<http://www.diva-portal.org>

Preprint

This is the submitted version of a paper presented at *53rd AIAA Aerospace Sciences Meeting*.

Citation for the original published paper:

Mengmeng, Z., Rizzi, A., Nangia, R. (2015)

Transonic airfoil and wing design using inverse and direct methods.

In: *53rd AIAA Aerospace Sciences Meeting* Kissimmee, Florida

53rd AIAA Aerospace Sciences Meeting

<http://dx.doi.org/10.2514/6.2015-1943>

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-168150>

Transonic Airfoil and Wing Design Using Inverse and Direct Methods

M. Zhang*, A. Rizzi†

Royal Institute of Technology (KTH), 10044 Stockholm, Sweden

R. Nangia‡

Nangia Aero Research Associates, Bristol BS9 2DA, UK

A hybrid inverse/direct-optimization method for subsonic/transonic airfoil and wing shape design is presented with application to a range of airfoil and wing cases, in preparation for the test cases defined for the Special Session of SciTech 2015. The method is hybrid in the sense that it combines the traditional inverse design technique with a gradient-based procedure to approach the optimum aerodynamic surface. This paper emphasizes the first part, the development of SCID, the Surface Curvature Inverse Design method, the theory upon which it is based, including many of the details involved with shocks, smoothing and cross flow. The application of SCID to wing design poses many challenges, and how they are met is discussed in the context of a number of inverse design test cases for airfoils and wings. But it also includes results from the adjoint optimization and compares them. The procedure works well for airfoils and the twist optimization for wings. The real benchmarks for our hybrid approach are the three Optimization Discussion Group design problems. Solutions are presented for the drag minimization of the airfoil test cases along with the wing twist optimization problem, and conclusions are drawn from the results obtained. A swept-back transonic wing is designed by SCID with encouraging results, showing that SCID works fine with wings. Work has started on the drag minimization of the CRM wing in transonic flight, and final results will be presented in a future paper.

I. Introduction

Aircraft design activities are concerned with the determination of designs that meet a priori specified performance features of the vehicle. The specified design objectives are traditionally met through an iterative process of analyses, evaluations and modifications of the design. In this sequential trial-and-error procedure, the designer must rely on experience, intuition and ingenuity for every re-design, and this makes aircraft design an exciting creative discipline. In practice, however, designers are often forced to depend on tried concepts to cut a path through an incomprehensible number of feasible designs, historically characterizing the process as a slow gradual improvement of existing types of concepts.

Current design processes in the aerospace industry rely on sequential decision making through the traditional route starting with conceptual design, going on to the preliminary design phase and then to the detailed design, wherein each subsequent phase sees a significantly reduced design freedom as a consequence of decisions frozen in the earlier phase. When interdisciplinary conflicts arise, as they will in a complex coupled system, current design practice relies on non-automated means to resolve these, which might lead to sub-optimal designs in a global sense.

*Ph.D. student, AIAA Senior Member, e-mail: mzha@kth.se

†Professor, AIAA Associate Fellow, e-mail: rizzi@kth.se

‡Consulting Engineer, AIAA Associate Fellow

A. MDO & Aerodynamic design approaches

Multidisciplinary Design Optimization, or MDO, combines analysis and optimizations in several individual disciplines with those of the entire system concurrently through formal mathematical processes. It puts into place a formal integrated system design process for better product quality by effectively exploiting the synergism of interdisciplinary couplings. MDO, as a discipline, itself comprises of many areas of research. It is “A methodology for design of complex engineering systems that are governed by mutually interacting physical phenomena and made up of distinct interacting subsystems (suitable for systems for which) in their design, everything influences everything else” (1).

One of the significant factors holding back the widespread adoption of MDO is its computational cost when the number of design variables becomes very large (*the curse of dimensionality*). The use of high-fidelity models can raise the cost from merely expensive to unbearable. Parallel computing helps, but cannot overcome computational inefficiency. The revolution in computing speed and memory capacity of digital computers together with persistent systematization of design methodology has led to tools for computational aircraft design (e.g. MDO) that aim at automation of the conventional design process through integration of numerical methods for analysis, sensitivity analysis and mathematical programming so that the best design in terms of a pre-defined criterion can be determined. Traditionally the process of selecting design variations has been carried out by trial-and-error, relying on the intuition and experience of the designer, the engineer in the loop. Increasing the level of automation by computational means has reduced, but not eliminated, the engineer-in-the-loop activities. The overall success of the design process depends heavily not only on reliability and accuracy of the computational methods but also on how well the designer has set his goals.

1. Aerodynamic Shape Optimization - Two approaches

Optimization of aircraft wings is not new. The thing that makes wings so hard to design is that their aerodynamics and structure are not just interdependent, they are variable. Computational aerodynamic design, one of the disciplinary subsets of the aircraft design process, aims directly at determining the geometrical shape of the aircraft hull that produces certain specified aerodynamic properties, with or without constraints on the geometry. Usually termed aerodynamic shape optimization (ASO), this is the subject of this paper. ASO is a very attractive technology because it replaces workable designs with optimal ones, and cuts down design times, thus enabling faster responses to the economic pressure of the marketplace.

For wing design there are generally two approaches: the inverse design (engineering-skill dependent) and the direct optimization design (mathematics-skill dependent). The former works by first finding a well-posed pressure distribution that fulfills the design requirements and then determining a geometry that yields this target pressure. Our previous SciTech paper (2) presented the SCID method as one example of this approach. The direct approach requires the user to define the cost function (usually the drag) along with the constraints, and then seeks the solution to the constrained optimization problem by mathematical algorithms for non-linear optimization. When the algorithm involves gradient searches, the sensitivities that indicate how to change the geometry in order to reduce the cost function can be computed also for very many design parameters by solution of an adjoint to the flow problem. The goal of this paper is to establish an algorithm/toolbox which, under user control, can employ both approaches, much in the manner of the LINDOP optimizer in the MSES(3) package but in both 2D and 3D cases.

II. Toolbox with Inverse and Direct Optimization Methods

The paper describes a hybrid scheme which combines inverse design with optimization of the aerodynamic shape based on the above as shown in Fig. 1. The “hybrid” means we use mathematical optimization with gradients produced by the adjoint technique in a loop together with inverse design that integrates the streamline curvature to produce the shape associated with the target pressure distribution to find the shape (right). One key point is that as the iteration proceeds, the engineer, with some insight from the direct optimization, can modify the current target pressure to guide the inverse design process.

The target pressure distribution on the wing planform is constructed by considering the span loads, isobar patterns, etc. (4), as well as other best practice guidelines provided by experience. Keeping the engineer in the loop emphasizes wing design rather than accurate solution of a (possibly not-so-well formulated) mathematical optimization task:

Figure 1. The feedback design loop

Engineer in loop to minimize drag by finding the best feasible target pressure distribution, for which a feasible shape can be found by inverse design.

The resulting surface curvature inverse design method called SCID from our previous work(2) is used for inverse design. Fig. 2 is re-produced here for convenience. It shows the SCID inverse design algorithm with some details of the design procedures. In SCID the CFD code MSES(3) is used for airfoils and EDGE^a is used for wings. The direct optimization design is carried out with the SU² (5) solver from Stanford University.

Figure 2. Flow chart for wing inverse design using SCID algorithm, re-produced from (2)

^awww.foi.se/en/Customer-Partners/.../Edge1/Edge/

A. Modal-Inverse design SCID: algorithms

The flow chart of SCID inverse design is shown in fig. 3. The SCID shape design is accomplished by a variant of the CDISC procedure developed by R.Campbell (6; 7). It connects streamline curvatures on the wing surface with pressure changes to iteratively modify an initial shape. It is combined with under-relaxation chosen to help convergence, and smoothing procedures to ensure a smooth surface and curvature.

To validate the methodology, it has been applied to symmetric and asymmetric airfoils and untwisted wings in subsonic and transonic inviscid flow with specified target pressure distributions (see our SciTech paper 2014(2)). The following sub-sections will show the recent updates on SCID, with improved inverse design algorithms so that it works well on the airfoils and swept/twisted wings by considering the target pressures.

Figure 3. Flow chart for wing design using SCID algorithm

1. Surface Curvature Equation SCE^b

The relation is derived from the normal component of the momentum equation for inviscid flow along the streamline on the wing surface as long as the flow is attached. Compressibility effects of local supersonic flow have been incorporated. The momentum equation along a streamline on the wing surface is

$$UU_s \mathbf{e}_s + U^2(c_n \mathbf{e}_n + c_g \mathbf{e}_t) + \nabla p / \rho = 0 \quad (1)$$

where $\mathbf{e}_s, \mathbf{e}_n, \mathbf{e}_t$ form the Darboux frame: the unit tangent, surface normal, and $\mathbf{e}_s \times \mathbf{e}_n$ vectors, s is the *arc-length*, and c_n and c_g are the *normal* and *geodesic* curvatures of the streamline. The normal component is:

$$c_n \rho U^2 + \frac{\partial p}{\partial n} = 0$$

from which we can derive a relation between small changes in curvature and pressure coefficient,

$$\Delta c_n = \frac{c_n + 1/L}{1 - C_p} \cdot \Delta C_p$$

where L is a length-scale for the pressure gradient. A stagnation point has $C_p = 1$ so it cannot be changed by local geometry modification, but its location can move.

The surface analogue of the Frenet-Serret formulas for the surface coordinates $\mathbf{y}(s)$ along the streamline is

$$\mathbf{y}_{ss} = c_n \mathbf{e}_n + c_g \mathbf{e}_t. \quad (2)$$

^bThis section is re-produced from (2) for completion.

Figure 4. Wing represented in the Darboux frame

For small change on the surface, it gives

$$\Delta \mathbf{y}_{ss} = \Delta c_n \mathbf{e}_n + c_n \Delta \mathbf{e}_n + \Delta c_g \mathbf{e}_t + c_g \Delta \mathbf{e}_t \quad (3)$$

where only the first term on the right hand side is kept. Note that the last two terms vanish for airfoils. The final result is the proportionality between changes in normal curvature and pressure coefficient, for unit chord-length, so curvature becomes non-dimensional,

$$\Delta c_n = F \Delta C_p \quad (4)$$

The F -coefficient was proposed by Barger and Brooks(8). Campbell(6) suggests the c_n -dependence $F = A(1 + c_n^2)^B$ to produce

$$\Delta c_n = A(1 + c_n^2)^B \Delta C_p \quad (5)$$

where A and B are adjustable constants. A is chosen as large as possible without creating divergence in the iteration. Reported values range from 0 to 0.5. Smaller values give slow convergence, larger values may cause divergence. The correct coefficients must be chosen as compromise between speed of convergence and robustness. An adjustment algorithm is applied to select A and B according to the status of convergence. With Δc_n from eqn. 5, the new shape $\mathbf{y}(s)$ is computed from the two-point boundary value problem.

$$\frac{d^2 \Delta \mathbf{y}}{ds^2} = \text{coeff} \Delta C_p \mathbf{e}_n, \quad \Delta \mathbf{y}(s_{0,\text{lower}}) = \Delta \mathbf{y}(s_{\text{max,upper}}) = 0 \quad (6)$$

The arc-length s starts from the trailing edge on the lower surface. The boundary conditions are applied to ensure a sharp and closed trailing edge. The section geometry is represented by point clouds vector $\mathbf{y}^i = (x, z)$, $i = 1, 2, \dots, N$, N is the number of total points of the airfoil.

2. Strong shock term^c

The convergence slows down with stronger shocks, in particular with local $M > 1.15$, so the relationship of C_p and airfoil curvature should be corrected when the flow becomes locally supersonic. Linearized theory for supersonic flow(9), over surfaces with small angles gives

$$C_p = 2 \frac{\tan \theta}{\sqrt{M_\infty^2 - 1}} \quad (7)$$

where θ is the airfoil surface angle. This can be rearranged to yield

$$\Delta C_p \mathbf{e}_n = \text{coeff} \cdot \frac{d \Delta \mathbf{y}}{ds} \mathbf{e}_n \quad (8)$$

SCID uses no modification.

^cThis section is re-produced from (2) for completion.

3. Solutions to SCE

The desired C_p -change is translated to a change in local curvature by a second-order Ordinary Differential Equation, *as if* the streamline is *chord-wise* (eqn. 4-10), from which the new airfoil shape can be integrated by

- i solving the second-order boundary value problem(2), or
- ii shearing the surface segments to change the curvatures as Campbell did in CDISC(6)

The first one has been discussed in our previous work(2) and it showed that it works well for airfoils. The second method is to shear the surface segment to change the curvature at a point, as Campbell did in his work(6). In this section the details of the three types of solutions to the problem are discussed.

FINITE DIFFERENCE The new shape(k+1) is computed by the old shape(k) plus the shape change $\Delta \mathbf{y}$:

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \Delta \mathbf{y} \quad (9)$$

The shape change $\Delta \mathbf{y}$ is computed by solving the nonlinear second-order BVP below by finite difference in second order accurate.

$$\frac{d^2 \Delta \mathbf{y}}{ds^2} = \text{coeff}(c_n) \cdot \Delta C_p \mathbf{e}_n, \quad \Delta \mathbf{y}(s_{0,\text{lower}}) = \Delta \mathbf{y}(s_{\text{max},\text{upper}}) = 0 \quad \text{on the old shape(k)} \quad (10)$$

The c_n , ΔC_p and the normal curvature $\mathbf{e}_n$ are computed from the old shape(k),

This method integrates from trailing edge via leading edge back to trailing edge, rotating the airfoil around *trailing edge* and re-scale it to make sure the leading edge is located at (0,0). The rotation angle α is the local angle of attack. No curvature discontinuity appears at the leading edge. However, the algorithm might produce self-intersecting profiles especially around the leading edge since the boundary condition only constrains the trailing edge, and it can not easily be avoided mathematically.

SHEARING THE SURFACE SEGMENTS It simply solves the relations algebraically, applied at each point i along both airfoil surfaces from the leading edge to the trailing edge, rotating polygon edges one after another by translating the endpoints while keeping its x constant. It allows to preserve the surface shape/curvature at other points. This is especially useful when we would like to preserve some particular geometric characteristics. The local curvature changes computed from the famous relation is following

$$\Delta c_n^i \mathbf{e}_n^i = F(c_n^i) \Delta C_p^i \mathbf{e}_n^i \quad \text{on the old shape k} \quad (11)$$

It can be translated by shearing the points aft of the current one (i) through an angle θ .

$$\theta^i = \Delta c_n^i \cdot (s^{i+1} - s^i) \quad (12)$$

With x keeping unchanged, shifting only the z by θ , we have

$$z_{k+1}^i = z_k^i + \Delta z^i(\theta^i, x^{i+1}, x^i), \quad (13)$$

The trailing edge of the upper surface is located at $\mathbf{y}(\text{TE}, \text{Upp}) = (1, \tan \alpha)$, the trailing edge of the lower surface will be rotated around leading edge to meet this point to ensure a closed trailing edge. This method can create discontinuity at leading edge, which will be treated by *smoothing*. Self-intersecting profiles may still develop, for instance if the profile near the trailing edge of the lower surface is very concave. When the upper and lower profiles intersect, we move the intersected part of the lower surface to preserve continuity. The resulting profile must be smoothed to avoid wiggles.

In this paper SCID uses the second method.

4. Geometry update

The new shape is developed by updating the geometry iteratively. The geometry is represented in *sumo*(10) and unstructured grids are generated by it. In order to avoid geometric failures and to get workable grids, geometry refinements must be done.

SMOOTHING The integration is sensitive to rounding errors and the computed pressure distribution is also sensitive to wiggles in the shape. Such short-wave wiggles are damped by smoothing. The chord-wise airfoil shape is smoothed by using a least-squares approximation of the generated points by two pieces of cubic Bezier curves on both upper and lower surfaces. The Bezier curve shapes may not be able to represent the optimal shape exactly, but the differences are generally very small between existing airfoils and their best approximation by Bezier curves, see e.g T.Melin (11). The leading edge discontinuity caused by the integration is smoothed out by setting the constraints on the control points. Note that the constraints of the control points on the second Bezier curve must be released to allow a cusp tail shape.

FOR WINGS There is a damping factor γ applied to the section-wise angle of attack α to update the local twist for one section (i), no matter which method is used.

$$\alpha_{i,\text{update}} = \gamma \cdot \alpha_i \quad (14)$$

The γ is chosen to be a constant for all the sections. From the author's experience a good under-relaxation $\gamma = 0.4$ for a wing with moderate to high aspect ratio is acceptable.

The local twist of each section may cause non-planarity. Equal-spaced sections are chosen span-wisely. To avoid geometric wiggles, the local twists of the sections are smoothed using a least-squares method to fit a quadratic curve by fixing the root and the tip twists.

5. User specified- C_p editing

The main challenge in the optimization process for the engineer is the definition of the target pressure distributions. The target C_p for the wing is chosen for each chord-wise section to ensure the wing has "straight isobars" (4). It is well known that elliptical wing loading lead to least induced drag for subsonic flow. Such constraints, as also constraints on lift and moment coefficients are easily incorporated into constraints on section-wise C_p by interpolation and integration. However, at transonic speed, the optimal loading is not known.

In the latest SCID there are two approaches to define, change and modify the target pressure distributions for each section of the wing. One is to input the target pressure distribution directly by user selected control points with the screen cursor for a n th-order B-spline curve. The other is to import existing pressure distributions, approximate them by B-splines and optionally modify them to define the target pressure.

Figure 5 shows an example of the estimated pressure distributions for RAE 2822 at Mach = 0.734 and $c_\ell = 0.824$ with 8 control points for the upper surface pressure and 7 control points for the lower surface pressure. The first and last points, at leading edge and trailing edge, for both surfaces coincide.

Figure 5. Pressure distribution represented by B-spline curves for RAE 2822 at Mach = 0.734 and $c_\ell = 0.824$

The resulting sectional profile is obtained from (one or several) SCID iterations, and local lift and span loading distributions as well as isobars are displayed in response. The engineer decides on modifications etc. for the next SCID iterations.

B. Direct mathematical optimization by SU²

In this work the direct optimization is carried out by the open-source Stanford University Unstructured Software SU² (5). SU² is a Reynolds-averaged Navier-Stokes solver for compressible, turbulent flows. Its C++ modules are embedded in a *Python* framework that is suited for constrained nonlinear optimization problems by calling *scipy.optimize*.

Figure 6. The direct optimization design loop using adjoint solver

The adjoint method was originally applied to aerodynamics by Jameson adapting ideas originally formulated by Lionson optimal control of systems governed by partial differential equations (PDEs). The adjoint equations can be conveniently formulated in a framework to calculate the sensitivity of a given objective function J (f for the RSM problem) to parameters α in a problem governed by the set of equations which can be represented by $G(U, \alpha) = 0$, where U is the solution.

The additional computational cost of solving the adjoint problem is generally of the order of one additional flow solution, and the adjoint variables represent the sensitivities of J to changes in all of the parameters that define the problem at every point in the domain. In contrast, though finite difference methods can also be used to find these sensitivities, they are in general significantly more expensive, requiring at least one additional flow solution per parameter in the domain. There are two main ways to characterize the adjoint approach, as a discrete method, in which the discretized governing equations are used to derive the adjoint equations, and as a continuous method, in which the adjoint equations are derived from the analytical PDEs. The adjoint equations for both approaches can be derived in two different ways, via the Primal-Dual Equivalence Theorem or via an optimization framework using Lagrange multipliers.

The discrete and continuous approaches are found to have relative advantages and disadvantages over each other. In theory a discrete method can handle PDEs of arbitrary complexity without significant mathematical development and can treat arbitrary functionals. However, this method requires the evaluation of discrete Jacobians, and there are two main ways to do this. The first is to analytically derive these terms from the discretized forms of the flow residuals and then develop code based on this, and the second is to use algorithmic Automatic Differentiation, either via source code transformation or operator overloading.

The former approach requires significant development, more than that generally required in the continuous method, while the latter is computationally expensive, with large memory storage requirements. In comparison, the continuous adjoint requires significant theoretical development but is better connected to the underlying physics and can be solved by a method independent of the flow solution scheme. However, it is more limited in the types of functionals and governing equations that can be treated, and the gradient calculated will differ more from that found by finite differencing. But as the mesh is refined, all three gradients, discrete, continuous and finite difference, should converge to the same limit.

III. Optimization Discussion Group Test Cases

The work here aims at establishing a robust approach to specifying the target pressure for the inverse design method, and thus improve upon the work reported in the previous SciTech paper (2). The enhanced SCID hybrid design will be demonstrated for both airfoil test cases and the induced drag minimization of rectangular wing.

In order to illustrate the pros and cons associated with the inverse design approach, we produce solutions to the airfoil test case with the direct design approach by comparing SU²(5) results with SCID. Thus the second goal is to investigate the design by direct approach with adjoint solver, and to compare experiences with these two approaches. The test case to be compared is the drag minimization of airfoils NACA0012 and RAE2822. The airfoils are parametrized by Hicks-Henne bump functions (12) and the wing is parametrized by Free-Form Deformation (13). The Hicks-Henne bumps used here are

$$f_k(x) = \left(x^{\frac{\log 0.5}{\log x_k}} \right)^4$$

with the max-points $x_k, k = 1, 2, \dots, n$ equally-distributed over $[0.5/n, 1 - 0.5/n]$, see fig. 7.

Figure 7. The Hick-Henne shape functions used for airfoil test cases

A. Drag Minimization of the NACA-0012 in Transonic Inviscid Flow

The drag is minimized at inviscid flow Mach number 0.85 at zero angle of attack, and the shape is required to contain the baseline NACA-0012 profile. The optimization problem is:

$$\begin{aligned} \text{min: } & c_d \\ \text{subject to: } & y \geq y_{\text{baseline}} \end{aligned} \tag{15}$$

where $y(x)$ is the upper surface and the airfoil is symmetric around $y = 0$

1. Results from direct optimization

SU² is run in inviscid mode. The mesh is shown in fig. 8 with 10216 nodes. The airfoil is parameterized by $n = 10$ max point locations (CP) of the Hicks-Henne's bump functions. Figure 9 reports the convergence history for c_ℓ , c_d and the total area of the airfoil. The Karush-Kuhn-Tucker necessary conditions are met after 35 design cycles. Figure 10 shows the pressure distributions and the airfoil shape for both baseline and the optimized airfoil. From table 1 it can be seen that the wave drag is reduced from around 460 drag *counts* (0.0001) to 220 drag *counts* while its maximum thickness is greater than the baseline. A wedge shape is obtained at the trailing edge. The reason is that this optimization problem is set up in SU² as:

$$\begin{aligned}
& \min: && c_d \\
& \text{subject to:} && \max(t) \geq \max(t_{\text{baseline}}) \\
& && \max(t_{1/4}) \geq \max(t_{1/4,\text{baseline}}) \\
& && \max(t_{1/2}) \geq \max(t_{1/2,\text{baseline}}) \\
& && \max(t_{3/4}) \geq \max(t_{3/4,\text{baseline}}).
\end{aligned} \tag{16}$$

With the shape functions defined in fig. 7. Note that for the optimized shape obtained the right most shape function (with max-point at $0.95c$) is not activated, where c is the chord length. In order to get a improved shape more shape functions between $[0.9c, c]$ should be placed. Note that the max-point locations do not necessarily have to be equally-distributed.

Shape	c_l	c_d	AREA
Baseline	0.04529	0.04568	0.08169
Optimized	-0.002512	0.02209	0.1032

Table 1. NACA 0012 optimization results table

Figure 8. NACA-0012 airfoil mesh with 10216 nodes

(a) Convergent history of c_l and the total area

(b) Convergence of cost function c_d

Figure 9. The design cycle history for NACA-0012 with 10 design variables

Figure 10. The resulting pressure distributions and the airfoil shape for NACA 0012. Black: baseline; Blue: optimized.

2. Results from inverse design

In order to carry out the inverse design, the target pressure must be defined. A very rough target pressure is generated using Campbell's method(7) without smoothing it. The initial airfoil has around 460 drag counts. As the pressure distribution is proceeding to the target, the drag decreases. After 30 iterations the drag is reduced to 230 counts. Further reduction of the drag is possible with more detailed and well-designed target pressure distribution.

Vassberg et al.(14) report good results on drag reduction. Comparing our target pressure distribution (Figure 11(a)) with Vassberg's optimal pressure distribution, we can see a peak (minimum C_p) around the leading edge. We believe that wave drag is sensitive to airfoil nose curvature and the shape differences there are responsible for the drag difference.

The inverse design gives equivalent drag reduction as the direct optimization but in a faster and easier way with a more reasonable airfoil shape. The proposed hybrid toolbox is intended to combine both direct and inverse design methodologies to compensate for the drawbacks of each approach.

Figure 11. Comparison of NACA0012 airfoil and its design *Iter-30* at transonic speed, $M = 0.85$, $\alpha = 0^\circ$

B. Drag Minimization of RAE-2822 Airfoil in Transonic Viscous Flow

The drag should be minimized at Mach number 0.734 and lift coefficient of 0.824, and the cross section area must exceed or equal that of the baseline. Initial angle of attack is 2.79° . The flow is viscous with Reynolds number $Re = 6.5 \times 10^6$.

The optimization problem is

$$\begin{aligned}
&\text{min: } c_d \\
&\text{subject to: } c_\ell = 0.824 \\
&\quad c_m \geq -0.092 \\
&\quad A \geq A_{baseline}
\end{aligned} \tag{17}$$

1. Results from direct optimization

The S-A turbulent model is used in this test case. Both a coarser grid and a much finer grid with 10 times nodes are studied. Fig. 12 shows the finer airfoil mesh with 140573 nodes. Table 1 shows the optimization solution table for RAE 2822 airfoil. For all the cases the optimization constraints are met. The KKT condition is met after 35 design cycles. The shock at around 55% chord is weakened, with a drag benefit of 70 drag *counts*. Figure 14 shows the pressure distribution and airfoil shape for both baseline and optimized airfoils. The C_p of the optimized shape has wiggles in between 0.5 to 0.6 chord, that the shock starts to rebuild a little. Figure 15 shows the Mach contours of both RAE 2822 and its optimized shape, with weakened shock on the optimized shape. Note that the Mach is re-developed between 0.5 to 0.6 chord.

Figure 12. RAE-2822 airfoil mesh with 140573 nodes

Nodes	CP nr.	c_ℓ	c_d	c_m	$c_{\ell_{opt}}$	$c_{d_{opt}}$	$c_{m_{opt}}$
13937	19	0.8128	0.01922	-0.09758	0.8147	0.01451	-0.08994
13937	10	0.8128	0.01922	-0.09758	0.8385	0.01323	-0.09151
140573 ^d	19	0.8092	0.01949	-0.09679	0.8431	0.01263	-0.08631

Table 2. RAE 2822 optimization results table

2. Results from inverse design

The target pressure is defined as a pressure distribution with weakened shocks. The target is found perfectly with 50 iterations by SCID inviscid mode to get quick convergence, see fig. 16. Due to zero pressure-gradient through boundary layer we would rather use SCID inviscid mode to compute once the target pressure is given. However there are form drag and skin friction drag that SU² can give while SCID-inviscid mode can not. A compromise is made by re-running the solution from inviscid SCID in MSES(3) viscous mode, see fig. 17. The drag is reduced from 170 *counts* to 115 *counts*, with c_m constraint perfectly held.

C. Wing twist optimization

For this test case a straight rectangular wing(15) with $AR = 6$ is studied. The optimization problem is

(a) Convergence of constraints on c_ℓ and $-10 * c_m$

(b) Convergence of cost function c_d

Figure 13. The design cycles history for RAE 2822 with 19 design variables on the fine grids

(a)

(b)

Figure 14. Pressure distributions and airfoil shapes for RAE 2822. Black: baseline; Blue: optimized.

(a) Baseline

(b) Optimized

Figure 15. Mach contour for RAE 2822 and its optimized shape at Mach 0.734, $c_\ell = 0.824$

Airfoil	c_ℓ	c_d	c_m
RAE2822	0.824	0.016987	-0.09394
Iter-50	0.824	0.011512	-0.05896

Table 3. RAE 2822 design by re-running the RAE 2822 and the solution in viscous mode

Figure 16. The resulting pressure distributions and the airfoil shape from SCID-inviscid for RAE-2822 airfoil

Figure 17. Re-run the solution Iter-50 in viscous mode

$$\begin{aligned}
 \min: & \quad C_D \\
 \text{subject to:} & \quad C_L(\gamma, u(\gamma)) = 0.375 \\
 & \quad \frac{d\gamma}{dy}(y = 3c) \geq -10^\circ/c
 \end{aligned} \tag{18}$$

where c is the chord length, the twist is denoted by γ , and the state variables by $u(\gamma)$. The flow condition is subsonic at Mach 0.5. The minimum mesh off-wall spacing is $0.004c$ with 3.7 M nodes, and all the calculations are made by Euler solver in Edge. By specifying the local lift C_{LL} to be elliptic to minimize induced drag, the twisted wing is found by matching the C_{LL} using simply lifting-line theory by changing the “effective angle of attack” (16; 17) after four design cycles. The span efficiency factor $e = \frac{C_L^2}{\pi AR C_D}$ is already high for the initial untwisted wing, and twisting the wing sections can only improve it marginally. Four design cycles were made, it shows a good convergence towards the design target. Fig. 18 shows the results for optimizing the twist of a rectangular wing to minimize the induced drag, or, with target that produces an elliptical lift distribution. We can see that the span efficiency factor e converged to a higher value from 0.9177 to 0.9324 after four design cycles (G04), and the wing gets around 1 *count* drag benefit by simply twisted to match the elliptical loading. Root section angle of attack is fixed at 4.262° .

IV. SCID on a Swept-back Wing: a test case

A test case for the latest SCID inverse design toolbox is carried out, on a joined-wing configuration that Nangia raised(18), which is designed to be a competitor to A321-100. Only the forward wing is considered

Figure 18. Results for optimizing the twist of a rectangular wing with NACA-0012 profiles and $AR = 6$, $M = 0.5$, $C_L = 0.375$, angle of attack fixed at $\alpha = 4.26^\circ$

now. It is a swept-back wing with twists and cambers, which is designed to fly at Mach 0.78 with $C_L \approx 0.46$. The C_p contour on the baseline wing and the resulting wing by SCID-Euler with 30 design cycles are shown below. A better design is achieved with 30 design cycles, that the lift coefficient is maintained at $C_L \approx 0.46$, with drag halved from 169.8 counts to 79.96 counts. Figures 19,20 show that the designed wing is “shock free” after 30 iterations in SCID. The target pressure distributions are obtained from the Euler solutions on the Common Research Model (CRM) wing(19) based on engineer’s knowledge that the CRM model is considered as optimized under the desired flight conditions. Note that a much thinner wing is obtained and the reduction of thickness contributes a lot to the drag reduction. Developing a thickness constrain option in SCID is one of our work in the future.

V. Future work: target pressure selection

How to define/select good target pressure distributions, still needs efforts to work with, especially for wings those are already well designed, for example the CRM wing (19). SCID new option *User specified- C_p editing* is still under-developing with the following features considered.

A. Adaptive refinement of geometry parametrization

The surface curvature of the geometry will be plotted for a converged design, that the engineer will know how to modify the numbers of design variables for geometry parametrization to get a more realistic geometry. For example to avoid the geometry that is obtained in fig. 9(b). The adaptive refinement of geometry parameterization will provide visual guidance to the engineer in selectively refining her choice of target

Figure 19. C_p on the baseline wing and SCID ITER-30 wing, Euler solution

Figure 20. Comparisons of pressure distributions for J-W forward wing, at $M = 0.78$ with $C_L \approx 0.46$, designed by SCID with 30 design cycles. The initial C_p distributions are marked by black solid lines, the target C_p distributions are marked by red solid lines, and the designed C_p distributions by SCID with 30 design cycles are marked by blue solid lines.

pressure.

B. Thickness and area constraints in SCID

The profile's thickness and area should be able to be constrained in SCID. Campbell used a "glove" option in the CDISC code(6) to constrain the thickness, we would like to continue with it and add an option to constrain the area.

Figure 21. Comparisons of airfoil coordinates for J-W forward wing, at $M = 0.78$ with $C_L \approx 0.46$, designed by SCID with 30 design cycles,

VI. Conclusions

A hybrid wing design toolbox combined both inverse design (SCID) and direct adjoint optimization (SU^2) is under developing, solutions are presented for the drag minimization of the airfoil cases along with the wing twist optimization problem. A test case of a swept wing inverse design by SCID in transonic speed is also carried out, showing that the latest SCID works well for both airfoils and wings. A number of future work should be carried out, together with the drag minimization of the CRM wing in transonic speed. The work is on progress and the results will be presented in a future paper.

References

- ¹ Sobieszczanski-Sobieski, J., *Multidisciplinary Design Optimization: An Emerging New Engineering Discipline*, Kluwer Academic Publisher, 1995.
- ² Zhang, M., Wang, C., Rizzi, A., and Nangia, R., “Hybrid feedback design for subsonic and transonic airfoils and wings,” *AIAA SciTech*, AIAA 2014-0414, National Harbor, Maryland, USA, 13–17 January 2014.
- ³ Drela, M., “Newton solution of coupled viscous/inviscid multielement airfoil flows,” *AIAA Aerospace Sciences Meeting*, 90-1470, 1990.
- ⁴ Takanashi, S., “An iterative procedure for three-dimensional transonic wing design by the integral equation method,” *AIAA Aerospace Sciences Meeting*.
- ⁵ Palacios, F., Colonno, M. R., Aranake, A. C., Campos, A., Copeland, S. R., Economon, T. D., Lonkar, A. K., Lukaczyk, T. W., Taylor, T. W. R., and Alonso, J., “Stanford University Unstructured (SU^2): An open-source integrated computational environment for multi-Physics simulation and design,” *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA 2013-0287, Grapevine, Texas, USA, 7–10 January 2013.
- ⁶ Campbell, R. L. and Smith, L. A., “A hybrid algorithm for transonic airfoil and wing design,” *AIAA Aerospace Sciences Meeting*, 87-2552, 1987.

- ⁷ Campbell, R. L., “An approach to constrained aerodynamic design with application to airfoils,” Tech. rep.
- ⁸ Barger, R. L. and Brooks, C. W. J., “A streamline curvature method for design of supercritical and subcritical airfoils,” Tech. rep., 1974, NASA TN D-7770.
- ⁹ Anderson, J. D., *Modern Compressible Flow with Historical Perspective*, third edition ed., 2004.
- ¹⁰ Tomac, M. and Eller, D., “From geometry to CFD grids. An automated approach for conceptual design,” *Progress in Aerospace Sciences*, Vol. 47, No. 11, 2011, pp. 589–596, doi: 10.1016/j.paerosci.2011.08.005.
- ¹¹ Melin, T., Amadori, K., and Krus, P., “Parametric wing profile description for conceptual design,” *CEAS Meeting*, Venice, Italy, 2011.
- ¹² Hicks, R. and Henne, P. A., “Wing Design by Numerical Optimization,” *AIAA*, AIAA 77-1247, Seattle, Washington, USA, 22–24 August 1977.
- ¹³ Sederberg, T. W. and Parry, S. R., “Free-Form Deformation of Solid Geometric Models,” *ACM Transactions on Mathematical Software*, Vol. 20, No. 4, 1986, pp. 151–160.
- ¹⁴ Vassberg, J. C., Harrison, N., Roman, D., and Jameson, A., “A systematic study on the impact of dimensionality for a two-dimensional aerodynamic optimization model problem,” *29th AIAA Applied Aerodynamics Conference*, AIAA 2011-3176, Honolulu, Hawaii, USA, June 2011.
- ¹⁵ Hicken, J. E., “Aerodynamic design optimization workshop: twsit optimization case,” Tech. rep.
- ¹⁶ Lamar, J. E., “A Vortex-Lattice Method for the Mean Camber Shapes of Trimmed Non-coplanar Planforms with minimum Vortex Drag,” Tech. rep., Washington, D. C., 1976, NASA TN D-8090.
- ¹⁷ Nangia, R. K., Palmer, M. E., , and Doe, R. H., “Aerodynamic Design Studies of Conventional & Unconventional Wings with Winglets,” *AIAA Aerospace Sciences Meeting*, AIAA 2000-3400, 2000.
- ¹⁸ Nangia, R. K., Palmer, M. E., Hyde, L., and Cooper, J. E., “Aerodynamic Efficient Configurations & Structural Design Challenges Arising - Joined Wings & Oblique Wings,” *RAeS / CEAS 2009 European Air & Space Conference*, Manchester, United Kingdom, 26–29 October 2009.
- ¹⁹ Osusky, L. M. and Zingg, D. W., “Lift-constrained drag minimization of a wing allowing section and twist variation with flow governed by the Reynolds-Averaged Navier-Stokes equations,” Tech. rep.