

European Congress on Computational Methods  
in Applied Sciences and Engineering (ECCOMAS 2012)  
J. Eberhardsteiner et.al. (eds.)  
Vienna, Austria, September 10-14, 2012

## EFFICIENT POLAR OPTIMIZATION OF TRANSPORT AIRCRAFT IN TRANSONIC RANS FLOW USING ADJOINT GRADIENT BASED APPROACH

Časlav Ilić<sup>1</sup>, Markus Widhalm<sup>1</sup>, and Joël Brezillon<sup>1</sup>

<sup>1</sup>German Aerospace Center (DLR)  
DLR AS, Lilienthalplatz 7, 38108 Braunschweig, Germany  
e-mail: {caslav.ilic,markus.widhalm,joel.brezillon}@dlr.de

**Keywords:** aircraft, optimization, gradient, adjoint, multi-point, polar

**Abstract.** *A major design requirement for transport aircraft is efficient cruise flight in the transonic region. From the aerodynamic viewpoint, this is achieved by favorable lift-to-drag ratio of the aircraft, both at the main design point and at off-design conditions. We therefore present a method to efficiently perform a multi-point optimization of representative 2D (transonic airfoil) and 3D geometry (wing-body configuration). All designs are evaluated with RANS CFD simulations, using SA turbulence model. In the 2D case, the airfoil is parametrized with 30 CST parameters, as well as with Z-movement of every mesh node. In the 3D case, the wing is parametrized with 40 free-form deformation control points, such that the planform is kept fixed. Gradient based methods (CG, SQP) are used to drive the optimization. Gradients of flow-related objective and constraint functions are computed with a discrete adjoint approach, in which flow and mesh adjoint equations are solved. Unlike with single point optimization, we obtain improved lift-to-drag ratio over most of the practical range of lift coefficients using multi-point optimization.*

## 1 INTRODUCTION

Transport aircraft, especially mid- and long-range, spend most of their mission time in cruise flight. Efficient cruise is therefore one of the main design goals for transport aircraft. Cruise can be conducted in several ways, the most efficient of which (if only ideal) is continuous cruise-climb. Flight Mach number and lift coefficient are kept constant throughout at their design values, which necessitates climbing into less dense air to reduce total lift as the fuel is expended and the aircraft becomes lighter. Performance for cruise-climb can be quantified using the Breguet range equation[22, p. 460]:

$$R = M \frac{a}{g} \frac{1}{c_T} \frac{C_L}{C_D} \ln \frac{m_i}{m_f} \quad (1)$$

where  $R$  is the cruise range,  $M$  the flight Mach number,  $a$  the speed of sound (constant for the usual cruise altitudes above tropopause),  $g$  the acceleration of gravity,  $c_T$  the thrust-specific fuel consumption,  $C_L/C_D$  the lift-to-drag ratio, and  $m_i/m_f$  the ratio of initial to final aircraft mass during cruise. This equation is an approximation since  $c_T$  and  $C_D$  can change with altitude, but not by much for typical cruise altitudes above tropopause.

Equation (1) combines influences of several aircraft design disciplines: factor  $MC_L/C_D$  belongs to aerodynamic design,  $1/c_T$  to engine design, and  $m_i/m_f$  to structural design. As such, this equation can be used in various ways for the purpose of aircraft design; for example, in the multi-disciplinary design context, maximum  $R$  can be sought by tuning several factors at once[15, 9, 4]. The present work is concerned with aerodynamic design alone, seeking to maximize the aerodynamic range factor  $MC_L/C_D$ . The higher this term is, the longer the range for the given fuel load, or the lower necessary fuel load for the given range, can be achieved.

However, it has been shown[5, 19] that maximizing aerodynamic performance at given design point ( $M^{\text{des}}$ ,  $C_L^{\text{des}}$ ) alone leads to unacceptable performance when flight is performed slightly off-design. For transport aircraft, off-design flight can happen simply due to different payload amounts, but also due to air traffic regulations (e.g. only step climb instead of continuous climb allowed) and variations in piloting technique. It is therefore necessary to assure acceptable off-design performance. In the present work, we achieve this through multi-point optimization[23].

Section 2 describes the methodology we use for multi-point optimization, including details of the analysis and optimization algorithms. In section 3 we apply this methodology to a 2D airfoil example, and in section 4 to a 3D wing-body configuration. Section 5 wraps up with some general observations based on our experience with outlined methodology.

## 2 METHODOLOGY

Aerodynamic design optimization can be performed at different fidelity levels of analysis and different degrees of freedom (design parameters) in the aerodynamic shape. The higher the fidelity of the analysis and the number of design parameters, the more expensive the optimization. We target the detailed aerodynamic design phase, where small but significant local improvements are to be obtained, while global features (such as the wing planform) have already been fixed. This implies the use of RANS CFD analysis, and tens to hundreds of design parameters for fine control over the shape. Adjoint gradient based methods are among the best suited for solving problems in this context[32, 8].

Figure 1 represents the process chain of adjoint gradient based optimization. The equations given within flow chart boxes establish the link between the tools in the process chain and

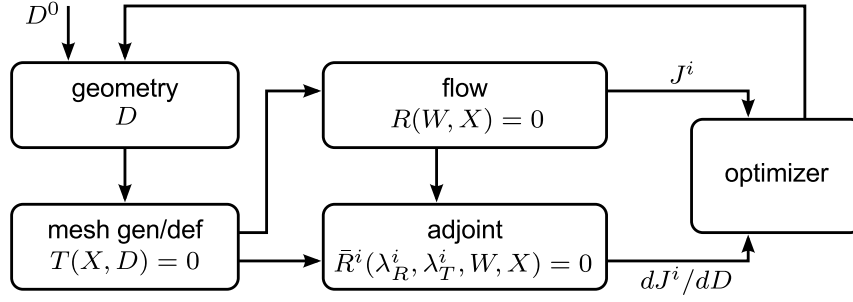


Figure 1: Adjoint gradient based optimization chain.

derivations presented in the following. The optimization starts by supplying the initial design parameter vector  $D^0$  to the geometry generator. Generated geometry is then passed to a mesh generator or deformation tool; in the present work mesh deformation is used throughout. The mesh is sent to the flow solver to compute the flow field and cost function (goal and constraint) values, which are then passed on to the adjoint solver to compute the cost function gradients. Cost function values and gradients are finally used by the optimization algorithm to produce an updated design parameter vector. The cycle repeats until the convergence criterion is satisfied.

## 2.1 Flow analysis

We evaluate aerodynamic characteristics of the design using the DLR TAU flow solver[10, 28]. The utility of TAU in aerodynamic optimization context has already been well established[2, 3]. For stationary turbulent viscous simulation, TAU offers several 1-equation, 2-equation, and RSM turbulence models. In the present work we use the Spalart-Allmaras[29] turbulence model for all flow simulations, as the best compromise between robustness and efficiency on the one side and accuracy on the other. TAU also contains discrete flow adjoint[7] and mesh adjoint[31] computation modes, which we use to efficiently compute gradients of cost functions.

## 2.2 Adjoint computation of the gradient

A straightforward way to compute the gradient is by finite differences. Gradient of  $m$  cost functions  $J$  with respect to  $n$  design parameters  $D$  can be approximated with the forward difference formula:

$$\frac{dJ_i(D)}{dD_j} \approx \frac{J_i(D + E^j) - J_i(D)}{\epsilon_j}, \quad i = 1 \dots m, \quad j = 1 \dots n \quad (2)$$

with  $E^j = (0 \dots \epsilon_j \dots 0)$  and  $\epsilon_j$  the finite difference step per design parameter. There are two problems with this approach. Firstly, to compute  $J_i(D + E^j)$  it is necessary to perform as many additional CFD simulations as there are design parameters, which is usually prohibitively expensive for anything more than single-digit number of design parameters. Secondly, it may be hard to determine proper  $\epsilon_j$  to achieve needed accuracy of the gradient[18]; this is especially so for highly non-linear cost functions, such as force coefficients in transonic RANS flows. Central instead of forward differences can be used to get better accuracy, but this will require twice the number of additional CFD simulations.

If there are much more design parameters than cost functions ( $n \gg m$ ), as is usual in aerodynamic optimization problems, the gradient can be computed at cost of approximately  $m$

flow simulations using the adjoint approach[13, 20]. To compute the value of the cost function  $J_i$  integrated over the computational mesh (such as force coefficients):

$$J_i(W, X) \quad (3)$$

where  $W$  is the flow field (density, velocity...) and  $X$  are the coordinates of the computational mesh, it is necessary to solve the following equations:

$$\begin{aligned} R(W, X) &= 0 \\ T(X, D) &= 0 \end{aligned} \quad (4)$$

$D$  is the design parameter vector,  $R$  is the residual vector of governing flow state equations (Euler, RANS...) and  $T$  is the residual vector of dependency of mesh coordinates to design parameters (the mesh deformation function). We introduce arbitrary fields  $\Lambda_R^i$  and  $\Lambda_T^i$ , respectively called flow and mesh adjoint, and use them to add zeros to cost function<sup>1</sup>:

$$J_i = J_i(W, X) + \Lambda_R^i R(W, X) + \Lambda_T^i T(X, D) \quad (5)$$

Deriving (5) with respect to  $D$  using chain rule and rearranging:

$$\begin{aligned} \frac{dJ_i}{dD} &= \frac{\partial J_i}{\partial W} \frac{dW}{dD} + \frac{\partial J_i}{\partial X} \frac{dX}{dD} + \Lambda_R^i \frac{\partial R}{\partial W} \frac{dW}{dD} + \Lambda_R^i \frac{\partial R}{\partial X} \frac{dX}{dD} + \Lambda_T^i \frac{\partial T}{\partial X} \frac{dX}{dD} + \Lambda_T^i \frac{\partial T}{\partial D} \\ &= \left( \frac{\partial J_i}{\partial W} + \Lambda_R^i \frac{\partial R}{\partial W} \right) \frac{dW}{dD} + \left( \frac{\partial J_i}{\partial X} + \Lambda_R^i \frac{\partial R}{\partial X} + \Lambda_T^i \frac{\partial T}{\partial X} \right) \frac{dX}{dD} + \Lambda_T^i \frac{\partial T}{\partial D} \end{aligned} \quad (6)$$

If  $\Lambda_R^i$  and  $\Lambda_T^i$  are computed to satisfy:

$$\frac{\partial J_i}{\partial W} + \Lambda_R^i \frac{\partial R}{\partial W} = 0 \quad (7)$$

$$\frac{\partial J_i}{\partial X} + \Lambda_R^i \frac{\partial R}{\partial X} + \Lambda_T^i \frac{\partial T}{\partial X} = 0 \quad (8)$$

then the derivatives  $dW/dD$  and  $dX/dD$  in (6) are eliminated and the final expression for the gradient becomes:

$$\frac{dJ_i}{dD} = \Lambda_T^i \frac{\partial T}{\partial D} \quad (9)$$

Equation (7) is the flow adjoint equation and (8) is the mesh adjoint equation, shortly written as:

$$\bar{R}^i(\Lambda_R^i, \Lambda_T^i, W, X) = 0 \quad (10)$$

Adjoint equations do not depend on design parameters  $D$  but only on cost functions  $J_i$ , and hence need to be solved for each cost function only. This makes the total cost of adjoint gradient computation only weakly dependent on the number of design parameters, the dependency being reduced to the  $\partial T / \partial D$  term in expression (9).

As  $T(X, D)$  we use linear elastic[6] mesh deformation throughout. Its form is:

$$X_S - K X_V = 0 \quad (11)$$

<sup>1</sup>At implementation time it becomes obvious where vectors and matrices should be transposed and in which order multiplications should be performed, so for simplicity we avoid making this explicit in the notation.

where  $K$  is the stiffness matrix, and  $X_V$  and  $X_S$  are the split of mesh points  $X$  into volume and surface points, respectively. Thus the needed derivatives for the mesh adjoint system (8) and gradient expression (9) are:

$$\frac{\partial T}{\partial X} = -K, \quad \frac{\partial T}{\partial D} = \frac{\partial X_S}{\partial D} \quad (12)$$

Linear elastic mesh deformation results in high quality meshes, but it can be computationally costly[12]. However, in multi-point optimization context, this cost is relatively decreasing as the number of operating points is increasing, since only one deformation is needed for all operating points.

Depending on the particular parametrization, the term  $\partial X_S / \partial D$ —derivative of surface mesh coordinates with respect to design parameters—may be problematic to evaluate. For example, if parametrization is provided by a CAD program, various methods are proposed to obtain this term[30, 24]. However, for all the parametrizations we use in this work, this term is readily analytically derivable. One of those parametrizations is z-position of every surface mesh node (free-node parametrization), for which it holds  $D = X_S e_z$  with  $e_z = (0 \ 0 \ 1)$ , and therefore (9) becomes simply  $dJ_i / dD = \Lambda_T^i e^z$ .

Finally, for moderate values of lift coefficients there is nearly linear dependency between the lift coefficient  $C_L$  and the angle of attack  $\alpha$ , which can be exploited to remove the explicit  $C_L$  constraint from the set of constraints seen by the optimizer, thus decreasing the difficulty of the problem. Another way to look at it, especially natural in the context of multi-point optimization, is as making  $C_L$  the input parameter of the flow solver instead of  $\alpha$ . To this end the flow solver must perform an internal fix-point iteration on  $\alpha$  to simultaneously solve:

$$\begin{aligned} R(W, X, \alpha) &= 0 \\ C_L(W, X, \alpha) - C_L^* &= 0 \end{aligned} \quad (13)$$

and, due to the additional equation to satisfy target lift  $C_L^*$ , gradients of cost functions need to be corrected[23] as:

$$\left( \frac{dJ_i}{dD} \right)_{\text{corr.}} = \frac{dJ_i}{dD} - \frac{dJ_i / d\alpha}{dC_L / d\alpha} \frac{dC_L}{dD} \quad (14)$$

Since  $\alpha$  is a boundary condition for the flow, the derivatives  $dJ_i / d\alpha$  necessary in the correction term do not depend on the shape, and can be computed based on the flow adjoint solution alone.

### 2.3 Gradient based optimization methods

For unconstrained optimization we use the conjugate gradient (CG) method[11] with quadratic interpolation line search. For constrained optimization, we use the sequential quadratic programming (SQP) method[21], with active set handling of inequality constraints, and back-tracking line search. These methods are implemented in-house within the DLR Pyranha optimization framework[4].

In the SQP method, we employ two approximations of the Hessian of the Lagrangian, needed for solving the local quadratic programming (QP) problem in each cycle. For CAD-type parametrizations, we use BFGS updates to approximate the Hessian. For free-node parametrizations in 2D, we use the discretization matrix arising from the Sobolev smoothing equation for the gradient[14][27, p. 79]:

$$\epsilon \frac{d^2 \tilde{g}}{ds^2} + \tilde{g} = g \quad (15)$$

Here  $\epsilon$  is the smoothing coefficient ( $\epsilon = 1$  is used in subsequent airfoil optimization example),  $g$  is the gradient of cost function with respect to design parameters (e.g.  $\partial C_L / \partial D$ ), and  $s$  is the curvilinear coordinate of the boundary. This equation is never actually solved to obtain the smoothed gradient  $\tilde{g}$ , instead the line search direction that comes out of solving the local QP problem ends up effectively smoothed due to insertion of the smoothing discretization matrix into the QP system.

The active set in the SQP method is maintained as follows. At start, if there are any violated inequality constraints they are included into active set, and others are left out. In each cycle, the first phase is the line search on the merit function, to compute the step in the direction of the QP solution. If the computed step violates one or more inactive constraints, “constraint snapping” is performed. In this phase, the step is shortened and stretched in the binary search manner, until at least one inactive violated constraint falls within the specified limits from the feasible and infeasible side (snapping limits), and all other previously violated inactive constraints fall even deeper inside the feasible region. Snapping limits are specified by the user, per constraint; for example, an airfoil spar height may be given snapping limits  $+0.01h^*/c$  (feasible side) and  $-0.0001h^*/c$  (infeasible side) relative to the target spar height  $h^*/c$ . After the snapping phase, those inactive constraints which are now located within their snapping limits are added to the active set, and next SQP cycle is performed. When the line search on the merit function does not produce a sufficient improvement, one active inequality constraint is dropped from the active set if possible, and the QP solving and line search phase are performed again. An inequality constraint can be dropped if its current Lagrange multiplier is negative (for  $\geq 0$  formulation of constraints), and the constraint with largest absolute negative multiplier is selected. If no constraints can be dropped from the active set, the optimization ends.

### 3 2D RESULTS: AIRFOIL OPTIMIZATION

Standalone airfoil design is still an important elementary block of aircraft design, hence in this section we present an example of multi-point optimization of a fully turbulent transonic airfoil.

As the starting shape we take an 11% thick airfoil designed in-house using a 2.5D inverse design method[1]. This airfoil was designed for strong single-point performance on a fully turbulent wing with  $\phi = 21.4^\circ$  sweep, to serve as a benchmark for other designs with additional considerations (e.g. increased laminarity). As such, this airfoil has a pronounced “shock bump” on the upper surface and a sharp performance peak around the design point. We attempt to redesign it for less sensitive behavior around the design point, without sacrificing other aerodynamic and structural characteristics. The CFD analysis is carried out in 2D, after transforming the original geometry and the design point as:

$$\frac{z^{2D}}{c} = \frac{z}{c} \frac{1}{\cos \phi}, \quad M^{2D} = M \cos \phi, \quad C_L^{2D} = C_L \frac{1}{\cos^2 \phi} \quad (16)$$

This transformation results in 11.7% thick airfoil (depicted on figure 2) and design point  $M^{\text{des}} = 0.7433$  and  $C_L^{\text{des}} = 0.741$ . Hybrid computational mesh with 32,000 points is used.

The optimization problem is illustrated by figure 3. The goal function is a weighted sum of aerodynamic range factor  $MC_L/C_D$  from (1), at the design point and  $n_{\text{off}}$  near off-design points, and the pitching moment coefficient  $C_M$  is constrained (nose-down negative) to be no worse than the baseline. We set the weight factor  $\omega = 1/(n_{\text{off}} + 1)$  for all points.  $n_{\text{off}} = 4$  off-design points are used: two at  $M^{\text{des}}$  with  $C_L = C_L^{\text{des}} \pm 0.1$ , and two at  $M = M^{\text{des}} \pm 0.02$

with  $C_L = (M^{\text{des}}/M)^2 C_L^{\text{des}}$  (i.e. at constant lift force rather than lift coefficient). To this we add  $n_{\text{low}} = 4$  low speed points at which the goal function is not considered, but the drag and pitching moment coefficients are constrained not to be worse than the baseline. These points are all at  $C_L = C_L^{\text{des}}$ , while  $M = 0.65, 0.60, 0.55, 0.50$  (constant lift force formulation is not used here because it would result in unattainably high lift coefficients). In total, there are  $1 + n_{\text{off}} + n_{\text{low}} = 9$  operating points considered. From the structural side, the height of front and rear spar (at  $0.3c$  and  $0.7c$ ), as well as the enclosed volume by spar-delimited section (front, mid, and rear), are constrained not to be lower than the baseline. Spar and volume constraints are evaluated directly on the CFD mesh, and their gradients are derived exactly on the discrete level.

We compare two parametrization strategies. The first is the CST[16], applied on thickness and camber distributions. There are 30 design parameters in total, distributed evenly between thickness and camber. The parametrization is applied to the displacement from the baseline shape rather than on the shape itself, i.e. all CST design parameters are initially zero. The second parametrization strategy is free-node[14], where each airfoil surface mesh node, except for the leading and trailing edge node, is allowed to move in  $z$ -direction. This results in 378 design parameters, but their movement is smoothed out as explained in section 2.3. This parametrization is also of the displacement type, with all design parameters initially being zero.

Optimization histories in terms of optimizer cycles are shown on figure 4. Each optimizer cycle is composed of one gradient evaluation and one or more value-only evaluations as requested by the line search (2 to 3 on average), for all cost functions and operating points. The stopping criterion, per accumulative active set of constraints, is that the weighted-sum lift-to-drag value increased by less than 0.1 for previous 5 cycles. Distinct kinks in history curves correspond to reaching convergence with current active set, after which a constraint is dropped and the optimization proceeds, if possible. Optimizations were performed on a single computing node with two Intel Xeon E5540 processors, giving 8 computing cores in total. Parallelization was only on the level of the flow solver, while different operating points were computed in serial.<sup>2</sup> In terms of wall clock time, these optimizations took 48–72 hours. This time can likely be reduced by using less conservative values of process parameters (e.g. we required very high convergences of flow and adjoint simulations).

Figure 5 compares performance of baseline and optimized airfoils in terms of drag coefficient and Mach-scaled lift-to-drag ratio. Satisfaction of pitching moment constraints for the optimized airfoils is given by figure 6, and of structural constraints by table 1. Shapes and curvature distribution of optimized airfoils are depicted on figure 7. Figures 8a and 8b compare optimized pressure distributions at the design point.

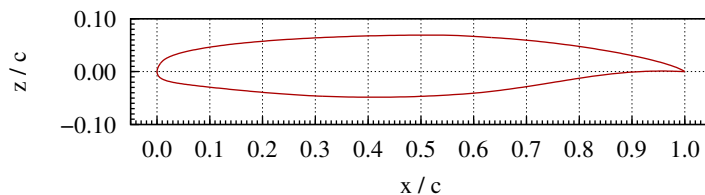


Figure 2: Baseline airfoil shape.

<sup>2</sup>Given the size of the CFD mesh, it would likely have been more efficient if two or more operating points were computed in parallel, with correspondingly reduced number of processes per point.

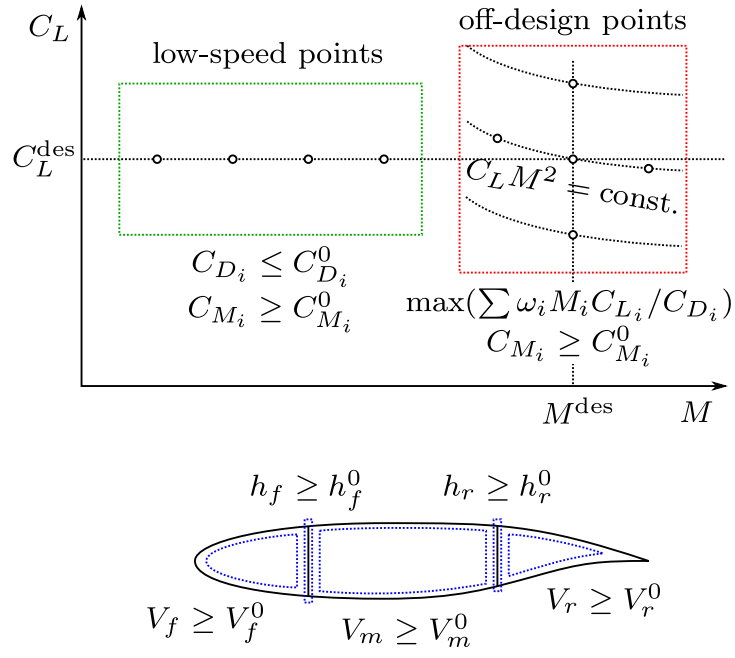


Figure 3: Airfoil optimization setup.

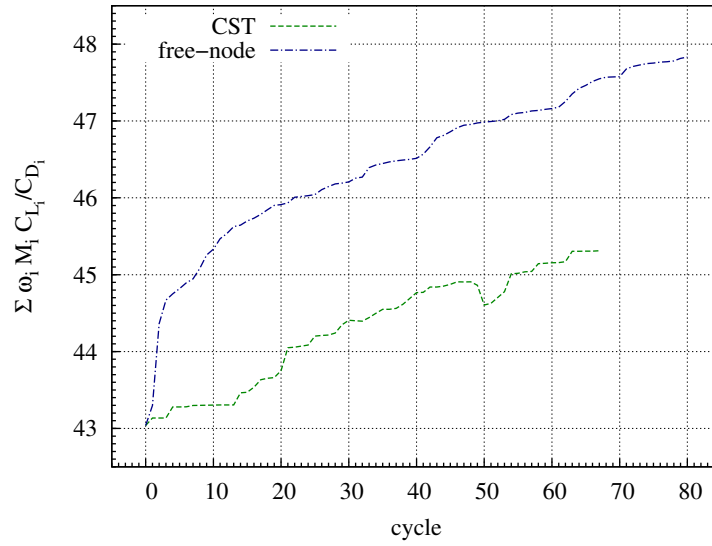


Figure 4: Airfoil optimization history.



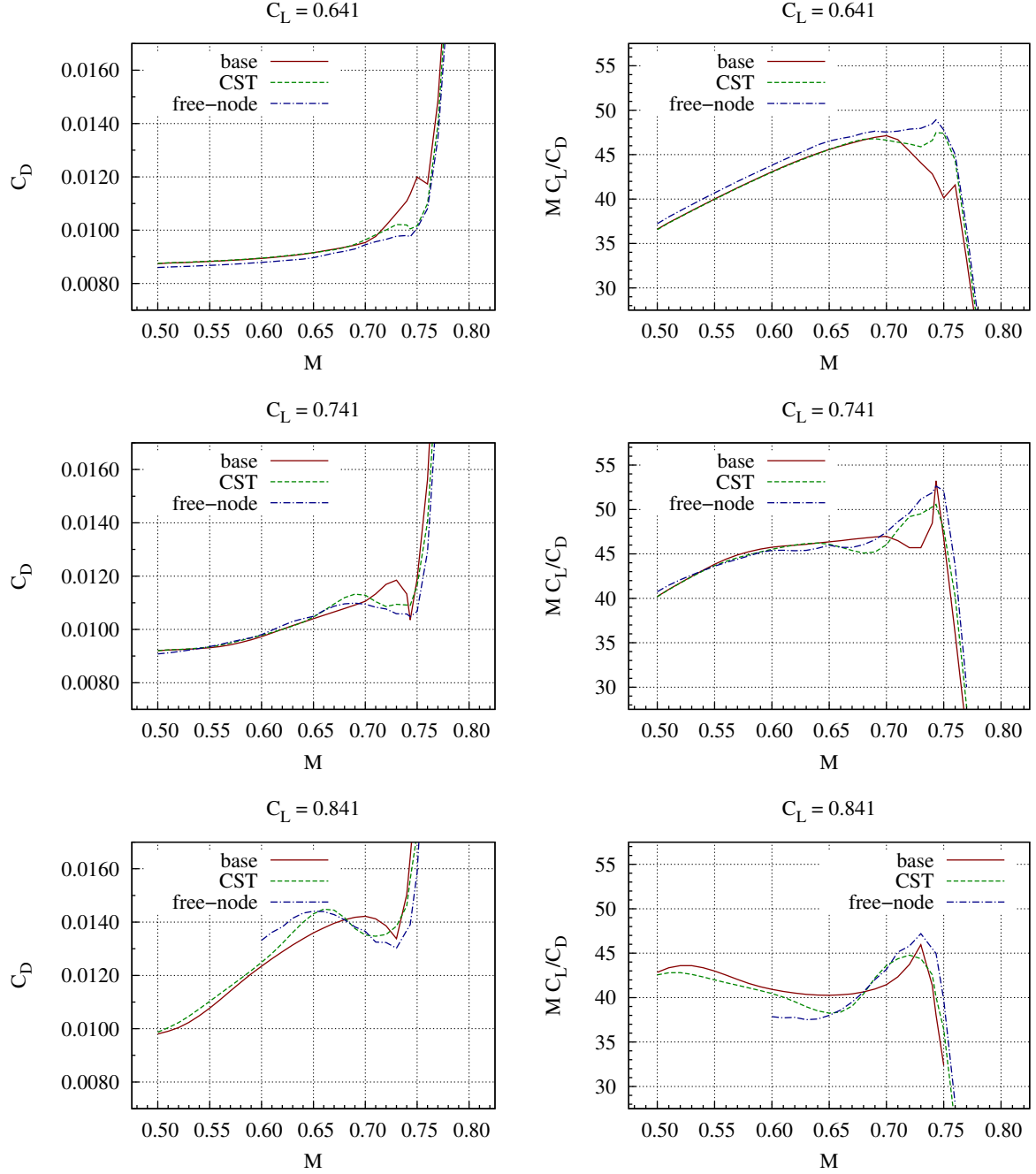


Figure 5: Drag coefficient and Mach-scaled lift-to-drag ratio of baseline and optimized airfoil.

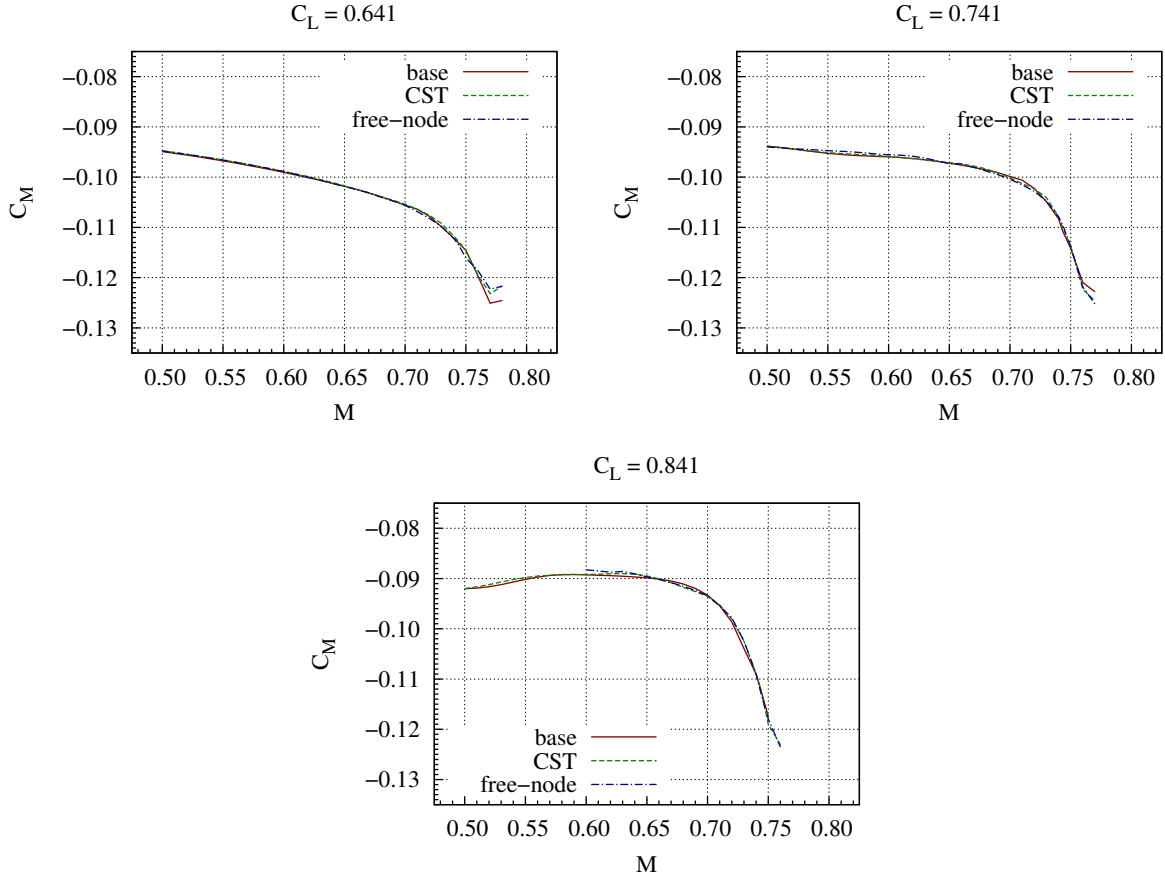


Figure 6: Pitching moment of baseline and optimized airfoil.

|                                    | baseline | CST             | free-node       |
|------------------------------------|----------|-----------------|-----------------|
| front spar height ( $h_f/c$ )      | 0.0964   | 0.0964 (−0.00%) | 0.0964 (+0.06%) |
| rear spar height ( $h_r/c$ )       | 0.0888   | 0.0888 (−0.00%) | 0.0901 (+1.45%) |
| front section volume ( $V_f/c^2$ ) | 0.0146   | 0.0146 (−0.00%) | 0.0146 (−0.07%) |
| mid section volume ( $V_m/c^2$ )   | 0.0549   | 0.0549 (−0.00%) | 0.0547 (−0.07%) |
| rear section volume ( $V_r/c^2$ )  | 0.0143   | 0.0143 (−0.00%) | 0.0143 (−0.08%) |

Table 1: Structural quantities of baseline and optimized airfoil.

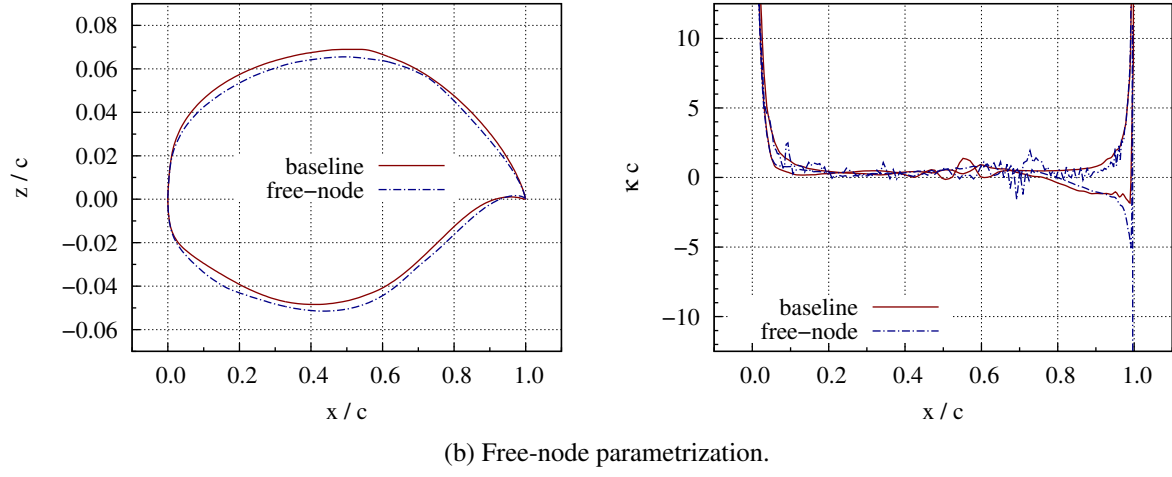
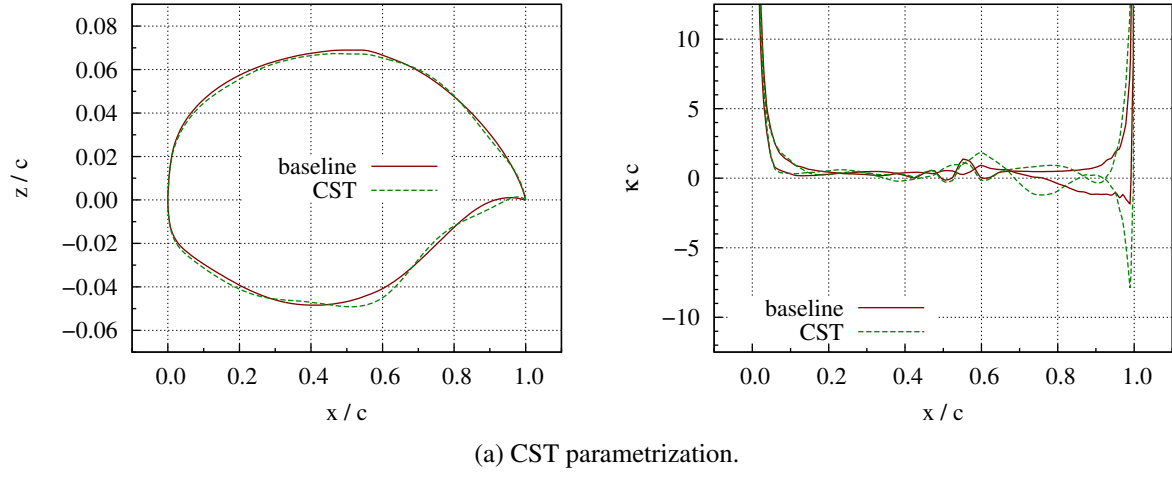


Figure 7: Shape and curvature of baseline and optimized airfoil.

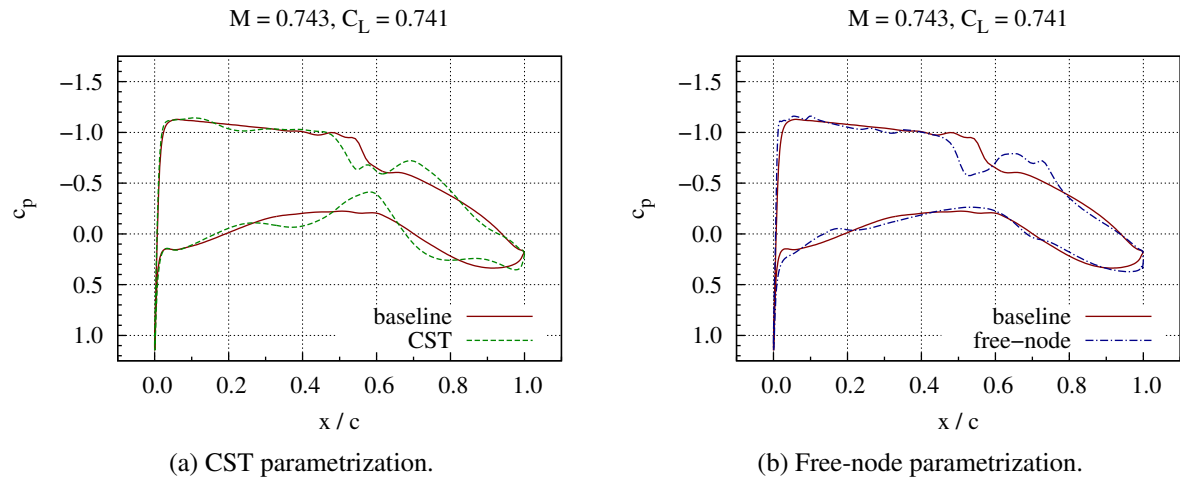


Figure 8: Pressure distribution at design point of baseline and optimized airfoil.

### 3.1 Discussion of airfoil optimization results

Gradient based optimization with many design parameters is prone to “badly surprising” the user. The optimizer will aggressively exploit any available freedom to improve the goal, frequently resulting in unacceptable deterioration (compared to baseline design) of important characteristics of the design which were neither explicitly nor implicitly constrained.

The set of operating points and constraints in this example has been iteratively picked to prevent such over-exploitation by the optimizer. For example, had there not been a constraint on the nose section volume, leading edge would have become much sharper, adversely affecting maximum  $C_L$ , especially at lower speeds; had there not been the constraint on tail section volume, trailing edge would have become more curved and thinner, adversely affecting both maximum  $C_L$  and structural feasibility. As it is, optimized airfoils are a clear improvement over the baseline shape, especially the one resulting from free-node parametrization. CST parametrization produces some waviness on the lower surface, as it does not seem to be penalized by any of the operating points or constraints. Free-node parametrization does not have this effect on the lower surface, and in general it tends to produce shapes which are nearest to the original shape. It seems to us that in this case there is no advantage in CST parametrization over free-node parametrization.

It is interesting to consider here Drela’s hypothesis[5], that the number of design parameters should be equal to the number of operating points and constraints to prevent optimizer over-exploitation. It was further proved[17] that, in unconstrained multi-point optimization, the weighted-sum goal can always be improved by adding more operating points up to the number of design parameters. With the CST parametrization there are 30 design parameters, about equal to the sum of 9 operating points and 18 explicit constraints, 27 in total, quite in line with Drela’s hypothesis. With the free-node parametrization, there is an order of magnitude more design parameters, but the smoothing Hessian approximation in the SQP algorithm plays the role of a moderator. In fact, without some sort of smoothing the optimizer would start to introduce geometry variations well below the resolution capability of the flow solver (e.g. by moving consecutive surface mesh points in alternating directions), quickly leading to loss of boundary regularity[26].

We also checked how much the baseline airfoil could be improved with single-point optimization at the design point. From the figure 9 it can be seen that the improvement is negligible. Single-point optimization does smooth out the pressure distribution somewhat in the shock region, but this translates into only 1–2 drag counts of performance gain. This demonstrates that established inverse design methods are quite capable of matching formal optimization methods in the context of single-point transonic airfoil design, and that more complicated cases should be considered the real target for formal optimization methods.

Further work on airfoil optimization will go into two complementary directions. One direction is to achieve yet smoother shapes in terms of curvature distribution, which can be seen to have low-amplitude high-frequency oscillation on figure 7. Post-smoothing could be used to this end, but this would also remove some of the gain delivered by the optimizer, and possibly worsen satisfaction of constraints. The other direction is using optimization “from scratch” as means of designing a transonic airfoil. Optimization would start from a very generic baseline shape, such as a NACA airfoil, and produce an airfoil with required characteristics. This will probably require yet more operating points and constraints, but also the capability to start from an infeasible initial shape. For example, the initial shape may not at all be capable of reaching one or more of the considered  $(M, C_L)$  operating points.

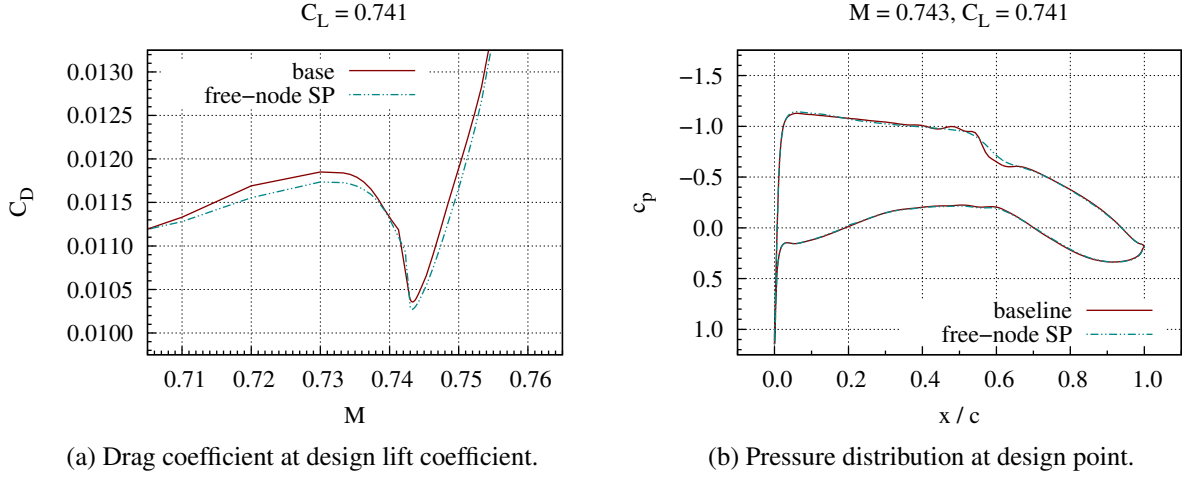


Figure 9: Baseline airfoil improved with single-point optimization.

#### 4 3D RESULTS: WING-BODY OPTIMIZATION

The baseline wing-body geometry (figure 10a) is taken from the MDORMEC[4] project. It is representative of a short-range jet airliner, with 28 m wing span. The design point considered is  $M^{\text{des}} = 0.82$ ,  $C_L^{\text{des}} = 0.554$ . For analysis of this geometry we use a hybrid mesh with 1.7 million points.

The optimization setup is a simplified version of the one described in section 3. There are two near off-design points, at  $M = M^{\text{des}}$  and  $C_L = C_L^{\text{des}} - 0.15$ ,  $C_L^{\text{des}} - 0.30$ , and no low-speed points. There are also no explicit structural constraints; instead wing section thickness is implicitly preserved by choice of parametrization. Free-form deformation[25] with 110 control nodes is used as parametrization, figure 10b. Only wing shape is changed during optimization, through 80 control nodes (colored red on figure 10b), while body shape is kept fixed. Corresponding upper and lower wing control nodes are moved together vertically to maintain the section thickness, resulting in 40 design parameters in total. Since the baseline shape in this case was not optimized for single-point performance, we also perform single-point optimization for comparison.

Figure 11 shows optimization histories for single- and multi-point problems, in terms of optimizer cycles (with same definition as in section 3). Since there are no explicit constraints, for these optimizations we use the CG optimizer. The stopping criterion is that the relative improvement in the goal function value between two consecutive cycles is less than  $10^{-9}$ . Problems were run on 4 computing nodes of the same type as in section 3, for total of 32 cores employed in the same fashion as before. The wall clock time for the single-point optimization was 43 hours, and for multi-point 87 hours. Single-point optimization took 52 flow and 14 adjoint simulations, and multi-point took 34 flow and 9 adjoint simulations per operating point.

Performance of baseline and optimized shapes is given by figures 12 and 13. This is a good place to explain the perhaps peculiar choice of off-design points: at first, we tried the more usual  $C_L = C_L^{\text{des}} \pm 0.05$  points, but that led to almost the same result as single-point optimization; then we switched to stated off-design points in order to recover some of the loss of performance against the baseline seen at off-design conditions. Spanwise load distributions are compared on figure 14, and airfoil shapes and pressure distributions at selected span stations are compared on

figure 15. It can be seen that the lift distribution of optimized shape is more elliptic, reducing the induced drag, and that shock strengths are reduced, reducing the wave drag. Figure 15 also shows that the optimized wing is effectively additionally twisted from root to tip, although none of the design parameters are directly related to twist.

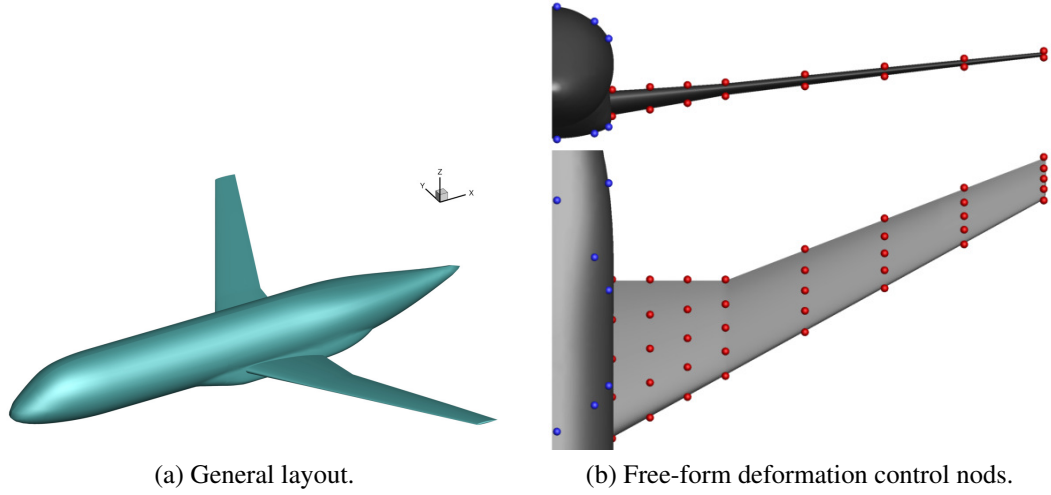


Figure 10: Baseline wing-body configuration.

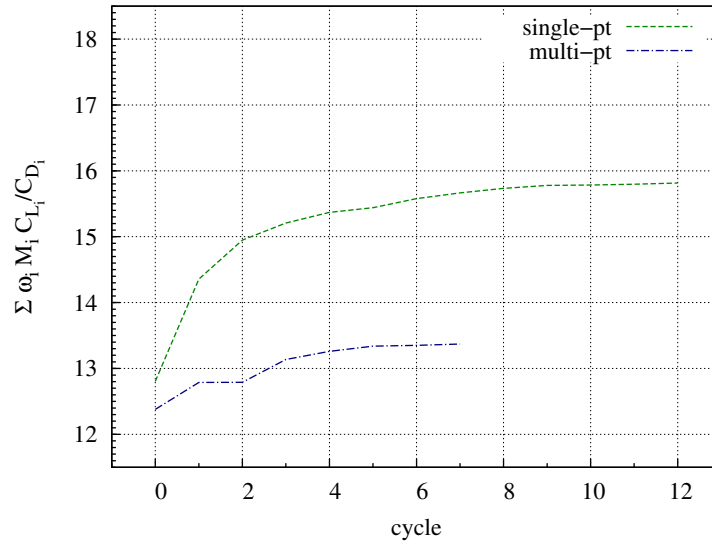


Figure 11: Wing-body optimization history.

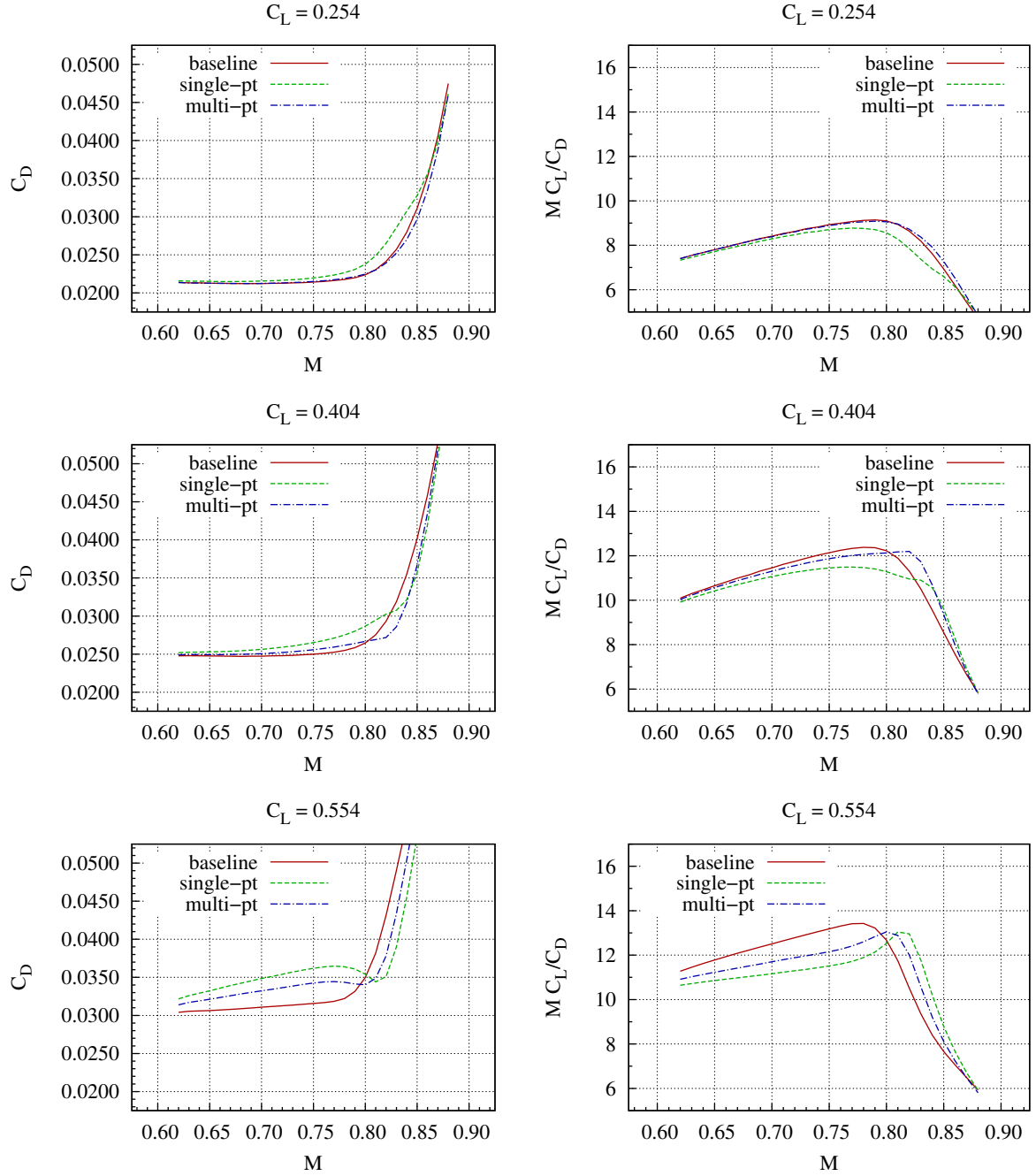


Figure 12: Drag coefficient and Mach-scaled lift-to-drag of baseline and optimized wing-body.

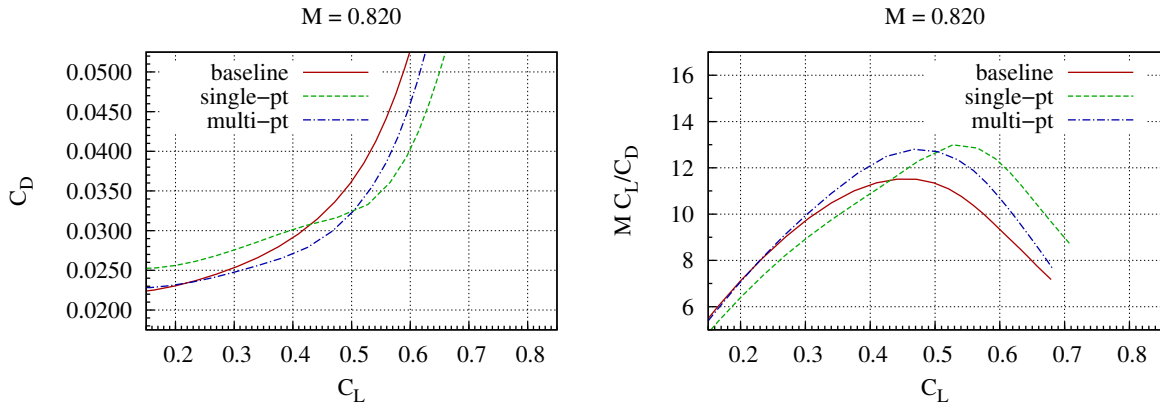


Figure 13: Drag coefficient and Mach-scaled lift-to-drag polars of baseline and optimized wing-body at design Mach number.

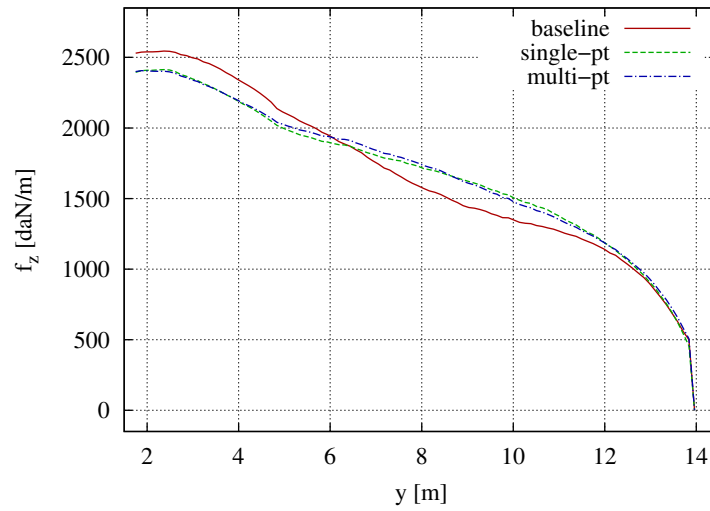


Figure 14: Spanwise wing lift distribution of baseline and optimized wing-body.



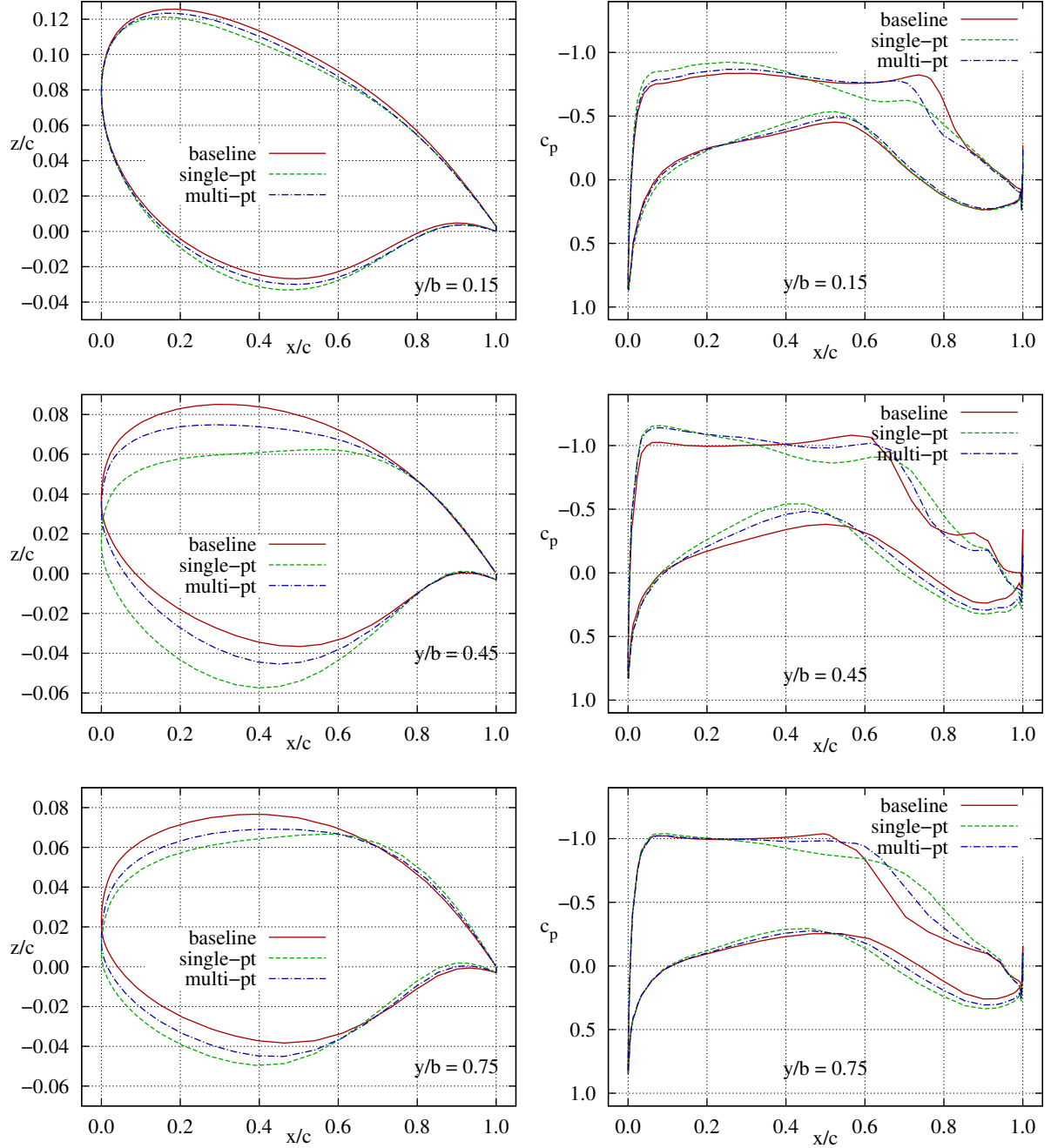


Figure 15: Section shapes and pressure distributions of baseline and optimized wing-body.

## 4.1 Discussion of wing-body optimization results

Unlike in the airfoil optimization example of section 3, in this example the number of operating points and constraints is much smaller than the number of design parameters. Yet, the resulting optimized shapes do not exhibit too high specialization to the considered operating points (figures 12 and 13). From this it seems that the clean wing-body problem is less prone to optimizer over-exploitation than the airfoil problem.

However, we must note that there are important elements missing from the demonstrated wing-body optimization setup to make it a practical design tool. Most importantly, since this is pure aerodynamic optimization, it is necessary to add constraints related to spanwise lift distribution. While wing section thickness is implicitly preserved by the parametrization, figure 14 shows that lift distribution tends to move outward, demanding stronger, heavier wing structure. The other important missing factor are trimming effects. Since the horizontal tail operates in the downwash of the wing, and given that we strive for high-fidelity analysis, it is questionable if trimming could be handled by simply prescribing pitching moment constraints on the standalone wing-body configuration; a full wing-body-tail configuration with trimming iteration may have to be used instead. Thus, further work on 3D configuration optimization will concentrate on implementing needed elements in the optimization chain for handling these additional constraints.

## 5 CONCLUSIONS

Gradient based adjoint optimization is capable of handling 2D and 3D aerodynamic design cases related to cruise flight with large number of design parameters, but some mechanism must be employed to avoid over-specialization of the design to single operating point which frequently happens with single-point optimization. Multi-point optimization may be used to this end, if the parametrization and the set of operating points and constraints are appropriately chosen. This may actually be trickier for 2D problems, as they seem to be more prone to over-specialization for single operating point.

Even if the flow solver is capable of delivering cost function values and gradients accurate to machine precision, such accuracy can be achieved only on test problems. In practice, flow/adjoint simulation convergence will stall much before machine precision on more complex geometries, and there will certainly be limits to time and computational resources. Therefore, optimization algorithms need to be devised or tweaked to handle some inaccuracy in the analysis results. This is particularly the case where gradient of constraints is the main driver for maintaining feasibility during optimization.

In classical aerodynamic design, many constraints are implicitly satisfied through interaction of parametrization and design methodology. This is not so with formal optimization methods—gradient based or otherwise—and therefore optimization chains need to be able to handle many kinds of explicit constraints. Implementers of adjoint capability in particular need to have this in mind, as gradients of most constraints need to have adjoint systems derived for them.

## 6 ACKNOWLEDGMENTS

We are thankful to our colleague Mohammad Abu-Zurayk who supplied us with the wing-body CFD mesh, which was economical in size and numerically very nicely behaving. We are also thankful to our colleagues Thomas Streit and Benjamin Rodax who gave us many pointers on airfoil and wing design.

## REFERENCES

- [1] W. Bartelheimer. An inverse method for the design of transonic airfoils and wings. *Inverse Problems in Engineering*, 4:21–52, 1996.
- [2] J. Brezillon, O. Brodersen, R. P. Dwight, A. Ronzheimer, and J. Wild. Development and application of a flexible and efficient environment for aerodynamic shape optimisation. In *ONERA-DLR Aerospace Symposium (ODAS)*, 2006.
- [3] J. Brezillon and R. P. Dwight. Applications of a discrete viscous adjoint method for aerodynamic shape optimisation of 3D configurations. *CEAS Aeronautical Journal*, (DOI: 10.1007/s13272-011-0038-0), 2011.
- [4] J. Brezillon, A. Ronzheimer, D. Haar, M. Abu-Zurayk, M. Lummer, W. Krüger, and F.-J. Natterer. Development and application of multi-disciplinary optimization capabilities based on high-fidelity methods. In *8th AIAA Multidisciplinary Design Optimization Specialist Conference*, number AIAA-2012-1757, 2012.
- [5] M. Drela. Pros and cons of airfoil optimization. In *Frontiers of Computational Fluid Dynamics 1998*, 1998.
- [6] R. Dwight. Robust mesh deformation using the linear elasticity equations. In *Computational Fluid Dynamics 2006*, pages 401–406. Springer, 2006.
- [7] R. P. Dwight and J. Brezillon. Efficient and robust algorithms for solution of the adjoint compressible navier-stokes equations with applications. *International Journal of Numerical Methods in Fluids*, 60(4):365–389, 2008.
- [8] B. Epstein, A. Jameson, S. Peigin, D. Roman, N. Harrison, and J. Vassberg. Comparative study of 3D wing drag minimization by different optimization techniques. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, number AIAA-2008-326, 2008.
- [9] K. A. Geiselhart, L. P. Ozoroski, J. W. Fenbert, and E. W. Shields. Integration of multifidelity multidisciplinary computer codes for design and analysis of supersonic aircraft. In *49th AIAA Aerospace Sciences Meeting*, number AIAA-2011-465, 2011.
- [10] T. Gerhold, M. Galle, O. Friedrichs, and J. Evans. Calculation of complex three-dimensional configurations employing the DLR TAU-Code. In *35th Aerospace Sciences Meeting and Exhibit*, number AIAA-97-0167, 1997.
- [11] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization*, 2(1), 2006.
- [12] J. E. Hicken and D. W. Zingg. Aerodynamic optimization algorithm with integrated geometry parameterization and mesh movement. *AIAA Journal*, 48(2):400–413, 2010.
- [13] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, 1988.
- [14] A. Jameson. Optimum aerodynamic design using CFD and control theory. In *AIAA 12th Computational Fluid Dynamics Conference*, number AIAA-95-1729, 1995.

- [15] A. Jameson, K. Leoviriyakit, and S. Shankaran. Multi-point aero-structural optimization of wings including planform variations. In *45th Aerospace Sciences Meeting and Exhibit*, number AIAA-2007-764, 2007.
- [16] B. M. Kulfan and J. E. Bussoletti. "Fundamental" parametric geometry representations for aircraft component shapes. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number AIAA-2006-6948, 2006.
- [17] W. Li, L. Huyse, and S. Padula. Robust airfoil optimization to achieve consistent drag reduction over a mach range. Technical Report CR-2001-211042, NASA ICASE, 2001.
- [18] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso. The complex-step derivative approximation. *ACM Transactions on Mathematical Software*, 29(3), 2003.
- [19] M. Nemec and D. W. Zingg. Multi-point and multi-objective aerodynamic shape optimization. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, number AIAA-2002-5548, 2002.
- [20] E. J. Nielsen and M. A. Park. Using an adjoint approach to eliminate mesh sensitivities in computational design. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*, number AIAA-2005-491, 2005.
- [21] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.
- [22] D. P. Raymer. *Aircraft Design: A Conceptual Approach*. AIAA, 2nd edition, 1992.
- [23] J. Reuther, J. Alonso, M. J. Rimlinger, and D. Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers. *Journal of Aircraft*, 36(2), 1999.
- [24] T. T. Robinson, C. G. Armstrong, H. S. Chua, C. Othmer, and T. Grahs. Optimizing parameterized CAD geometries using sensitivities based on adjoint functions. *Computer-Aided Design and Applications*, 9(3), 2012.
- [25] A. Ronzheimer. Shape based on freeform deformation in aerodynamic design optimization. In *ERCRAFT Design Optimization International Conference*, 2004.
- [26] S. Schmidt, Č. Ilić, V. Schulz, and N. Gauger. Shape gradients and their smoothness for practical aerodynamic design optimization. Technical Report SPP1253-10-03, DFG Forschungsbericht, 2008.
- [27] S. Schmidt and V. Schulz. Impulse Response Approximations of Discrete Shape Hessians with Application in CFD. *SIAM Journal on Control and Optimization*, 48(4):2562–2580, 2009.
- [28] D. Schwamborn, T. Gerhold, and R. Heinrich. The DLR TAU-Code: Recent Applications in Research and Industry. In *ECCOMAS CFD 2006*, 2006.
- [29] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th Aerospace Sciences Meeting and Exhibit*, number AIAA-92-0439, 1992.

- [30] K. Takenaka and K. Hatanaka. Efficient aerodynamic design of complex configurations by patch-surface approach. *Journal of Aircraft*, 48(5), 2011.
- [31] M. Widhalm, J. Brezillon, Č. Ilić, and T. Leicht. Investigation on Adjoint Based Gradient Computations for Realistic 3D Aero-Optimization. In *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, 2010.
- [32] M. Widhalm, A. Ronzheimer, and M. Hepperle. Comparison between gradient-free and adjoint based aerodynamic optimization of a flying wing transport aircraft in the preliminary design. In *25th Applied Aerodynamics Conference*, number AIAA-2007-4060, 2007.