Multi–point Algorithm for Inverse Aerodynamic Design

V. M. Freitas Jr.
PME–EPUSP
vander.freitas@poli.usp.br

G. L. Oliveira
EMBRAER – Propulsion Systems Engineering
guilherme.oliveira@embraer.com.br

L. C. C. Santos
EMBRAER – CFD Environmental Systems – IME– USP
luis.castro@embraer.com.br

E. V. Volpe
PME–EPUSP
ernani.volpe@poli.usp.br

Abstract. *Over the last two decades, CFD has played an increasingly important role in aerospace design. For it provides a very cost–effective means for one to analyze different configurations. More recently, the development of inverse design methods has opened up new possibilities. On combining those methods with CFD, one can specify design goals and search for configurations that meet them. Among these methods, the well–known Modified Garabedian Mcfadden (MGM) is an attractive alternative, for its simplicity and effectiveness. In essence, it compares the pressure distribution on control stations along the wing to target distributions. The location of those stations and the target distributions are both specified by the user. The method then changes those stations geometry so as to obtain the desired distributions. Wing lofting is gotten by interpolation between control stations. The procedure makes for a single point design, in that it allows one to meet requirements for a specific flight condition, such as cruise. However, that goal is often achieved at the expense of some performance degradation under off–design conditions. This paper discusses a method for multi–point design, which enables one to seek a geometry that meets a weighed combination of flight conditions. Application examples of dual-point airfoil designs are presented and discussed*

keywords: *MGM, Inverse Design, Multi-point*

1. Introduction

The enormous progress that has been achieved in computational fluid dynamics (cfd) over the last two decades has raised it to a crucial role in the aerospace industry (MacCormack, 1993; Santos, 1993). Flow simulation codes are used in virtually all phases of aircraft design, from conception to components design. As Jameson, 1997, notes, flow simulations now play a complementary role to that of wind tunnel tests in the aerospace industry.

In particular during the conceptual design phase, cfd makes for a very cost–effective means of analyzing different configurations, when compared to experimental testing. Even so, the cumulative costs of complex flow simulations can become prohibitive. That is the case when a trial–and–error approach is adopted to explore a large variety of possible designs, and the corresponding flow simulations build up in number. In addition, the odds of success of a such an approach rely heavily on the designer's experience, and they grow dimmer as the configurations variety increases, or as more elaborate conditions are imposed on the design.

Inverse design methods can have a very positive effect in this situation. On combining them with cfd codes, one makes for a more efficient way to explore the design space. In general, these methods compare actual surface pressure distributions to target distributions. Then they estimate geometry changes that should lead to the desired results. A number of inverse design methods have been proposed in the last two decades. Among them, the so–called Modified Garabedian–MacFadden method (MGM) remains an attractive alternative, for its simplicity and reliability. More recently, a hybrid formulation of MGM has been developed by Santos, 1993, which enables one to impose geometric constraints on the inverse design. Rho *et al.* (H. J. Kim and Rho, 1999; Kim and Rho, 1997) then made use of that formulation to give MGM the capability of multi–point inverse design.

The multi–point capability can be extremely advantageous in the context of inverse design. As the literature reports, wings that are designed for a specific flight condition may experience some performance degradation, when subject to different flight regimes. The rationale behind Rho's work is to pursue a weighed mix of target distributions, which represent different flight conditions, as opposed to just one.

The purpose of this paper is to discuss an implementation of MGM multi–point inverse design that is based on Rho's method. It presents airfoil design applications, in which the algorithm is used to pursue two target pressure distributions. The aim is to inverse design airfoils that are subject to prescribed combinations of two distinct flight conditions.

2. The Modified Garabedian–MacFadden Method

The conceptual foundations of MGM lie in the classic 2–D wavy–wall problem. In effect, it is based on the linearized small disturbance potential flow solution to that problem. It treats the airfoil upper and lower sides separately, and considers each one of them as a portion of a wavy–wall. For each portion, MGM compares the actual pressure coefficient distribution to a target distribution, which is previously specified by the user. On the basis of the difference between the two, it estimates geometry changes that should have the airfoil attain the desired distribution. Convergence is achieved when that difference falls below a prescribed level. The geometry changes are estimated by the model equation,

$$A(\delta z) + B\frac{\partial(\delta z)}{\partial x} - C\frac{\partial^2(\delta z)}{\partial x^2} = C_{pt} - C_p \tag{1}$$

where C_p represents the actual pressure coefficient distribution, C_{pt} stands for the corresponding target distribution. The x coordinate runs along the airfoil chord, from its leading edge to the trailing edge, and δz represents the airfoil contour changes. The coefficients A, B and C are arbitrary constants that are picked so as to make the inverse design cycles stable and, possibly, to accelerate convergence (Santos, 1998; Santos, 1999). Homogeneous Dirichlet boundary conditions are imposed at both ends of the domain: the leading and trailing edges.

$$\delta z|_{x=0} = \delta z|_{x=c} = 0 \tag{2}$$

The symbol c represents chord length. In the literature the RHS of eq. (1) is often replaced by the difference between squared velocity distributions $q_t^2 - q^2$, where q represents the dimensionless perturbation velocity in the flow direction $q \equiv u/U_\infty$ (Silva and Sankar, 1992). Both forms of the model equation are equivalent, since the relation between q^2 and C_p for steady potential flow is given by

$$q^2 = 1 - \frac{2}{(\gamma-1)M_\infty^2}\left[\left(1 + \frac{\gamma M_\infty^2}{2}C_p\right)^{\frac{(\gamma-1)}{\gamma}} - 1\right] \tag{3}$$

where M_∞ represents the far–field Mach number and γ is the specific–heat coefficients ratio $\gamma = c_p/c_v$. Owing to its conceptual foundations, MGM is suitable for the transonic flow regime.

The only piece of information from the flow solution the method actually needs is the C_p distribution. In that lies one of the most important features of MGM, which is its independence from the flow solver. It implies that an inverse design loop can be assembled on coupling the MGM routine with geometry and mesh generators, a flow solver and a post–processing unit to compute C_p distributions, as is shown in fig. 1

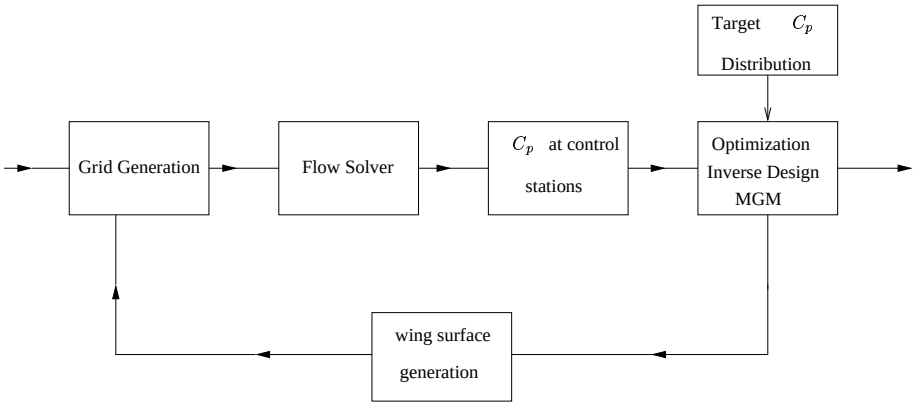

Figure 1: Inverse design loop. The MGM routine is nested in a loop along with geometry and mesh generators, a flow simulation code and a post–processing routine to evaluate C_p distributions

On accounting for its 2–D character, the use of MGM for wing design involves specifying a set of control stations along its span, as is depicted in fig. 2.a. The method is then applied to each one of them separately, thus changing their geometry— fig. 2.b. The wing lofting is gotten by surface interpolation between control stations.

Naturally, the fact that MGM is based on 2–D linearized potential flow raises important questions as to how effective it could be in 3–D applications, such as wing design. In addition to that, there is the question regarding the limitations of such a flow model, specially when the flow simulation may involve more complex models— such as the full Navier–Stokes equations, for example.

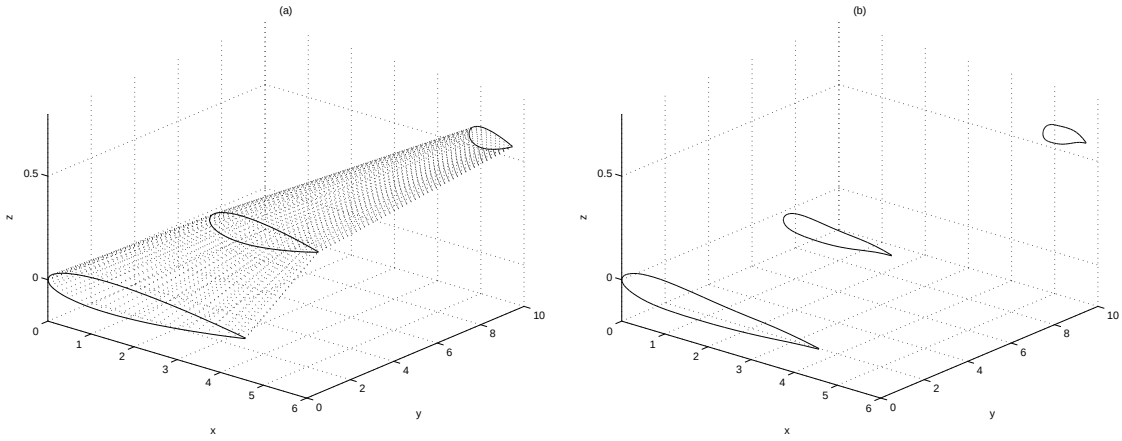

Figure 2: Schematic representation of wing inverse design: (a) original geometry showing three control stations. (b) geometry of control stations after 50 inverse design cycles. The wing tip region is not shown here.

As a matter of fact, both dimensionality and model limitations may indeed hinder or even prevent inverse design cycles from converging. That is likely to happen in regions where the flow field is strongly 3–D or where there is boundary layer separation.

On the other hand, for pressure driven flows without separation regions, the success of MGM is widely acknowledged in the literature (Bartelheimer, 1995; N. Hirose and Kawai, 1987; Santos, 1993; Silva and Sankar, 1992). Good results are reported even for flow fields that are moderately 3–D. That happens because the method responds primarily to pressure field variations. When those are dominant, then MGM can capture the most relevant part of the flow physics.

As a first account of our studies on the subject, this paper focuses on implementation issues, and the applications are limited to 2–D airfoil design tests. Wing design and other 3–D applications have been left for a later paper on the subject.

2.1. Numerical Implementation

Owing to its linearity, eq. (1) could be solved analytically, on making use of Green's function method for instance. However, finite differences solutions are more appropriate for the hybrid formulation and the multi–point algorithm, as will be shown below.

Previous works have shown that one can get sufficiently accurate solutions on nonuniform grids (Santos, 1993). Under such conditions, the best results are gotten on approximating the first order derivative by a backward difference operator and the second derivative by a central one (Volpe, 2004). On making use of such differencing scheme, one can easily cast the MGM equation in the form of a tridiagonal linear system (Hirsch, 1994).

$$\mathbf{M}\delta z = \mathbf{R} \tag{4}$$

where $\mathbf{M}$ is the finite difference operator matrix, δz is a vector of point–wise geometry changes (δz_i) and $\mathbf{R}$ represents the RHS of the MGM equation, which can be either $(C_{pti} - C_{pi})$ or $(q_{ti}^2 - q_i^2)$. The above set (4) can certainly be tackled by Thomas algorithm (W. H. Press and Flannery, 1992), and that is precisely what is done in the conventional implementation of MGM.

On the other hand, the hybrid formulation does not seek to solve the set. Instead, it attempts to find δz that minimizes the error with respect to the actual solution, while satisfying a set of constraints. Hence it gives MGM the character of an optimization method. The finite differences form (4) makes it simple for one to approach the problem that way, and the overall framework allows for the imposition of geometric constraints (Santos, 1993). The error function is defined so as to ensure that only the actual solution results in the absence of constraints. On abiding by the usual notation and defining the variable $\mathbf{X} \equiv \delta z$ as an estimate of geometry corrections, the error function is defined as

$$\mathrm{F} = \frac{1}{2}\left|\mathbf{MX} - \mathbf{R}\right|^2 \tag{5}$$

It is a quadratic objective function, for which the gradient and Hessian are both given in closed form, by exact expressions with low computational cost

$$\nabla \mathrm{F}(\mathbf{X}) = \mathbf{M}^T(\mathbf{MX} - \mathbf{R}) \tag{6}$$

$$\mathbf{H}(\mathbf{X}) = \mathbf{M}^T\mathbf{M} \tag{7}$$

The above expressions for F and its gradient can be cast in an alternative form, which suits our purposes best

$$\mathrm{F} = \frac{1}{2}\mathbf{X}^T\mathbf{H}\mathbf{X} - \mathbf{R}^T\mathbf{M}\mathbf{X} - \frac{1}{2}\mathbf{R}^T\mathbf{R} \tag{8}$$

$$\nabla\mathrm{F} = \mathbf{H}\mathbf{X} - \mathbf{M}^T\mathbf{R} \tag{9}$$

The quadratic character of F, along with the fact that both its gradient and Hessian are readily available, seem to favor the use of a second order minimization method. Yet, our objective is to introduce the multi–point algorithm in the exact same implementation that allows for the imposition of geometric constraints. So that the former can be seen as a natural extension of the latter. The imposition of constraints, in turn, gets very simple and robust when use is made of two well–known algorithms: The Preconditioned Conjugate Gradient (PCG) and the Augmented Lagrange Multiplier (ALM)— Volpe, 2004. Hence the combination PCG–ALM is the method of choice to tackle both the imposition of geometric constraints and the multi–point inverse design. The ALM algorithm is not relevant for the multi–point design and will not be presented here. The reader is referred to (Vanderplaats, 1984) for details.

In essence, the Conjugate Gradient (CG) algorithm takes an initial estimate of the solution $\mathbf{X}$, choses a search direction $\mathbf{S}$ that depends on ∇F, and performs an 1–D search for the extremum in that direction. On finding the extremum along $\mathbf{S}$, it restarts the process from that point on. The iterations go on until a local extremum of F is gotten. In cases where ∇F and $\mathbf{H}$ are known in closed form, one can compute the 1–D step size α so as to reach an extremum along $\mathbf{S}$ in a single step— since it implies that $\nabla\mathrm{F}\perp\mathbf{S}$ (Vanderplaats, 1984).

The number of iterations it takes for the above algorithm to converge is of the order of $\sqrt{\kappa(H)}\ln(\sqrt{2/\varepsilon})$. Where $\kappa(H)$ represents the Hessian spectral condition number and ε is the ratio between initial and final errors. In this application, both the difference operator matrix and the Hessian ultimately depend on grid spacing Δx_i. It can be shown that a severely nonuniform grid can cause κ to reach very high values, thus hampering convergence (Volpe, 2004).

A preconditioning algorithm can be used to circumvent the problem. In essence, it performs a similarity transformation on the CG equations. The procedure leads to a space where the Hessian is transformed into a matrix of lower κ, and where the extremum is actually sought. The inverse transformation then takes the results back into the original system (Axelsson and Barker, 1984). The CG equations can be fully integrated with those of the preconditioning algorithm to form the PCG algorithm. On defining $\mathbf{g}^q \equiv \nabla\mathrm{F}(\mathbf{X}^q)$, to simplify the notation, one gets

$$\begin{cases} \alpha_q = \frac{-(\mathbf{g}^q)^T.\mathbf{h}^q}{(\mathbf{S}^q)^T\mathbf{H}\mathbf{S}^q} \\ \mathbf{X}^{q+1} = \mathbf{X}^q + \alpha_q\mathbf{S}^{q+1} \\ \mathbf{g}^{q+1} = \mathbf{H}\mathbf{X}^{q+1} - \mathbf{R}\mathbf{M} \end{cases} \qquad \begin{cases} \mathbf{h}^{q+1} = \mathbf{C}^{-1}\mathbf{g}^{q+1} \\ \beta_q = \frac{(\mathbf{g}^{q+1})^T.\mathbf{h}^{q+1}}{(\mathbf{g}^q)^T.\mathbf{h}^q} \\ \mathbf{S}^{q+1} = -\mathbf{h}^{q+1} + \beta_q\mathbf{S}^q \end{cases} \tag{10}$$

The main difference between this and the original CG algorithm lies in the definition of vector $\mathbf{h}$, which is used to compute the search direction $\mathbf{S}$. It is gotten as a product between the gradient $\mathbf{g}$ and the inverse of the preconditioning matrix $\mathbf{C}$. The matrix $\mathbf{C}$, in turn, is a function of the Hessian $\mathbf{H}$ (Axelsson and Barker, 1984). For both algorithms CG and PCG, the first iteration search direction is taken as $\mathbf{S}^1 = -\mathbf{g}^1$.

Convergence is verified on the basis of two complementary criteria: the magnitude of the objective function gradient, and the magnitude of that function variation between consecutive iterations. The primary criterion is obviously the gradient magnitude, which should be the only one effective in the absence of constraints. However, the second criterion is essential in the presence of constraints. For the minimum of Φ can be at the boundary of the feasible region, in which case it may not be an actual extremum ($\nabla\Phi \neq 0$). Then the rationale is to set the criteria up in a hierarchy, so as to ensure the gradient prevails in all but those cases, where the constraints actually hamper convergence. That is accomplished on prescribing the objective function variation an accuracy level that is much more stringent than that of the gradient.

3. Multi–point Inverse Design

The method that was proposed by Rho *et al.* (H. J. Kim and Rho, 1999; Kim and Rho, 1997) involves a significant change in the objective function. As an extension to its original form, eq. (5) or eq. (8), the objective function now represents a weighed sum of quadratic error terms with respect to several target C_p distributions. Each target distribution represents a distinct flight condition. In particular for the case of two target distributions, C_{pt_a} and C_{pt_b}, one has

$$\begin{aligned} \mathrm{F} &= \frac{\omega}{2}\left|\mathbf{M}\mathbf{X} - \mathbf{R}_a\right|^2 + \frac{(1-\omega)}{2}\left|\mathbf{M}\mathbf{X} - \mathbf{R}_b\right|^2 \\ &= \frac{1}{2}\mathbf{X}^T\mathbf{H}\mathbf{X} - \left[\omega\mathbf{R}_a^T + (1-\omega)\mathbf{R}_b^T\right]\mathbf{M}\mathbf{X} - \frac{1}{2}\left[\omega\mathbf{R}_a^T\mathbf{R}_a + (1-\omega)\mathbf{R}_b^T\mathbf{R}_b\right] \end{aligned} \tag{11}$$

where ω is the weighing factor, which is previously set by the user. On making use of the same notation as in eq. (4), the vectors $\mathbf{R}_a$ and $\mathbf{R}_b$ represent the RHS of the MGM equation with respect to each target distribution, a and b respectively.

The expanded form of (11) shows clearly its quadratic character and its similarity to (8). It can be seen that the terms within square brackets in the former are analogous to the $(\mathbf{R}^T)$ and $(\mathbf{R}^T\mathbf{R})$, which appear in the latter. The gradient is directly obtained from (11), and it reads

$$\nabla \mathrm{F}(\mathbf{X}) = \mathbf{H}\mathbf{X} - \mathbf{M}^T[\omega\mathbf{R}_a + (1-\omega)\mathbf{R}_b] \tag{12}$$

which is, again, entirely similar to (9). The term within square brackets plays a role in the expression that is analogous to that of $(\mathbf{R})$ in (9). The Hessian, in turn, remains the same as in eq. (7).

The similarity between this and the hybrid formulation allows one to make use of the same procedures for both. It only implies replacing the terms that involve $\mathbf{R}$ in the hybrid formulation, by the corresponding weighed combinations of $\mathbf{R}_a$ and $\mathbf{R}_b$, as was pointed out above. The same algorithms PCG–ALM can be used, and all capabilities of the hybrid formulation are kept, including the imposition of geometric constraints.

Only, the loop must be changed to allow for the simulation of distinct flight conditions, as opposed to just one. The post–processing routine should also generate several C_p distributions, corresponding to all control stations at each flight condition. Other than for those changes, the loop basic structure remains the same, as is depicted in fig. 3.

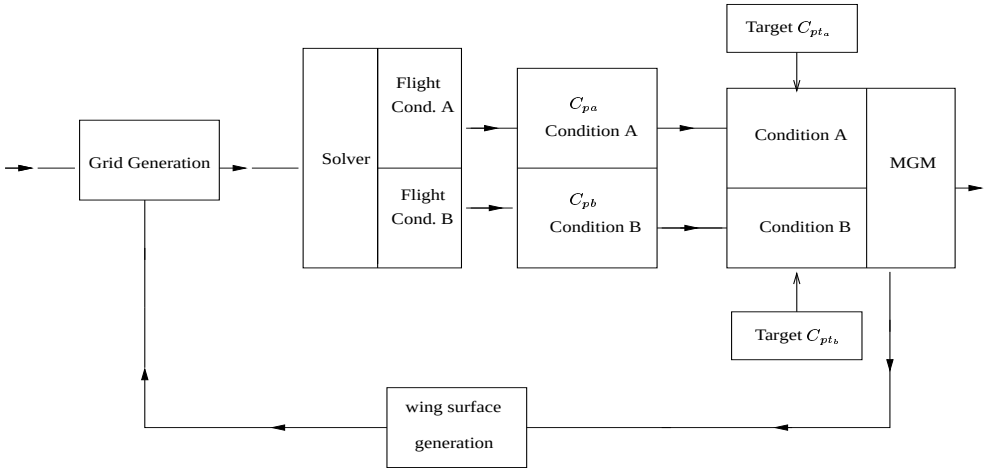

Figure 3: Multi–point inverse design loop. Two distinct flight conditions are simulated (a and b). For each one of them, the post–processing routine computes C_p distributions at the control stations k: C_{pa_k} and C_{pb_k}. The Multi–point MGM routine compares each C_p to a corresponding target distribution.

The same convergence criteria that are used for the hybrid remain valid for the multi–point design. At least in principle, if all the target C_{pt} could be realized by a single geometry (under corresponding flight conditions), then the algorithm would converge to $\nabla \mathrm{F} = 0$. However, it is more likely that no single geometry can meet all C_{pt} exactly. In that situation, the method would seek the contour that represents the best approximation to the various C_{pt}. Therefore, convergence would rather be verified by the variation of F than by its gradient.

The same reasoning could be pursued to extend the method for more than two flight conditions. The expressions for F and its gradient, (11) and (12), can be re–derived for a larger set of C_{pt}, and the loop could be changed accordingly. The problem with the extension is that the odds of success are bound to diminish. As the number of target distributions increases the likelihood of finding a geometry that meets all of them should become lower.

4. Airfoil Application Tests

Validation tests were performed on taking a known airfoil as the starting point and picking the targets C_{pt_a} and C_{pt_b} as distributions that correspond to another known airfoil, under flight conditions A and B. The procedure ensures that there is a single known geometry that realizes both target distributions.

Application tests, on the other hand, implied picking C_{pt_a} and C_{pt_b} as distributions of two distinct airfoils, under the two flight conditions. Some application test results are presented and discussed below. In these tests condition A involves $M_\infty = 0.7$, $Re = 10^6$ and zero angle of attack, while condition B involves the same Reynolds number and angle of attack, but $M_\infty = 0.9$.

The first test was done on taking Onera M6 as the original geometry, picking C_{pt_a} as that from Rae 2822 and C_{pt_b} as the one corresponding to Naca 0012, under conditions A and B, respectively. The weighing factor was set up to $\omega = 0.5$. Results are shown in figs. 4 and 5. The corresponding estimates for lift, drag and pitching moment coefficients are presented in table 1, 6th column. To some extent, the resulting airfoil approaches the target C_p distributions under both

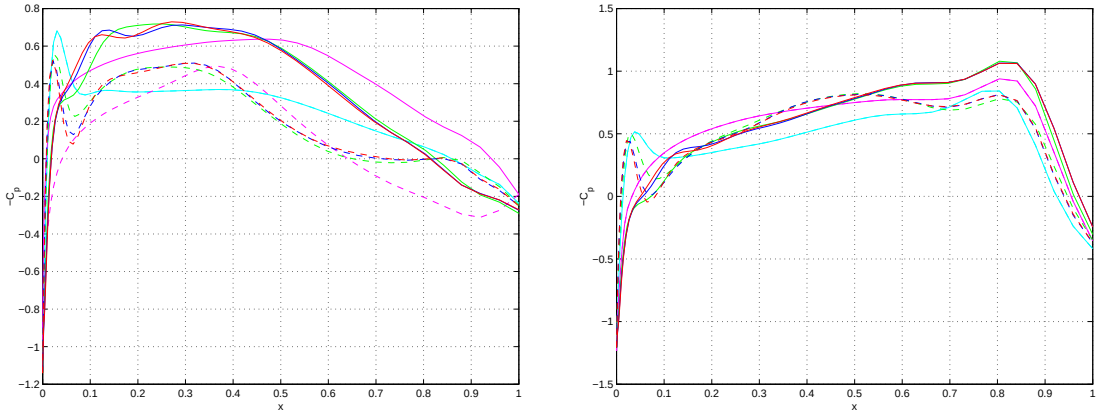

Figure 4: From Onera M6 to Rae 2822 at $M_a = 0.7$ and Naca 0012 at $M_b = 0.9$ ($\omega = 0.5$). Right C_{pa}; left C_{pb}. Upper side, solid line; lower side, dashed line. Both C_{pt}, magenta. Inv. des. cycles: Cyan, 1st; green, 2nd; 20th, blue; 30th, red.

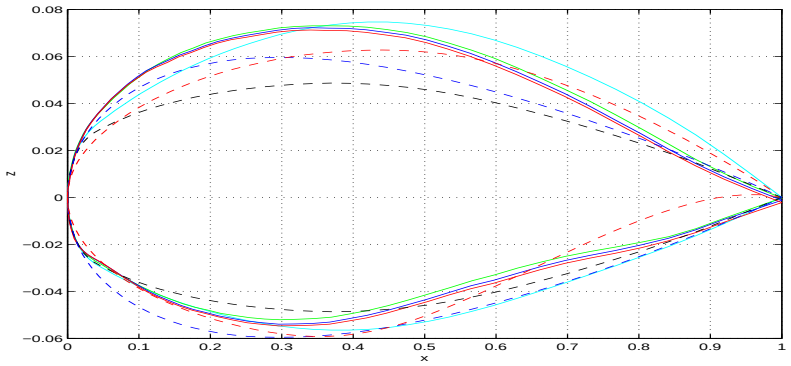

Figure 5: From Onera M6 to Rae 2822 at $M_a = 0.7$ and Naca 0012 at $M_b = 0.9$ ($\omega = 0.5$). Dashed lines, airfoils: Onera M6, black; Rae 2822, red; Naca 0012, blue. Solid lines, cycles: Cyan, 1st; green, 10th; blue, 20th; red, 30th.

flow conditions. From C_{pt_a}, it captures the main trends on the mid–portion, but departs from it at the leading and trailing edge regions. With regard to C_{pt_b} it seems to show better agreement, except for the leading edge neighborhood on the lower side. The proximity of the 20th and 30th cycles indicates that convergence was gotten on the basis of the objective function variation. Table 1 shows that C_l and C_m become non–zero, as a result of asymmetry, and C_d grows under both conditions. But they do not match the values of their targets exactly.

A test was done with the same targets, but on choosing another airfoil to start with, the asymmetric Naca 23012. At first, the weighing factor ω was assigned the same value 0.5. The test was designed to ascertain to what extent the initial geometry could have a bearing on the results. Figures 6 and 7 show that these results are similar to those of the previous test. It seems that the differences between them could be attributed to convergence related issues. They should grow smaller with increasing accuracy. Coefficient values in the 7th column of table 1 seem to corroborate that finding.

Another test was done with the same geometry and targets as before, but on setting ω up to 0.8. The objective was to assess inverse design sensibility to variations of that parameter. The results for C_p and geometry are presented in figs. 8 and 9. Indeed, they show significant differences when compared to the previous figs. 6 and 7. It appears that a better agreement has been obtained with respect to C_{pt_a}, as a result of the higher ω. Only, the trailing edge region has a distinctive shape that seems to combine influences of both Rae 2822 and Naca 0012. It certainly has strong bearings on the pressure recovery under both flow conditions.

5. Conclusions

The above results are only illustrative of the prospective applications. Yet they represent an attempt to highlight some of the most relevant features of the multi–point algorithm. Such is the case with the independence of the initial geometry, the convergence criteria, and the sensibility of the results to the weighing parameter. However, they hint at an underlying aspect of utmost importance. That is, convergence implicitly depends on whether the various target distributions can be realized by a single geometry. The answer to that question is usually not known beforehand in the applications.

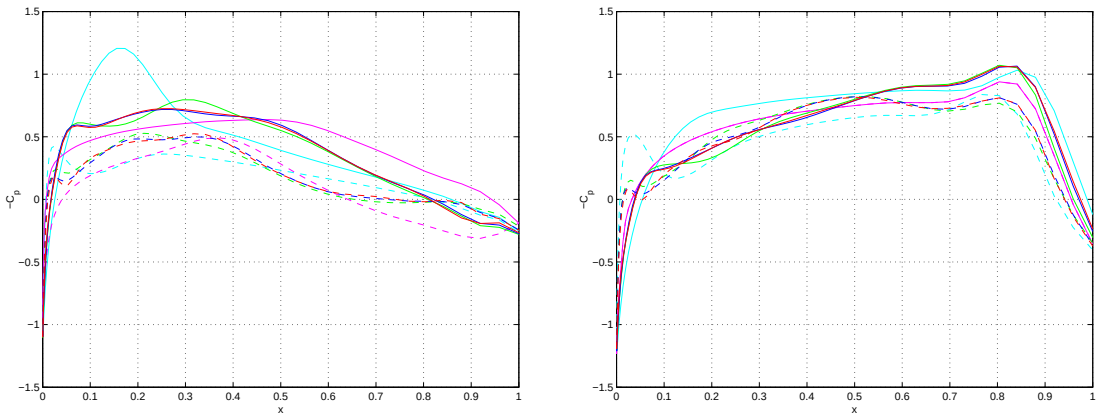

Figure 6: From Naca 23012 to Rae 2822 at $M_a = 0.7$ and Naca 0012 at $M_b = 0.9$ ($\omega = 0.5$). Right C_{pa}; left C_{pb}. Upper side, solid line; lower side, dashed line. Both C_{pt}, magenta. Inv. des. cycles: Cyan, 1st; green, 2nd; 20th, blue; 30th, red.

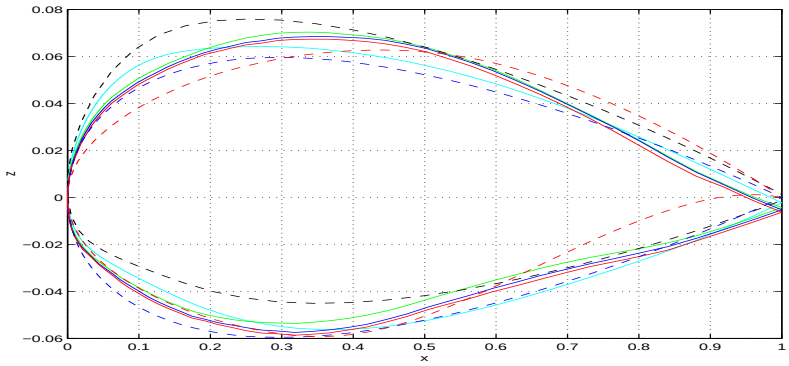

Figure 7: From Naca to Rae 2822 at $M_a = 0.7$ and Naca 0012 at $M_b = 0.9$ ($\omega = 0.5$). Dashed lines, airfoils: Onera M6, black; Rae 2822, red; Naca 0012, blue. Solid lines, cycles: Cyan, 1st; green, 10th; blue, 20th; red, 30th.

Table 1: Lift, drag and pitching moment coefficients – C_m with respect to LE

Airfoil	Onera M6	Naca 0012	Naca 23012	Rae 2822	Res. 1	Res. 2	Res. 3
C_{la}	0.0	0.0	0.2270	0.3515	0.1676	0.1796	0.2866
C_{da}	0.0032	0.0029	0.0044	0.0026	0.0037	0.0033	0.0036
C_{ma}	0.0	0.0	-0.0680	-0.1903	-0.0653	-0.0661	-0.1359
C_{lb}	0.0	0.0	0.2062	0.1689	0.1066	0.0854	0.1093
C_{db}	0.0527	0.0926	0.0997	0.1055	0.0711	0.1061	0.1067
C_{mb}	0.0	0.0	-0.1412	-0.1764	-0.0749	-0.0777	-0.1176

6. Acknowledgments

The authors are grateful for the support provided by EMBRAER and FAPESP, under grant no. 2000/13768–4. Moreover, the first author (V. M.) gratefully acknowledges the support provided by FAPESP, under grant no. 04/03015–0.

7. References

Axelsson, O. and Barker, V. A., 1984, “Finite Element Solution of Boundary Problems: Theory and Computation”, Computer Sceince and Applied Mathematics, Academic Press, Inc., Cambridge, MA, 1st edition.

Bartelheimer, W., 1995, An Improved Integral Equation Method for the Design of Transonic Airfoils and Wings, Published by the AIAA, Inc. with permission.

H. J. Kim, C. K. and Rho, O. H., 1999, Multipoint Inverse Design Method for Transonic Wings, “Journal of Aircraft”, Vol. 36, No. 6, pp. 941–947.

Hirsch, C., 1994, “Numerical Computation of Internal and External Flows”, Vol. I of “Wiley Series in Numerical Methods

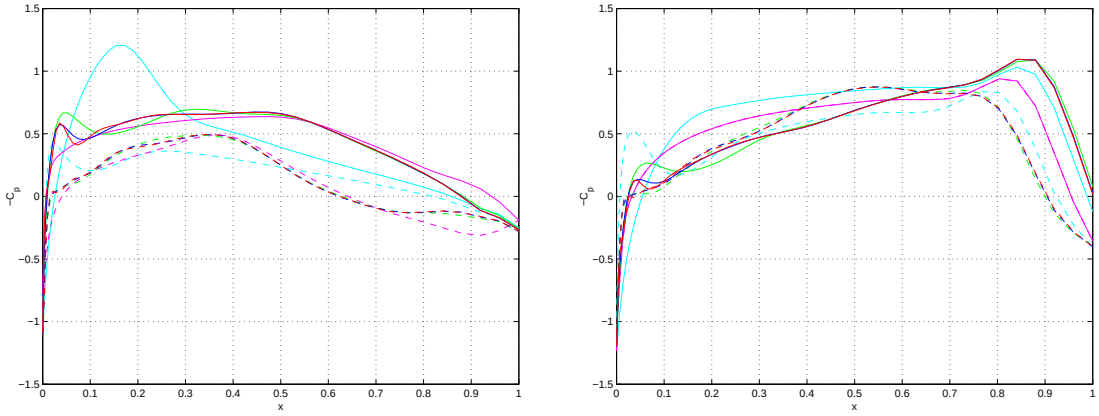

Figure 8: From Naca 23012 to Rae 2822 at $M_a = 0.7$ and Naca 0012 at $M_b = 0.9$ $(\omega = 0.8)$. Right C_{pa}; left C_{pb}. Upper side, solid line; lower side, dashed line. Both C_{pt}, magenta. Inv. des. cycles: Cyan, 1st; green, 2nd; 20th, blue; 30th, red.

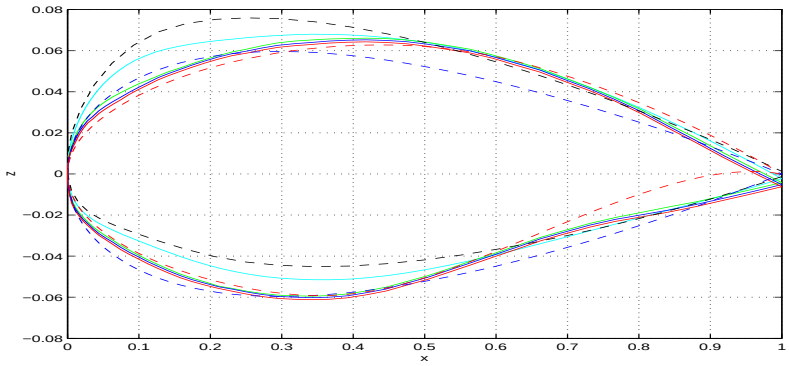

Figure 9: From Naca 23012 to Rae 2822 at $M_a = 0.7$ and Naca 0012 at $M_b = 0.9$ $(\omega = 0.8)$. Dashed lines, airfoils: Onera M6, black; Rae 2822, red; Naca 0012, blue. Solid lines, cycles: Cyan, 1st; green, 10th; blue, 20th; red, 30th.

in Engineering", John Wiley & Sons, NY, 1st edition, Fundamentals of Numerical Discretization.

Jameson, A., 1997, Re–Engineering the Design Process through Computation, "35th Aerospace Sciences Meeting & Exhibit", Reno, NV. American Institute of Aeronautics and Astronautics, AIAA, AIAA-97-0641.

Kim, H. J. and Rho, O. H., 1997, Dual–Point Design of Transonic Airfoils Using the Hybrid Inverse Optimization Methods, "Journal of Aircraft", Vol. 34, No. 5, pp. 612–618.

MacCormack, R., 1993, A perspective on a Quarter Century of CFD Research, "11th AIAA Computational Fluid Dynamics Conference", Orlando, FL. American Institute of Aeronautics and Astronautics, AIAA, AIAA-93-3291.

N. Hirose, S. T. and Kawai, N., 1987, Transonic Airfoil Design Procedure Utilizing a Navier-Stokes Analysis Code, "AIAA Journal", Vol. 25, No. 3, pp. 353–359.

Santos, L. C. C., 1993, "A Hybrid Inverse Optimization Method for Aerodynamic Design of Lifting Surfaces", PhD thesis, Georgia Institute of Technology.

Santos, L. C. C., 1998, Modelling Auxiliary Geometric Equations for Inverse Design Metods, Published by the AIAA, Inc. with permission, AIAA98-2405.

Santos, L. C. C., 1999, Convergence Acceleration Strategies for an Inverse Design Method, Published by the AIAA, Inc. with permission, AIAA99-0184.

Silva, D. H. and Sankar, L. N., 1992, An Inverse Method for the Design of Transonic Wings, "1992 Aerospace Design Conference", number 92–1025 in proceedings, pp. 1–11, Irvine, CA. AIAA.

Vanderplaats, G. N., 1984, "Numerical Optimization Techniques for Engineering Design: With Applications", Series in Mechanical Engineering, McGraw-Hill, N.Y., 1st edition.

Volpe, E. V., 2004, A11 – Inverse Aerodynamic Design Module, Technical Report, EMBRAER – FAPESP – EPUSP, SP, Research Project: Advanced Applications of Computational Fluid Dynamics to High Performance Aircraft.

W. H. Press, S. A. Teukolsky, W. T. V. and Flannery, B. P., 1992, "Numerical Recipes in C: The Art of Scientific Computing", Cambridge University Press, New York, NY, 2nd edition.