# Comparison of Exact and Approximate Discrete Adjoint for Aerodynamic Shape Optimization

Giampietro Carpentieri[1], Michel J.L van Tooren[2], and Barry Koren[3]

[1] Delft University of Technology `G.Carpentieri@tudelft.nl`
[2] Delft University of Technology `M.J.L.vanTooren@tudelft.nl`
[3] Delft University of Technology `B.Koren@tudelft.nl`

**Summary.** The effect of approximations in the discrete adjoint on constrained shape optimization is investigated. The different approximations are compared with the exact discrete adjoint, which is capable of producing exact gradient/sensitivity information. The purpose of such a comparison is to understand whether or not approximate adjoint codes can be effective for shape optimization in spite of the error in the computed gradient.

## 1 The Discrete Adjoint

The discrete adjoint method is well established in the field of aerodynamic shape optimization [7]. In order to implement the method, the residuals vector $\mathbf{R}$ must be differentiated to obtain the residuals Jacobian, $\partial \mathbf{R}/\partial \mathbf{U}$. The latter must be transposed and used to solve a linear system of equations with the adjoint variables $\mathbf{\Lambda}$ as unknown:

$$\frac{\partial \mathbf{R}}{\partial \mathbf{U}}^T \mathbf{\Lambda} = \frac{\partial J}{\partial \mathbf{U}}^T , \tag{1}$$

where $\mathbf{U}$ is the conservative variables vector and $J$ a functional (e.g. lift, drag or pitching Moment coefficient) that is to be optimized. The vectors $\mathbf{U}$, $\mathbf{R}$ and $\mathbf{\Lambda}$ have dimensions $N \times nvar$, with $N$ the number of nodes in the mesh and $nvar$ the number of variables (for the 2D Euler equations $nvar = 4$). The practical implementation of the method can be challenging due to the differentiation of the residuals vector and the derivation of the assembly for the left–hand side of Eq. (1), which preferably is to be performed in a matrix–free fashion due to the large storage that otherwise would be required.

In the present work, the flow solver is based upon a median–dual discretization and a MUSCL–type reconstruction scheme. On each edge of the mesh the second order numerical fluxes are computed by means of Roe's approximate Riemann solver using reconstructed primitive variables. Reconstruction

is performed using a least–squares gradient and a multi–dimensional type of limiter [2]. For the purpose of deriving the transposed residuals Jacobian, the vector of numerical fluxes, $\mathbf{H}$, and the vector of reconstructed left and right states, $\mathbf{U}_L$ and $\mathbf{U}_R$, can be introduced [3]. The three vectors have dimensions $E \times nvar$, with $E$ the number of edges in the mesh. According to the MUSCL–type scheme the residuals vector has a dependency on the conservative variables vector of the type $\mathbf{R} = \mathbf{R}(\mathbf{H}(\mathbf{U}_L(\mathbf{U}), \mathbf{U}_R(\mathbf{U})))$. Therefore, by means of the chain rule and transposition, the left–hand side of Eq. (1) becomes:

$$\frac{\partial \mathbf{R}}{\partial \mathbf{U}}^T \mathbf{\Lambda} = \left( \frac{\partial \mathbf{U}_L}{\partial \mathbf{U}}^T \frac{\partial \mathbf{H}}{\partial \mathbf{U}_L}^T + \frac{\partial \mathbf{U}_R}{\partial \mathbf{U}}^T \frac{\partial \mathbf{H}}{\partial \mathbf{U}_R}^T \right) \frac{\partial \mathbf{R}}{\partial \mathbf{H}}^T \mathbf{\Lambda}, \tag{2}$$

where $\partial \mathbf{R}/\partial \mathbf{H}$ is a dummy matrix that only contains $-1$ and $1$. $\partial \mathbf{H}/\partial \mathbf{U}_L$ and $\partial \mathbf{H}/\partial \mathbf{U}_R$ are diagonal matrices containing the differentiation of the fluxes with respect to the left and the right states respectively. $\partial \mathbf{U}_L/\partial \mathbf{U}$ and $\partial \mathbf{U}_R/\partial \mathbf{U}$ are sparse rectangular matrices that represent the reconstruction operator, i.e., the differentiation of the reconstructed states with respect to the conservative variables.

In practice, none of the matrices introduced in Eq. (2) is stored but more likely the assembly is performed on the edges similarly to what is done for the residuals vector. Details about the edge–based assembly can be found in [4] where Eq. (2) is derived in detail, rather than only in abstract matrix form. Exactness of the differentiation was demonstrated showing that the residuals Jacobian is capable of yielding quadratic convergence when employed in Newton iterations. Quadratic convergence can only be attained when the Jacobian is exact [2].

Approximations can be introduced in the differentiation in order to simplify the implementation of the discrete adjoint. Three approximations are considered in this work: (*i*) The first approximation is obtained by neglecting the differentiation of the limiter in the reconstruction operator. The simplification is appreciable since the limiter implemented here requires a construction phase which is quite involved compared to that of mono–dimensional limiters [2]. (*ii*) The second approximation is obtained by neglecting the differentiation of the Jacobian matrix in the Roe flux. The approximation saves lot of human work since the differentiation is cumbersome [1]. (*iii*) The third approximation is that obtained by ignoring the complete reconstruction operator, which makes the implementation of the adjoint trivial.

The price to pay for the simplifications is the detrimental effect on the accuracy of the computed gradient. For instance, for the test cases presented below, the first and second approximations show a percentage error of 0.1–2.5% compared to the exact adjoint code. The error increases to 10–30% for the third approximation. However, the error on the gradient can't be used to judge whether or not an approximation is acceptable. It seems more reasonable to consider the effect that the gradient has on the behavior of the optimization process [6, 4].

## 2 Numerical results

Numerical results for constrained shape optimization are presented. The purpose is to minimize the drag coefficient while keeping the lift coefficient of the airfoil constant by means of an equality constraint. Also, lower inequality constraints are enforced on the relative maximum thickness, on the nose radius and on the trailing edge angle. Unconstrained optimization algorithms that include the constraints as penalty terms in the objective could be used. However, their accuracy is known to be poor and moreover they can be ill–conditioned. Therefore, optimization algorithms capable of handling directly equality and inequality constraints are necessary. Two algorithms have been employed: a widely used sequential quadratic programming (SQP) algorithm and a sequential linear programming (SLP) algorithm known as the method of centers [8].
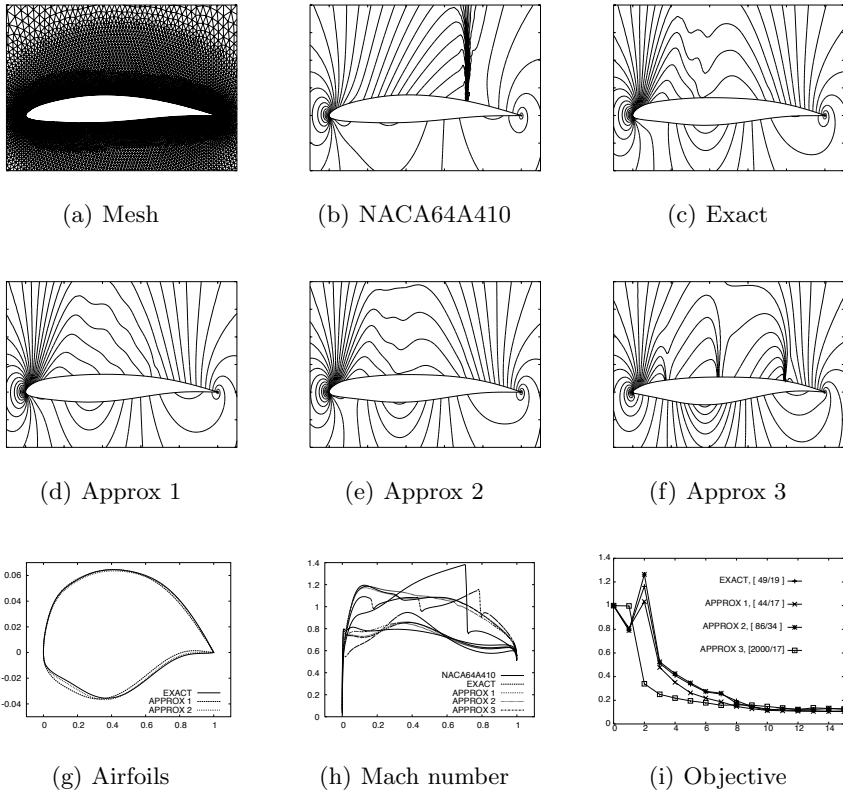


(a) Mesh          (b) NACA64A410          (c) Exact

(d) Approx 1      (e) Approx 2            (f) Approx 3

(g) Airfoils      (h) Mach number         (i) Objective

**Fig. 1.** Optimization of the NACA64A410 airfoil.

(a) RAE2822          (b) Exact          (c) Approx 1



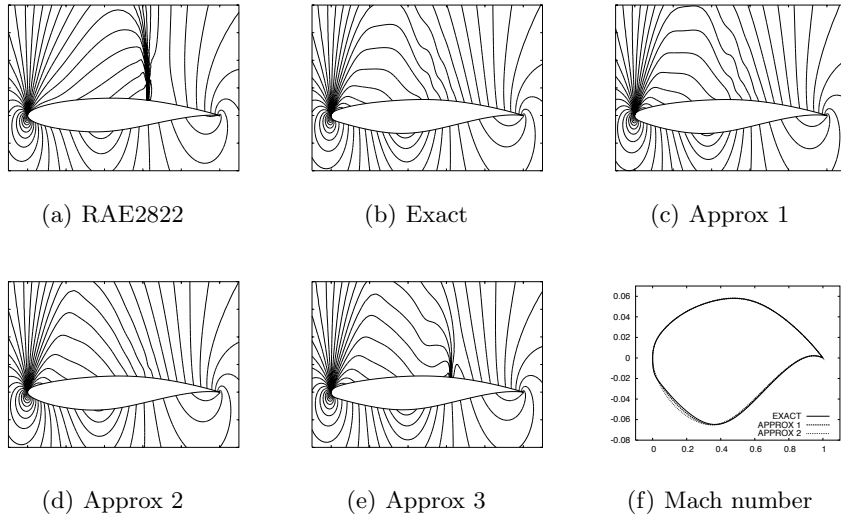(d) Approx 2          (e) Approx 3          (f) Mach number

**Fig. 2.** Optimization of the RAE2822 airfoil.

The flow and the adjoint equations are solved using an implicit pseudo–time stepping scheme. During the optimization, the two solvers are converged until the residual norm is reduced about 6 orders of magnitude. A spring analogy is used to deform the mesh and automatic differentiation in forward mode has been used to derive the geometric sensitivities, i.e., the derivatives with respect to the shape parameters. The shape of the airfoil is parameterized by means of Chebyschev polynomials. Further details can be found in [4, 5].

The computations presented below have been performed on an unstructured mesh of triangles, see Fig. 1.a, with 12161 nodes, 500 of which are on the airfoil. The mesh has been deformed in order to be used for all airfoils. In order to make sure that the mesh was capable of capturing weak shocks, a mesh of 30092 nodes, 700 of which are on the airfoil, has been used to check the optimization results. The shock–free pressure distributions obtained on the first mesh after optimization have been re–obtained on the second mesh. The SQP algorithm from the Matlab Optimization Toolbox[4] has been used for the NACA64A410 and RAE2822 cases whereas the SLP algorithm [5] has been used for the NACA0012 case. For the two algorithms a tolerance of $10^{-5}$ for both the objective function and the constraints has been adopted as criterion to stop the optimization.

*NACA64A410, $M_\infty = 0.75$ and $\alpha = 0\,deg$ angle of attack.* The pressure contours for the original airfoil are shown in Fig. 1.b. The airfoils optimized using the exact and the first two approximations are shock–free, see Figs. 1.c, 1.d

---

[4] Matlab©1984–2004, The Mathworks, Inc.

and 1.e. As can be seen from Fig. 1.g the three airfoils are different. Compared to the airfoil obtained from the exact adjoint, the other two airfoils show a maximum difference in $y$–coordinates of $5.6 \times 10^{-3}$ and $2 \times 10^{-3}$ respectively. However, they all satisfy the design problem accurately. The airfoil obtained using the third approximation shows three weak shocks on the upper side because its optimization has stalled, see Fig. 1.f. In Fig. 1.i the objective function, which is the drag divided by its initial value, is shown at each gradient iteration and, in the legend, the ratio of function/flow evaluations to gradient/adjoint evaluations is shown. The ratio is around 2.6 except for the third approximation. The latter has exceeded the maximum number, 2000, of allowed function calls.

*RAE2822, $M_\infty = 0.75$ and $\alpha = 2\, deg$ angle of attack.* The pressure contours for the original airfoil are shown in Fig. 2.a. As before, the airfoils obtained from the exact and the first two approximate adjoint codes are shock–free, see Figs. 2.b, 2.c and 2.d. The geometry of the three airfoils is different, especially for the first 40% of the chord on the lower side, see Fig. 2.f. Maximum difference in $y$–coordinates are of the order of $10^{-3}$. The optimization stalled when the third approximation was used. The airfoil still exhibits a weak shock, see Fig. 2.e.
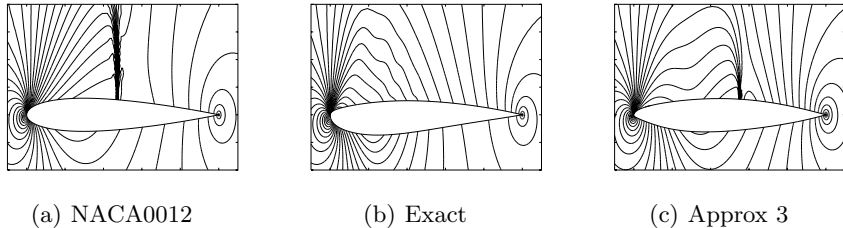


(a) NACA0012          (b) Exact          (c) Approx 3

**Fig. 3.** Optimization of the NACA0012 airfoil.

*NACA0012, $M_\infty = 0.75$ and $\alpha = 2\, deg$ angle of attack.* The pressure contours for the original airfoil are shown in Fig. 3.a. The SLP algorithm seems to be insensitive to the approximations and identical shock–free airfoils are obtained when using the exact and the first two approximations. The airfoils can be said to be identical since the maximum difference between the $y$–coordinates is of the order of $10^{-4}$. Since they are identical, only one of them is shown, see Fig. 3.b. When using the third approximations, also the SLP algorithm stalls and the resulting airfoil exhibits a shock on the upper side, see Fig. 3.c.

## 3 Conclusions

It has been shown that the first two approximate adjoint codes considered in this work can be effective for shape optimization purposes. However, the SQP algorithm has shown to converge to different airfoils, i.e., different local minima, for each approximation. Nevertheless, each of these airfoils satisfied the design problem. The SLP algorithm appeared to be insensitive to the two approximations and converged to a single airfoil.

The third approximation, which neglects the reconstruction operator, has shown to be ineffective. For the third approximation, both the SQP and the SLP algorithms have stalled during the optimization process, failing to converge to a shock–free airfoil.

## References

1. Barth, T.J.:Analysis of Implicit Local Linearization Techniques for Upwind and TVD Algorithms. AIAA Paper 87–0595 (1987).
2. Barth, B.J.: Aspects of Unstructured Grids and Finite–Volume Solvers for the Euler and Navier–Stokes Equations. VKI Lecture Series, 1991-06 (1991).
3. Barth, T.J., Linton S.W.: An Unstructured Mesh Newton Solver for Compressible Fluid Flow and its Parallel Implementation. AIAA Paper 95–0221 (1995).
4. Carpentieri, G., van Tooren, M.J.L., Koren, B.: Improving the Efficiency of Aerodynamic Shape Optimization on Unstructured Meshes. AIAA Paper 2006-298 (2006).
5. Carpentieri, G., van Tooren, M.J.L., Koren, B.: Aerodynamic Shape Optimization by Means of Sequential Linear Programming Techniques. ECCOMAS CFD 2006 Paper, to appear (2006).
6. Dwight, R.P., Brezillon, J.: Effect of Various Approximations of the Discrete Adjoint on Gradient–Based Optimization. AIAA Paper 2006-690 (2006).
7. Giles, M.B., Pierce, N.A.: An Introduction to the Adjoint Approach to Design. Flow, Turbulence and Combustion, **65**, 393–415 (2000)
8. Vanderplaats, G.N.: Numerical Optimization Techniques for Engineering Design. $3^{th}$ Edition, Vanderplaats Reasearch & Development (2001)