

Optimal Control of Unsteady Flows Using a Discrete and a Continuous Adjoint Approach

Angelo Carnarius¹, Frank Thiele², Emre Özkaya³, Anil Nemili³, and
Nicolas R. Gauger³

¹ Institut für Strömungsmechanik und Technische Akustik, TU Berlin,
Müller-Breslau-Str. 8, Berlin, 10623, Germany
angelo.carnarius@cfd.tu-berlin.de

² CFD Software Entwicklungs- und Forschungsgesellschaft mbH, Wolzogenstr. 4,
Berlin, 14163, Germany
frank.thiele@cfd-berlin.com

³ Computational Mathematics Group, CCES, RWTH Aachen University, Schinkelstr.
2, Aachen, 52062, Germany
{ozkaya,nemili,gauger}@mathcces.rwth-aachen.de

Abstract. While active flow control is an established method for controlling flow separation on vehicles and airfoils, the design of the actuation is often done by trial and error. In this paper, the development of a discrete and a continuous adjoint flow solver for the optimal control of unsteady turbulent flows governed by the incompressible Reynolds-averaged Navier-Stokes equations is presented. Both approaches are applied to testcases featuring active flow control of the blowing and suction type and are compared in terms of accuracy of the computed gradient.

Keywords: optimal control, active flow control, discrete adjoint, continuous adjoint, unsteady turbulent flows, URANS

1 Introduction

For many aerodynamic applications in aviation and automotive industry, flow separation has to be taken into account. The lift of an airfoil at a high angle of attack, for instance, decreases drastically, if the flow separates on the suction side.

Many studies in the past decades have shown that the aerodynamic behaviour of a body can be improved by using active flow control [4]. However, the choice of the control parameters is very case-specific and not trivial. An efficient method of finding the optimal set of actuation parameters is the gradient-based optimisation, which requires the calculation of the gradient of the cost function with respect to the control parameters. The control variables are then updated in an iterative manner according to a descent direction, which can be obtained from the gradient vector.

A very efficient way of computing the gradient is by using adjoint methods. Compared to simpler approaches such as Finite Differences or the Complex Taylor Series Expansion (CTSE) [13], adjoint-based methods compute the gradient

vector at a fixed expense independent of the number of actuation parameters. Adjoint methods are commonly divided into the continuous and the discrete approach.

In the continuous adjoint method [10], first the optimality system for a given objective function is derived and the resulting PDEs are then discretised and solved numerically. This procedure is called *first optimise then discretise*. The continuous approach is numerically efficient but it is known to suffer from consistency problems. The gradient can become inaccurate for insufficient time steps and grid spacing, which can be disadvantageous for complex configurations. Furthermore, most statistical turbulence models required for the unsteady Reynolds-averaged Navier-Stokes equations (URANS) are non-differentiable. The common approach is to use the so-called *constant eddy viscosity* or *frozen turbulence* assumption, i.e. the eddy viscosity is treated as independent of the control parameters and therefore taken from the primal solution. This assumption can lead to significant errors in the computed gradient [1].

The concept of the discrete adjoint method [3, 6, 12] is to *first discretise then optimise*, i.e. the discretised governing equations are used to derive the optimality system. This approach allows the generation of a fully consistent optimality system independent of the grid size, time step and turbulence model, as it does not require analytical differentiability [11]. Furthermore, Automatic Differentiation (AD) techniques [8] can be used to develop the discrete adjoint solver for a given simulation code in a semi-automatic fashion.

In this paper, we present the development of a continuous and a discrete adjoint solver for the optimal control of unsteady turbulent flows governed by the incompressible URANS equations. Both approaches, which are presented in more detail in sections 3 and 4, are based on the same flow solver ELAN [16]. For the current study, the adjoint solvers are applied to testcases which feature active flow control of the blowing and suction type.

2 Flow Model

Governing equations For this study, the unsteady, incompressible, turbulent flow in the domain Ω is described by the Reynolds-averaged Navier-Stokes equations⁴

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (1)$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j} \left[(\mu + \mu_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] = 0, \quad (2)$$

where u_i and p are the Reynolds-averaged velocity and pressure, respectively. The density ρ and the dynamic viscosity μ are constant for the cases shown here.

⁴ In the following, the Einstein summation convention is used, which implies summation from 1 to 3 over indices which appear twice in a single term. Indices, which appear only once take the value 1, 2 and 3 individually.

The eddy viscosity μ_t is obtained from the Wilcox-k- ω -model [15], which consists of transport equations for the turbulent kinetic energy k and the turbulent frequency ω .

Boundary Conditions At the farfield boundaries (Γ_i), u_i , k and ω were prescribed. On the body surface (Γ_b), u_i and $\partial k/\partial n$ were set to zero, whereas a high-Re boundary condition [16] was used for the turbulent frequency. At the outflow (Γ_o), the gradient of the turbulent quantities normal to the boundary was set to zero. For the Navier-Stokes equations, we want the sum of normal and friction forces to vanish at the outlet, i.e.

$$-pn_i + (\mu + \mu_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_j = 0. \quad (3)$$

At the control segment (Γ_c), the actuation velocity $c_i(b_j)$ as well as the turbulent quantities were prescribed. The dependence of $c_i(b_j)$ on the vector of the actuation parameters, b_j , is case-specific and is given in the testcase descriptions.

3 Continuous Adjoint Approach

Let J be the objective function to be minimised. Then the optimisation problem can be stated as

$$J(u_i, p, c_i(b_j)) \rightsquigarrow \min \text{ over } (u_i, p, c_i(b_j)) \text{ subject to } R(u_i, p, c_i(b_j)) = 0, \quad (4)$$

where R represents the state equations including the boundary conditions. In the cases presented here, the objective function can be written as⁵

$$\begin{aligned} J = & -\frac{1}{T} \int_0^T \int_{\Gamma_{b,c}} \left[(\mu + \mu_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_j - pn_i - \rho u_i u_j n_j \right] e_i dA dt \\ & + \frac{\gamma}{u_\infty} \frac{1}{T} \int_0^T \int_{\Gamma_c} \rho u^2 \sqrt{(u_i n_i)^2} + \epsilon dA dt. \end{aligned} \quad (5)$$

If the unity vector e_i is parallel to the mean flow, eq. 5 is the time-averaged drag. If e_i is oriented normal to the mean flow, eq. 5 represents the time-averaged downforce. The second integral is a penalty term which accounts for the energy consumption of the actuation and can be scaled by the factor γ . The parameter ϵ is only required for the differentiability of the penalty term, i.e. $1 \gg \epsilon > 0$.

To solve the minimisation problem, one first introduces the Lagrange function

$$L(u_i, p, c_i(b_j), v_i, q) = C J(u_i, p, c_i(b_j)) + \int_0^T \int_{\Omega} q R_\rho dV dt + \int_0^T \int_{\Omega} v_i R_u dV dt, \quad (6)$$

⁵ Note, that the negative sign of the first integral is a result of the convention that the normal vector n_i is directed out of the wall-adjacent control volume, i.e. into the body surface.

where the Lagrange multipliers v_i and q are the adjoint velocity and pressure, respectively, and C is a scaling factor to fix the units. By setting the variation of L with respect to the state variables, $\frac{\partial L}{\partial u_k} \delta u_k$ and $\frac{\partial L}{\partial p} \delta p$, to zero, one can obtain the adjoint equations. First the variations δu_k and δp have to be separated from other terms by using integration by parts and the boundary conditions have to be applied to the boundary integrals. As the resulting equations have to be fulfilled for any variation δu_k and δp , all integrals have to vanish individually, which gives the adjoint equations and boundary conditions. Setting the variation of L with respect to the control to zero and using the same procedure gives the equation for the gradient calculation. Due to the page limitation, a detailed derivation has to be omitted and the adjoint system can only be summarised. The adjoint PDEs read

$$\begin{aligned}
& \frac{\partial v_i}{\partial x_i} = 0 \quad \Omega \\
& -\frac{\partial \varrho v_i}{\partial t} + \varrho v_j \frac{\partial u_j}{\partial x_i} - \frac{\partial \varrho u_j v_i}{\partial x_j} + \frac{\partial q}{\partial x_i} - \frac{\partial}{\partial x_j} \left[(\mu + \mu_t) \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] = 0 \quad \Omega \\
& v_i = 0 \quad \Gamma_i \\
& \varrho v_i u_j n_j - q n_i + (\mu + \mu_t) \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) n_j = 0 \quad \Gamma_o \\
& v_i + \frac{C}{T} e_i = 0 \quad \Gamma_{b,c},
\end{aligned} \tag{7}$$

with the initial condition $v_i = 0$ at $t = T$. The gradient w.r.t. the actuation parameters can be evaluated from

$$\begin{aligned}
\frac{dJ}{db_n} &= \int_0^T \int_{\Gamma_c} \left[-\varrho u_j v_j n_m - q n_m + (\mu + \mu_t) \left(\frac{\partial v_m}{\partial x_j} + \frac{\partial v_j}{\partial x_m} \right) n_j \right] \frac{\partial c_m}{\partial b_n} dA dt \\
&+ \frac{\gamma}{u_\infty} \frac{C}{T} \int_0^T \int_{\Gamma_c} \left[\frac{\varrho c_k c_k}{\sqrt{c_i n_i c_j n_j} + \epsilon} c_l n_l n_m + 2\varrho c_m \sqrt{c_i n_i c_j n_j} + \epsilon \right] \frac{\partial c_m}{\partial b_n} dA dt.
\end{aligned} \tag{8}$$

Note, that the frozen turbulence assumption has been used, i.e. an adjoint turbulence model is not required.

4 Discrete Adjoint Approach

If we consider the discrete implementations of the objective function J and the state equations R , the discrete optimisation problem can be stated as:

$$J_d(y, b_i) \rightsquigarrow \min \text{ over } (y, b_i) \text{ subject to } R_d(y, b_i) = 0, \tag{9}$$

where $y = (u_i, p)$ is the discrete state vector and J_d, R_d denote the discrete implementations of J and R . Note, that in the discrete realisation the actuation

variables b_i are chosen as independent variables. The gradient of J_d with respect to the actuation parameters b_i can be computed from

$$\frac{dJ_d}{db_i} = \frac{\partial J_d}{\partial b_i} - \psi^\top \frac{\partial R_d}{\partial b_i}, \quad (10)$$

where the adjoint vector ψ can be determined by solving the adjoint system

$$\left(\frac{\partial R_d}{\partial y} \right)^\top \psi = \frac{\partial J_d}{\partial y}. \quad (11)$$

One way of constructing the adjoint system is by computing $\partial R_d/\partial y$ and $\partial J_d/\partial y$ using Finite Differences. The linear system of equations is then hand-coded and solved by an iterative method (e.g. GMRES). The resulting adjoint variables are used to calculate the gradient vector in eq. 10. A more promising way of developing the adjoint system is by employing the reverse mode of AD, which has the major advantage that it constructs the adjoint system consistently and computes the gradient vector dJ_d/db_i accurate to machine precision. In the present work, the discrete adjoint solver is developed by employing the AD tool TAPENADE [9] in reverse mode of differentiation.

If the reverse mode of AD is applied in a black-box fashion, the resulting adjoint code will have tremendous memory requirements. In order to reduce the excessive memory demands, we apply the reverse accumulation and checkpointing strategies for the Automatic Differentiation of the underlying flow solver. The solution strategy for the incompressible URANS equations mainly consists of two iterative loops: the time evolution step and the iterations for the velocity-pressure coupling scheme. Inside each time step, the velocity-pressure coupling iterations are performed, which are commonly known as *outer iterations* in the CFD community. It may be noted, that the outer iterations for the velocity-pressure coupling scheme converge to a fixed point in each time step. If AD is applied to these outer iterations in a black-box fashion, the flow solutions at each outer iteration of the primal solver must be saved for the adjoint part. However, the adjoint iterations require only the converged primal solution. Therefore, a lot of memory and run time can be saved, if we make use of the iterative structure and store only the converged flow solution in each physical time step. This can be achieved by employing the reverse accumulation approach [2, 5], the details of which are presented in the following.

Consider the total derivative of a discrete objective function J_d with respect to the control b_i at the converged state solution y^* for any time step:

$$\frac{dJ_d(y^*, b_i)}{db_i} = \frac{\partial J_d(y^*, b_i)}{\partial b_i} + \frac{\partial J_d(y^*, b_i)}{\partial y^*} \frac{dy^*}{db_i}. \quad (12)$$

On the other hand, if we have a fixed point for the state solution $y^* = G(y^*, b_i) \Leftrightarrow R_d(y^*, b_i) = 0$, we get

$$\frac{dy^*}{db_i} = \frac{\partial G(y^*, b_i)}{\partial b_i} + \frac{\partial G(y^*, b_i)}{\partial y^*} \frac{dy^*}{db_i} = \left(I - \frac{\partial G(y^*, b_i)}{\partial y^*} \right)^{-1} \frac{\partial G(y^*, b_i)}{\partial b_i}. \quad (13)$$

Multiplying on both sides with $\frac{\partial J_d(y^*, b_i)}{\partial y^*}^\top$, we obtain

$$\left(\frac{\partial J_d(y^*, b_i)}{\partial y^*}\right)^\top \frac{dy^*}{db_i} = \underbrace{\left(\frac{\partial J_d(y^*, b_i)}{\partial y^*}\right)^\top \left(I - \frac{\partial G(y^*, b_i)}{\partial y^*}\right)^{-1}}_{:=\bar{y}^{*\top}} \frac{\partial G(y^*, b_i)}{\partial b_i}. \quad (14)$$

From the definition of $\bar{y}^{*\top}$ in equation (14) and making use of equation (13), the adjoint fixed point iteration can be written as

$$\bar{y}^{*\top} = \bar{y}^{*\top} \frac{\partial G(y^*, b_i)}{\partial y^*} + \left(\frac{\partial J_d(y^*, b_i)}{\partial y^*}\right)^\top. \quad (15)$$

The first term on the right hand side of the above equation is the adjoint of a single outer iteration. This can be generated by applying the reverse mode of AD to the wrapper subroutine G , which combines all the steps done within one outer iteration of the flow solver. The gradient vectors $\partial J_d/\partial y^*$ and $\partial J_d/\partial b_i$ come from the adjoint of the post-processor, which is computed only once for each time iteration.

We now focus our attention on adjoining the time iterations. In general, the computation of the unsteady adjoint solution over the time interval $[0, T]$ with N time steps requires the storage of flow solutions at time steps T_0 to T_{N-1} . The stored solutions are then used in solving the adjoint equations from T_N to T_0 . For many practical aerodynamic configurations with millions of grid points and a large number of unsteady time steps, the storage costs may become prohibitively expensive.

One way of circumventing the excessive storage cost is by employing a checkpointing strategy [8], where the flow solutions are stored only at selective time steps known as checkpoints. These are then used to recompute the intermediate states that have not been stored. In the present example, we chose r ($r \ll N$) checkpoints. We then have $0 = T_0 = T_{C_1} < T_{C_2} < \dots < T_{C_{r-1}} < T_{C_r} < T_N = T$. Here, T_{C_r} represents the time step at r^{th} checkpoint. During the adjoint computation over the subinterval $[T_{C_r}, T_N]$, required flow solutions at intermediate time steps are recomputed by using the stored solution at T_{C_r} as the initial condition. The above procedure is then repeated over other subintervals $[T_{C_{r-1}}, T_{C_r}]$ until all adjoints are computed. It may be noted, that the checkpoints can be reused when they become free. We designate them as intermediate checkpoints.

Various checkpointing strategies have been proposed based on the storage criteria. If all the checkpoints are stored in main memory, it is called single-stage checkpointing. In yet another approach called multi-stage checkpointing [14], the checkpoints are stored both in main memory and on hard-disk, thus reducing the number of flow recomputations. In the present work, we have used the single-stage binomial checkpointing strategy, which is implemented in the algorithm `revolve` [7] and generates the checkpointing schedules in a binomial fashion, so that the number of flow recomputations is proven to be optimal.

5 Numerical Results

5.1 Cylinder with Pulsed Blowing and Suction

The first application is the unsteady laminar flow around a circular cylinder at a Reynolds-number of $Re = 100$, based on the cylinder diameter D and the freestream velocity u_∞ . The objective is to reduce the drag by applying pulsed blowing or suction according to

$$c_n = u_a \sin [2\pi f (t - t_0)] - u_a \quad (16)$$

on 15 slits, which are equidistantly distributed in 75% of the cylinder surface, see fig. 1(a). In eq. 16, c_n is the actuation velocity normal to the slit surface

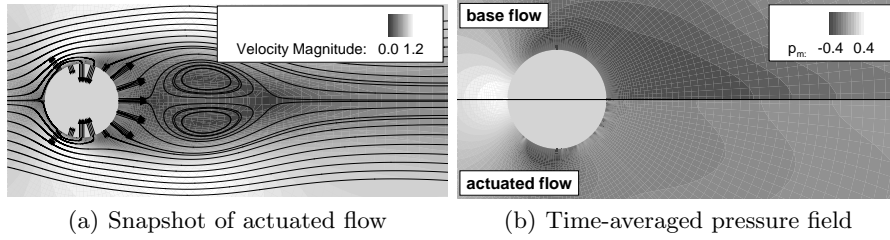


Fig. 1. Contour plots for the cylinder flow

and u_a , f and t_0 are the amplitude, frequency and phase shift, respectively. The actuation mode, i.e. blowing or suction, is set by the sign of the amplitude. For the case studied here, the actuation amplitudes at all slits are the parameters to be optimised, while the frequency and phase shift were fixed to $f = 1 u_\infty / D$ and $t_0 = 0 D / u_\infty$, respectively.

Only the continuous adjoint flow solver was applied to this testcase in order to test its accuracy in the unsteady laminar mode, i.e. without the influence of the frozen turbulence assumption. A numerical mesh consisting of about 25000 control volumes (CV) and a time step of $\Delta t = 0.04 D / u_\infty$ was used for the computations. In every iteration of the optimisation, which was performed with the steepest descent method, the primal solution was integrated over 15000 time steps. For the calculation of the objective function and the gradient, the first 5000 time steps were neglected to remove the initial transient. The optimisation was terminated when all sensitivities had dropped by two orders of magnitude.

As can be seen from fig. 2(a), the drag coefficient of the cylinder decreases from $c_d = 1.336$ to $c_d = 0.899$ when actuated with the optimal control parameters, which is a reduction of more than 30%. The comparison of the sensitivities at the first optimisation step shows a good agreement of the adjoint-based gradient with Finite Differences, see fig. 2(b). There are only small deviations, which can be attributed to the insufficient grid spacing and time step. Note, that only

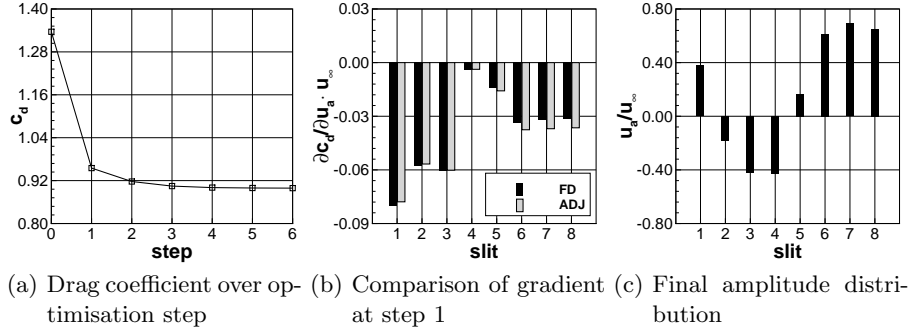


Fig. 2. Optimisation results for the cylinder flow

the slits on the upper half of the cylinder are shown, as the optimisation leads to a symmetric actuation. The same holds for the optimal amplitude distribution, which is presented in fig. 2(c). The blowing and suction at slits one to four generates a symmetric vortex pair which is pushed away from the rear of the cylinder by the blowing at slits five to eight. As is obvious in fig. 1(b), this increases the pressure level behind the cylinder, thus reducing the pressure drag.

5.2 NACA0015 Airfoil with Synthetic Jet Actuation

The second application is the lift maximisation for the unsteady turbulent flow around a NACA0015 airfoil at $Re = 10^6$, based on the cord length c and the freestream velocity u_∞ . The angle of attack (AoA) is $\alpha = 20^\circ$, leading to a massive separation on the suction side. In this case, sinusoidal blowing and suction (also called synthetic jet), which can be modelled according to

$$c_i = u_a r_i \sin [2\pi f (t - t_0)], \quad r_i = \begin{pmatrix} \cos(\beta - \theta) \\ \sin(\beta - \theta) \end{pmatrix}, \quad (17)$$

is applied at four slits on the suction side of the airfoil with a constant frequency of $f = 1.28 u_\infty / c$. Compared to the pulsed actuation (eq. 16) the blowing angle β can now be varied. The angle of the slit surface, θ , is fixed by the geometry of the airfoil. Computations with the discrete and the continuous adjoint solver were performed on a coarse mesh with 9500 CV and $\Delta t = 0.005 c / u_\infty$ over 100 time steps, including the initial transient.

The comparison of the sensitivity gradients, summarised in tab. 1, reveals an excellent agreement between the forward and reverse mode AD, giving only very small differences of approx. 1×10^{-5} . Compared to the AD-based solver, the results of the continuous adjoint code are significantly less accurate. One reason for this is the insufficient grid spacing, which is known to cause consistency problems with the continuous adjoint approach [11]. Furthermore, this can also be attributed to the frozen turbulence assumption. The active flow control modifies the separation on the suction side of the airfoil considerably, which has a

control parameter	forward mode AD	reverse mode AD	continuous adjoint
amplitude slit 1	0.132843488475446	0.132869414677547	0.070114751633607
amplitude slit 2	0.167065662623720	0.167070460770784	0.091718158272631
amplitude slit 3	0.181252126166289	0.181247271988999	0.103029635268416
amplitude slit 4	0.155843813170431	0.155844489164031	0.078944639252318
angle slit 1	0.005209720130677	0.005212791392634	0.000849278587241
angle slit 2	0.006705398122871	0.006697219503597	0.000654244527370
angle slit 3	0.006841527973356	0.006841492789784	0.002024334722777
angle slit 4	0.007246750396135	0.007246751418587	0.001287773996372
phase slit 1	0.204178681978258	0.204246051913213	0.278962118275295
phase slit 2	0.244693324906123	0.244791572866917	0.295707942189874
phase slit 3	0.244819168327026	0.244817849004966	0.304905513643976
phase slit 4	0.125955476539906	0.125957535080716	0.150374244523809

Table 1. Comparison of the sensitivities for the NACA0015 testcase

strong impact on the turbulence field. This is completely neglected by the frozen turbulence assumption.

6 Summary and Outlook

In this paper, the development of a continuous and a discrete adjoint flow solver for the optimal control of unsteady, turbulent flows governed by the incompressible URANS equations was presented. For the continuous adjoint approach, the wide-spread frozen turbulence assumption was used, while the AD-based discrete approach is fully consistent independent of the grid size, time step and turbulence model, as it does not require analytical differentiability. The numerical efficiency of the discrete solver has been improved by employing the reverse accumulation technique and the binomial checkpointing, which allows the application of the discrete adjoint solver to practical configurations.

The numerical results of the drag reduction of the cylinder flow showed, that the continuous adjoint method works well for unsteady laminar flows. However, it gives fairly inaccurate sensitivity gradients when applied to the turbulent flow around a NACA0015 airfoil at a high Re-number due to the frozen turbulence assumption and insufficient grid spacing. In contrast to this, the sensitivities obtained from the AD-based adjoint solver are of excellent accuracy and match the forward mode AD nearly perfectly.

In future studies, the different approaches will be applied to more complex geometries such as multi-element high-lift configurations or simplified car models, aiming at a more detailed comparison of the adjoint methods in terms of accuracy and numerical efficiency.

Acknowledgments This research was partly funded by the German Science Foundation (DFG) within the scope of the project *Instationäre Optimale Strömungskontrolle aerodynamischer Konfigurationen*.

References

1. Carnarius, A., Thiele, F., Özkaya, E., Gauger, N.R.: Adjoint approaches for optimal flow control. AIAA Paper 2010-5088 (2010)
2. Christianson, B.: Reverse accumulation and attractive fixed points. *Optimization Methods and Software* 3, 311–326 (1994)
3. Elliot, J., Peraire, J.: Practical 3D aerodynamic design and optimization using unstructured meshes. *AIAA Journal* 35(9), 1479–1485 (1997)
4. Gad-el-Hak, M.: The Taming of the Shrew: Why Is It so Difficult to Control Turbulence. In: *Active Flow Control, Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, vol. 95, Springer (2007)
5. Gauger, N.R., Walther, A., Moldenhauer, C., Widhalm, M.: Automatic Differentiation of an Entire Design Chain for Aerodynamic Shape Optimization. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design* 96, 454–461 (2007)
6. Giles, M.B., Ghattas, D., Duta, M.C.: Algorithm Developments for Discrete Adjoint Methods. *AIAA Journal* 41(2), 198–205 (2003)
7. Griewank, A., Walther, A.: Algorithm 799: Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Trans. Math. Software* 26(1), 19–45 (2000)
8. Griewank, A., Walther, A.: *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation* 2nd edition. SIAM, Other Titles in Applied Mathematics, Philadelphia (2008)
9. Hascoët, L., Pascual, V.: TAPENADE 2.1 user’s guide. Technical Report, INRIA (2004)
10. Jameson, A.: Aerodynamic Design via Control Theory. *Journal of Scientific Computing*, vol. 3, pp. 233–260 (1988)
11. Nemili, A., Özkaya, E., Gauger, N. R., Carnarius, A., Thiele, F.: Optimal Control of Unsteady Flows Using Discrete Adjoints. AIAA-Paper 2011-3720 (2011)
12. Nielsen, E., Anderson, W.K.: Aerodynamic design optimization on unstructured meshes using the Navier-Stokes equations. *AIAA Journal* 37(11), 957–964 (1999)
13. Squire, W., Trapp, G.: Using complex variables to estimate derivatives of real functions. *SIAM Rev.*, vol. 10, no. 1, pp. 110–112 (1998)
14. Stumm, P., Walther, A.: Multistage approaches for optimal offline checkpointing. *SIAM J. Scientific Computing* 31(3), 1946–1967 (2009)
15. Wilcox, D.C.: Reassessment of the scale determining equation for advanced turbulence models. *AIAA Journal*, vol. 26, no. 11 (1988)
16. Xue, L.: Entwicklung eines effizienten parallelen Lösungsalgorithmus zur dreidimensionalen Simulation komplexer turbulenter Strömungen. PHD thesis, Institut für Strömungsmechanik und Technische Akustik, Technische Universität Berlin (1998)