Evolutionary and Deterministic Methods for Design,
Optimization and Control with Applications to Industrial and Societal Problems
EUROGEN 2005
R. Schilling, W.Haase, J. Periaux, H. Baier, G. Bugeda (Eds)

DISCRETE ADJOINT OF THE NAVIER-STOKES EQUATIONS FOR AERODYNAMIC SHAPE OPTIMIZATION

J. Brezillon, R. Dwight

Institute of Aerodynamics and Flow Technology
DLR, Braunschweig, Germany
Email: joel.brezillon@dlr.de, richard.dwight@dlr.de - Web page: http://www.dlr.de/as

Key words: Optimization, Aerodynamics, Adjoint, Profile, High-Lift, Numeric.

Abstract. *A discrete adjoint of the Navier-Stokes equations has been developed in the unstructured finite-volume solver the DLR-TAU-code. The method consists of the explicit construction of the exact Jacobian of the spatial discretization with respect to the unknown variables allowing the adjoint equations to be formulated and solved. A wide range of the spatial discretizations available in TAU have been differentiated, including the Spallart-Almaras-Edwards one-equation, and the Wilcox k-omega two-equation turbulence models.*
The aim of this paper is to give an overview of the capabilities of the discrete adjoint to perform aerodynamic shape optimization in viscous flow. The strategy developed is extensively validated on 2D cases. The adjoint based design method is first validated by comparing the gradients of the drag, lift and pitching moment it produces with the approximate gradients obtained by finite-differences. Then the accuracy and efficiency of the approach are demonstrated for transonic airfoil design by considering geometric as well aerodynamic constraints, single- as well as multi-point design. Finally, the flap design of a multi-element airfoil in take off configuration confirms the capability of the discrete adjoint to solve wide range of aerodynamic problems.

1 INTRODUCTION

Numerical shape optimisation will play a critical strategic role in future aircraft design. It offers the possibility of designing or improving aircraft components with respect to a prespecified figure of merit, subject to geometrical and physical constraints. However, the extremely high computational cost of straightforward methodologies currently in use prohibits the application of numerical optimisation for industrially relevant problems. Optimisation methods based on the calculation of the derivatives of the cost function with respect to the design variables suffer from high computational costs if many design variables are used. However, these gradients can be efficiently obtained by solution of the adjoint flow equations. Within the framework of the German aerospace research program MEGADESIGN [7], the DLR plays an active role in developing efficient numerical methods for shape design and the development of the discrete adjoint approach is one important issue.

The aim of the paper is to give an overview of the capability of the discrete adjoint to perform aerodynamic shape optimization in viscous flow. In the next chapter, the discrete adjoint formulation and its implementation in the DLR-*TAU* code is outlined. The developed strategy is then validated on realistic 2D cases. The first step in validating the adjoint based design method is to compare the gradients of the drag, lift and pitching moment it produces with the approximate gradients obtained by finite-differences. Then the capability of the approach is demonstrated on both single- and multi-element airfoils, under consideration of both geometric and aerodynamic constraints. This paper will assess the superiority of the adjoint approach in terms of accuracy and efficiency as compared to other approaches, and its limits for solving more challenging industrial problems will be outlined. Future developments needed to overcome these problems will be proposed at the end of the paper.

2 DISCRETE ADJOINT FORMUATION

The advantage of the adjoint method is its ability to evaluate the gradient of a single cost function with respect to a large number of design variables with an effort that scales weakly with the number of design variables.

2.1 Theory of the Method

Let the optimization problem be stated as

$$\min_{w.r.t.\ D} I(W, X, D), \tag{1}$$

under the constraint

$$R(W, X, D) = 0, \tag{2}$$

where here I is a cost function such as lift or drag, D is a vector of design variables, $X(D)$ the computational mesh, $W(X, D)$ the vector of flow variables, and R a discretization of the Navier-Stokes equations on X.

The goal is then to find $\mathrm{d}I/\mathrm{d}D$. By stating the constraint in terms of the discretized flow equations – rather than the continuous Navier-Stokes equations on a given continuous domain

– the discrete variety of the adjoint procedure will be derived.

Consider the *Lagrangian*:

$$L = I + \Lambda^T \cdot R, \tag{3}$$

where Λ is a vector of *adjoint variables* of the same magnitude as W (and hence R). Since equation (2) is valid for all D, the Lagrangian is identical to I and the same holds for their derivatives with respect to D, for all D and all Λ. Differentiating L by the chain rule and rearranging gives:

$$\begin{aligned}\frac{dL}{dD} &= \left\{\frac{\partial I}{\partial W}\frac{dW}{dD} + \frac{\partial I}{\partial X}\frac{dX}{dD} + \frac{\partial I}{\partial D}\right\} + \Lambda^T \left\{\frac{\partial R}{\partial W}\frac{dW}{dD} + \frac{\partial R}{\partial X}\frac{dX}{dD} + \frac{\partial R}{\partial D}\right\}, \\ &= \left\{\frac{\partial I}{\partial W} + \Lambda^T \frac{\partial R}{\partial W}\right\}\frac{dW}{dD} + \left\{\frac{\partial I}{\partial X} + \Lambda^T \frac{\partial R}{\partial X}\right\}\frac{dX}{dD} + \left\{\frac{\partial I}{\partial D} + \Lambda^T \frac{\partial R}{\partial D}\right\}.\end{aligned} \tag{4}$$

All partial derivatives in the above equation may be evaluated by hand, their numerators all being known and explicit functions of their denominators. The quantities dW/dD and dX/dD however must be found by applying the chain rule to equation (2) and the grid deformation equations respectively – thus resulting in a separate equation for each D. In the case of the grid deformation equations this is not critical as the equations are typically simple and rapidly solved, but in the case of the flow equations this evaluation would dominate the effort of the gradient calculation. Instead eliminate dW/dD from (4) by setting Λ such that

$$\left(\frac{\partial R}{\partial W}\right)^T \Lambda = -\left(\frac{\partial I}{\partial W}\right)^T, \tag{5}$$

a linear equation for the adjoint variables.

The remaining unknown term dX/dD may be reliably evaluated by finite-differences as

$$\begin{aligned}\frac{\partial I}{\partial X}\frac{dX}{dD}\Delta D &\approx I(W, X(D+\Delta D), D) - I(W, X, D), \\ \frac{\partial R}{\partial X}\frac{dX}{dD}\Delta D &\approx R(W, X(D+\Delta D), D) - R(W, X, D),\end{aligned} \tag{6}$$

although it is possible to use an adjoint approach here as well, to avoid the necessity of deforming the grid in response to each shape-modifying design variable [8].

2.2 Implementation in the flow solver *TAU*

There are two principal difficulties in the implementation of the above method: the evaluation of the partial derivatives in (4), and the solution of the resulting linear system (5). Typically it is the case that discretized residual of the flow equations R may be written explicitly in terms of W, X and D, and hence the partial derivatives with respect to these terms may be calculated per hand straightforwardly; while at the same time R is extremely complex (as implemented in the DLR *TAU*-Code [6]), containing non-linear fluxes, gradient

calculations, limiters, state equations, turbulence models, etc., etc., and thus the differentiation is extremely tedious and time-consuming.

To provide an insight into the implementation aspect, the somewhat personal process developed by the authors after much trial and error is offered here, in the hope that it may guide initial efforts of others. It may be summarized in the following six steps:

1. ***Divide*** the code for the non-linear residual R into parts that may be differentiated independently – either because R is a simple sum of e.g. inviscid and viscous fluxes, or by means of the product and chain rules. All parts of the code that do not influence the solution of $R = 0$, are superfluous.
2. ***Copy out*** the definition for e.g. a particular flux function F, directly from the code onto paper. Begin with the definition of F in terms of intermediate variables – at the end of the code fragment – and proceed upwards. This order will aid the application of the chain rule in step 4. In order that the resulting Jacobian be the exact derivative of R it is the authors' experience that it is not sufficient to take flux function definitions from technical reports, articles etc., as the reality is often considerably different.
3. ***Decide*** which parts of the definition of F to neglect as insignificant, too time-consuming to differentiate, or as a means of providing a more efficient implementation. This step is optional.
4. ***Differentiate*** the simplified version of F on paper, which should now be a matter of working top to bottom on the page. Making use of the chain rule wherever possible (i.e. for every intermediate variable) tends to simplify expressions and helps avoid errors. Given a one page definition, a roughly two page derivative can be expected.
5. ***Implement*** the derivative calculated in step 4. As a basis use the original function for F, as many intermediate variables will appear in undifferentiated form in the expression for the derivative; also the inputs should be identical. Write original expressions and their derivatives together. Comment the derivative with the original function name, w.r.t. which variables are differentiated, and any assumptions made during step 3.
6. ***Compare*** the hand-coded derivative against a finite-difference approximation using the original function. You have made at least one mistake: probably in step 5.

This must be done not only for every element of R but also for each desired I, see (5). Note that the derivatives are considerably simplified by choosing primitive variables as working variables; because the equations remain in conservative form this choice has no effect on the final adjoint solution. As each component of the derivative of R is calculated a contribution is made to the explicitly stored Jacobian matrix.

Storing the matrix explicitly has the disadvantage of requiring approximately six times the memory of the standard *TAU*-code, see table 1, reducing the capacity of a node with 1GB of memory from 2 million points to 300 thousand points, which given the relatively small size of optimization problems compared to single flow calculations, is not a serious handicap. However this result is only valid in 2D.

	Standard *TAU*	+ Jacobian Storage	+ Lin. Solver Storage
Memory (Bytes)	25M	165M	290M
Factor increase	x1	x6.6	x11.6
Points in 1GB	$2x10^6$	$300x10^3$	$170x10^3$

Table 1 : Comparison of the peformance

2.3 Solution of the adjoint equation

Given an explicitly stored Jacobian – which must only be constructed once per gradient evaluation, even for multiple cost functions – assessment of the residual of the linear equation (5) reduces to a matrix-vector product. Further the availability of the matrix allows the application of the ILU preconditioner to a Krylov subspace method, a procedure which is often preferred in aerodynamic applications for the solution of the linear systems resulting from a Newton iterative method for the non-linear equations [3] and again this preconditioner must only be constructed once.

As the eigenvalues of the Jacobian and transpose Jacobian are identical, the convergence rates achieved with Krylov subspace methods such as GMRES are guaranteed to be identical, and so the experience gained applying these methods in implicit schemes may be carried over [4]. The result is that the calculation of the adjoint solution requires only approximately 10% of the time required for a non-linear flow calculation - and so forms an insignificant component of the total time for the optimization. Given that the gradient is much cheaper than the line search, optimization strategies that rely on many gradient evaluations, such as Quasi-Newton Trust Region (QNTR), become more attractive. The ILU preconditioner and GMRES method have an associated memory cost, in this case the requirements for ILU with four levels of fill-in and GMRES(30) are given in table 1.

3 VALIDATION OF THE DISCRETE ADJOINT

The first step in validating the adjoint based design method is to compare the gradients of the drag, lift and pitching moment produced with those obtained by finite-differences. One of the problems of such a comparison comes from the inaccuracy of the gradient obtained with finite-differences, in particular the well known problem of the dependence of the result on the chosen finite step size. Too small a change in the geometry would produce too a change in the aerodynamic loads which could be dominated by the numerical noise: the resulting gradient is then inexact. In contrast, too big a step does not allow neglection of the second-order truncation error and no agreement with the adjoint approach could be achieved. Therefore, to find an adequate step, a lot of trial and error is required. In contrast, the finite-differences used in the adjoint gradient evaluation (6) are relatively step-size independent as a result of the higher level of convergence achieved in the mesh deformation equations on one hand, and their linear nature on the other.

The case considered for the validation of the discrete adjoint is the RAE2822 airfoil in viscous transonic flow. The flight conditions are M_∞=0.73, α=2.8° and Re=$6.5x10^6$ as Mach number, angle of attack and Reynolds Number respectively. The flow is computed using the

TAU-code which is an unstructured grid finite-volume Reynolds Averaged Navier-Stokes solver. It operates on 2D grids consisting of triangles and quadrilaterals or 3D grids of hexahedra, prisms, pyramids and tetrahedra, of which a dual grid is constructed in a preprocessing step so that the metric is cell-vertex. The dual grids use an edge-based data-structure so that the solver itself has only edge-to-point connectivity information. For the present flow simulation, the 2D viscous mode is employed with the Spallart-Almaras-Edwards one-equation turbulence model.

The design variables are 30 control points which deform the thickness (the 10 first design variables) and the camberline of the airfoil (the remainder). A more precise description of the parameterization is given in Section 4.1. The finite-differences gradients are provided by forward differencing. Thus 31 viscous solutions are required to obtain the complete gradients. In order to avoid noise in the evaluation of the aerodynamic coefficients, a convergence of 8 orders of magnitude in RMS density residuals is necessary for each computation.

For the adjoint approach just one viscous and three corresponding discrete adjoint solutions are necessary. As mentioned in the previous chapter, the calculation of the adjoint solution requires only approximately 10% of the time required for a viscous flow calculation. By including the metric sensitivities in equation (6), the total time needed for the evaluation of all four gradients is equivalent to 50% of a single viscous flow computation.

Figure 1 shows the components of the gradients of the total drag, viscous drag, lift, and pitching moment, according to the design variables. All four figures show a very good conformity between the finite differences and adjoint gradients.

As expected, the adjoint approach allows considerable CPU time saving and it took 20 times less time to evaluate the 4 gradients than with finite differences.

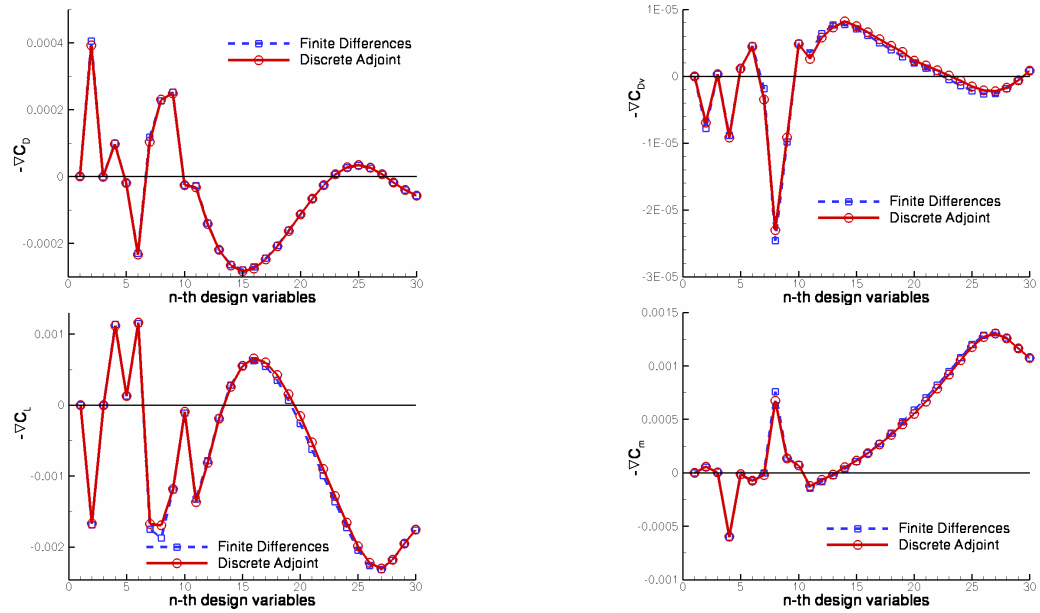

Figure 1: Gradient of the total drag, viscous drag, lift and pitching moment for the RAE2822 airfoil (Thickness and Camberline parameterization - M_∞=0.73,α=2°,Re=6.5×10^6)

4 AIRFOIL OPTIMISATION

The aim of this chapter is to assess the capability of the adjoint approach to optimize the airfoil drag in transonic flow, considering geometric as well as aerodynamic constraints, and single- as well as multi-point conditions. For all optimizations, the aerodynamic conditions are the same as in the previous section except for the multi-point design where two additional conditions are considered. Before showing the results obtained on the RAE2822 airfoil, the main components of the aerodynamic chain (parameterization, mesh generation and flow solver) are described.

4.1 Optimization strategy

Airfoil design is deeply influenced by the parameterization of the geometry and without any geometrical constraint, airfoil improvements can easily be made. For instance, decreasing the drag is straightforwardly obtained by decreasing the maximum thickness of the airfoil and a desired lift can be reached by modifying the angle of the trailing edge. In order to avoid such a design, geometrical constraints have to be retained in order to really assess the capability of the optimization chain for airfoil design. Some challenging designs were proposed in the European project AEROSHAPE [9] and we retain here the geometrical constraints set for the design of the RAE2822 test case:

- the maximum thickness is frozen,
- the thickness at 5% chord should not be less than 96% of the initial one,
- the leading edge radius should be equal to or greater than 90% of the initial radius,
- the trailing edge angle should not be less than 80% of that of the initial geometry.

An appropriate parameterization was developed at DLR and successfully used to handle such a problem [1]. The airfoil parameterization is split into two parts, allowing the deformation of the camberline independently of the thickness. For smooth changes of the camberline, Hicks-Henne bump functions are used. The thickness is parameterized by B-Splines, properly set to automatically fulfill all geometrical constraints. For the single point optimization, only the change of camberline with 20 design variables is activated while for the multi-point optimization 29 design variables are used in order to broaden the design space.

The grid deformation algorithm used for this case is based on an empirical advancing front algorithm with special considerations for solid body motions. As a result it can only be applied robustly to simple geometries for small deformations, but has the advantage of being extremely rapid. The baseline mesh is shown in figure 2.

The aerodynamic state is then evaluated using the *TAU*-code in viscous mode with the Spallart-Almaras-Edwards one-equation turbulence model. The required gradients are computed using the corresponding discrete adjoint as described in Section 2. The aerodynamic state and the gradient information are then passing to the user defined optimizer.

The SynapsPointer® Pro optimization framework [5] has been retained as optimization framework because it allows an easy construction of the aerodynamic chain and manages the connection to user defined optimizers. Furthermore, this framework automatically runs processes in parallel which allows a decrease of the turn around time.

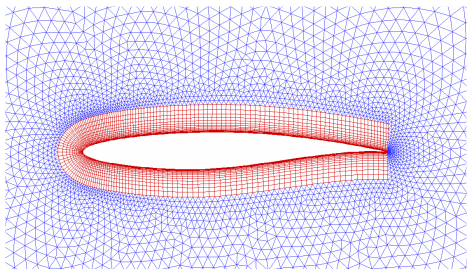

Figure 2: Details of the mesh around the RAE 2822 airfoil.

4.2 Unconstrained optimization

As first test case, we consider drag reduction at constant lift. The desired lift is here maintained by adjusting the angle of attack. This is an unconstraint optimization in the sense that the aerodynamic constraint is automatically handled by the *TAU*-code. For solving this problem, three optimization strategies of increasing efficiency are considered: steepest descent, conjugate gradient and Quasi-Newton Trust Region (QNTR). The evolution of the aerodynamic coefficients according to the call of the flow solver is given in figure 3. One can see that the *TAU*-Code is able to maintain the desired lift by adjusting the angle of attack. In all cases, a decrease of more than 60 drag counts is obtained and the main characteristics of each optimization are recorded in table 2. As expected, the steepest descent is the least efficient method with the lowest drag improvement. In contrast, the QNTR approach finds a better optimum with 5 times fewer flow evaluations but with more gradient evaluations, made possible by the use of the discrete adjoint approach for the gradients. The conjugate gradient is also able to attain this best optimum but with more state evaluations and fewer gradient evaluations. Depending of the optimization strategies, the same optimizations with finite differences would be 3 to 8 times slower. The comparison of the pressure distribution in figure 4 shows that the final design is a shockless profile. The pressure distribution is also characterized by an increase in the suction peak and an increase of the rear loading in order to keep the lift constant. An analysis of the optimal geometry evidences a lower camber in the front part and a higher one in the rear part. Only small differences are noticeable between all designs. This test case gives a first idea of the efficiency of the adjoint approach coupled with appropriate optimization strategies. The good convergence of the optimization process and the resulting design confirm the high accuracy of the gradients computed with the discrete adjoint.

	Number of state evaluations	Number of gradients	Drag improvement in drag counts
Steepest descent	99	15	-60.5 (36,4 %)
Conjugate Gradient	74	10	-63.8 (36,8 %)
QNTR	21	21	-63.8 (36,8 %)

Table 2 : Comparison of peformances for the unconstrained optimization test case

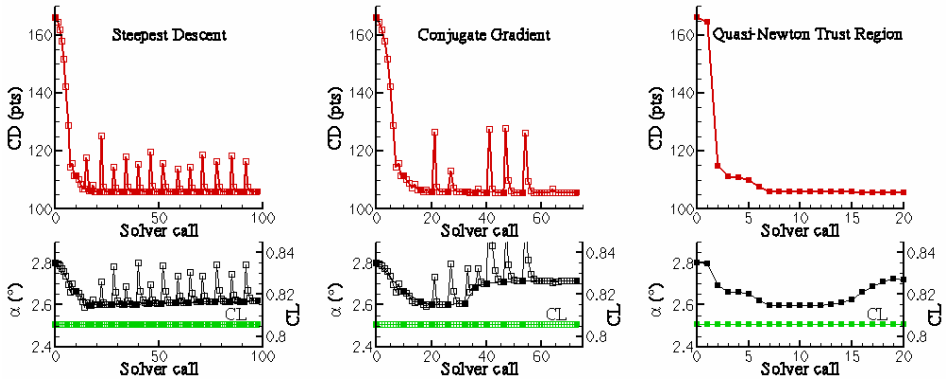

Figure 3: Evolution of the aerodynamic coefficients.

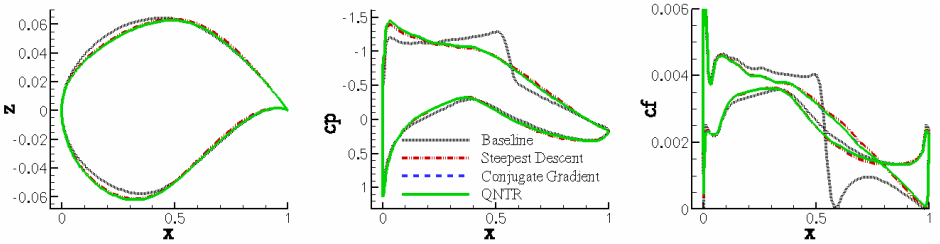

Figure 4: Initial and final geometries, pressure distribution and skin friction.

4.3 Single point constrained optimization

A second test is drag reduction at constant lift, pitching moment *and* angle of attack. The optimization strategy is now a projected steepest descent where the search direction is a projection of the gradient of the drag onto the hyperplane normal to the gradients of the lift and the pitching moment. Following this direction, one expects to decrease the drag and to maintain the other values, so long as second-order effects remain small. Here, the accuracy of the gradients plays a crucial role in the variation of the constraints. The left part of figure 5 shows the evolution of the drag, lift and pitching moment over the optimization steps. An optimization step consists of evaluation of the aerodynamic state, the necessary gradients and the line search for finding a better minimum. In order to converge the optimization problem, 11 evaluations of gradients and 63 evaluations of the aerodynamic coefficients are necessary. This optimization is more than 4 times faster than the classical finite-differences approach.

Upon convergence, a decrease of more than 60 drag counts is achieved while both the constraints (lift and pitching moment) remain unchanged. Due to the non-linearity of this optimization problem and to the magnitude of the design step at the beginning of the optimization, small deviations in the constraints do occur. Therefore the pitching moment and the lift are corrected at the third step to retrieve their initial values and this correction introduces an increase of the cost function as can be seen on the left of figure 5. The optimization procedure is then continued until convergence. As expected, the strong shock appearing on the RAE2822 airfoil is also eliminated at the end of the design process as shown on the right of figure 5.

It can be seen that this optimization behavior is identical to that previously performed on inviscid flow with a corresponding continuous adjoint [1].

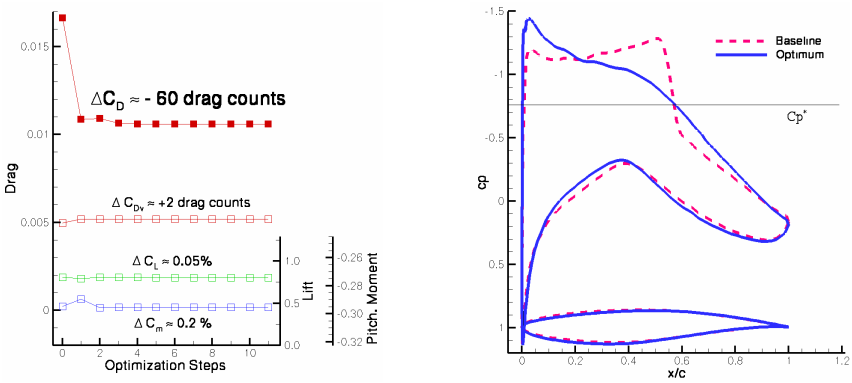

Figure 5: History and pressure distribution.

4.4 Multi-point optimization

The design problem is now the minimization of a linear combination of the drag at three flow conditions for the RAE2822 airfoil under aerodynamic and geometric constraints, as defined within the AEROSHAPE project. The aerodynamic design conditions with the corresponding weighting factor read as follow:

- First design point : α_1=2.8°, M_1=0.73, Re_1=6.5x10^6, W_1=2,
- Second design point : α_2=2.8°, M_2=0.75, Re_2=6.2x10^6, W_2=1,
- Third design point : α_3=1.8°, M_3=0.68, Re_3=5.7x10^6, W_3=1.

The geometrical constraints are those introduced in Section 4.1. Aerodynamic constraints are set at each design point and imposed that the lift is not allowed to decrease, the pitching moment is only allowed to vary in a range of ±2% and the angle of incidence is fixed.

For solving such problem, a robust optimization strategy is required and good experiences have already been achieved [1] with the “modified method of feasible direction”.

The complete optimization requires 8 optimizations steps, which represent 71 evaluations of the goal function (i.e. 71x3 aerodynamic computations) and 8x3x3 adjoint computations for the gradient of the objective function and constraints. About 17 hours wall clock time were necessary to perform all computations on a 3 nodes Linux cluster where each nodes computed a single design point. At the end of the optimization process, the two first design points show a decrease of about 40 and 50 drag counts respectively, whereas the drag for the third design point increases slightly by about 2 drag counts. All geometric and aerodynamic constraints are fulfilled. Table 3 summarizes the aerodynamic changes.

	Design pt 1	Design pt 2	Design pt 3
ΔC_D	-39.9 pts / -24.0%	-49.7 pts / -19.8%	+2.0 pts / +2.3%
ΔC_L	+4.5 %	+2.7 %	0.0 %
ΔCm	+1.9 %	-0.6 %	+1.0 %

Table 3 : Airfoil design for the multi-point test case.

The pressure distributions plotted in figure 6 show a reduction of the shock strength, which can be particularly seen on the main design point. An increase of the suction peak can be seen for all 3 design points.

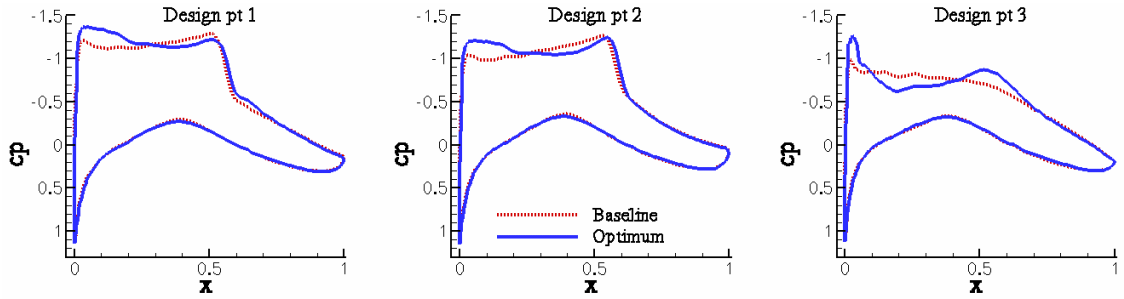

Figure 6: Pressure distribution for the multi-point design.

5 HIGH-LIFT FLAP DESIGN

The test case considered now is the flap design of a multi-elements airfoil in take-off configuration (M_∞=0.1715, Re=14.70x10^6). The initial geometry and the aerodynamic conditions were defined within the European project Eurolift II [10]. The parameterization and the structured mesh generation are identical to [2] and here adjusted for evaluating the flow with the *TAU*-code. The goal function is the drag reduction at constant lift; the lift is kept constant by changing the angle of incidence. For solving this problem, the conjugate gradient optimization strategy is used. The evolution of the drag, lift and angle of incidence according to the optimization step is given on the left of figure 7 and geometries and pressure distributions at the initial and optimized configuratrion are on the right. The optimization requires at least 10 steps to converge (42 state evaluations). The designed flap has 43 drag counts less at the same lift coefficient. This exercise has demonstrated the capability to use the discrete adjoint for solving a 2D high-lift problem.

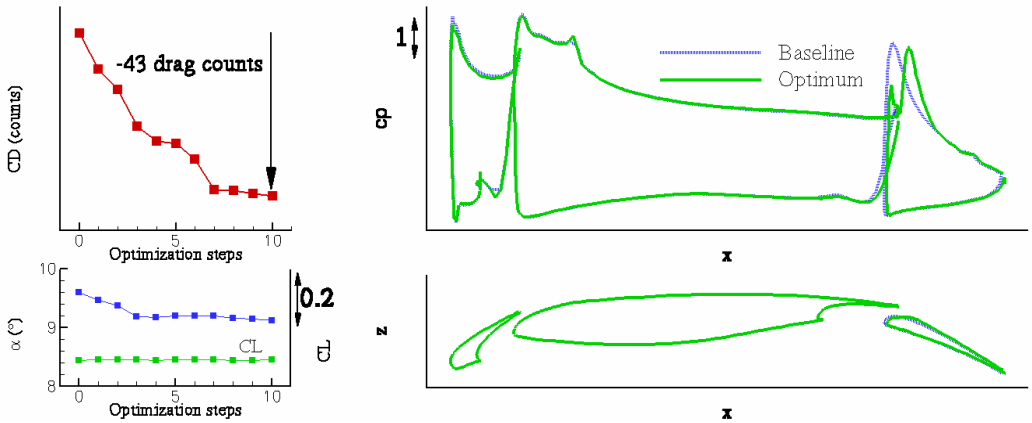

Figure 7: History and pressure distribution for the high-lift flap design.

6 CONCLUSION

The capability of the discrete adjoint to perform 2D viscous aerodynamic shape optimization has been demonstrated. For instance, combining the rapidly solved adjoint approach with the Quasi-Newton optimization strategy allows to optimize configurations using 20 design variables with a similar number of flow solutions.

The extension of the approach to 3D configurations presents problems in the formulation and solution of the adjoint system. It has been seen that the memory requirements of the adjoint method heavily dominate those of the flow solver itself, and effect that is more pronounced in 3D due the increase number of neighbors of a point, particularly in tetrahedral meshes. Given that the explicit system matrix may therefore not be explicitly available it is also necessary to develop an alternative means of solving the adjoint equations. It is known that iterative methods for the original discretization of the flow equations may be adapted for solution of the adjoint equations though a different sort of adjointing procedure and this method will be pursued in future.

ACKNOLEDGEMENTS:

The author wants to thanks J. Wild for his support for the high-lift mesh generation and the BMWA for financial support under the grant number MEGADESIGN/20A0302A.

7 REFERENCES

[1] Brezillon, J., Gauger, N.R. “2D and 3D aerodynamic shape optimization using the adjoint aproach”, Aerospace Science and Technology, 8, 8, pages 715-727, 2004.

[2] Brezillon, J , Wild, J., “Evaluation of different optimization strategies for the design of a high-lift flap device”, EUROGEN 2005, Munich, September 2005

[3] Cantariti, F., Woodgate, M., Badcock, K., Richards, B “Solution of the Navier-Stokes equations in three dimensions using a fully unfactored method”, Glasgow University, Department of Aerospace Engineering, (Technical Report 9908), 1999.

[4] Dwight, R.P., “An implicit LU-SGS scheme for finite-volume discretizations of the Navier-Stokes equations on hybrid grids”, DLR FB 2005-05, ISSN 1434-8454, 2005.

[5] Frommann O., “SynapsPointerPro V2.50”, Synaps Ing.-GmbH, Germany, 2002.

[6] Gerhold, T., Galle, M., Friedrich, O., Evans, J, “Calculation of Complex 3D Configurations Employing the DLR *TAU*-Code”, AIAA paper 97-0167, 1997.

[7] Kroll, N., Gauger, N. R., Brezillon, J., Becker K., Schulz V., “Ongoing Activities in Shape Optimization within the German Project MEGADESIGN”, ECCOMAS 2004, Jyväskylä, Finland, 24-28 July 2004.

[8] Nielsen, E., Kleb, B., “Efficient construction of discrete adjoint operators on unstructured grids using complex variables”, AIAA 2005-0324, Reno, USA, 2005.

[9] Selmin, V. ,“Multi-point Aerodynamic Shape Optimization: The AEROSHAPE Project”, Proceedings of ECCOMAS, Barcelona, Spain, 2000.

[10] Wild J., Mertins R., Quagliarella D., Brezillon J., Quest J., Amoignon O., Moens F., “Applying numerical optimization to realistic high-lift design of transport aircraft-An overview of the aerodynamic design optimization investigation with the EUROLIFT II project”, EUROGEN 2005, Munich, September 2005.