AIAA JOURNAL
Vol. 44, No. 12, December 2006

Effect of Approximations of the Discrete Adjoint on Gradient-Based Optimization

Richard P. Dwight* and Joël Brezillon†
DLR, German Aerospace Research Center, D-38108 Braunschweig, Germany

DOI: 10.2514/1.21744

An exact discrete adjoint of an unstructured finite-volume solver for the Reynolds-averaged Navier–Stokes equations has been developed. The adjoint is exact in the sense of being based on the full linearization of all terms in the solver, including all turbulence model contributions. From this starting point various approximations to the adjoint are derived with the intention of simplifying the development and memory requirements of the method; considered are many approximations already seen in the literature. The effect of these approximations on the accuracy of the resulting design gradients, and the convergence and final solution of optimizations is studied, as it applies to a two-dimensional high-lift configuration.

Nomenclature

D = design variables
$|D|$ = number of design variables
E = total energy
e = internal energy
f^c = exact convective flux
f^v = exact viscous flux
$\hat{f}^c$ = numerical flux function
$\mathcal{G}$ = mesh deformation operator
H = total enthalpy
I = cost function
L = pseudo-Laplace operator
$\mathcal{L}$ = Lagrangian
n = face normal vector
p = pressure
R = discrete residual of flow equations
$\mathcal{R}$ = residual of governing equations
U = velocity vector
t = time
W = conservative flow variable vector
X = computational grid
γ = ratio of specific heats
μ_e = effective viscosity
κ_e = effective thermal conductivity
Λ = adjoint variables
Ω = volume of grid cell
ρ = density
τ = viscous shear-stress tensor

I. Introduction

As computational aerodynamics reaches maturity, and computational power advances, design optimization tools using high-fidelity Navier–Stokes simulations are winning an increasingly important place in the aircraft design process. Because of the particular demands of aircraft design, in which a large number of design variables are needed in order to parameterize the shape, gradient-based optimization strategies are favored. Gradient-based optimization using the steepest descent method requires two steps: first, the evaluation of the search direction, the gradient of the cost function with respect to the design variables, which results in the most rapid improvement of the design locally; and second a 1-D search in this direction, consisting of repeated evaluations of the cost function until a minimum is found in this 1-D space. This basic process is repeated until an optimum is found.

The ultimate goal of the present work is gradient-based optimization of three-dimensional high-lift transport aircraft configurations using an unstructured Reynolds-averaged Navier–Stokes (RANS) code. This is an extremely ambitious objective requiring the resolution of a number of very significant problems before it becomes practicable, and is one of the goals of the German project MEGADESIGN [1].

Some of the difficulties involved are the following: the large grids necessary to accurately resolve the wakes of the individual elements of the wing (which control the onset of separation on the upper surfaces) and the associated high computational costs [2], the many design variables needed for parameterization [3], grid deformation which can robustly handle adjacent bodies with large relative motion, calculating gradients of the maximum lift $C_L^{\max}$ [4], and taking unsteady solutions in the design process into account. Further the adjoint method for the gradients, which is absolutely necessary for optimizations involving a large number of design variables, has trouble handling turbulence modeling, in the case of the continuous formulation [5], or is too memory hungry to be applied to large 3-D grids, in the case of the discrete formulation [6].

This report tackles solely the latter problem in the context of 2-D high-lift optimization. There are two principle difficulties with the use of the discrete adjoint. First, formulating the adjoint requires differentiating the corresponding flow solver per hand, including discrete boundary conditions, gradient calculations, turbulence models, etc., which although a straightforward exercise, is laborious in the extreme. Second, depending on the manner of constructing the adjoint residual, storage of the full discrete flux-Jacobian may be required, which is acceptable in 2-D only.

Alternatives to hand differentiation are arising, for example complex variable finite differences [7,8], and algorithmic differentiation [9], but the solution many authors have used is to perform only an approximate differentiation of a flow solver. For example, by treating the coefficient of artificial viscosity in the Jameson–Schmidt–Turkel (JST) scheme [10] as constant it is possible to reduce the construction of the adjoint residual to two sweeps over the faces of the grid [11]. Another example is the practice of assuming that the eddy viscosity is constant, thereby obviating the differentiation of the turbulence model [12].

Presented as Paper 0690 at the 4fourth AIAA Aerospace Sciences Meeting and Exhibit, Reno NV., 9–12 January 2006; received 13 December 2005; revision received 23 June 2006; accepted for publication 3 July 2006. Copyright © 2006 by R.P. Dwight. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code $10.00 in correspondence with the CCC.

*Research Scientist, Institute of Aerodynamics and Flow Technology, Department of Numerical Methods, Lilienthalplatz, 7; richard.dwight@dlr.de.

†Research Scientist, Institute of Aerodynamics and Flow Technology, Department of Numerical Methods, Lilienthalplatz, 7; joel.brezillon@dlr.de.

There have, however, been few studies examining the effect these approximations have on the resulting gradients and optimizations; and this is consequently a matter of pressing interest to the community. It is the purpose of this report to discover which of these simplifications are acceptable in the context of aerodynamic optimization, in the sense of how the convergence of the optimization and its result is affected. In particular, the approximations considered are

1) Adjoint solution based on a first-order accurate discretization,
2) Adjoint solution with thin shear-layer (TSL) viscous fluxes,
3) Assumption of constant coefficients in the JST scheme,
4) Assumption of constant eddy viscosity,
5) Adjoint solution with alternative turbulence model.

Each of these approximations is either based on the assumption that the derivatives of the particular terms are negligible, or that they may be replaced with the derivatives of similar related terms. By examining a high-lift configuration—in which a wide variety of flow phenomena are represented, and for which both compressible and viscous effects, as well as the choice of turbulence model are critical—as well as a single-element airfoil, where the design problem essentially consists of the removal of the shock, it is anticipated that the results obtained will be valid for general 2-D aerodynamic optimizations.

With this goal the exact discrete adjoint to the unstructured finite-volume RANS solver, the DLR *Tau* code [13], has been evaluated by hand for a wide variety of the spatial discretizations available in the code. The gradients obtained from the exact adjoint are verified against those obtained using finite-differences on the original nonlinear routines. Where the approximate adjoint formulations are the result of using a related discretization, as for the adjoint based on first-order fluxes, finite differences have again been used to verify the implementation. Gradient evaluations and thereafter full optimizations are then performed with the various adjoint approximations, and variations in gradients, convergence rates and solutions attained are compared.

The solution of the linear adjoint problem is performed using a Krylov subspace method with incomplete lower–upper factorization (ILU) preconditioning, allowing the solution of the adjoint problem in a CPU time equivalent to about 5% of the time required for the main problem. In addition the adjoint fields for multiple cost-functions may be computed simultaneously, further reducing the CPU-time cost of the gradient evaluation.

II. Statement of the Problem

The optimization problem of interest is stated, as are the equations governing the fluid flow, and a summary of their finite-volume discretization.

A. Aerodynamic Design Problem

The optimization problem may be stated as follows: minimize $I(W, X, D)$ a cost function with respect to some set of design variables D, subject to the constraints $R(W, X, D) = 0$ and $\mathcal{G}(X, D) = 0$; whereby W and X are functions of D, and R and $\mathcal{G}$ are general nonlinear operators.

Here I is typically an aerodynamic force integrated over the geometry such as lift, drag, or pitching moment, W represents the flow variables, X the computational mesh, R the residual resulting from the discretization of the fluid flow equations, and G a mesh deformation operator. In particular R is here the finite-volume discretization on an unstructured mesh of the Navier–Stokes equations. Finally, D is some parameterization of the geometry and onflow conditions.

For problems with a large number of design variables, the most efficient algorithms are gradient -based, and require the evaluation of $\mathrm{d}I/\mathrm{d}D$ for each design variable. These gradients may in turn be efficiently evaluated using the adjoint approach. Additional constraints on the problem are also common, such as the specification of constant lift and constant pitching moment.

B. Finite-Volume Discretization

The 2-D Favre averaged Navier–Stokes equations in conservation form may be written

$$\frac{\partial W}{\partial t} + \frac{\partial}{\partial x_i} f_i^c(W) - \frac{\partial}{\partial x_i} f_i^v(W) = \frac{\partial W}{\partial t} + \mathcal{R}(W) = 0 \tag{1}$$

where summation convention is applied and the conservative state vector is defined by $W = (\rho, \rho u, \rho v, \rho E)^T$. The vectors of convective fluxes are given by

$$f_x^c = \begin{bmatrix} \rho u \\ \rho u u + p \\ \rho u v \\ \rho H u \end{bmatrix}, \qquad f_y^c = \begin{bmatrix} \rho v \\ \rho v u \\ \rho v v + p \\ \rho H v \end{bmatrix} \tag{2}$$

and the vector of viscous fluxes by

$$f_x^v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ U_i \tau_{xi} - \kappa_e (\nabla T)_x \end{bmatrix}, \qquad f_y^v = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ U_i \tau_{yi} - \kappa_e (\nabla T)_y \end{bmatrix} \tag{3}$$

where $U = (u, v)^T$ is the velocity vector, $H = E + p/\rho$ is the total enthalpy, and $\tau(W, \nabla W)$ is the viscous shear-stress tensor

$$\tau = \mu_e \{\nabla U + \nabla U^T - \tfrac{2}{3} \nabla \cdot U I\} \tag{4}$$

where μ_e and κ_e represent the effective viscosity and thermal conductivity, respectively, the turbulent quantities being obtained using an eddy-viscosity model. The equation of state for a calorically perfect gas, $p = (\gamma - 1)\rho e$, where e is the internal energy, closes the system.

The equations are solved using an unstructured cell-vertex finite-volume method, applying a central numerical flux with mixed second- and fourth-order dissipation operators after the scheme of Jameson, Schmidt, and Turkel [10],

$$\begin{aligned} \hat{f}^c(W_i, W_j; n_{ij}) = {} & \tfrac{1}{2}[f^c(W_i) + f^c(W_j)] \cdot n_{ij} - \tfrac{1}{2}|A| \{\epsilon_2 \theta_2 [W_j - W_i] \\ & - \epsilon_4 \theta_4 [L_j(W) - L_i(W)]\} \end{aligned} \tag{5}$$

where f^c is the exact inviscid flux, $|A|$ is either a scalar or matrix measure of the absolute convective speeds, the ϵ control the absolute levels of the two dissipation operators, and together the θ act as a shock switch. The pseudo-Laplacian L is given by

$$L_i(W) = \sum_{j \in \mathcal{N}(i)} (W_j - W_i) \tag{6}$$

$\mathcal{N}(i)$ being the set of immediate neighbors of node i. In the following $|A|$ is taken to be the maximum eigenvalue of the convective flux-Jacobian, that is, scalar dissipation.

The spatial gradients needed in the expression for the viscous fluxes and in the turbulence model are normally obtained using

$$\nabla \phi_i \approx \frac{1}{2\Omega_i} \sum_{j \in \mathcal{N}(i)} (W_i + W_j) n_{ij} \tag{7}$$

where Ω_i is the volume of the control-volume about node i. The values of the gradients from two neighboring cells are averaged onto their common face To model the viscous flux there.

Throughout Spalart–Allmaras–Edwards [14] is the turbulence model used. Strong boundary conditions are used on the viscous wall, and characteristic boundary conditions on the far field. In the following the discretization of $\mathcal{R}$ described here will be denoted R.

III. Gradients via Discrete Adjoint

The adjoint approach allows the rapid evaluation of $\mathrm{d}I/\mathrm{d}D$ for a large number of design variables $|D|$. It can be readily understood by contrasting it with the direct or primal approach.

A. Primal Approach

The most direct approach to evaluation of the gradient is to apply the chain rule to $\mathrm{d}I/\mathrm{d}D$, to give

$$\frac{\mathrm{d}I}{\mathrm{d}D}=\frac{\partial I}{\partial W}\frac{\mathrm{d}W}{\mathrm{d}D}+\frac{\partial I}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}D}+\frac{\partial I}{\partial D}$$

which is an expression for $\mathrm{d}I/\mathrm{d}D$ in terms of $\mathrm{d}W/\mathrm{d}D$ and $\mathrm{d}X/\mathrm{d}D$ (the remaining quantities being readily calculable). By noting that $\mathrm{d}R/\mathrm{d}D=0$, as the condition $R=0$ holds for all D, we have

$$\frac{\mathrm{d}R}{\mathrm{d}D}=\frac{\partial R}{\partial W}\frac{\mathrm{d}W}{\mathrm{d}D}+\frac{\partial R}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}D}+\frac{\partial R}{\partial D}=0 \tag{8}$$

a linear system for $\mathrm{d}W/\mathrm{d}D$, based on the linearization of the discretized flow equations. Hence to find the sensitivity of I to $|D|$ design variables it is necessary to solve (8) $|D|$ times, and in practice this effort dominates the total cost of the calculation.

B. Adjoint Approach

Instead of applying the chain rule to I, apply it to the *Lagrangian*:

$$\mathcal{L}(W,X,D,\Lambda)=I(W,X,D)+\Lambda^{T}R(W,X,D)$$

where Λ are known as the *adjoint variables*. Because $R=0$ for all D, $\mathcal{L}=I$ for all Λ and all D. Hence

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}D}=\frac{\mathrm{d}I}{\mathrm{d}D},\quad \forall\ \Lambda,D$$

and so, applying the chain rule to $\mathcal{L}$

$$\begin{aligned}\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}D}&=\left\{\frac{\partial I}{\partial W}\frac{\mathrm{d}W}{\mathrm{d}D}+\frac{\partial I}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}D}+\frac{\partial I}{\partial D}\right\}\\&\quad+\Lambda^{T}\left\{\frac{\partial R}{\partial W}\frac{\mathrm{d}W}{\mathrm{d}D}+\frac{\partial R}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}D}+\frac{\partial R}{\partial D}\right\}\\&=\left\{\frac{\partial I}{\partial W}+\Lambda^{T}\frac{\partial R}{\partial W}\right\}\frac{\mathrm{d}W}{\mathrm{d}D}+\left\{\frac{\partial I}{\partial X}+\Lambda^{T}\frac{\partial R}{\partial X}\right\}\frac{\mathrm{d}X}{\mathrm{d}D}\\&\quad+\left\{\frac{\partial I}{\partial D}+\Lambda^{T}\frac{\partial R}{\partial D}\right\}\end{aligned} \tag{9}$$

after rearranging. The unknown quantity $\mathrm{d}W/\mathrm{d}D$ may then be eliminated by choosing Λ such that

$$\left(\frac{\partial R}{\partial W}\right)^{T}\Lambda=-\left(\frac{\partial I}{\partial W}\right)^{T} \tag{10}$$

whereby the first bracketed term of (9) is zero. This is the *adjoint equation*, and must be solved only once to evaluate the gradient of a single I with respect to any number of design variables. Given Λ the gradient is

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}D}=\left\{\frac{\partial I}{\partial X}+\Lambda^{T}\frac{\partial R}{\partial X}\right\}\frac{\mathrm{d}X}{\mathrm{d}D}+\left\{\frac{\partial I}{\partial D}+\Lambda^{T}\frac{\partial R}{\partial D}\right\}$$

whereby $\partial I/\partial D$ and $\partial R/\partial D$ are zero for shape-based design variables, and the remaining unknown term $\mathrm{d}X/\mathrm{d}D$ may be reliably evaluated by finite differences,

$$\begin{aligned}\frac{\partial I}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}D}\Delta D_i&\approx\frac{I[W,X(D+\epsilon\Delta D_i),D]-I(W,X,D)}{\epsilon},\\\frac{\partial R}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}D}\Delta D_i&\approx\frac{R[W,X(D+\epsilon\Delta D_i),D]-R(W,X,D)}{\epsilon}\end{aligned}$$

as the result is relatively insensitive to choice of ϵ, in contrast to direct approximation of $\mathrm{d}I/\mathrm{d}D$ by finite-differences. It is also possible to use an adjoint approach for $\mathrm{d}X/\mathrm{d}D$, thereby removing the necessity of deforming the grid in response to each shape-modifying design variable, and eliminating finite differences from the algorithm completely. This can be particularly straightforwardly realized if the grid deformation is a linear operator [15].

C. Implementation of the Method

Implementation of the above procedure requires the ability to evaluate the quantities $(\partial R/\partial W)^{T}\cdot\Lambda$—the *adjoint residual*—and $\partial I/\partial W$.

The Jacobian is evaluated by hand, which is a straightforward exercise as R may be written explicitly in terms of W, while being time consuming as R is often extremely complex. As R is a sum of convective fluxes, viscous fluxes, boundary conditions, etc., each of these may be handled independently, and by application of the chain rule may be further subdivided into manageable chunks. The derivatives are further simplified by choosing primitive variables as working variables. Because the equations remain in conservative form this choice has no effect on the final solution.

Using finite differences to evaluate some of the terms—especially the turbulence models—seems attractive, but in fact it is difficult to evaluate the dot product of the transpose of the Jacobian with a vector in such a manner, though possible [8]. The proper choice of step size poses a second problem, and the technique is not applied here.

Strong boundary conditions such as specification of zero velocity on viscous walls are handled by rewriting the flow equations $R(W)=0$ as

$$(I-B)\cdot R(W)=0,\qquad B\cdot W=0$$

where B is a projection matrix that extracts the velocity at the boundary, and then differentiating [16]. The accuracy of the derivatives of the individual fluxes is verified by applying finite differences to the original flux routines to approximate their derivatives, and comparing with the hand-calculated Jacobian for a variety of inputs. As each flux derivative is calculated, a contribution is made to the explicitly stored Jacobian.

Storing the matrix explicitly has the disadvantage of requiring approximately 6 times the memory of the standard code, see Table 1, reducing the capacity of a node with 1 GB of memory from 2×10^{6} points to 300,000 points, however, because the size of grids used in optimization loops lags behind those used for single flow solutions (due to the otherwise too great computational effort) this limitation is not regarded as fatal. Whereby it is important to emphasize that this result is only valid in 2-D, the situation in 3-D being worse due to the greatly increased numbers of next-neighbors of a point.

Given an explicit Jacobian, which must be constructed only once per gradient evaluation, even for multiple cost-functions, assessment of the adjoint residual 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 nonlinear equations [17,18], and again this preconditioner must only be constructed once.

As the eigenspectrum of the Jacobian and transpose Jacobian are identical, the convergence rates achieved with Krylov subspace methods for the two resulting linear systems are guaranteed to be identical, and so the experience gained applying these methods in implicit schemes may be directly carried over [19]. The result is that the calculation of the adjoint solution requires only approximately 5% of the time required for a nonlinear 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) [20], become more attractive. The ILU preconditioner and generalized minimal residual (GMRES) method have an associated memory cost that further reduces the size

Table 1 The memory requirements of the adjoint code with exact Jacobian storage compared with the standard *Tau*-code—measured for a 2-D unstructured grid with 50 × 10^3 points

	Std. *Tau*	+ Jac. storage	+ Linear sol. storage
Memory (bytes)	25M	165M	290M
Factor increase	×1.0	×6.6	×11.6
Points in 1 GB	2×10^{6}	300×10^{3}	170×10^{3}

of grid that may be calculated in a given memory, in this case the requirements for ILU with four levels of fill-in and GMRES(30) are given in Table 1.

IV. Approximations of the Discrete Adjoint

In an attempt to reduce the memory requirements of the scheme, the manual effort required to adjoint new spatial discretizations, and the computational effort required to solve the resulting system, approximations to the adjoint equations are made. The cost is inaccuracies in the resulting gradients. Each approximation is here described and its advantages noted, while its accuracy will be numerically assessed in Sec. VI. Throughout, no approximation of the boundary conditions is undertaken, as all are numerically treated with a stencil of a single (boundary) point, and therefore appear only on the diagonal of the Jacobian.

A. First-Order Approximation

The aim of the first-order approximation (FOA) is to reduce the stencil of the discretization to immediate neighbors of a node only, leading to a corresponding reduction in the sparse fill-in of the Jacobian, and the stiffness of the linear system. Such a simplification results in a reduction of the memory requirements of the adjoint solver to less than twice those of the nonlinear solver, and the system can be solved in a time equivalent to 1% of that of the nonlinear system. Further, the immediate neighbor fill-in allows the storage of the off-diagonal entries of the Jacobian on the edges of the unstructured grid, making a matrix–vector multiplication a familiar loop over edges.

This stencil reduction may be achieved by constructing the Jacobian for the adjoint method from derivatives of first-order convective fluxes and TSL viscous and turbulence diffusion fluxes (see Sec. IV.B). The stencil of the turbulence discretization consists only of immediate neighbors anyway, and therefore is treated without approximation.

The most sensitive issue is the simplification of the convective fluxes. For an upwind discretization the first-order generalization is natural; the use of constant face-reconstruction rather than affine reconstruction reduces the stencil in the required manner. For the central scheme used here the situation is not as clear. There are two principal possibilities: a) use purely second-order dissipation, neglecting the term involving ϵ_4 in (5), so that

$$\hat{f}^c(W_i, W_j; n_{ij}) = \tfrac{1}{2}[f^c(W_i) + f^c(W_j)] \cdot n_{ij} - \tfrac{1}{2}|A|\{\epsilon_2(W_j - W_i)\} \tag{11}$$

is differentiated, or b) explicitly neglect derivatives of the dissipation with respect to next-neighbors, whereby derivatives with respect to immediate neighbors include fourth-order dissipation terms.

Given that the solution is smooth almost everywhere, the true residual will contain a fourth difference operator almost everywhere, and neither of these simplifications is satisfactory. Consider Table 2, which gives the weights of second differences, fourth differences, and the "chopped" fourth differences in 1-D. The operator of option a) is completely at variance with the fourth difference operator, with even the signs of the corresponding weights differing. On the other hand choosing option b) (the last row of Table 2) results in a system that is considerably more difficult to solve than the full unapproximated system. This could be understood to be a result of fact that the chopped fourth-order dissipation operator contains a large second-order antidissipation component.

Table 2 Second and fourth difference operators in one dimension. Also a fourth difference that has been chopped to reduce its stencil

	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}
Second difference		$+1$	-2	$+1$	
Fourth difference	$+1$	-4	$+6$	-4	$+1$
Fourth chopped		-4	$+6$	-4	

In the absence of a really satisfactory approximation, the pure second-order dissipation operator is used in the following.

B. Thin Shear-Layer (TSL) Assumption

The formulation of the viscous fluxes given in Sec. II.B has a stencil including next neighbors as a result of the use of gradients based on (7). This stencil may be reduced if an alternative discretization is used, inspired by thin shear-layer approximations in structured methods, whereby only the component of the gradient normal to a face is used to construct the viscous flux on that face, that is,

$$\nabla\phi \approx \frac{\partial\phi}{\partial\eta} n$$

where η is the direction normal to the face. A finite-difference approximation is then trivial:

$$(\nabla\phi)_{\text{face}} \approx \frac{1}{|\Delta x_{ij}|}(\phi_j - \phi_i)n \tag{12}$$

and the corresponding Jacobian will have immediate neighbor fill-in only. Diffusion occurring in turbulence model transport equations is treated identically.

Note that this is not the same as the method commonly called TSL used in structured methods, in which only viscous fluxes normal to viscous walls are considered, and which is therefore inappropriate for modeling general viscous phenomena. Here viscous fluxes in all directions are considered and in practice results obtained with formulations (7) and (12) are almost identical, even for multielement aerofoils where boundary-layer separation and interaction of shear layers are crucial flow elements. Of course this does not necessarily imply that their gradients will be identical.

C. Constant JST Coefficients Approximation (CCA)

By making the assumption that the ϵ, θ, and $|A|$ in (5) are constant, the following simplification of the adjoint residual is possible [11]. Let $L(W)$ be as defined in (6), and treated in the expression for the convective fluxes as an independent variable, so that the derivative and adjoint of (5) may be written, respectively

$$\frac{\mathrm{d}\hat{f}^c}{\mathrm{d}W} = \frac{\partial\hat{f}^c}{\partial W} + \frac{\partial\hat{f}^c}{\partial L} \cdot \frac{\partial L}{\partial W} \tag{13}$$

$$\left(\frac{\mathrm{d}\hat{f}^c}{\mathrm{d}W}\right)^T = \left(\frac{\partial\hat{f}^c}{\partial W}\right)^T + \left(\frac{\partial L}{\partial W}\right)^T \cdot \left(\frac{\partial\hat{f}^c}{\partial L}\right)^T \tag{14}$$

whereby all matrices on the right-hand sides above have immediate neighbor fill-in only, $\partial\hat{f}^c/\partial L$ is symmetric and $\partial L/\partial W$ is trivial.

Memory requirements are then approximately 1.6 times those of the FOA scheme if advantage is taken of the symmetry of $\partial\hat{f}^c/\partial L$, and $\partial L/\partial W$ is calculated on-the-fly. Also, as for the FOA scheme, the matrices may be stored on the edges and nodes of the grid, given which the adjoint residual may be evaluated in two loops over all edges by introducing an intermediate variable Λ^* as follows:

$$\Lambda^* = \left(\frac{\partial\hat{f}^c}{\partial L}\right)^T \cdot \Lambda,$$

$$\left(\frac{\mathrm{d}\hat{f}^c}{\mathrm{d}W}\right)^T \cdot \Lambda = \left(\frac{\partial\hat{f}^c}{\partial W}\right)^T \cdot \Lambda + \left(\frac{\partial L}{\partial W}\right)^T \cdot \Lambda^*$$

TSL viscous and turbulence diffusion fluxes are used to similarly reduce the memory requirements of the viscous flux Jacobian, while the discrete turbulence Jacobians are formed exactly.

The CCA approximation may be justified by noting that the terms in the derivative that are neglected due to the approximation are of higher order in the grid spacing than the remaining terms. Therefore in the limit of zero cell size the influence of the approximation tends to zero.

Compared with the FOA approximation this approach seems very favorable; for only sightly more memory the convective fluxes are sensibly approximated. However the resulting system is almost as poorly conditioned as the exact system and therefore similarly powerful linear solvers are required.

D. Constant Eddy-Viscosity (CEV) Assumption

One of the most demanding parts of the spatial discretization to differentiate by hand is the turbulence model, partly due to the wide variety of blending functions, limiters, vortex corrections, etc., but mainly because of the many coupling points to the mean-flow equations, and the enormous selection of models available. More seriously it is very difficult to treat turbulence models in a continuous adjoint framework without resorting to continuous-discrete hybrids [5], hence some simplifying assumption must be made.

By assuming that derivatives of all turbulence quantities with respect to all flow variables are negligible all turbulence terms are eliminated. For Spalart–Allmaras the derivatives of the eddy viscosity are taken to be zero, for $k - \omega$ and $k - \epsilon$ models derivatives of k and eddy viscosity are taken to be zero.

One place where this assumption might be invalid is in the adverse pressure gradient region following the shock on a transonic airfoil, where the large eddy viscosity increases the surface shear-stress, directly affecting the aerodynamic forces.

E. Use of an Alternative Turbulence Model (ATM)

The only significant benefit that the CEV assumption confers is the avoidance of hand differentiating the turbulence model. The similarity in the formulation and results of, for instance, the original Spalart–Allmaras model (SA) [21], and the modified Spalart–Allmaras–Edwards (SAE) [14], suggest that it may be reasonable to use the Jacobian of one as an approximation of the Jacobian of the other, and that this may be a better approximation than ignoring turbulence derivatives altogether. If so only one or two models of each type must be differentiated in a large code.

SA and SAE differ only in the formulation of the turbulence production term: for SA it is based on a measure of the flow vorticity, SAE uses a measure of the shear stress. Within the boundary layer the source terms are dominated in both models by terms involving the wall-distance, hence the models are expected to differ most significantly in detached shear-layers and vortices.

V. Results and Discussion

We wish to determine which, if any, of the previously described approximate Jacobians results in an adjoint gradient evaluation method that is accurate enough for use in aerodynamic optimization. This is a question that is difficult to answer theoretically due to the complexity of the complete optimization process, and the somewhat arbitrary nature the Jacobian approximations. Instead we consider concrete optimization problems numerically and examine how each approximate adjoint solver performs in each case. In an attempt to obtain results that have a relatively general validity, two significantly different 2-D optimization problems are considered representing a variety of industrially relevant flow physics, and, just as importantly, two different optimization algorithms are applied.

The first case is drag reduction of a transonic RAE 2822 single-element aerofoil at a Reynolds number of 6.5×10^6, and a Mach number of 0.730, whereby the lift must be held constant at a lift coefficient of 0.8. The computational grid is shown in Fig. 1. The geometry is parameterized using 20 design variables which modify the camberline of the aerofoil with Hicks–Henne bump functions [22]. The thickness of the aerofoil is not permitted to change, and as a result no additional geometrical constraints are necessary. The baseline geometry has a strong shock on the upper surface which is the main source of pressure drag; the optimization problem therefore substantially consists of the removal of the shock. Two gradient-based algorithms, the conjugate gradient (CG) method and the QNTR method are applied to the problem.

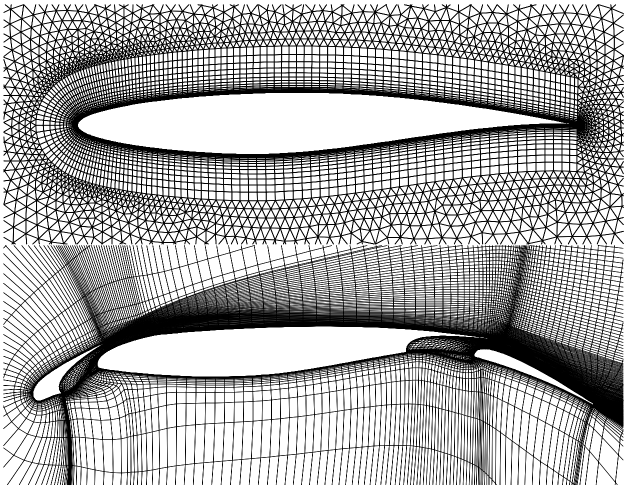

Fig. 1 The two baseline grids under consideration, a RAE2822 and the EUROLIFT high-lift configuration.

The lift constraint is enforced explicitly by varying the angle of attack during the evaluation of the drag in the nonlinear RANS solver, the so-called *target-lift* mode. Because we wish to minimize the drag at the target-lift C_L^*, rather than at the preexisting lift C_L, the objective function must be modified to consider the lift constraint consistently [23]

$$I = C_D - \frac{(\partial C_D/\partial \alpha)}{(\partial C_L/\partial \alpha)}(C_L - C_L^*) \tag{15}$$

a consequence of which is that the accuracy of the gradients of lift are also important for the optimization.

The second optimization test case is also drag reduction at constant lift, but of the three-element high-lift geometry in takeoff configuration. The baseline geometry and flow conditions are derived from a test case defined in the European project EUROLIFT II [24], and operate at a Reynolds number of 14.70×10^6 and a Mach number of 0.17146. Constant lift is ensured in the same manner as for the RAE case. The computational grid is shown in Fig. 1 and is structured, whereby the structured topology is not used by the solvers, all algorithms being implemented for general unstructured grids. Only the flap of the configuration is parameterized, and in such a manner that no modification to the composite clean wing is possible [25]. The position and angle of attack of the flap relative to the main element may be changed (the so-called *setting* parameters), as may the sharpness of the flap's nose and the shape of the portion of the flap hidden by the main element (*shape* parameters). Only the conjugate gradient algorithm is examined for this case.

A. Transonic RAE2822 Aerofoil

First exact adjoint gradients are compared with direct finite-difference approximations to $\mathrm{d}I/\mathrm{d}D$; the gradients of total and viscous drag are plotted in Fig. 2, whereby the 20 design variables parameterize the aerofoil camberline from the leading edge to the trailing edge. The trailing edge itself remains fixed. Both finite differences and the discrete adjoint approximate the gradient of the *discretized* cost function, and should therefore agree with each other exactly on any given grid. It may be seen that the agreement is very good, the discrepancies apparent in the viscous drag gradients may be attributed to the rounding error in finite differences as a result of the very small absolute values of the gradients. This discrepancy as a percentage of the total drag gradient is less than 0.5%. The discrete adjoint using the exact Jacobian is thereby taken to be verified for this case.

Next the gradients using the adjoint method with various Jacobian approximations are compared with those of the exact adjoint in

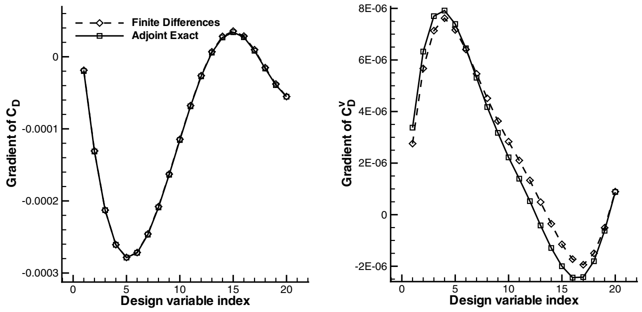

Fig. 2 Gradients of force components obtained with finite differences and exact adjoint for the RAE2822.

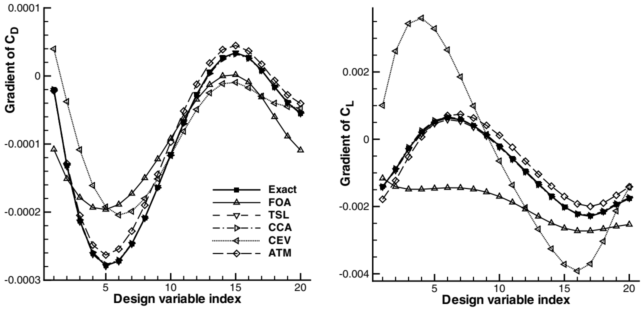

Fig. 3 Gradients of force components obtained with various adjoint approximations for the RAE2822.

Fig. 3. TSL and CCA gradients agree with the exact gradients to within a relative error of 1%, whereby TSL is slightly more accurate than CCA, an expected effect as TSL is a subset of the CCA approximation. The considerable inaccuracy of the FOA gradients might be attributed to the presence of the shock, which would be heavily smeared if first-order fluxes were used in the nonlinear calculation. It might be possible to improve FOA by modulating and reducing its coefficient of dissipation in an attempt to reproduce the level of dissipation in the exact problem, but this has the side effect of increasing the stiffness of the problem, and a major advantage of FOA—that the system is easy to solve—would be lost. The error in the CEV gradients might be attributed to the importance of turbulence, an effect which is apparently well captured by the ATM model. More detailed explanations might be constructed by comparing the adjoint fields for each approximation, but physically interpreting the adjoint field is a delicate and difficult matter, and rather we move directly to numerical investigations.

1. Conjugate Gradient Optimization

The CG algorithm, an improvement on the method of steepest descent, uses conjugate gradients rather than the local gradient to determine a direction in which to search for the minimum. Given this search direction, a line search using repeated evaluations of the cost function is performed on the resulting one-dimensional subspace. Once the minimum on the line is found, a new search direction is calculated.

If no cost-function improvement is obtained in the CG direction the algorithm is restarted, and the search continues in the gradient direction. This is done to prevent stalling of the algorithm in the case that the cost function is not a pure quadratic form, but has the effect of also preventing the accumulation of errors from inaccurate gradients on each iteration. As a result the algorithm is particularly robust, and will fail only if no reduction in the cost function can be found in either the CG direction or the gradient direction, which could only be the result of a very poor gradient, or a design point close to a local minimum.

This robustness can be seen in the convergence of the RAE case, Figs. 4 and 5, whereby the force coefficients are plotted against the number of evaluations of the cost function (i.e., nonlinear RANS computations) performed, and therefore approximately represent the cost of the calculation, given that gradient evaluations are relatively cheap and seldom. Gradient evaluations are denoted on the convergence curves by symbols. In convergence plots of α and C_D^v in Fig. 5 it is clear that none of the CG methods are fully converged, however, after this point the change in the cost function is very small —less than one-tenth of a drag count. As a result full convergence is rarely considered necessary in practical applications, and the partial convergence used here better reflects engineering practice.

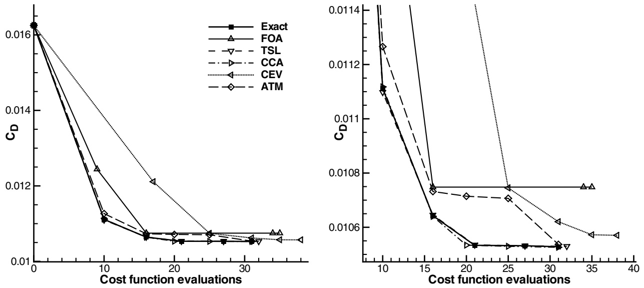

Fig. 4 Convergence of conjugate gradients for drag minimization at constant lift of the RAE2922.

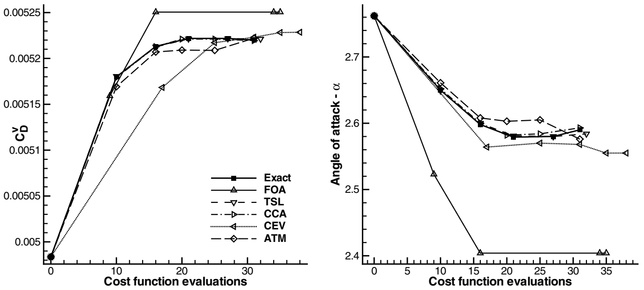

Fig. 5 Convergence of viscous drag and angle-of-attack for the optimization of Fig. 4.

The most striking feature of the convergence, seen in the left-hand graph of Fig. 4, is that all approximate gradient optimizations converge to approximately the same value of C_D with approximately the same effort. All optimal solutions are within 2.5 drag counts of each other—if FOA is disregarded, 0.5 drag counts—and obtained within 30–40 cost-function evaluations. Apparently large deviations in gradients have little effect on CG for this relatively simple case, although it is the case that the approximations that showed more accurate gradients have better results, TSL and CCA in particular being indistinguishable from the exact gradient optimization.

FOA was the worst performer, at iteration 34 it was unable to find a better solution in the CG direction, and at iteration 35 also in the gradient direction and therefore stopped, at a point far from the optimum found by the other methods, as seen in Fig. 5. To establish whether this design point was an alternative local minimum, or stalling due to an inaccurate gradient, an optimization was restarted using exact gradients from the iteration at which FOA stalled, the convergence is shown in Fig. 6 plotted against number of gradient evaluations. The restarted optimization rapidly finds a better solution, indicating that poor FOA gradients were responsible for the 2.5 drag count deficit.

For completeness the force coefficients of the best design found by each approximate optimization process are given in Table 3. Some optimal geometries and pressure distributions are shown in Fig. 7, whereby the results of TSL and CCA are indistinguishable from the exact optimization. All approximations removed the shock, as might have been deduced from the drag convergence behavior, but the resulting pressure distribution for the exact gradient optimization is significantly smoother than that of the poorer approximations.

As described above the constant lift coefficient was enforced explicitly; an alternative method that is not considered here involves restricting the search direction to be in the hyperplane normal to the

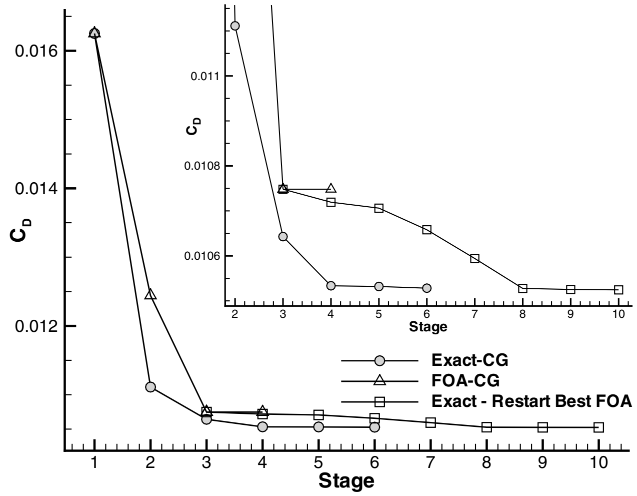

Fig. 6 Convergence of the optimization of Fig. 4 with exact and FOA adjoint gradients.

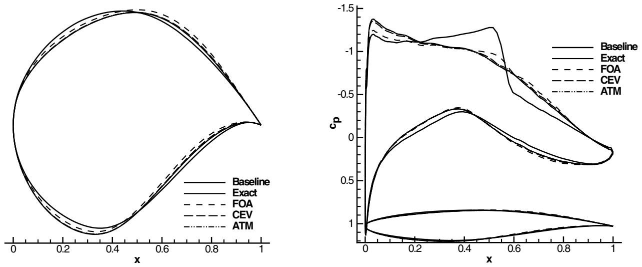

Fig. 7 Baseline and optimized geometries and pressure distributions for several approximate adjoint gradients.

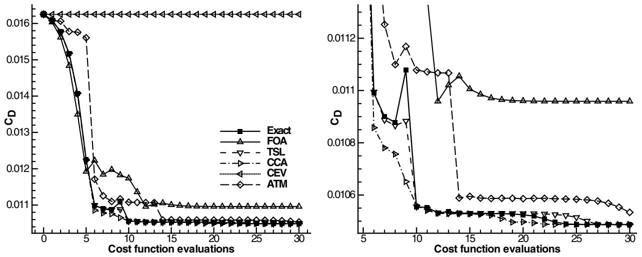

Fig. 8 Convergence of QNTR for drag minimization of the RAE2822 at constant lift.

gradient of the lift. For sufficiently small updates, the lift then varies in proportion to the square of the magnitude of the update step (if the lift has "wandered too far" after several iterations then an optimization iteration purely for the correction of the lift is performed). This process is of course very sensitive to the accuracy of the gradient, poor gradients implying the need for more correction steps. However, the results *directly* reflect the disparities in the gradients themselves, and the performance of the scheme may be roughly judged by considering the error in the gradient.

2. *Quasi-Newton Trust-Region Optimization*

The quasi-Newton trust-region method attempts to improve on the convergence of the CG method by approximating the cost function on the design space by a quadratic form [20]. The gradient of the cost function is computed at every iteration, and based on a Broyden–Fletcher–Goldfarb–Shanno (BFGS) update, an approximation to the Hessian of the cost-function is built. The next design point is chosen as the minimum of this approximation whereby the minimum must lie within the *trust region*, effectively a limit on the size of the design step. The size of the trust region is increased or decreased based upon the accuracy with which the approximation matches the real function, judged using the discrepancy between the previous cost-function evaluation and its corresponding estimate.

In practice this method is less robust than CG, and strongly dependant on the accuracy of the Hessian, which in turn depends upon the gradients at all previous steps. One poor gradient could damage the Hessian approximation, and hence the convergence of the method, significantly. QNTR represents therefore a more demanding test of gradient accuracy than CG.

This method is applied to the same RAE drag reduction case as before and similar optima are achieved, the convergence is shown in Fig. 8. With this algorithm the convergence of the exact, TSL and CCA approximations are no longer identical, testifying to the increased sensitivity of the method to the gradient, while being similar enough to have confidence in the use of these gradients in QNTR. The fact that CCA converges slightly faster than the exact method is not significant, but rather noise as a result of the complexity of the system.

The complete lack of convergence of CEV in this case cannot be attributed to the poor initial drag gradient, which is not substantially worse than that of FOA. But upon examining the gradient of the corrected cost function of (15) it is seen that while the shape of the gradient is correct, the scaling is completely wrong. This is of no consequence for CG, which uses only directional information from the gradient, but is fatal for QNTR, whose cost-function approximation becomes increasingly inaccurate, leading to a reduction in the size of the trust region on every iteration. Examining the gradient of CEV at each step of the CG method of the previous section, it is apparent that the gradient improves substantially after the first iteration, after which the strength of the shock has been reduced considerably.

The results for the optimal designs using the QNTR method are given in Table 4. The best drag coefficients for TSL and CCA show a slight improvement over those of the CG optimization due to the more highly converged state of the system.

3. *Comparison of Conjugate Gradient and Quasi-Newton Trust Region*

It is interesting to compare the relative performance of CG and QNTR for this adjoint code, which, due to the extremely rapid adjoint solution, makes gradient evaluation considerably cheaper than cost-function evaluation. The CPU time required for full evaluation of the gradient, including (exact) adjoint calculations for C_D and C_L, and deformation of the mesh for each design variable, is approximately 17.5% of the time required for a single nonlinear flow solution. Hence the QNTR overhead of a gradient evaluation at every iteration is minimal.

Table 3 Results of optimizations of the RAE2822 aerofoil using the conjugate gradient method. Given are iterations counts and flow coefficients for the best geometry obtained using each gradient approximation

	Cost-fn. evals.	Gradient evals.	C_D (counts)	C_D^v (counts)	C_L (counts)	C_M	α
Baseline	——	——	162.51	49.83	79.99	−0.29327	2.7620
Exact	31	5	105.28	52.20	79.99	−0.29921	2.5899
FOA	35	4	107.48	52.50	79.99	−0.31165	2.4040
TSL	32	5	105.30	52.21	79.99	−0.29956	2.5840
CCA	31	5	105.26	52.19	79.99	−0.29903	2.5929
CEV	38	5	105.70	52.28	80.00	−0.30269	2.5550
ATM	31	5	105.36	52.22	79.99	−0.29995	2.5759

Table 4 Results of optimizations of the RAE2822 aerofoil using the Quasi-Newton trust-region method. Values given are as for Table 3

	Cost-fn. evals.	Gradient evals.	C_D (counts)	C_D^v (counts)	C_L (counts)	C_M	α
Baseline	——	——	162.51	49.83	79.99	−0.29327	2.7620
Exact	32	32	104.86	51.92	79.99	−0.29142	2.7070
FOA	12	12	109.58	52.51	80.00	−0.31582	2.3499
TSL	32	32	104.84	51.93	79.99	−0.29145	2.7109
CCA	37	37	104.87	51.88	79.99	−0.29038	2.7349
CEV	6	6	162.51	49.83	79.99	−0.29327	2.7620
ATM	36	36	105.28	52.21	79.99	−0.29945	2.5780

The convergence of CG and QNTR using exact adjoint gradients are shown in Fig. 9. Convergence of the drag to within a tenth of a drag count of the optimum occurs for CG and QNTR within 50 and 25 cost-function evaluations, respectively. However, it is apparent from the development of the pitching moment C_M, on the right-hand side of Fig. 9, that while the QNTR solution is completely stationary after 30 iterations, CG is still modifying the geometry after 60 iterations. This property of QNTR, while very attractive, is not of immediate practical relevance, as the minimization of drag is the single objective of the optimization. Hence we can say that for this case QNTR delivers the optimal solution in a CPU-time equivalent to about 50% of that of CG, while being less robust to poor gradients.

B. Three-Element High-Lift Configuration

Again gradients obtained by the adjoint method are compared with finite-difference gradients in Fig. 10, whereby the latter are difficult to determine accurately in this case, and convergence of the difference could not be achieved for all design variables. Instead the finite-difference gradients are plotted for three distinct step sizes. The agreement with the adjoint is very good, even for C_D^v, and the adjoint is thereby taken to be verified for this case.

As before the adjoint approximations are compared in Fig. 11. TSL and CCA are again almost indistinguishable from the exact gradients, CEV agrees well, corroborating the evidence that for subsonic cases its accuracy is good, while FOA and in particular ATM make extremely large systematic errors. The poor performance of ATM here is surprising given its good performance for the RAE aerofoil, but might be explained by the strong influence of detached shear-layers in this case, in which regions the SA and SAE models are most likely to differ, as previously discussed. It seems that neither CEV nor ATM can be considered really reliable adjoint approximations as regards the gradient.

Figure 12 shows a typical nonlinear solver convergence plot for this case. The time-stepping algorithm, a lower upper-symmetric Gauss–Seidel scheme (LU-SGS) smoothed multigrid iteration [26], requires about 5000 cycles to achieve a reduction in the residual of the discretization of 6 orders of magnitude, at which point the drag is converged to an accuracy well under a drag count, although changes

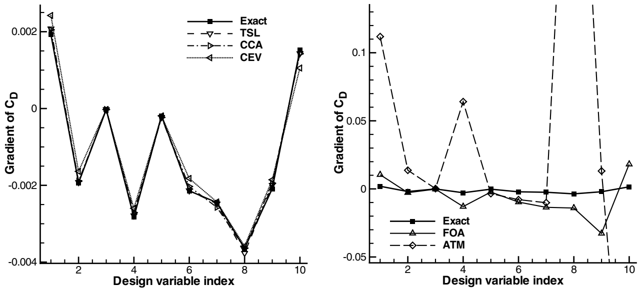

Fig. 11 Gradients of total drag for the high-lift case with various adjoint approximations.

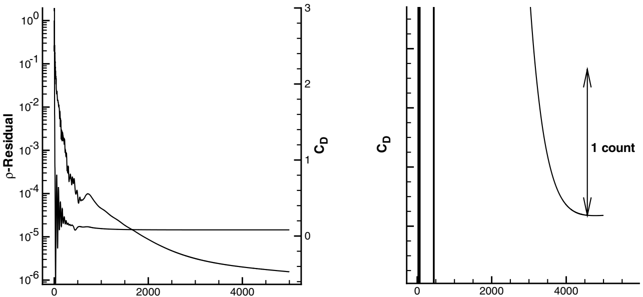

Fig. 12 Convergence of residual and drag for the nonlinear RANS high-lift configuration.

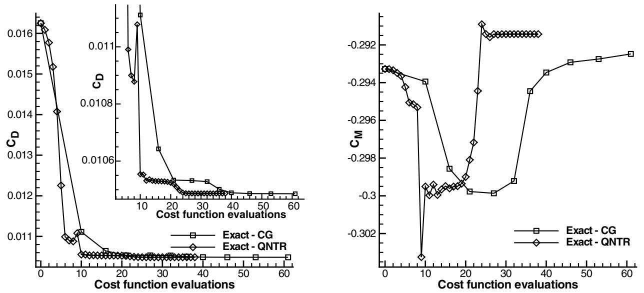

Fig. 9 Drag and pitching moment convergence for CG and QNTR with the exact adjoint.

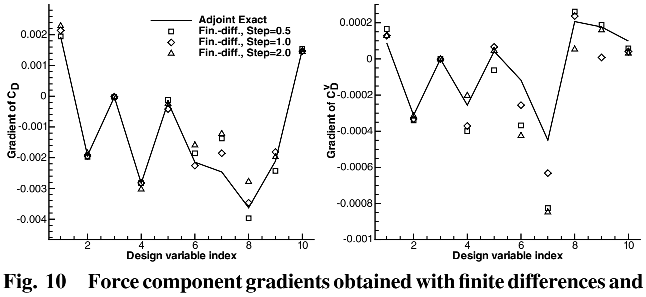

Fig. 10 Force component gradients obtained with finite differences and exact adjoint for the high-lift case.

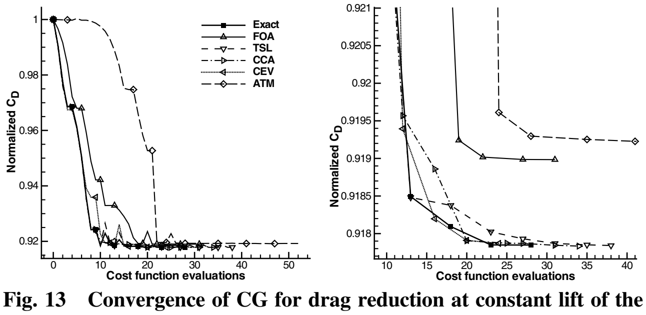

Fig. 13 Convergence of CG for drag reduction at constant lift of the high-lift case.

Table 5 Results of optimizations of the high-lift configuration using the conjugate gradient method and a variety of adjoint gradient approximations. Values given are changes in the coefficients, the coefficients themselves are as in Table 3

	Cost-fn. evals	Gradient evals	ΔC_D (counts)	ΔC_D^v (counts)	ΔC_M	$\Delta\alpha$
CCA	34	8	−66.69	−5.80	−0.0432	−0.587
Baseline	——	——	——	——	——	——
Exact	28	6	−66.68	−5.80	−0.0432	−0.587
FOA	31	7	−65.76	−6.16	−0.0445	−0.518
TSL	38	9	−66.68	−5.85	−0.0433	−0.582
CEV	23	6	−66.66	−5.80	−0.0434	−0.584
ATM	44	9	−65.59	−5.94	−0.0452	−0.509

in the drag are still visible in the right-hand plot of Fig. 12. Further convergence of the nonlinear solution is desirable, as partially converged solutions are a major source of noise in the optimization process, however CPU-time constraints do not permit this.

An optimization using CG is performed for each adjoint gradient approximation, and the convergence histories are plotted in Fig. 13, whereby the drag has been normalized using the baseline drag. Again convergence is obtained in all cases, testifying to the robustness of CG. Convergence and optimal solution are very similar for the three approximations that showed accurate gradients: TSL, CCA, and CEV. On the other hand FOA and even ATM achieve optima within one drag count of the best solution found; the total drag reduction over the baseline geometry amounts to 66 counts. Whether this discrepancy is significant depends upon the significance of one drag count to the engineering problem, bearing in mind also the limited accuracy of the CFD model.

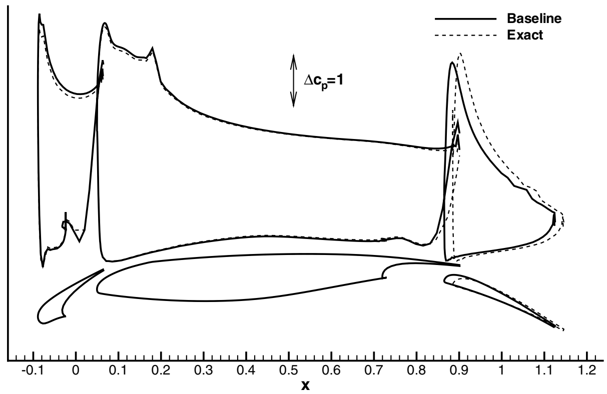

Fig. 14 Baseline and optimized pressure distributions for the high-lift case.

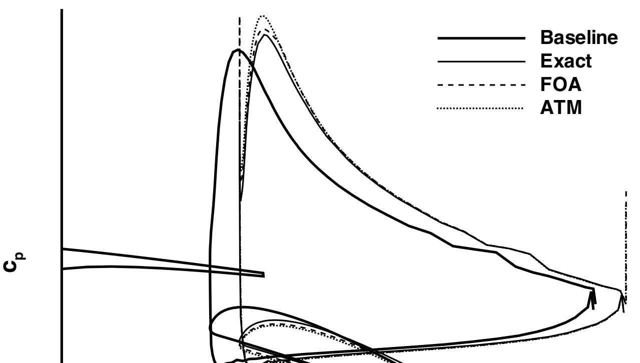

Fig. 15 Baseline and optimized pressure distributions for the high-lift flap for several optimizations.

The details of each of the optimal solutions are given in Table 5. The optimal geometry and pressure distribution achieved with the exact gradients are shown in Fig. 14, where it should be noted that the baseline angle-of-attack was 9.61 deg, and the optimized value 9.02 deg. The means by which the drag may be reduced is not as clear as for the RAE case, however, an important aspect is certainly the prevention of separation and maintenance of high speed flow over the upper side of the main element near the trailing edge. This may be achieved by, for example, moving the flap downstream until the suction peak at the flap stagnation point is at, or downstream of, the main element trailing edge. In Fig. 14 it can be seen that exactly this effect has been obtained.

The optimal geometries and pressure distributions obtained with exact gradients, FOA and ATM are shown in Fig. 15, the remaining approximate results being identical to the exact gradient result. The CG method was ultimately able to perform effective optimizations with all the gradient approximations tested here.

VI. Conclusions

A method for construction and solution of the exact adjoint problem has been described which permits convergence within 5% of the CPU time for the nonlinear problem, and therefore typical full gradient calculations (including two adjoint solutions) within 20% of the time for a single nonlinear problem solution. Memory requirements are, however, 6–7 times higher than the nonlinear solver due to the storage of the Jacobian, effectively limiting the method to 2-D cases.

In an effort to reduce the memory requirements while maintaining the efficiency of the method several approximations to the discrete adjoint have been proposed and studied with respect to optimizations on two 2-D cases. Two of the approximations studied, TSL and CCA, showed consistently minimal variance from the exact gradients, and as such may be used with confidence as substitutes for the exact adjoint method. On the other hand it has been seen that approximating the adjoint of the turbulence model, either with constant eddy viscosity (CEV) or the adjoint of a very similar model (ATM), leads to gradients that are good in some cases, but exceptionally poor in other cases. In particular CEV produced poor gradients for the RAE baseline case with a strong shock but excellent gradients in the high-lift case; for ATM the situation was reversed. These results suggest that construction of reliable adjoint method requires exact consideration of the particular turbulence model used.

A remaining question of importance is whether the CCA approximation performs similarly well in the case of matrix dissipation. Given the similarity of the two methods the authors strongly expect this to be the case, but numerical experiments that could confirm this have not been performed as yet.

Optimizations were performed with the conjugate gradient (CG) algorithm, which converged reasonably well for all cases and with all gradients, no matter how poor. Optimal results obtained with variously inaccurate gradients resulted in optimal solutions differing by no more than 2.5 drag counts for the RAE case and 1 drag count for the high-lift case. A consequence is that an extremely poor (and correspondingly cheap) adjoint approximation, such as FOA, could be useful in situations where high accuracy is not required.

It has also been shown that the possibility for rapid gradient evaluation allows efficient use of the Quasi-Newton trust-region method, which can reduce the overall optimization time by 50% in comparison to CG, whereby the method is much more sensitive to poor gradients.

Based on the results of this study an adjoint solver using the CCA approximation will be further developed, and a reduction in memory requirements to about twice those of the nonlinear solver are expected.

Acknowledgments

This work is partially supported by funds from the 3rd German Aerospace Research Program in the context of the MEGADESIGN project. The authors would also like to thank Jochen Wild for his assistance with respect to the high-lift configuration.

References

[1] Kroll, N., Gauger, N., Brezillon, J., Becker, K., and Schulz, V., "Ongoing Activities in Shape Optimization within the German Project MEGADESIGN," *Proceedings of ECCOMAS*, ECCOMAS, Jyväskylä, 2004.

[2] Rudnik, R., Heinrich, R., Eisfeld, B., and Schwarz, T., "DLR Contributions to Code Validation Activities Within the European High Lift Project EUROLIFT," *DGLR, New Results in Numerical and Experimental Fluid Dynamics IV*, Springer–Verlag, Berlin, 2004, pp. 42–49.

[3] Wild, J., "On the Potential of Numerical Optimization of High-Lift Multi-Element Airfoils based on the Solution of the Navier–Stokes Equations," *Proceedings of the ICCFD II Conference*, Springer–Verlag, Sydney, 2003, pp. 267–273.

[4] Kim, S., Alonso, J., and Jameson, A., "Design Optimization of High-Lift Configurations Using a Viscous Continuous Adjoint Method," AIAA Paper 2002-0844, 2002.

[5] Nadarajah, S., and Jameson, A., "A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization," AIAA Paper 2000-0667, 2000.

[6] Brezillon, J., and Dwight, R., "Discrete Adjoint of the Navier–Stokes Equations for Aerodynamic Shape Optimization," *Proceedings of Evolutionary and Deterministic Methods for Design, EUROGEN*, EUROGEN Committee, Munich, 2005.

[7] Nielsen, E. J., and Kleb, W. L., "Efficient Construction of Discrete Adjoint Operators on Unstructured Grids Using Complex Variables," *AIAA Journal*, Vol. 44, No. 4, 2006, pp. 827–836.

[8] Burdyshaw, C., and Anderson, W., "A General and Extensible Unstructured Mesh Adjoint Method," AIAA Paper 2005-0335, 2005.

[9] Griewank, A., *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation*, No. 19, Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[10] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time-Stepping Schemes," AIAA Paper 81-1259, 1981.

[11] Mavriplis, D. J., "Multigrid Solution of the Discrete Adjoint for Optimization Problems on Unstructured Meshes," *AIAA Journal*, Vol. 44, No. 1, 2006, pp. 42–50.

[12] Kim, C., Kim, C., and Rho, O.H., "Feasibility Study of the Constant Eddy Viscosity Assumption in Gradient-Based Design Optimization," *Journal of Aircraft*, Vol. 40, No. 6, 2003, pp. 1168–1176.

[13] Gerhold, T., Galle, M., Friedrich, O., and Evans, J., "Calculation of Complex 3-D Configurations Employing the DLR TAU-Code," AIAA Paper 97-0167, 1997.

[14] Edwards, J., and Chandra, S., "Comparison of Eddy-Viscosity Transport Turbulence Models for Three-Dimensional Shock-Seperated Flowfields," *AIAA Journal*, Vol. 34, No. 4, 1996.

[15] Nielsen, E., and Park, M., "Using an Adjoint Approach to Eliminate Mesh Sensitivities in Computational Design," AIAA Paper 2005-0491, 2005.

[16] Giles, M., Duta, M., Muller, J., and Pierce, N., "Algorithm Developments for Discrete Adjoint Methods," *AIAA Journal*, Vol. 41, No. 2, 2003, pp. 198–205.

[17] Chisholm, T., and Zingg, D., "A Fully Coupled Newton–Krylov Solver for Turbulent Aerodynamic Flows," International Council of the Aeronautical Sciences Paper 333, 2002.

[18] Cantariti, F., Woodgate, M., Badcock, K., and Richards, B., "Solution of the Navier–Stokes Equations in Three Dimensions Using a Fully Unfactored Method," Department of Aerospace Engineering, Technical Rept. 9908, Glasgow University, 1999.

[19] Dwight, R., "An Implicit LU-SGS Scheme for Finite-Volume Discretizations of the Navier–Stokes Equations on Hybrid Grids," DLR Forschungsbericht, FB 2005-05, 2005.

[20] Geiger, C., and Kanzow, C., *Numerische Verfarhren zur Loesung unrestringierter Optimierungsaufgaben*, Springer, New York, 1999.

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

[22] Hicks, R., and Henne, P., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412.

[23] Reuther, J., Alonso, J., Rimlinger, M., Sanders, D., and Jameson, A., "Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 51–60.

[24] Wild, J., Mertins, R., Quagliarella, D., Brezillon, J., Quest, J., Amoignon, O., and Moens, F., "Applying Numerical Optimization to Realistic High-Lift Design of Transport Aircraft: An Overview of the Aerodynamic Design Optimization Investigations Within the EUROLIFT II Project," *Proceedings of EUROGEN 2005*, EUROGEN Committee, Munich, 2005.

[25] Brezillon, J., and Wild, J., "Evaluation of Different Optimization Strategies for the Design of a High-Lift Flap Device," *Proceedings of EUROGEN 2005*, EUROGEN Committee, Munich, 2005.

[26] Dwight, R., "Application of Approximately Factored Implicit Schemes to Unsteady Navier–Stokes Calculations," *Proceedings of the ICCFD3 Conference*, Springer, Berlin, 2006.

D. Gaitonde
Associate Editor