13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference 13 - 15 Sep 2010, Fort Worth, Texas

Investigation on Adjoint Based Gradient Computations for Realistic 3d Aero-Optimization

Markus Widhalm[*], Joël Brezillon[*], Časlav Ilić [*] and Tobias Leicht[*]

German Aerospace Center (DLR), Braunschweig, D-38108 Germany.

A discrete adjoint method for efficiently computing gradients for aerodynamic shape optimizations is presented. The chain itself involves an unstructured mesh Reynolds-Averaged Navier-Stokes solver, and is suitable for the optimization of complex geometries in three dimensions. In addition to the discrete flow adjoint the method introduces a second adjoint equation for the mesh deformation. Using the adjoint chain it is possible to evaluate the gradients of a cost function for the cost of one adjoint flow solution and one adjoint volume mesh deformation, without performing any (forward) mesh deformation. By choosing a suitable mesh deformation operator, like linear elasticity, the chain may be readily constructed by hand. Furthermore, this adjoint chain can be subsequently used with parameterized surface grids. The accuracy and the computational savings of the resulting procedure is examined for the gradient-based shape optimization of a wing in inviscid flow.

I. Introduction

With the increasing availability, efficiency and reliability of adjoined versions of flow solvers, the dominant cost involved in setting up and performing a gradient-based optimization has expanded from flow sensitivities towards other aspects of the problem.

Only a single adjoint flow computation must be performed in order to evaluate sensitivities of a single cost function with respect to any number of design variables N_d, however if a variable describes a geometric change then the sensitivities of the volume mesh with respect to the change in the surface mesh are needed; effectively the derivatives of the volume mesh deformation are required. In the past these quantities have been obtained using finite differences, by deforming the volume mesh with respect to each design variable in turn, incurring a cost of N_d mesh deformations per gradient evaluation.[1–3] This is a reasonable approach: the cost of a mesh deformation is typically much less than that of a flow solution, and its behavior might be expected to be more linear, suggesting that finite differences should be less sensitive to the choice of increment size.

However as the geometric complexity of optimization problems increases, this reasoning becomes increasingly less valid. If the cost of a mesh deformation is one nth of an adjoint solution, then mesh deformation will begin to dominate the total cost of a gradient evaluation at $N_d = n$. While n depends strongly on the algorithm used, volume mesh deformation is a difficult problem and there is always a trade-off to be made between efficiency and robustness, the latter being increasingly expensive for complex geometries. Furthermore, searching for an acceptable range of finite difference increment sizes, suitable for evaluating adjoint gradients, are time consuming and become even more difficult when switching from inviscid to viscous meshes. As a result, prior to starting each optimization, a parameter study must be performed on this increment. Finally, and most critically, the presence of the mesh deformation step in the gradient evaluation can include many geometry modifications, for instance parameterization of a wing shape with a fixed fuselage or the wing-fuselage intersection line varying as the wing varies. Another example is in 3D high-lift applications, where the topology of the domain can change as slats and flaps move.

An elegant solution to these issues was first proposed by Nielsen and Park[4] who suggested constructing the adjoint of the volume mesh deformation equations in addition to the flow adjoint. Thereby the sensitivities

[*] Research Scientist, Institute of Aerodynamics and Flow Technology - Dept. Numerical Methods, Braunschweig.

of the volume mesh with respect to the surface mesh are eliminated from the expression for the gradient, whose evaluation becomes almost completely independent of N_d. A single adjoint mesh deformation equation replaces the N_d forward mesh deformations.[5–7]

However the evaluation of the gradients of the cost function with respect to the design variables is still not complete, as the sensitivities of the surface mesh to the design variables is unknown. Again these may be evaluated by finite differences, or alternatively all surface mesh points may be used as design variables. However, the latter imposes restrictions on the type of geometry modifications possible, e.g. wing-fuselage intersection line changes are again excluded.

We pursue a similar approach to that of Nielsen and Park, and extend the usage of all surface mesh points by a CAD based surface parameterizations which has several advantages. As the control points do not have to coincide with the geometry or can even lie within the flow domain, definition of a parameterization is particularly simple. This admits the possibility of adaptive parameterizations, which vary during the optimization in order to better resolve regions of the geometry with a large influence on the cost function.[8] Another advantage is that in general smooth geometries will result from this approach.

The paper will first briefly discuss the evaluation of sensitivities using a general formulation of the volume mesh deformation operator. Then we introduce the discrete adjoint of the flow solver developed in previous work.[9] This procedure will then be applied to a linear elasticity deformation operator which may be written as a linear system, followed by an investigation of the solution vector on different types of meshes. Methods of obtaining the remaining terms in the expression for the gradient, in particular the derivatives of the flow discretization with respect to the mesh, will be discussed.

The adjoint chain will be first validated for a representative test case against gradients obtained using finite differences. In addition, the influence of replacing the linear elasticity deformation by a fast and robust volume deformation suitable for complex geometries is examined. Finally the effectiveness of the resulting tool will be demonstrated with application to a three-dimensional configuration. Benefits of the new approach will be examined quantitatively.

II. Sensitivities from the Adjoint Mesh Deformation

II.A. Adjoint of a Generic Deformation Operator

A general volume mesh deformation equation may be written

$$T\left[x^0, s^0\right](\tau(\alpha), \sigma(\alpha)) = 0, \tag{1}$$

where x^0 represents the initial volume mesh, s^0 its surface, τ the unknown volume mesh node displacements, and σ some known (possibly indirect) description of the surface mesh node displacements. In the above τ and σ have been taken to be functions of a geometry parameterization α, e.g. the control points from a spline based deformation. The two initial mesh parameters have been set in square brackets to emphasize that in the following we are concerned only with the behavior of T with respect to τ and σ.

Now a Lagrangian is constructed with respect to not only the flow equation $R(w) = 0$, but also this mesh deformation equation $T(\tau, \sigma) = 0$:

$$L(w, x, \alpha; \Lambda, \hat{\Lambda}) = I + \Lambda R(w, x, \alpha) + \hat{\Lambda} T(\tau, \sigma), \quad \forall \Lambda,\ \hat{\Lambda},$$

whereby two adjoint variables Λ and $\hat{\Lambda}$ appear. Differentiating with respect to α gives

$$\begin{aligned} \frac{\mathrm{d}L}{\mathrm{d}\alpha} &= \frac{\mathrm{d}I}{\mathrm{d}\alpha} + \Lambda \frac{\mathrm{d}R}{\mathrm{d}\alpha} + \hat{\Lambda} \frac{\mathrm{d}T}{\mathrm{d}\alpha}, \\ &= \left\{ \frac{\partial I}{\partial \alpha} + \frac{\partial I}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}\alpha} + \frac{\partial I}{\partial w}\frac{\mathrm{d}w}{\mathrm{d}\alpha} \right\} + \Lambda \left\{ \frac{\partial R}{\partial \alpha} + \frac{\partial R}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}\alpha} + \frac{\partial R}{\partial w}\frac{\mathrm{d}w}{\mathrm{d}\alpha} \right\} + \hat{\Lambda} \left\{ \frac{\partial T}{\partial \sigma}\frac{\mathrm{d}\sigma}{\mathrm{d}\alpha} + \frac{\partial T}{\partial \tau}\frac{\mathrm{d}\tau}{\mathrm{d}\alpha} \right\}. \end{aligned} \tag{2}$$

By noting that derivatives of x are identical to derivatives of τ:

$$\frac{\mathrm{d}x}{\mathrm{d}\alpha} = \frac{\mathrm{d}}{\mathrm{d}\alpha}(x^0 + \tau) = \frac{\mathrm{d}\tau}{\mathrm{d}\alpha},$$

terms involving $\mathrm{d}w/\mathrm{d}\alpha$ and $\mathrm{d}x/\mathrm{d}\alpha$ in Eq. (2). We can now group the terms in Eq. (2) that contribute to the same variation and write

$$\frac{\mathrm{d}L}{\mathrm{d}\alpha} = \left\{ \frac{\partial I}{\partial \alpha} + \Lambda \frac{\partial R}{\partial \alpha} + \hat{\Lambda} \frac{\partial T}{\partial \sigma}\frac{\mathrm{d}\sigma}{\mathrm{d}\alpha} \right\} + \left\{ \frac{\partial I}{\partial w} + \Lambda \frac{\partial R}{\partial w} \right\} \frac{\mathrm{d}w}{\mathrm{d}\alpha} + \left\{ \frac{\partial I}{\partial x} + \Lambda \frac{\partial R}{\partial x} + \hat{\Lambda} \frac{\partial T}{\partial \tau} \right\} \frac{\mathrm{d}x}{\mathrm{d}\alpha}, \tag{3}$$

If we set the term multiplying $dw/d\alpha$ and $dx/d\alpha$ to zero and choosing the adjoint variables Λ and $\hat{\Lambda}$ arbitrary, we obtain two adjoint equations: an adjoint flow problem

$$\frac{\partial I}{\partial w} + \Lambda \frac{\partial R}{\partial w} = 0 \tag{4}$$

for unknowns Λ, and an adjoint mesh deformation problem

$$\frac{\partial I}{\partial x} + \Lambda \frac{\partial R}{\partial x} + \hat{\Lambda} \frac{\partial T}{\partial \tau} = 0, \tag{5}$$

for unknowns $\hat{\Lambda}$ given Λ. After solving both equations we are left with the total derivative of the cost function I as a function of the design parameters. Given the solution of the above equations the cost function gradient may be written

$$\frac{\mathrm{d}I}{\mathrm{d}\alpha} = \frac{\mathrm{d}L}{\mathrm{d}\alpha} = \frac{\partial I}{\partial \alpha} + \Lambda \frac{\partial R}{\partial \alpha} + \hat{\Lambda} \frac{\partial T}{\partial \sigma} \frac{\mathrm{d}\sigma}{\mathrm{d}\alpha},$$

In case of no direct derivatives, e.g. angle of attack or Mach number, both derivatives

$$\frac{\partial I}{\partial \alpha} = \Lambda \frac{\partial R}{\partial \alpha} = 0$$

vanish and the equation reduces further to

$$\frac{\mathrm{d}I}{\mathrm{d}\alpha} = \hat{\Lambda} \frac{\partial T}{\partial \sigma} \frac{\mathrm{d}\sigma}{\mathrm{d}\alpha}, \tag{6}$$

for design variables which modify the geometry only.

II.B. Adjoint Flow Problem

The flow adjoint applied here is a discrete adjoint[9–14] of the unstructured DLR TAU-Code.[15,16] The unsteady TAU-Code solves the compressible, three-dimensional Reynolds-Averaged Navier-Stokes equations using a finite volume formulation. All aspects of the spatial discretization including the Spalart-Allmaras and $k-\omega$ turbulence models have been differentiated by hand and are available in the adjoint discretization. In order to achieve a reasonable compromise between CPU time and memory requirements, the Jacobian $\partial R/\partial w$ is not stored in its entirety. Rather a reduced version containing only immediate neighbor information is stored, and the remaining entries are evaluated on-the-fly each time the *adjoint residual*, $\partial R/\partial w^T \Lambda$, is required. As a result an adjoint residual evaluation is slightly cheaper than a non-linear residual evaluation, and the adjoint code requires roughly three times the memory of the flow code.

The discrete adjoint system is solved with the adjoint of the iterative method applied to the non-linear equations. This choice guarantees that the adjoint will converge at exactly the same rate as the nonlinear solver. The iteration itself is a LU-SGS smoothed multigrid cycle.[9,17] Several such multigrid cycles are applied as a preconditioning operator for a Generalized Minimal Residual algorithm (GMRES).[18–20] The purpose of this Krylov method is stabilization of the linear solver, especially when the non-linear simulation could not been converged to an appropriate residual to fulfill $R(w) = 0$ for the adjoint flow equations.

Experience with the code has shown that, in particular in three-dimensions, consideration of the full linearization of the turbulence models in the adjoint results in exceptionally poor conditioning of the flux jacobian, and in some cases causes a divergence of the adjoint flow simulation. Hence, in the following the full linearization of the adjoint is used for two-dimensional cases and for three-dimensional applications the eddy-viscosity is taken to be constant. The error introduced in the cost function gradients by this approximation has been systematically examined.[21]

II.C. Linear Mesh Deformation Operators

Many common volume mesh deformation techniques, for example the spring analogy, truss analogy and linear elasticity,[22–24] may be written in the simple linear form

$$T(\tau, \sigma) = \Pi \sigma - A\left(x^0, s^0\right)\tau = 0, \tag{7}$$

where A is some square $3n^v \times 3n^v$ matrix (encoding for example the stiffness of grid edges), and Π is an $3n^v \times 3n^s$ matrix projecting a surface node vector onto a volume node vector. Here, n^v and n^s denote the number of volume and surface nodes, respectively. In particular, if $N^s(k)$ is the surface node number for entry k in the surface node displacement vector σ , $N^v(l)$ is the volume node for entry l in the volume node displacement vector τ and $\mathcal{P}(m)$ denotes the volume node corresponding to surface node m, then we typically have

$$\Pi_{ij} = \begin{cases} 1 & N^v(i) = \mathcal{P}(N^s(j)) \\ 0 & \text{otherwise} \end{cases}.$$

Given this form of T the adjoint deformation is particularly easy to construct, as $\partial T/\partial\tau = -A$ and $\partial T/\partial\sigma = \Pi$, so only the transpose of A is required to build the adjoint.

In this context we will consider a finite element discretization of the linear elasticity equations with linear finite elements on the original unstructured CFD mesh, thus the system is directly formulated in the required volume node displacements. Each node has a direct coupling, based on a stiffness analogy, to the nodes to which it is connected by an element.

Similar to what is done in Ref.[24] the stiffness of small elements is increased in order to yield a more robust scheme, however, the unphysical modification of the constitutive law described there is not appropriate in 3d and thus not considered. Boundary conditions on all bounding surfaces are incorporated in a form compliant with (7) by setting the diagonal entries corresponding to surface nodes to unity, whereas all other entries on the corresponding line of A are cancelled. Symmetry planes can be incorporated by only treating the displacement normal to the plane as a boundary condition with zero Dirichlet value, but with the displacement components within the plane kept free.

If the resulting sparse matrix is to be stored in memory it should be stored in its transpose form as the typically employed compressed row storage (CRS) format is not suited for transpose matrix vector products. The transpose can easily be accounted for during the insertion of the dense element-wise matrices into the sparse global one.

II.D. Constructing the Adjoint Mesh Deformation

Firstly, we have to solve the flow adjoint equations to derive Λ which holds the flow adjoint variables for all volume points. After that, we are able to solve the second adjoint equation, again for the complete flow domain. Then system of linear equations, Eq. (5), may be rewritten as:

$$\left(\frac{\partial T}{\partial \tau}\right)^T \hat{\Lambda} = -\left(\frac{\partial I}{\partial x}\right)^T - \left(\frac{\partial R}{\partial x}\right)^T \Lambda \tag{8}$$

The right hand side, shortly referred as **b**, of Eq. (8) is evaluated by hand differentiating. The objective function I and the flow residual R may be directly or indirectly dependent on the mesh metrics, like the control volume face normals n or the dual control volume vol. By subsequently using the chain rule, the complete derivatives may be written as follows:

$$\frac{\partial I(x,n,vol)}{\partial x} + \Lambda^T \frac{\partial R(x,n,vol)}{\partial x} = \frac{\partial I}{\partial x} + \frac{\partial I}{\partial n}\frac{\partial n}{\partial x} + \frac{\partial I}{\partial vol}\frac{\partial vol}{\partial x} + \Lambda^T \frac{\partial R}{\partial x} + \Lambda^T \frac{\partial R}{\partial n}\frac{\partial n}{\partial x} + \Lambda^T \frac{\partial R}{\partial vol}\frac{\partial vol}{\partial x} \tag{9}$$

The derivatives from the objective function I, like drag, lift or pitching moment, are split into a pressure and viscous part. While the pressure part is only effective on the surface and otherwise zero, the viscous part involves additional derivatives from interior points whenever gradients from the flow variables are needed to evaluate the viscous forces. Considering the computation of surface forces, which are integrated with the surface normals n_s, the direct derivatives vanish and finally $\frac{\partial I}{\partial x}$ can be written as

$$\frac{\partial I}{\partial x} = \frac{\partial I_p}{\partial n_s}\frac{\partial n_s}{\partial x} + \frac{\partial I_v}{\partial n_s}\frac{\partial n_s}{\partial x} + \frac{\partial I_v}{\partial grad}\frac{\partial grad}{\partial n}\frac{\partial n}{\partial x} + \frac{\partial I_v}{\partial grad}\frac{\partial grad}{\partial vol}\frac{\partial vol}{\partial x} \tag{10}$$

where I_p and I_v corresponds to the pressure part and the viscous part, respectively. The last two terms in Eq. (10) refer to the gradient computation with a Gaussian integral theorem which depends on face normals n of the control volumes and their corresponding volume.

The derivatives of the flow residual R w.r.t. the mesh coordinates are evaluated with matrix-vector multiplications. Again, the flow residual can be split into an inviscid R_i and visous part R_v. In case of a central scheme the derivatives of the dissipation may be added additionally. The final derivatives for the flow residual can be written in short form:

$$\frac{\mathrm{d}R}{\mathrm{d}x} = \frac{\mathrm{d}R_i}{\mathrm{d}x} + \frac{\mathrm{d}R_v}{\mathrm{d}x} \tag{11}$$

The terms $\partial n/\partial x$ and $\partial vol/\partial x$ in Eq. (9) involve linearizations of algorithms which generate dual mesh information like face normals, areas and control volumes.

The evaluation of the derivatives are conducted by first solving the cost function I and matrix vector multiplications of the residuals w.r.t. face normals for the flow solver routines, like $\partial I/\partial n$, $\partial R/\partial n$ and $\partial R/\partial vol$. For the residual derivatives w.r.t normals this will generate an array with a size of number of all face normals times 3 for each coordinate. In case of a viscous RAE2822 two-dimensional grid, later on described in the section IV, it contains about 14.000 points and about 36.800 faces which imposes an increased memory consumption. In same manner the final vector b is assembled in the dual grid processing routines. For accuracy reasons the original flow solver and metric algorithms are differentiated by hand and no simplifications are incorporated to fulfill the duality principle, mentioned in previous work,[10,12,25] to establish an exact discrete adjoint solver.

The validation of each derivative component of the gradient is made by comparing with finite differences which was throughout the differentiation of the particular routines a straight forward procedure. A more accurate way was proposed by Martins[26] using the complex step derivative approximation which will be considered to validate the derivatives in the future.

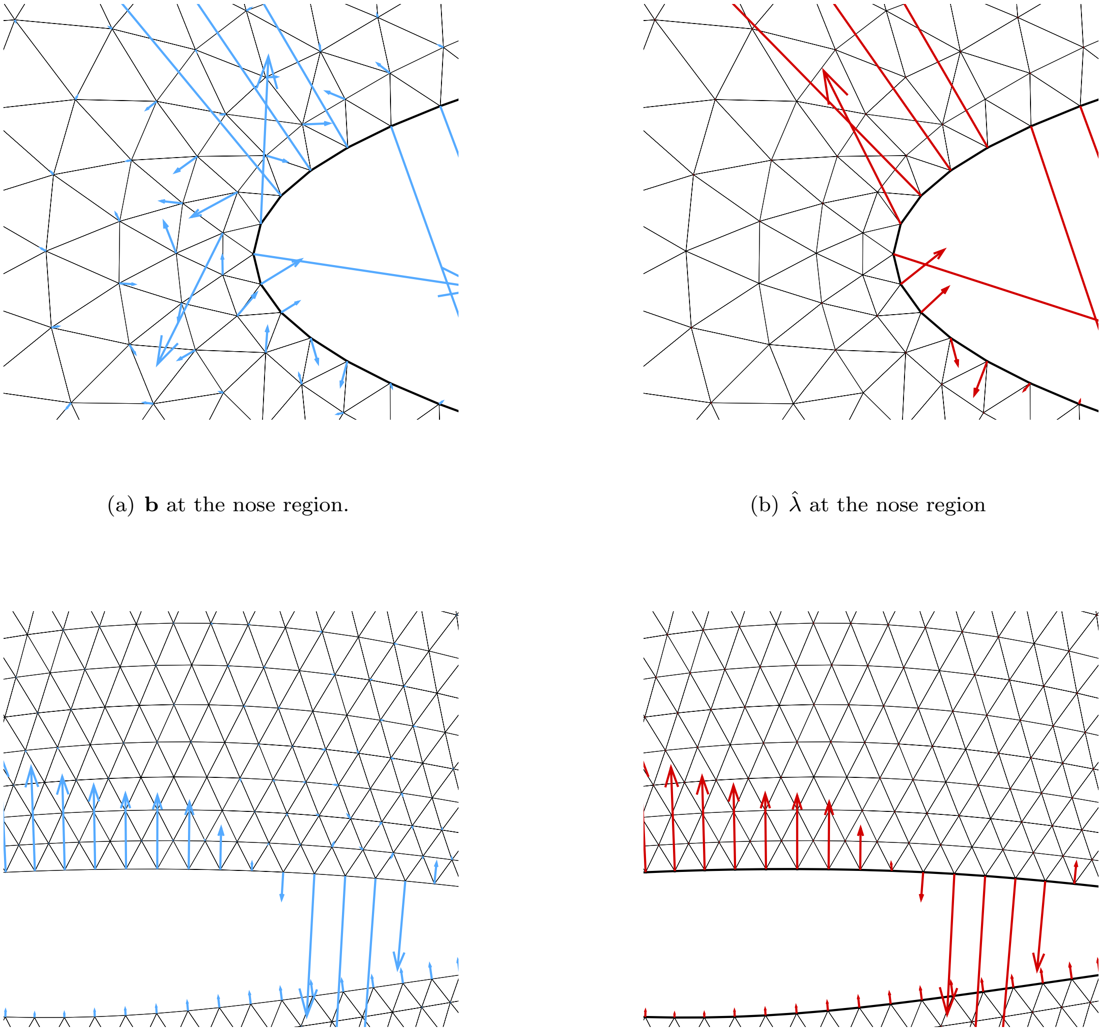
The system given in Eq. (8) is solved using the same GMRES algorithm as the flow adjoint.

II.E. Behavior of the right hand side vector b and solution vector $\hat{\Lambda}$ on different mesh types.

One reason for this investigation is to provide a better insight of how the terms in Eq. (8) influence the final solution vector $\hat{\Lambda}$ at the surface. Another reason comes from the chosen approach. The evaluation of sensitivities relies on the differentiation of the complete mesh dependencies while a surface formulation evaluates the sensitivities directly from information at the surface only. As grid sizes rise it may become memory expensive and reducing the amount of mesh points to an appropriate area around the aerofoil might be useful if memory is limited. As the matrix $\partial T/\partial x$ depends on grid point dimensions, it is difficult to extract the necessary information from the matrix to present the propagation of a deformation from the surface into the interior mesh. Apart from that, the vector **b** can be analyzed and depicted in figures to conclude about the influence on the final solution vector $\hat{\Lambda}$.

Firstly, we examine the influence on a typical unstructured mesh containing triangles used for inviscid flow simulation. The extracted details of the mesh can be seen in Figure 1. The mesh mainly consists of equally spaced elements and the initial grid sizing is comparatively large because of the neglected boundary layer region. The right hand side vector **b** is shown in Figure 1(a) and (c) and the vector $\hat{\Lambda}$ in Figure 1(b) and (d) for the nose and shock region. At the nose, Figure 1(a), the mesh metric influence represented by **b** reduces in about four to five neighboring cells to a negligible small value. The first neighboring cell from the surface causes the largest decay. In comparison to Figure 1(c), the shock region influences the surface values only. In the vicinity of the shock values are barely seen for the interior points. In conjunction with that behavior and independent of the values in the matrix $\partial T/\partial x$, the solution of the linear deformation equations will be mainly influenced from the values at the surface for that inviscid mesh. In that particular case the initial spacing is large in comparison to a realistic three-dimensional grid. It can therefore not be a priori stated that this will be the same for all inviscid meshes.

The next type of mesh is a grid used for viscous flows with the typical boundary layer resolution with very highly stretched elements. In the same manner, we have extracted the nose region and a vertical stack of points from the boundary layer region on the upper side of the surface mesh in the shock region, seen in detail in Figure 2. Figure 2(a) displays the vector **b** at the leading edge. The slow decrease of **b** from the surface to the 15^{th} to 18^{th} (total of 20) structured layer causes a larger influence from the neighboring interior points on the final solution vector. Remarkable is in Figure 2(c) for the shock region is that the vector facing downwards on the upper surface corresponds to the next neighbor of the surface point and the slightly smaller vector facing upwards is the one at the surface. The resulting solution vector for the extracted mesh detail is seen in Figure 2(b) and (d) and is nothing like the right hand side vector **b** on an inviscid mesh.



(a) **b** at the nose region.

(b) $\hat{\lambda}$ at the nose region

(c) **b** at the shock region.

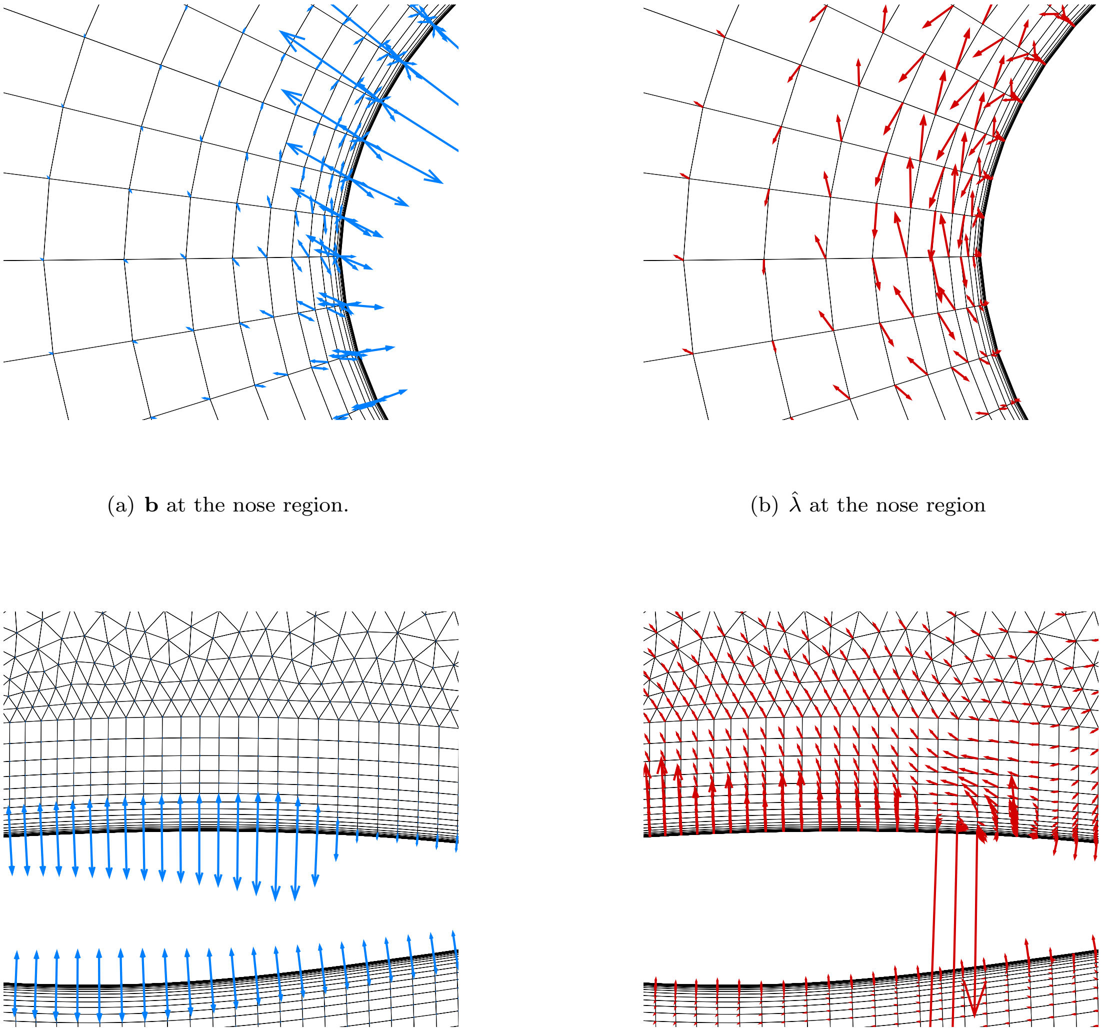
(d) $\hat{\lambda}$ at the shock region.

Figure 1. Right hand side vector b, left, and solution vector $\hat{\Lambda}$, right, for the nose and shock region displaying the propagation from the surface into the interior mesh points for a inviscid RAE2822 aerofoil. The cost function is the drag coefficient.

Due to this dependency of the mesh sensitivity on the complete boundary layer it is non-practical to establish the effort for reducing the mesh size to a certain area around the aerofoil.

Furthermore we present the distribution of the right hand side vector **b** and solution vector ($\hat{\Lambda}$) on the surface for the two types of meshes. Figure 3 shows the vectors for a mesh around an inviscid RAE2822 aerofoil. The flow simulation was performed for transonic flow conditions and the surface pressure distribution is depicted as a reference for the shock position. The cost functions chosen were drag and lift. Obviously, both vectors are very similar in length and direction except at the leading edge and trailing edge where they differ. During this investigation it might be possible to replace the matrix $\partial T/\partial \tau$ by an identity matrix because of their small difference and the iterative solution process for the linear system can be omitted. Later in Section IV an optimization for both approaches are conducted for an inviscid ONERA M6 wing[27] to complete this investigation.

In comparison to the above the viscous solution behaves completely differently. The solution is shown again for a RAE2822 aerofoil with transonic flow conditions in Figure 4 for drag and lift as cost function. It



(a) **b** at the nose region.

(b) $\hat{\lambda}$ at the nose region

(c) **b** at the shock region.

(d) $\hat{\lambda}$ at the shock region.

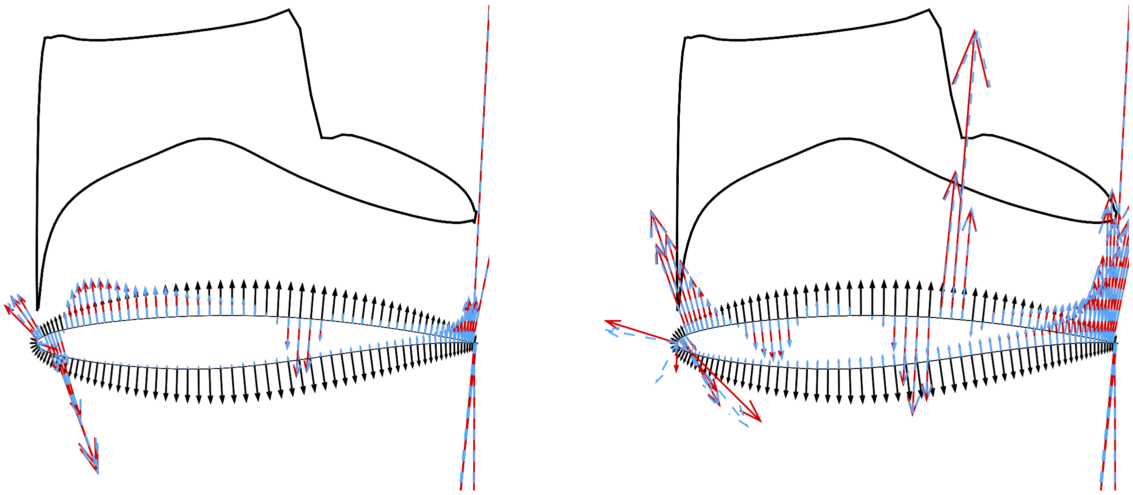
Figure 2. Right hand side vector b, left, and solution vector $\hat{\Lambda}$, right, for the nose and shock region displaying the propagation from the surface into the interior mesh points for a viscous RAE2822 aerofoil. The cost function is the drag coefficient.

should be further mentioned that the solution vector $\hat{\Lambda}$ got scaled by 100 to make them visible in one figure. In relation to inviscid meshes, any simplification for the matrix $\partial T/\partial \tau$ is not applicable for viscous meshes.

Another effect to the inviscid and viscous case is the non smoothness of vector $\hat{\Lambda}$ over the surface for both drag and lift. It becomes evident that it is difficult to use all surface mesh points directly in an optimization process. Firstly the surface will be erratically distorted and secondly any off design point will usually lead to a worse performance than the original configuration. Instead it becomes necessary to smooth the sensitivities by an additional numerical algorithm or to use a kind of parameterization which can provide a smooth surface for a subset of all surface points.

III. Surface Parametrization and Deformation

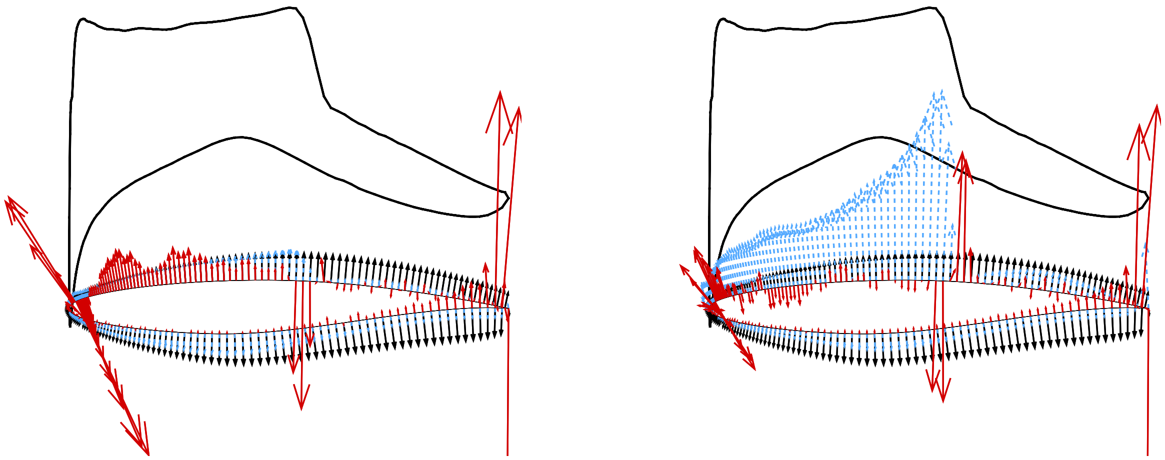
One of the advantages of the present approach is gaining the sensitivities on each surface mesh node but apart from that it has been shown using the sensitivities directly can lead to unusable geometries.[28, 29] As



(a) Drag

(b) Lift

Figure 3. Vector $\hat{\Lambda}$ (solid red), b (dashed blue), surface normals n (solid black) with the pressure distribution over the surface of an inviscid RAE2822 aerofoil.



(a) Drag.

(b) Lift.

Figure 4. Vector $\hat{\Lambda}$ (solid red), b (dashed blue), surface normals n (solid black) with the pressure distribution over the surface of a viscous RAE2822 aerofoil for drag (left) and lift (right).

more clearly seen in Figure 3 and Figure 4 the vector $\hat{\Lambda}$ is not smooth over the surface. To avoid an irregular surface we have decided for a parameterization technique.

In this approach the computation of mesh sensitivities is dependent on the deformation of the volume grid, which furthermore depends on the surface grid deformation. The surface grid deformation is then performed via freeform deformation (FFD).[30–32] The parameterization is represented by the coordinates of the lattice box points controlling the freeform deformation. One of the main advantages of this approach is the ability to create smooth surfaces with a low number of lattice box points.

In general, freeform deformation is performed in two steps. At first, all vertex points of the initial surface grid in cartesian space are mapped into an initial trivariate B-spline volume, Figure 5 (a). In a second step the initial B-spline volume is replaced by a new B-spline volume with new knot points. All grid vertices are then remapped to cartesian space to yield a new surface. Consequently FFD can be deemed as an indirect method since the changes of the shape are at last determined by the changes of the control point lattices,

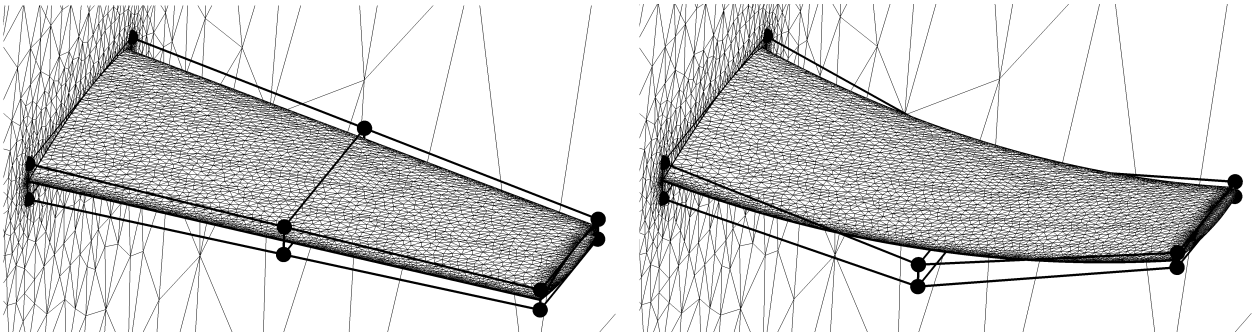


Figure 5. Initial (left) and deformed (right) ONERA M6 wing with the initial and adjacent lattice box

Figure 5 (b).

This approach made it necessary to split the mesh deformation module into two parts. The first part has been demonstrated in Figure 5 for the surface and the second part deforms the interior grid points in the flow domain with an advancing front algorithm.[16] The advancing front algorithm is fast and robust even if large surface movements are required.

The sensitivities of the control points of the freeform box are then evaluated by an inexpensive finite difference and summing up the contributions from the vector $\hat{\Lambda}$ for each deformed surface point. The value of the finite difference step size has been proved to be nearly independent over a wide range.

IV. Results

Firstly we present results of the correctness and accuracy of the adjoint chain for a representative test case. It is examined by comparing the gradients produced with those obtained using finite differences at the level of the flow solver. An investigation which uses parts of the derivatives from the right hand side of Eq. (8) will show their influence on the complete gradient. At the end, a gradient-based optimization of an ONERA M6 wing for inviscid flow is considered. Figure 6 shows the test cases used for validation purposes.

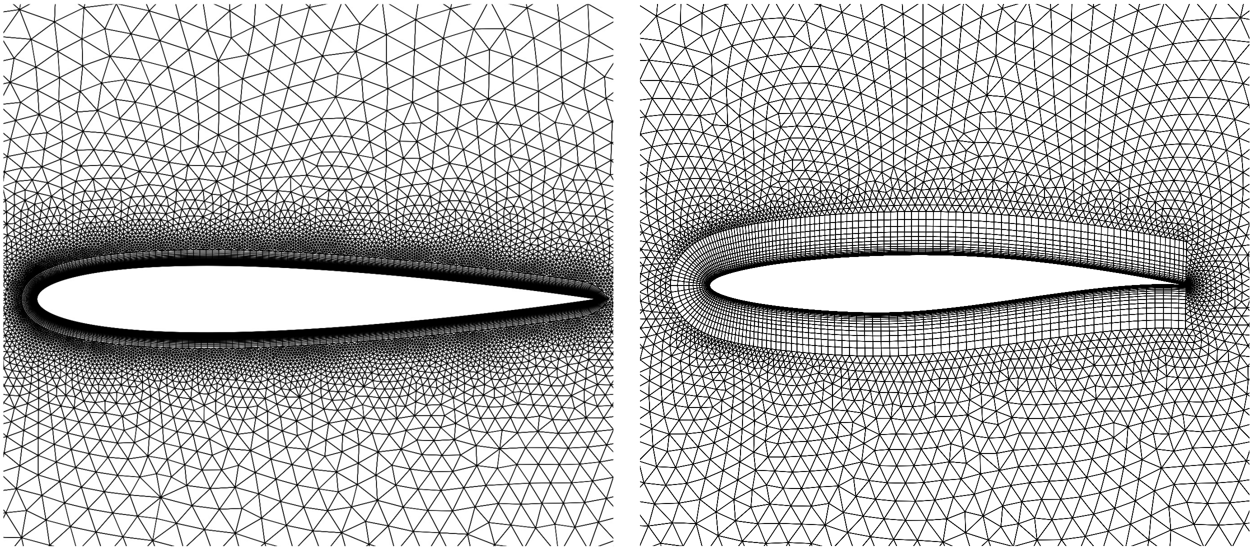


Figure 6. Unstructured two-dimensional grids used for the simulation of laminar subsonic (left) and turbulent transonic flow (right).

IV.A. Verification of Gradients

The first case considered is a transonic viscous RAE2822 single element aerofoil with a single shock on the upper surface. The grid contains about 13400 mesh points. The freestream Mach number is 0.73, Reynolds number is 6.5 million and the angle of attack is 2.31 degree with the Spalart-Allmaras one-equation turbulence model.[33]

With this approach all surface mesh of the aerofoil can be used as design parameters. While the adjoint chain allows for evaluation of gradients with respect to all surface points without difficulty, performing finite differences presents a problem. In order to make it suitable, a flow solution must be computed on a grid with just one single displaced surface mesh point and that sequentially for all surface points. To perform such a comparison more quickly the number of design parameters was reduced. A parameterization was defined consisting of 22 freeform control points and finite differences are applied to these variables, see Figure 7.

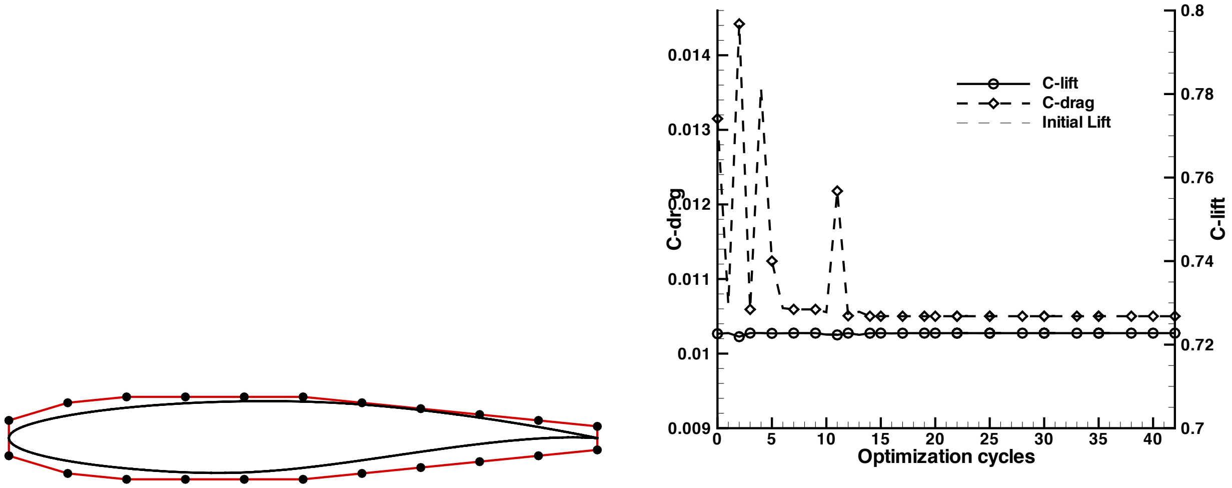


Figure 7. Surface mesh points of the RAE2822 and freeform box around aerofoil and optimization history.

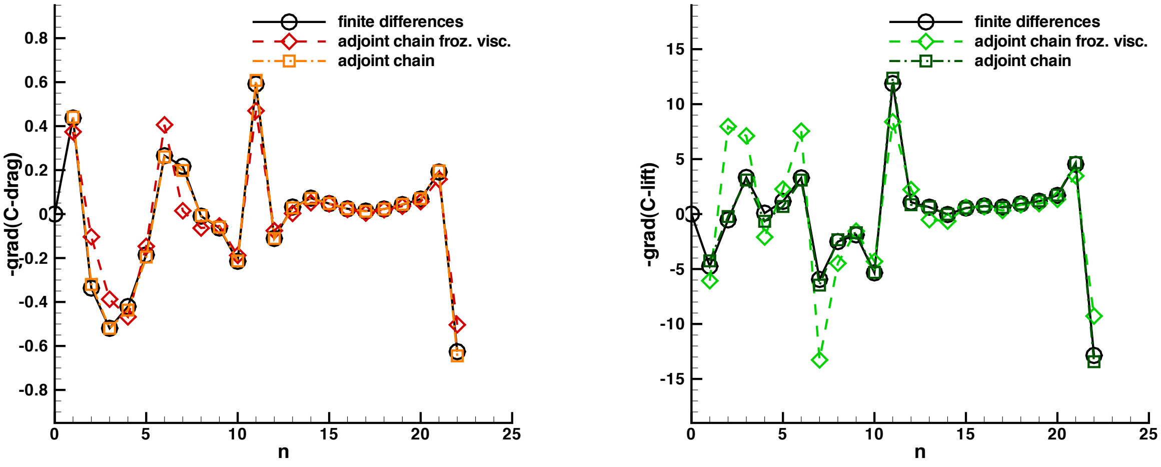


Figure 8. Gradients of lift (left) and drag (right) coefficients with respect to design variables for the RAE2822 aerofoil. Comparison of finite difference gradients and adjoint mesh deformation (adjoint chain) gradients for frozen viscosity and full approximations.

A comparison of the results can be seen in Figure 8 for lift and drag where the design variables traverse

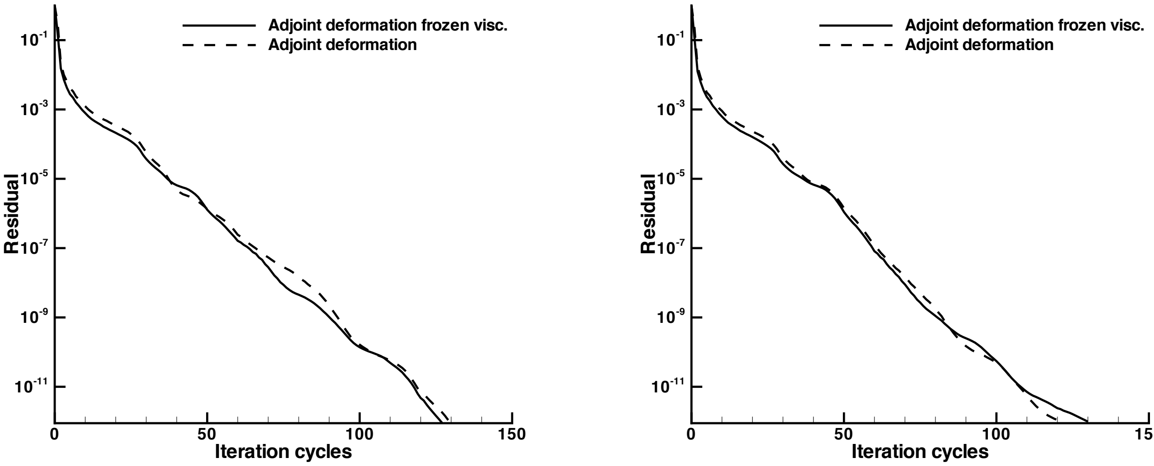


Figure 9. Convergence history of the adjoint mesh deformation computation using the simplification of frozen viscosity and full linearization of the flow adjoint for a RAE2822 aerofoil. Left figure for the drag and right for the lift coefficient.

first the upper surface from nose to trailing edge and then the lower surface from nose to trailing edge. The adjoint chain was applied with the exact derivatives from the viscous and turbulent terms in the flow adjoint. Further, the figures display the difference when evaluating the gradients with the simplification of frozen viscosity and the full linearization. The upper surface shows a discrepancy at the nose (1-3) and at the shock and post shock region (6-8). The lower surface agrees very well. The frozen viscosity approach for the flow adjoint equations is often used for complex three-dimensional cases instead of using the full linearizations. In addition, when solving then the linear adjoint deformation equations the convergence rate was nearly the same for both procedures, seen in Figure 9. Apart from the memory costs, the computation of the right hand side vector **b** is comparably fast in relation to solving the linear equations. The only difficulties observed so far for solving the linear system of equations is the stiff and sparse matrix $\partial T/\partial \tau$. For the cases performed it was not diverging as long it was not spread out in parallel for too many processors and the ratio of points per domain did not get smaller than a certain value.

Finally, the difference between the finite difference gradients and the gradients obtained with the adjoint mesh deformation (adjoint chain) is well within the variation of finite differences. In addition to the linear elasticity as the forward deformation tool we examined the error made using a different, more efficient, deformation based on the advancing front algorithm. Figure 10 shows the gradients and the difference of the absolute error for the drag and lift coefficients which is described as:

$$\text{error} = \nabla \phi_{adjchain} - \nabla \phi_{adjfd}.$$

where the gradient from the adjoint mesh deformation is $\nabla \phi_{adjchain}$ and $\nabla \phi_{adjfd}$ is the gradient evaluated using the flow adjoint with finite differencing for the mesh sensitivities with the advancing front deformation. The error introduced by the alternative deformation tool is at the shock region and at the trailing edge. In fact, if the grid deformations introduced are small, which can be assumed during shape optimizations, the error is small and the advancing front algorithm can be used.

IV.B. Contributions of Gradients

The linearization of the flux jacobian from the flow solver w.r.t. the flow variables, and the mesh jacobian w.r.t. the cartesian coordinates leads us to investigate the influence of partial derivatives. Especially the amount of derivatives from the cost function and residual in the complete chain will be of interest for gaining experience for further complex optimizations. The following figures were generated by using a flow simulation with a central spatial discretization with scalar dissipation, then the flow adjoint was computed exactly. With these pre-requisites the different partial sensitivities were obtained.

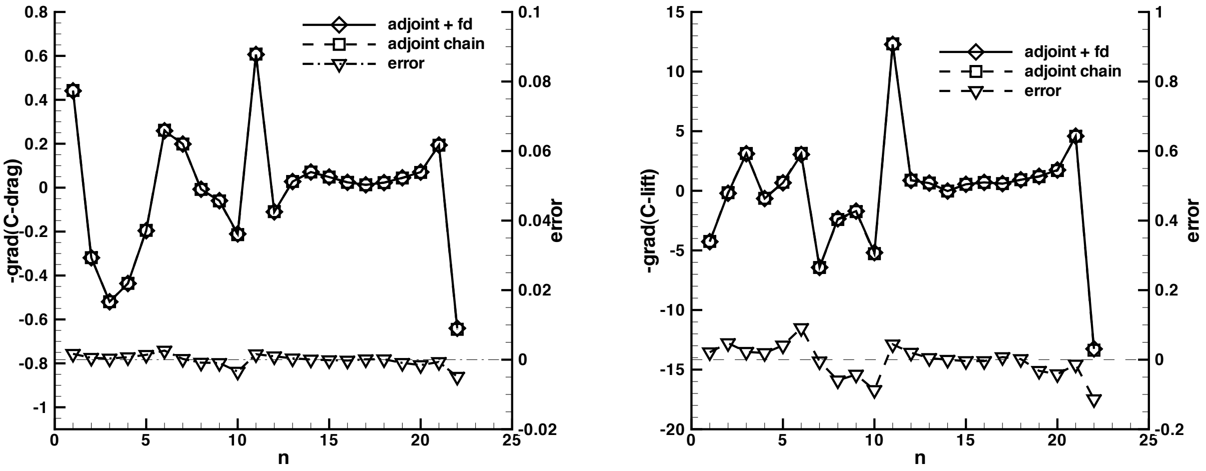
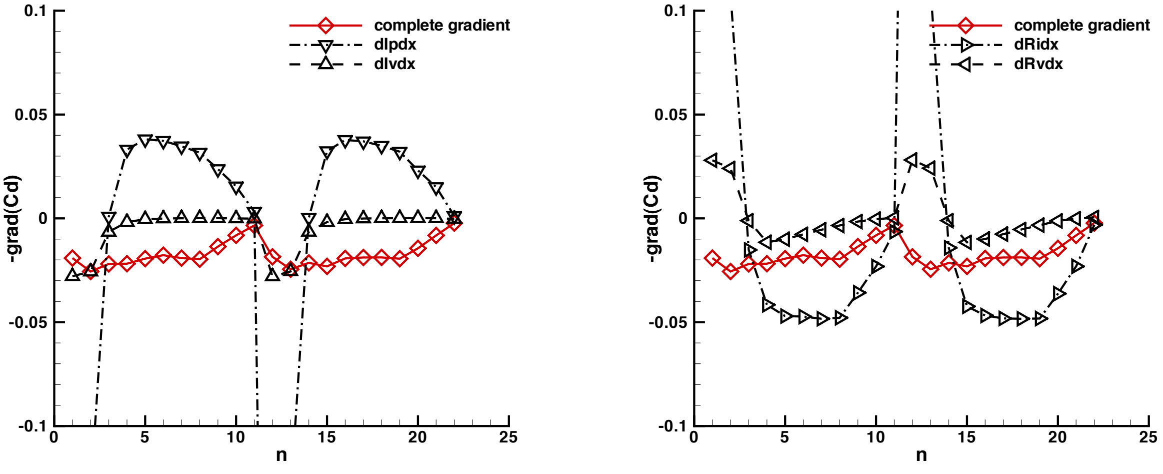


Figure 10. Gradients of lift (left) and drag (right) coefficients with respect to design variables for the RAE2822 aerofoil using the complete adjoint chain with linear elasticity and the TAU deformation for the finite differences.



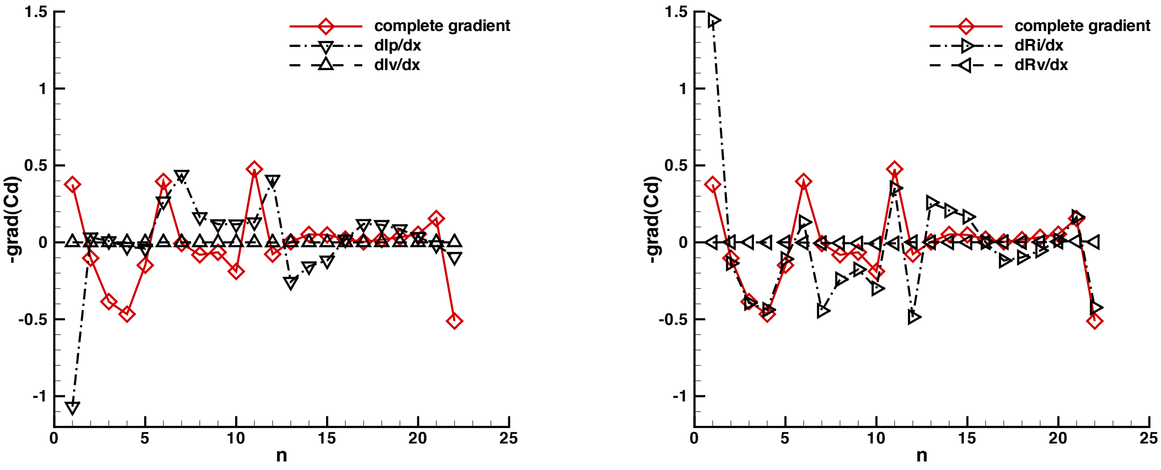
(a) Cost function derivatives.

(b) Residual derivatives

Figure 11. Mesh sensitivities from each contribution for drag as cost function for a laminar flow around NACA0012 aerofoil. Right the derivatives from the cost function in relation to the complete gradient, left the derivatives from residuals.

At first a subsonic laminar NACA0012 aerofoil Figure 6(a) is considered with drag as the cost function. The freestream Mach number is 0.3, Reynolds number is 3000 and the angle of attack is 0 degrees. From the low Reynolds number, the viscous effect on the complete gradient is substantial at the nose region, as can be seen in both Figure 11 either for the cost function I and and the flow residual R. Particularly the derivatives from the residual R_v have a considerable effect on the final gradient.

Because of the symmetric aerofoil and the angle of attack of 0 degrees it was especially suited for validating the derivatives of the viscous adjoint mesh deformation. Now we apply the same procedure for the transonic viscous RAE2822 single element aerofoil and again the cost function will be the drag coefficient. Left of Figure 11 shows the derivatives of the cost function I split into the pressure part $\partial Ip/\partial x$ and viscous part $\partial Iv/\partial x$, see Eq. (10). The right Figure 11 shows the derivatives of the term $\Lambda(\partial R/\partial x)$ again split into an inviscid part and viscous part of the flow residual R, see Eq. (11). In contrast to the laminar case, in both figures the viscous derivatives are nearly negligible.



(a) Cost function derivatives

(b) Residual derivatives

Figure 12. Mesh sensitivities from each contribution for drag as cost function for a turbulent transsonic flow around RAE2822 aerofoil. Right the derivatives from the cost function in relation to the complete gradient, left the derivatives from residuals.

The derivatives of the residual R_i and R_v w.r.t. coordinates, normals and volumes show only a summation of all partial derivates. R_i can be further split into partial derivatives of scalar dissipation, when using a spatial central discretization, and the boundary fluxes. Further on, there are many other contributions to R_v like turbulent diffusion and turbulent source terms but as found out during the implementation and testing stage these pieces had a very small effect on the final gradient. Nevertheless, it was more difficult to be able to validate these derivatives by finite differences due to their small magnitude. Usually for the direct (coordinates) and normal derivatives step sizes of about 10^{-6} were sufficient however for the volume derivatives and turbulent source terms finite differences steps had to be decreased to about 10^{-9} to be able to make a comparison. In that sense, the depicted splitting of the partial derivatives present a negligible influence from the viscous mesh derivatives for the transonic test case with a relatively high Reynolds number. However, for laminar flows the viscous mesh derivatives are not negligible due to their influence in the leading and trailing edge region seen from Figure 11.

IV.C. Shape optimization of a transsonic turbulent RAE2822 profile.

To demonstrate the described approach in section II an optimization for the viscous RAE2822 test case is performed. An objective function for drag has been used while keeping the lift constant. It is necessary to evaluate the gradients for both, drag and lift. The design parameters are the freeform control points depicted in the left Figure 7 (20 design parameter) which are free to move up and down only and neglecting the two control points at the trailing edge. An in-house developed optimization suite based on a conjugate gradient method was used to perform the design process. The flow simulations have been converged to a density residual of 10^{-9} and the same order of residual magnitude for both adjoint simulations, respectivley. The results from the design process are shown in Figure 7(right). The drag was decreased by about 20% while keeping the lift constant. The final shape and pressure distribution C_p can be seen in Figure 13. The strong shock on the upper side of the aerofoil has been entirely removed. The optimization process required 9 gradient evaluations and the total time was 4 hours and 45 minutes on a single processor machine. The savings of using the complete adjoint chain is negligible because of the low number of design parameters and fast finite differences for this two-dimensional case.

IV.D. Optimization of an inviscid ONERA M6 wing

The optimization procedure is now applied to an ONERA M6 wing[27] for an inviscid flow. The grid contains 226,515 points and about 1.4 million tetrahedra. This fine grid was chosen to elaborate the influence of the

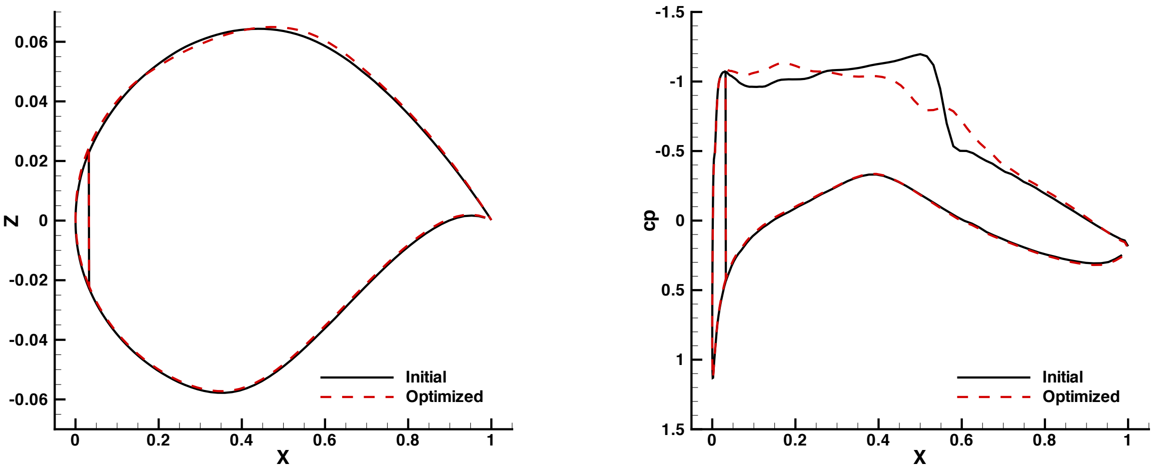


Figure 13. Initial and final aerofoil (left) and pressure distribution C_p (right) of the RAE-2822 aerofoil.

simplification by replacing the deformation matrix from Eq. (8) by an identity matrix I in comparison of using the complete adjoint deformation. The freestream Mach number is 0.84 and the angle of attack is 3.06 degrees. The following computations present optimizations for minimization of the drag while keeping the lift constant. For the case where the deformation operator was set to unity, and the case with the full adjoint chain. The main focus of the following comparisons is on how close the results of the two approaches are, and what runtime savings are obtained.

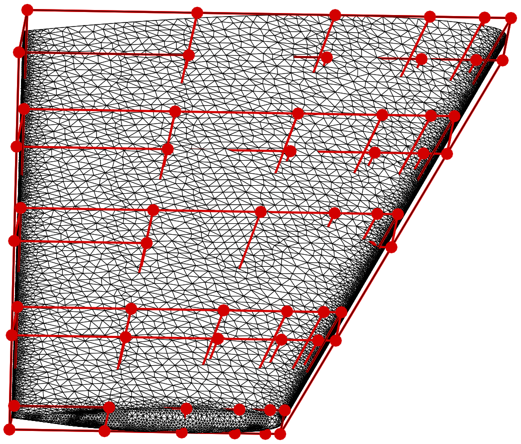


Figure 14. Initial surface grid and freeform box control points

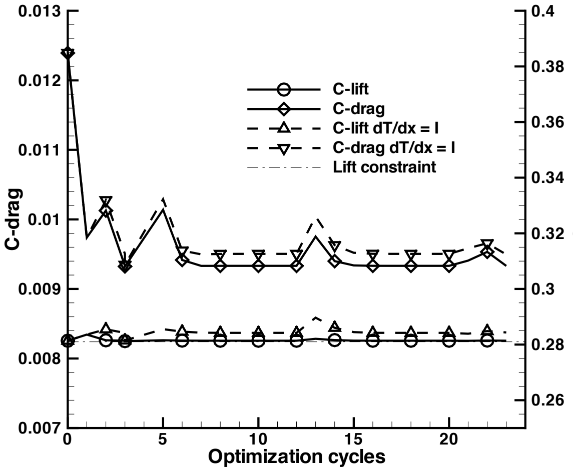


Figure 15. Drag and lift coefficients history during the optimization.

Figure 14 displays the fine Onera M6 mesh with the freeform box with 60 control points unevenly spaced over the wing based on the knowledge of the λ-shock structure and expected changes in the nose region. Again, the control points can move up and down only. The trailing edge control points were fixed because of their large and erratic gradients which would make the optimization mainly move up and down the trailing edge and fail to find an optimum. The optimization history can be seen in Figure 15 for both approaches and two gradient computations were performed to conduct the design process. The simplification $\partial T/\partial x = I$ has an offset in the gained drag reduction and for the lift in comparison to the exact second adjoint computation which was caused by the inaccurate gradients. Furthermore, it was observed during the optimization with $\partial T/\partial x = I$ mainly failed to converge to a preselected minimum density residual. The density residual for both was imposed to be 10^{-9}.

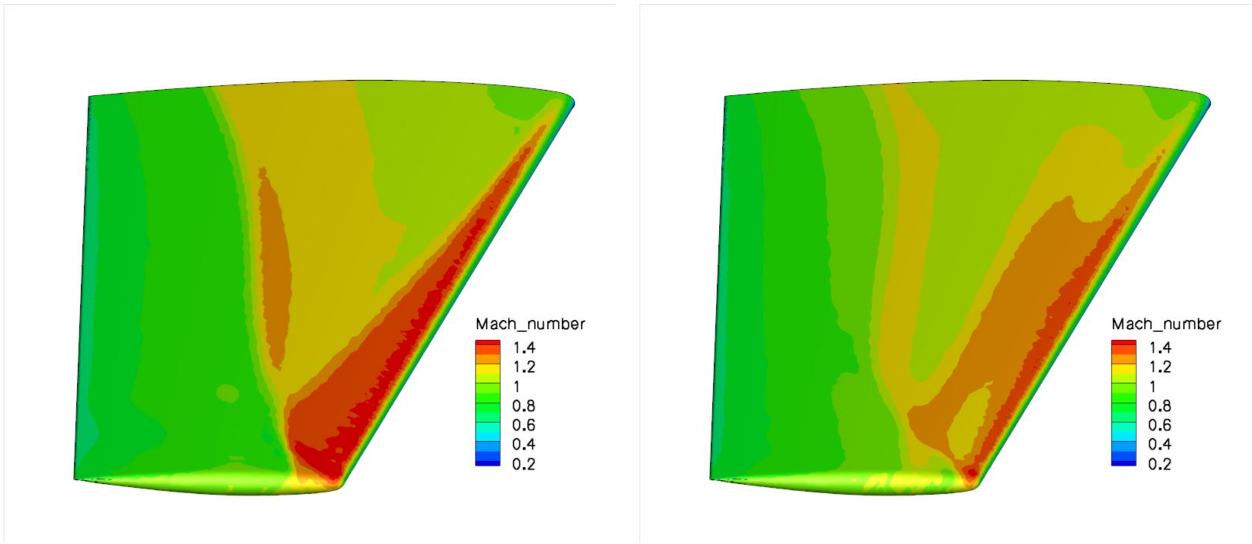


Figure 16. Initial (left) and final (right) Mach number distribution over the upper side of wing.

Finally, looking at the optimization history in Figure 15 the exact approach reached 24.7% of drag reduction and the simple one reached 23.3%. As seen from Table 1 the time savings for the simple approach compared to the exact exact are negligible. The highest time savings are reached with the adjoint mesh deformation computing the sensitivities in relation of using a finite differences procedure which is factor of 16.5. The earlier optimization with an adjoint flow and evaluating the final gradients by finite differences had 3 gradient evaluations instead of two like the following ones. In Figure 16 the initial and optimized wing with the Mach number distribution are shown.

Table 1. Timings of components within the optimization procedure in minutes.

Test Case	Flow Simulation	Adjoint Flow	Adjoint Mesh	Total time
M6 $\partial T/\partial \tau = I$	13	17	0:12	339
M6	13	17	2	355
M6 fd *	13	17	33	452

* Previously performed with adjoint flow and finite differences for mesh sensitivities.

V. Conclusion

The efficient computation of sensitivities with a discrete adjoint approach has been developed and implemented in the DLR TAU-Code in this work. Since one obtains the sensitivity information on all surface points, independently of the type of mesh, this makes it suitable for unstructured grids. Especially when using a large number of design parameters, the cost for deriving the mesh dependent sensitivities for each design parameter by a finite difference approach, including mesh deformations, dual grid processing and residual evaluations, is replaced by a single adjoint deformation. The resulting sensitivities from the discrete adjoint approach have shown good agreement applying the analytic differentiation for the mesh metric terms in comparison to finite differences. However, the implementation was time consuming but mainly straightforward. The optimization performed with the simplification of the adjoint deformation equations has shown no significant difference in the obtained drag reduction compared to the exact approach, but remains applicable for inviscid cases only. On the other hand the iterative solution process is computationally inexpensive and so far will be taken into account in general. The time saved for the inviscid case applying the discrete adjoint formulation shows a rigorous reduction in the computational time by performing an optimization. The adjoint discrete framework will be further applied for complex three-dimensional configurations in turbulent flows.

Acknowledgments

The first author wishes to thank Richard Dwight from the Technical University of Delft for his mathematical based discussions during the preparation of the second chapter. Furthermore I appreciated Jen-Der Lee's help starting up the manuscript and his support on optimizations procedures.

References

[1] Brezillon, J. and Dwight, R. P., "Discrete Adjoint of the Navier-Stokes Equations for Aerodynamic Shape Optimization," *Evolutionary and Deterministic Methods for Design.*, edited by E. 2005, FLM, TU Munich, Germany, 2005.

[2] Amoignon, O. and Berggren, M., "Discrete adjoint-based shape optimization for an edge-based finite-volume solver," *Computational Fluid and Solid Mechanics*, June 2003, pp. 2190–2193.

[3] Carrier, G., "Multi-disciplinary Optimisation of a Supersonic Transport Aircraft Wing Planform," *European Congress on Computational Methods in Applied Sciences*, ECCOMAS, Jyväskylä, Finnland, 24-28 July 2004 2004.

[4] Nielsen, E. J. and Park, M. A., "Using an adjoint approach to eliminate mesh sensitivities." *AIAA-2005-0491*, 2005.

[5] Yang, Z. and Mavriplis, D., "A Mesh Deformation Strategy Optimized by the Adjoint Method on Unstructured meshes." *45th Aerospace Sciences Meeting and Exhibit*, No. AIAA-2007-557, Reno, NV, January 8-11 2007.

[6] Jakobsson, S. and Amolgnon, O., "Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization." *Computers and Fluids*, Vol. 36, No. 6, 2007, pp. 1119–1136.

[7] Hicken, J. E. and Zingg, D. W., "Aerodynamik Optimization Algorithm with Integrated Geometry Parameterization and Mesh Movement," *AIAA Journal*, Vol. 48, No. 2, February 2010, pp. 400–413.

[8] Duvingeau, R., "Adaptive Parameterization using Free-Form Deformation for Aerodynamic Shape Optimization." Tech. Rep. 5949, INRIA, 2006.

[9] Dwight, R., Brezillon, J., and Vollmer, D., "Efficient Algorithms for Solution of the Adjoint Compressible Navier-Stokes Equations with Applications." *Proceedings of the ONERA-DLR Aerospace Symposium (ODAS), Toulouse*, 2006.

[10] Giles, M. B., Duta, M. C., and Müller, J.-D., "Adjoint Code Developments Using the Exact Approach," *15th Computational Fluid Dynamics Conference*, No. AIAA 2001-2596, Anaheim, California, 11 June - 14 June 2001 2001.

[11] Nielsen, E. J. and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Grids." *AIAA Journal*, Vol. 40, 2002, pp. 1155–1163.

[12] Mavriplis, D., "A Discrete Adjoint-Based Approach for Optimization Problems on Three-Dimensional Unstructured Meshes." *AIAA-2006-0050*, 2006.

[13] Nemec, M., Zingg, D. W., and Pulliam, T. H., "Multipoint and Multi-Objective Aerodynamic Shape Optimization." *AIAA Journal*, Vol. 42, 2004, pp. 1057–1065.

[14] Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J., "High-Fidelity Aerostructural Design Optimization of a Supersonic Business Jet," *Journal of Aircraft*, Vol. 41, No. 3, May-June 2004, pp. 523–530.

[15] Gerhold, T., Galle, M., Friedrichs, O., and Evans, J., "Calculation of Complex Three-Dimensional Configurations employing the DLR TAU-Code," *AIAA-97-0167*, 1997.

[16] Schwamborn, D., Gerhold, T., and Heinrich, R., "The DLR TAU-Code: Recent Applications in Research and Industry." *ECCOMAS CFD 2006*, Egmond aan Zee, The Netherlands, September 5-8 2006.

[17] Jameson, A. and Yoon, S., "Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations," *AIAA Journal*, Vol. 25, No. 7, July 1987, pp. 929–935.

[18] "PETSc Web page," 2009.

[19] Balay, S., Buschelman, K., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Smith, B. F., and Zhang, H., "PETSc Users Manual," Tech. Rep. ANL-95/11 - Revision 3.0.0, Argonne National Laboratory, 2009.

[20] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F., "Efficient Management of Parallelism in Object Oriented Numerical Software Libraries," *Modern Software Tools in Scientific Computing*, edited by E. Arge, A. M. Bruaset, and H. P. Langtangen, Birkhauser Press, 1997, pp. 163–202.

[21] Dwight, R. and Brezillon, J., "Effect of Approximations of the Discrete Adjoint on Gradient-Based Optimization." *AIAA Journal*, Vol. 44, No. 12, December 2006, pp. 3022–3071.

[22] Baker, T. J. and Cavallo, P. A., "Dynamic adaption for deforming tetrahedral meshes." *37th AIAA Aerospace Sciences Meeting and Exhibit*, Vol. AIAA-1999-3252, Reno, NV, January 1999.

[23] Yang, Z. and Mavriplis, D., "Unstructured dynamic meshes with higher-order time integration schemes for the unstready Navier-Stokes equations." *43th AIAA Aerospace Sciences Meeting and Exhibit*, No. AIAA-2005-1222, Reno, NV, January 10-13 2005.

[24] Dwight, R., "Robust Mesh Deformation using the Linear Elasticity Equations." *Computational Fluid Dynamics. 2006*, No. 4, Springer Berlin Heidelberg, 2006, pp. 401–406.

[25] Nielsen, E. J., Lu, J., Park, M. A., and Darmofal, D. L., "An implicit, exact dual adjoint solution method for turbulent flows on unstructured grids," *Computers and Fluids*, Vol. 33, 2004, pp. 1131–1155.

[26] Martins, J. R. R. A., Sturdza, P., and Alonso, J. J., "The Complex-Step Derivative Approximation," *ACM Transactions on Mathematical Software*, Vol. 29, No. 3, September 2003, pp. 245–262.

[27] Schmitt, V. and Charpin, F., "Pressure Distributions on the ONERA M6 Wing at Transsonic Mach Numbers," Experimental Database for Computer Program Assessment AR-138, B1-1-B1-44, AGARD, May 1979.

[28] Jameson, A., Siriam, Martinelli, L., and Haimes, B., "Aerodynamic Shape Optimization of Complete Aircraft Configurations using Unstructured Grids." *AIAA-2004-0533*, 2004.

[29] Schmidt, S., Ilic, C., Schulz, V., and Gauger, N., "Shape Gradients and their Smoothness for Practical Aerodynamic Design Optimization," Tech. Rep. SPP1253-10-03, DFG Forschungsbericht, 2008.

[30] Samareh, J. A., "A Survey of Shape Parameterization Techniques," Tech. rep., NASA, 1999.

[31] Ronzheimer, A., "Post-Parametrisation of CAD-Geometries Using Freeform Deformation and Grid generation Techniques." *Notes on Numerical Fluid Mechanics and multidisciplinary Design*, Vol. 87, 2002, pp. 382–389.

[32] Widhalm, M., Ronzheimer, A., and Hepperle, M., "Comparison between Gradient-free and Adjoint based Aerodynamic Optimization of a Flying Wing Transport Aircraft in the Preliminary Design." *25th Applied Aerodynamics Conference*, No. AIAA 2007-4060, Miami, Florida, 25-28 June 2007.

[33] Spalart, P. R. and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows." *AIAA 92-0439*, 1992.