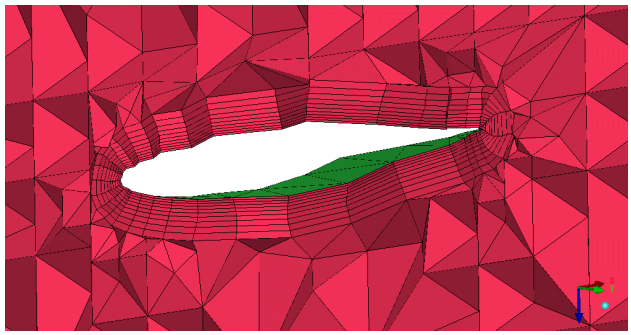

Fig. 8. A station-cut in the 3-D mesh generated over the wing surface.

3.3 Solver Module

The solver module is responsible for performing the numerical setup to the following numerical (commercial) solvers: Fluent [16], CFD++ [17], Xfoil [18], Mses [19], Blwf [20].

4. Output Files

Whenever a optimization is performed, a set of files is generated to supply the user with all sort of information for the post-processing stage. Based on these files, one can find the

Pareto-fronts, the decoded-variables, the binary strings, the objective functions, among other important results of the calculation. Figure 9 shows an example of an output file from the optimization process. This specific file contains information about the mean fitness, the individual fitness and the binary string.

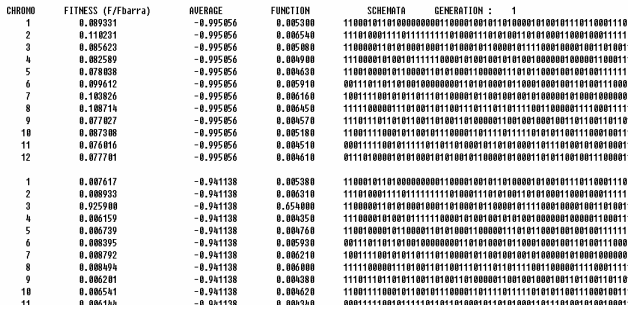

Fig. 9. Example of the output information from the optimization process.

5. Graphical Interface

The setup of input files can be accomplished using the graphical interface or using a text editor, since the input files are ASCII files.

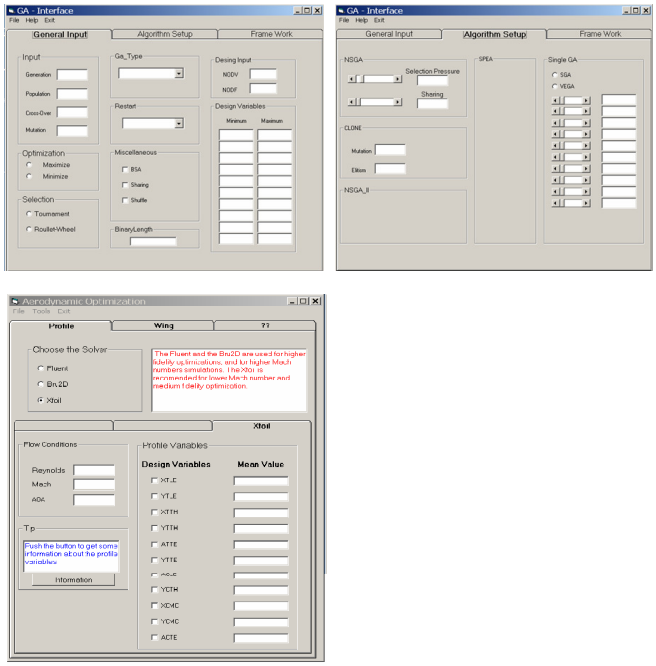

Fig. 10. Part of the graphical interface under development for the framework.

Figure 10 presents a view of the Windows-platform graphical interface which is used to generate the input file. Due to the current continuous development of the framework, and, hence, the modifications in the input file format, the graphical interface is updated only after major module implementations are finalized.

6. Validation

6.1 Analytical Functions

The validation of the implemented GA's is the first step to be performed before any type of real optimization problem can be undertaken. In order to perform this validation, a group of analytical functions was selected. Among these functions, there are single- and multi-objective test problems.

Table 1 summarizes the results obtained with the algorithms implemented for single-objective optimization. One can notice that the real-coded representation of the design variables failed in getting the right optimized solution for two of the functions. These test cases are being further analyzed in other to find out the reasons that led to such unsatisfactory results.

Table 1. Tests for single-objective functions.

		Binary	Real - BLX
	Funções	Minimize	Minimize
1	Michalewicz (N=5)		
2	Michalewicz (N=10)		
3	De Jong's (function 2)		
4	Schwefel (N=5)		
5	Schwefel (N=10)		
6	Rastrigin		
7	Goldstein-Price (N=2)		
8	Goldstein-Price (N=4)		
9	Hyper-Elipsoid		
10	Branin		
11	Potencias		
12	Eason		
13	MF1		
14	MF2		
15	MF3		

	passed
	failed

For the multi-objective validation cases, some of the literature best-known test problems were selected. One of the characteristics of these test problems is the fact that they evaluate the capability of the algorithm to obtain the true Pareto-optimal front, while still maintaining the diversity of solutions. Moreover, the capability to handle different shapes of the Pareto front, such as convex, nonconvex or discontinuous fronts, are also tested. The following test functions were selected: ZDT1, ZDT2, ZDT3, ZDT4 [21], FON [22], and SCH [23]. Such test

cases have 2 objective functions and the number of design variables varies from 2 up to 30.

Definitely, among these problems, the most complex functions are ZDT1, ZDT2 and ZDT3, due to the large number of design variables. In Fig. 11, one can appreciate the sensitiveness of the results with regard to changes in the setup of the algorithm parameters. One can further notice that the true Pareto-optimal front can only be achieved with a modification in the mutation parameter. Changes in the others parameters produced no effect at all on the results. Figure 12 shows the results obtained for the ZDT3 function, yielding similar behavior with regard to the optimization parameter effects.

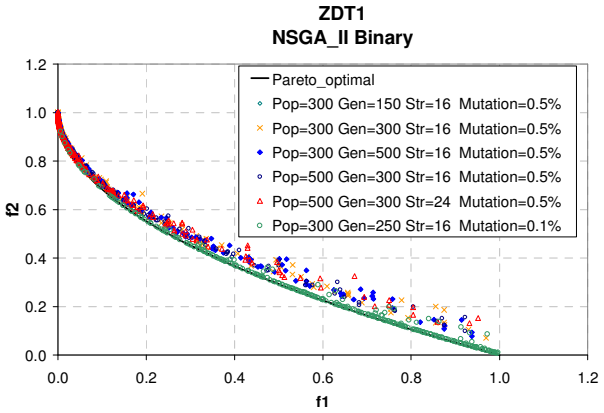

Fig. 11. ZDT1 multi-objective test problem.

The increase in the number of design parameters sometimes leads to such sensitivity in relation to the algorithm setup. Moerover, several times, it is not obvious which parameter should be changed in order to obtain the correct results. In other words, it is not true that every setup will lead to the true Pareto-optimal front. Hence, those that use optimization algorithms as a black-box should be aware of this fact.

The validation using SCH, FON and ZDT4 functions has also yielded good results. These functions have only 10 design variables, which makes the problem easier when compared to the ones with ZDT1, ZDT2 and ZDT3 functions. For the former functions, even with a mutation value above the one practiced in the evaluations with the later functions, the true Pareto-optimal front is achieved.

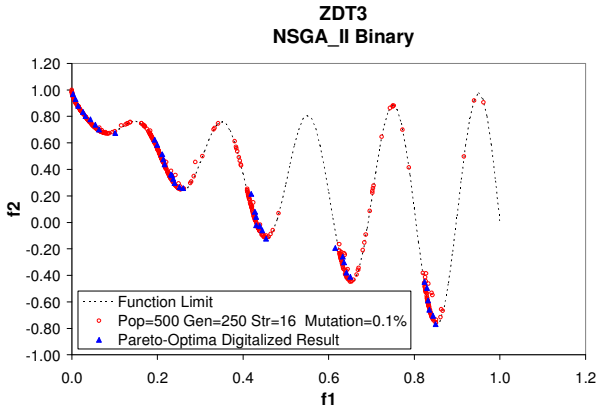

Fig. 12. ZDT3 multi-objective test problem.

6.2 Airfoil Shape Optimization

It is correct to state that the GA algorithms implemented provided good performance for most of the single- and multi-objective test functions considered in the analyses. Such encouraging results have allowed the authors to proceed into the next step, which consists in addressing aerodynamic optimization problems.

The first set of simulations performed, with aerodynamic interest, consists in airfoil shape optimization using the drag coefficient (Cd) as the objective function. These simulations are accomplished using the Bezier, Boeing and Parsec geometric parametrizations. Calculations consider a single flight condition. The solver selected for such evaluations is XFOIL, due to its lower computational costs.

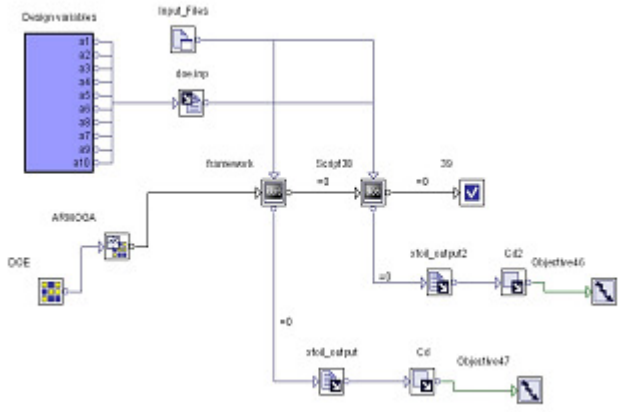

Fig. 13. ModeFrontier design environment used to validate the implemented framework.

In order to evaluate the quality of the results so obtained, the ModeFrontier [24] design environment is used as a checking tool.

For such a task, a design environment is created in which the framework is connected to the ModeFrontier optimization algorithms, as one can see in Fig. 13.

Figures 14-16 show, for the three geometric parametrizations here adopted, the comparison among the airfoil shapes obtained by the framework, using the NSGA algorithm, and by ModeFrontier, using the ARMOGA algorithm. There are slightly differences on the final airfoil shapes, if Bezier and Boeing parametrizations are used. However, for Parsec parametrization, the final airfoil shape obtained with the current framework is identical to the one given by the ModeFrontier algorithm. Such verification yields further confidence on the reliability of the implemented algorithms.

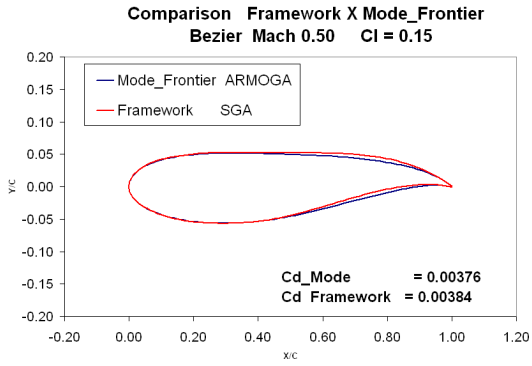

Fig. 14. Comparison between the airfoil shape obtained by the ModeFrontier and the airfoil shape obtained by the framework – Bezier Parametrization.

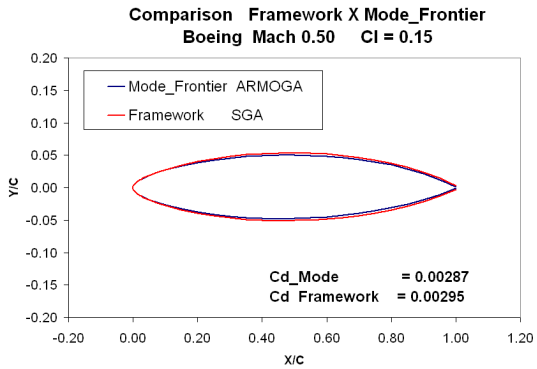

Fig. 15. Comparison between the airfoil shape obtained by the ModeFrontier and the airfoil shape obtained by the framework – Boeing Parametrization.

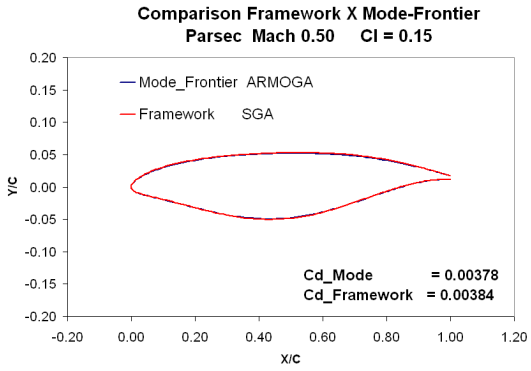

Fig. 16. Comparison between the airfoil shape obtained by the ModeFrontier and the airfoil shape obtained by the framework – Parsec Parametrization.

Results for multi-objective optimizations are shown in Figs. 17-19. Figure 17 shows the Pareto fronts obtained by ModeFrontier and the present framework, for Bezier parametrization. Similar results are shown in Figs. 18 and 19, for the Boeing and Parsec parametrizations, respectively. The comparison of the results is typically very good.

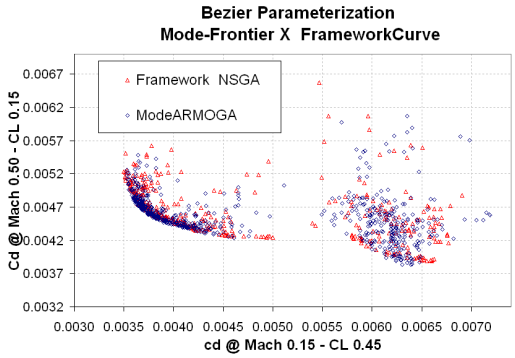

Fig. 17. Comparison between the framework and ModeFrontier for a multi-objective optimization using the Bezier parametrization.

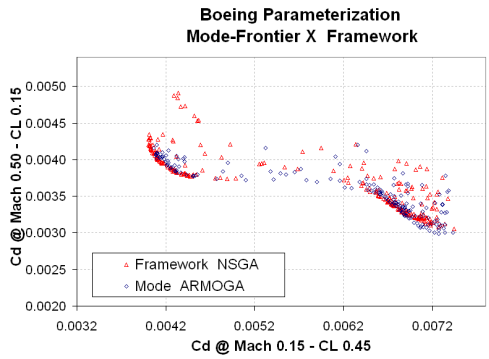

Fig. 18. Comparison between the framework and ModeFrontier for a multi-objective optimization using the Boeing parametrization.

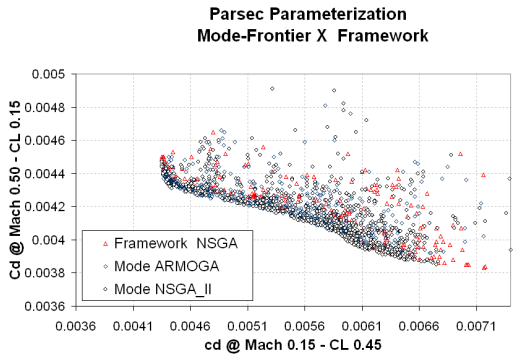

Fig. 19. Comparison between the framework and ModeFrontier for a multi-objective optimization using the Parsec parametrization.

7. Results

In the present work, numerous simulations are performed to evaluate the influence of the following aspects on the optimization process:

- effect of the parametrization;
- effect of the number of individuals;
- effect of the boundary layer transition.

7.1 Parametrization Study

Bezier, Boeing and Parsec parametrizations are used in order to study the effect that these different methodologies have on the geometric optimization process. The optimization study is performed with the Xfoil solver at the following flight conditions: Mach number of 0.50 and lift coefficient of 0.15; and Mach number of 0.15 and lift coefficient of 0.45. In the present study, the objective function is the minimization of the drag coefficient.

In Fig. 20, one can observe that different geometric parametrizations lead to different airfoil shapes. This is not a surprising behavior since each geometric parametrization has a particular form of generating the airfoil shape.

Typically, parametrizations that have more degrees of freedom to generate the airfoil shape have an easier time achieving the optimized configuration. The improvement associated with the geometric flexibility occurs due to the capability of creating a large amount of different airfoil shapes. On the other hand, as the parametrization gets more flexible, the number of control points, necessary to generate the airfoil shape, increases. This fact leads to a higher computational cost.

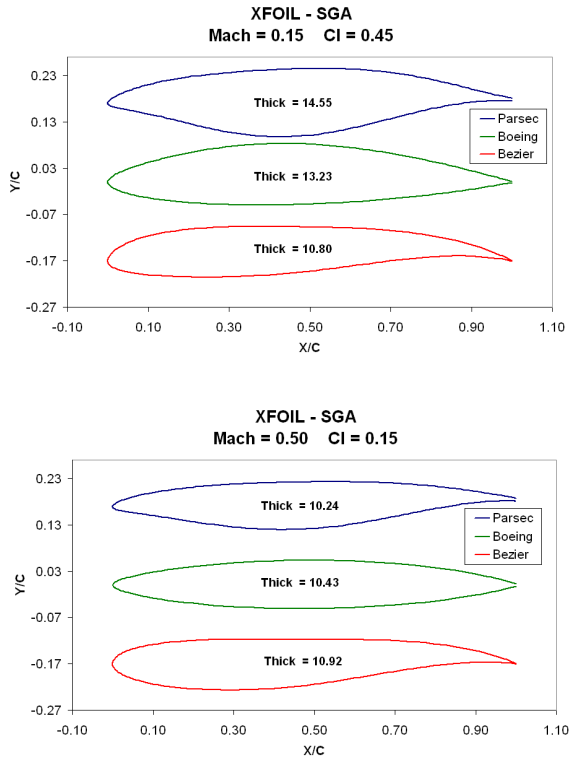

Fig. 20. Airfoil shapes obtained from different parametrization methodologies.

The previous observation does not imply that less flexible parametrizations cannot yield comparable optimized shapes. One must have in mind, however, that it is possible to obtain equal values for the objective function with different geometric shapes, as evidenced in the results presented in Table 2 for the Bezier and Parsec parametrizations. Hence, the conclusion is that the ideal parametrization is one which provides geometric flexibility to generate the airfoil, but at the same time have few design control points.

Table 2 presents the drag coefficient, Cd, for each of the airfoil shapes obtained by the three parametrizations. Part of the difference in Cd can attributed to the airfoil thickness. In this study, no constraints are imposed in terms of the minimum airfoil thickness. Thus, the thickness is allowed to vary within a pre-specified range. The results are actually showing that there is a large amount of possible geometric solutions for an aerodynamic optimization problem, and the chosen geometric parametrization considerably affects the final result.

Table 2. Drag coefficients for the optimal airfoil shapes obtained with the three parametrization methodologies studied.

	Boeing	Bezier	Parsec
Mach 0.15 @ Cl = 0.45	40.3	36.3	43.8
Mach 0.50 @ Cl = 0.15	29.5	38.9	38.9

7.2 Study of the Number of Individuals

The number of individuals, that is included in the population of a GA, is of fundamental importance on the optimization process. In some cases, it is just not possible to obtain the true optimum solution due to the insufficient number of individuals during the optimization. Usually, the necessary number of individuals to perform a certain optimization is not known a priori [25]. In order to obtain some measure of the effect of this parameter, a single-objective optimization, using the Parsec parametrization and the solvers Xfoil and Mses, is performed.

Figure 21 presents the effect of different number of individuals on the resulting objective function value. In this particular case, the aerodynamic drag coefficient, Cd, is selected as the objective function. One can clearly observe that the value of the objective function varies quite extensively if a small population is considered. It is also clear that the values of Cd start to level off only with a number of individuals above 500. However, as also indicated along the two curves in Fig. 21, the computational cost, in minutes per generation, increases rapidly as the population increases. Therefore, in this study case, the use of 300 individuals would be recommended, since the time per generation almost double and the difference in Cd is small enough to be attributed to the solver numerical accuracy, when comparing the results with 300 and 500 individuals.

The selection of the solver fidelity plays an important role in the optimization procedure, and it is intrinsically connected with the type of physical phenomena expected to be captured. If the physical phenomena can be well represented with a lower fidelity solver, the use of a higher fidelity solver does not bring any improvement into the optimization process, but it can certainly increase the computational cost.

An important aspect that must be always remembered is the relation between an adequate solver formulation and the final optimized shape for more severe flight conditions. For example, flight conditions with massively separated flow or with strong shock waves are particularly worrisome. For such cases, fidelity compatible with Navier-Stokes solvers is required, and issues such as an adequate mesh refinement and appropriate turbulence models are important aspects for the final results. Such questions, however, are beyond the scope of the present paper and they will be addressed by the authors in future work.

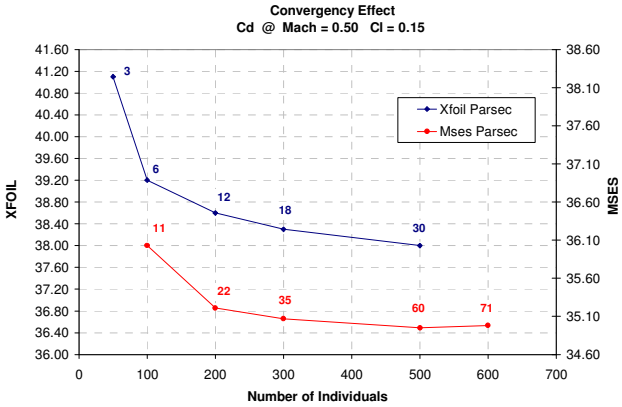

Fig. 21. Effect of the number of individuals over the objective function value and the computational cost.

7.3 Study of Boundary Layer Transition

The transition of the boundary layer and its location have large effects on the final geometric shape obtained by the optimization process. It is observed that, if the transition is not imposed, and flight conditions do not yield the appearance of massively detached flow, the optimization process searches for a geometric shape with the most extended laminar boundary layer as possible. On the other hand, if the flight condition is more severe, the optimization process searches for a geometric shape that withstands the pressure gradient and, at the same time, presents the lowest pressure drag coefficient. Typically, the capability to resist the adverse pressure gradient is achieved with a turbulent boundary layer. Thus, for such cases, although transition is not imposed the final

geometric shape is the one which causes a rapid transition from a laminar to a fully turbulent boundary layer.

An interesting situation occurs when transition is imposed near the leading edge and the flight condition is not severe enough to produce separation of the flow. In this case, the optimization process looks for the lowest pressure drag coefficient. However, due to a lack of more severe drag generating phenomena, all geometric shapes yield almost the same level of drag coefficient. Hence, for such cases, the optimization process might converge to any geometric shape.

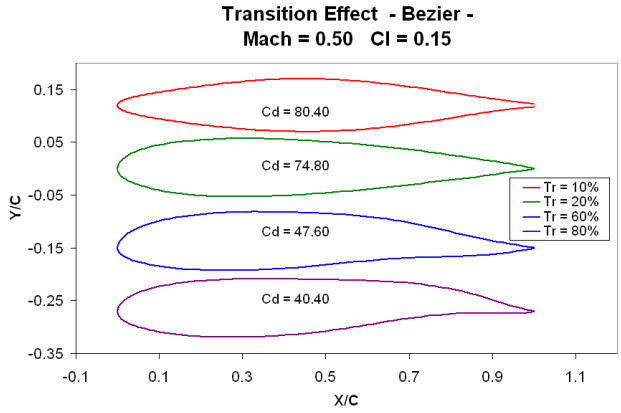

Fig. 22. Different optimized profile shapes obtained by Bezier parametrization for different position of the boundary layer transition.

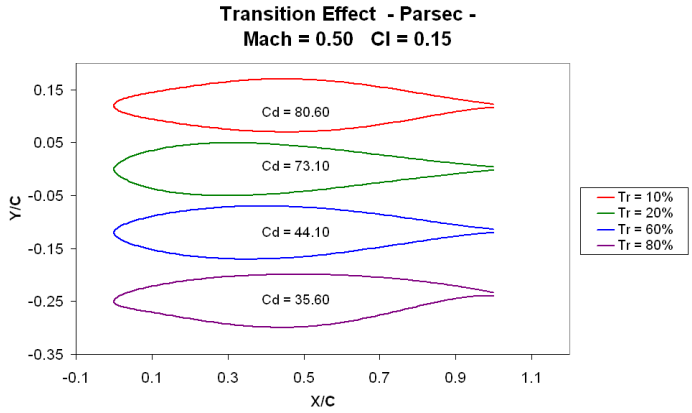

Fig. 23. Different optimized profile shapes obtained by Parsec parametrization for different position of the boundary layer transition.

Figure 22 shows different airfoil shapes, obtained using the Bezier parametrization, for different transition positions. The optimization is performed for Mach number of 0.50 and lift coefficient of 0.15. Similarly, Fig. 23 presents the different airfoil shapes, obtained with the Parsec parametrization, for the same previous flight condition. The objective function in both cases is the drag coefficient.

8. Conclusions

The construction of the framework has proven to be a very complex task of coupling different tools. Part of this complexity comes about from the use of compiled languages and scripts to implement the connection. However, there is a substantial gain in performance, with the present implementation, when compared to visual languages, such as those available from commercial solutions.

The genetic algorithms implemented for the present framework are validated with the use of well-known analytical test functions, and, afterwards, with the aid of the ModeFrontier commercial software. This is an important step for the assurance of the reliability of the present framework for its use in the future in actual industrial environments.

The performed simulations have shown that, depending on the approach used during the optimization process, different results can be obtained. Part of this variability of solutions is attributed to the aspects which have been investigated in greater detail here, namely, the geometric parametrization, location of boundary layer transition, the number of individuals in a population, and solver fidelity, although this last subject clearly deserves further study. The most significant effects in the optimization results are brought about by the geometric parametrization and the location of boundary layer transition. Knowledge acquired with the present study has led to other questions, particularly the effect of more severe flight conditions.

9. Acknowledgments

The authors acknowledge the partial support of Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, under the Project Research Grant No. 312064 /2006-3. Further support was also provided by Fundação

de Amparo à Pesquisa do Estado de São Paulo, FAPESP, under Process No. 2004/16064-9.

References

[1] Antoine, N., and Kroo, I., "*Framework for Aircraft Conceptual Design and Environmental Performance Studies*", *AIAA Journal*, Vol. 43, No. 10, October 2005.

[2] Reklaitis, G. V., Ravindran, A., and Ragdell, K. M., *Engineering Optimization Methods and Applications*, Wiley, New York, 1983.

[3] Mokhtar, B. S., and Shetty, C. M., *Nonlinear Programming. Theory and Algorithms,* Wiley, New York, 1979.

[4] Schrijver, A., *Theory of Linear and Integer Programming*, Wiley, New York, 1998.

[5] Hestenes, M. R., and Stiefel, E., "Methods of Conjugate Gradients for Solving Linear Systems," *Journal of Research of the National Bureau of Standards*, Vol. 49, 1952, No. 6, pp.409-436.

[6] Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

[7] Adami, C., *Introduction to Artificial Life*, Springer-Verlag, New York, 1998.

[8] Dorigo, M., and Di Caro, G., "The Ant Colony Optimization Meta-Heuristic," In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, Chapter 2, McGraw-Hill, London, UK, 1999, pp. 11-32.

[9] Deb, K., *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, New York, 2003.

[10] Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York, 1989.

[11] Mortenson, M., E., *Geometric Modeling*, Wiley, New York, 1985.

[12] Kulfan, B., M., and Bussoletti, J., E., "Parametric Geometry Representation for Aircraft Component Shapes," AIAA Paper No. 2006-6948, *Proceedings of the 11th AIAA/AISMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, Virginia, Sept. 2006.

[13] Sobieczky, H., *Parametric Airfoils and Wings, Recent Development of Aerodynamic Design Mehodologies*, Inverse Design and Optimization, Vieweg & Sons Verlagsgesellschaft, Braunschweig/ Wiesbaden, Germany, 1999, pp. 72-74.

[14] Ansys, Inc, ICEM-CFD™, http://www.icemcfd.com/.

[15] Silva, A.L.F.S., Oliveira Neto, J.A., Antunes, A.P., Mendonça, M.T., Azevedo, J.L.F., and Silveira Neto, A., "Numerical Study of Two-Dimensional High-Lift Configurations Using the MSES Code," *Proceedings of the 18th International Congress of Mechanical Engineering – COBEM 2005*, Ouro Preto, MG, Brazil, Nov. 2005.

[16] Fluent 6.0 User's Guide, 2001, Fluent Inc.

[17] Metacomp Technologies, Inc, CFD++, http://www.metacomptech.com/.

[18] Harold Youngren, Aerocraft, Inc http://web.mit.edu/drela/Public/web/xfoil

[19] Drela, M., "A User's Guide to MSES 2.95," MIT Computational Aerospace Sciences Laboratory, Cambridge, MA, 1996.

[20] Karas, O. V., and Kovalev, V. E., BLWF 43 User's Guide, 2002.

[21] Zitzler, E., Deb, K., and Thiele, L., "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, Vol.8, No. 2, March 2000, pp. 173-195.

[22] Fonseca, C. M., and Fleming, P.J., "Multiobjective Optimization and Multiple Constraint handling with Evolutionary Algorithms – Part II: Application Example," *IEEE Transactions on System, Man, and Cybernetics: Part A: Systems and Humans*, Vol. 28, Jan. 1998, pp.38-47.

[23] Schaffer, J.D., "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms," *Proceedings of the First International Conference on Genetic Algorithms*, J.J Grefensttete, editor, Hillsdale, New Jersey, July 1987, pp. 93-100.

[24] Esteco, Inc, Mode-Frontier, http://www.esteco.com/.

[25] Goldberg, D.E., Deb, K., and Clark, J.H., "Genetic Algorithms, Noise, and the Sizing of Populations," *Complex Systems*, Vol.6, 1992, pp. 333-362.

Copyright Statement