

Airfoil design for compressible inviscid flow based on shape calculus

Stephan Schmidt · Caslav Ilic · Volker Schulz ·
Nicolas R. Gauger

Received: 7 April 2008 / Accepted: 17 April 2011 / Published online: 12 May 2011
© Springer Science+Business Media, LLC 2011

Abstract Aerodynamic design based on the Hadamard representation of shape gradients is considered. Using this approach, the gradient of an objective function with respect to a deformation of the shape can be computed as a boundary integral without any additional “mesh sensitivities” or volume quantities. The resulting very fast gradient evaluation procedure greatly supports a one-shot optimization strategy and coupled with an appropriate shape Hessian approximation, a very efficient shape optimization procedure is created that does not deteriorate with an increase in the number of design parameters. As such, all surface mesh nodes are used as shape design parameters for optimizing a variety of lifting and non-lifting airfoil shapes using the compressible Euler equations to model the fluid.

Keywords Shape derivative · Preconditioning · One-shot · Drag reduction · Euler flow

S. Schmidt (✉) · V. Schulz
Department of Mathematics, University of Trier, Trier, Germany
e-mail: Stephan.Schmidt@uni-trier.de

V. Schulz
e-mail: Volker.Schulz@uni-trier.de

C. Ilic
German Aerospace Center (DLR), Braunschweig, Germany
e-mail: caslav.ilic@dlr.de

N.R. Gauger
Computational Mathematics Group, RWTH Aachen University, Aachen, Germany
e-mail: gauger@mathcces.rwth-aachen.de

1 Introduction

Numerical optimization in general and aerodynamic shape optimization in particular is a field that has seen much progress in the past decade. Two major advancements have been the introduction of gradient computation via the adjoint method (Gauger 2003; Jameson 1988) the optimization by the one-shot method (Gherman 2008; Hazra and Schulz 2004, 2005, 2006; Ta'asan 1991; Ta'asan et al. 1991). Since the adjoint approach makes gradient computation in principle independent of the number of design parameters, it appears natural to use the maximum possible degrees of freedom in the optimization, and that means using the coordinates of the mesh nodes themselves. In addition to the very costly computation of the “mesh sensitivity” Jacobian when using such a high number of design parameters, a high frequency noise in the shape is also often introduced. In order to reduce this noise, Jameson was one of the first to introduce the notion of gradient smoothing in this context (Jameson 1988, 1990, 1994).

Our aim is to prepare one-shot optimization for a large-scale 3D application. Key ingredients are the use of sensitivity-free shape derivatives and the correct smoothing procedure. Without shape derivatives, the computation of mesh sensitivities consumes any speedup the one-shot method might provide. Since the one-shot method is based on an inexact Newton scheme, information about the shape Hessian is needed for which we use the Laplace–Beltrami operator as suggested by operator symbol calculus in Arian and Ta'asan (1996), Arian and Vatsa (1998). Since the Laplace–Beltrami operator is a pseudo-differential operator of order $+2$, it also preserves the regularity of the shape by removing any high frequency noise. Sometimes, this approximative Newton scheme is also called gradient smoothing or Sobolev-gradient Method. By using shape derivatives and an appropriate shape Hessian approximation, a numerical shape optimization method is created, which efficiently operates without any parameterization such as the popular Hicks–Henne functions (Hicks and Henne 1978) or b-splines. Instead, the airfoil shape is allowed to morph freely without prior user knowledge. This can for example be seen in the automatic formation of a sharp leading edge in the supersonic case in Sect. 6. Here, we focus on airfoil design for compressible Euler flows. For the incompressible viscous regime see Schmidt and Schulz (2010).

The outline is as follows: Before Sect. 3 briefly recapitulates shape sensitivity analysis and the Hadamard theorem, the classical approach based on smooth ansatz functions is discussed in Sect. 2. Next, the one-shot optimization method is discussed in Sect. 4 and before the numerical results are presented in Sect. 6, shape sensitivity analysis for fluid forces using the compressible Euler equations to model the fluid is considered in Sect. 5.

2 Classical approach

In the classical approach, the aircraft or airfoil is usually parametrized by some smooth ansatz functions. Very popular is a deformation based on the Hicks–Henne functions (Hicks and Henne 1978), which can either be applied to the upper and lower

side of the airfoil separately or to the camberline. Application to the camberline has the advantage of the geometry constraint of constant thickness being embedded in the parameterization and need not be taken care of by the optimizer, making it quite popular (Kroll et al. 2009). Thus, we follow this approach for our reference calculations presented in this work. However, similar results to those presented here were also achieved when the Hicks–Henne functions were used on the lower and upper airfoil curve separately. Let $\text{cam}(x)$ be the position of the airfoil camber at chord position x . A new airfoil shape is then constructed according to

$$\text{cam}_{k+1}(x) = \text{cam}_k(x) + \sum_{i=1}^{n_q} q_i \cdot h_i(x), \quad (2.1)$$

where the q_i are the design parameters to be found by the optimization and h_i are the smooth ansatz functions, usually the Hicks–Henne bump functions. Geometric or engineering constraints such as the popular constant thickness constraint are usually hidden in the particular choice of ansatz functions and are not visible to the optimization. Using this approach, the shape optimization nature of the airfoil design problem is thus discarded and one considers a constrained optimization problem of the type

$$\min_{(u,q)} F(u, q),$$

where u is the state variable of the fluid model. For this standard optimization problem, the gradient can be computed by the Lagrangian

$$\begin{aligned} \frac{dF}{dq} &= \frac{\partial F}{\partial q} - \lambda^T \frac{\partial c}{\partial q}, \\ \left[\frac{\partial c}{\partial q} \right]^T \lambda &= \frac{\partial F}{\partial u}, \end{aligned} \quad (2.2)$$

where $c(u, q) = 0$ symbolizes the flow equation constraint. With the shape optimization nature removed by parametrization, the partial derivatives $\frac{\partial c}{\partial q}$ in the above Lagrangian are very problematic to compute. Essentially, the expression $\frac{\partial c}{\partial q}$ requires the derivative of the flow discretization, i.e. the flow solver, in chain rule with the derivative of the procedure that deforms the field mesh nodes to the new aircraft surface. While approaches based on (2.2) have been successfully used in complex optimization problems including compressible, turbulent flows (Nemec and Zingg 2002) or multi-point optimizations (Nemec et al. 2003), the required mesh sensitivities $\frac{\partial c}{\partial q}$ are almost always computed by finite differences, which is numerically extremely expensive because a perturbed mesh must be made for each design parameter and each optimization step. In fact, for higher order central finite differences, which are often preferred, even two perturbed meshes must be made per design parameter. Especially for three dimensional problems, this is prohibitive even though only the flow residual $\frac{\partial c}{\partial q}$ must be evaluated and not the actual flow sensitivities $\frac{\partial u}{\partial q}$.

In the following, we present an alternative approach based on shape calculus, which does not forsake the shape optimization nature of the problem by parametriza-

tion. Instead, shape gradients are used in a very efficient one-shot optimization approach, which bypasses the need for these partial derivatives and thus allows one to use the maximum degrees of freedom for the shape, i.e. the position of the surface mesh nodes.

3 Elements from shape calculus

3.1 Perturbed domains

In the following let the domain occupied by the fluid be given by $\Omega \subset \mathbb{R}^2$ compact with sufficiently smooth boundary $\partial\Omega$. Furthermore, let $\Gamma \subset \partial\Omega$ be the variable part of the boundary, i.e. the airfoil surface. We first consider a boundary objective, which does not yet depend on the geometry of the domain itself:

$$J(\Omega) := \int_{\Gamma} g(x) dS(x),$$

where $g : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a sufficiently smooth function not depending on the shape of Ω . A perturbed domain Ω_t is then given by

$$\Omega_t := T_t(\Omega) := \{T_t(x) : x \in \Omega\}.$$

The actual deformation is thus determined by the particular choice of the family T_t of bijective mappings. In the literature (Delfour and Zolésio 2001; Sokolowski and Zolésio 1992) T_t is usually chosen as either the perturbation of identity

$$T_t[V](x) := x + tV(x)$$

or the speed method, where $T_t[V] : x_0 \rightarrow T_t[V](x_0) := x(t, x_0)$ is the solution of the differential equation

$$\frac{\partial x}{\partial t} = V(t, x), \quad x(0, x_0) = x_0.$$

Here, V is a Lipschitz-continuous vector field that takes the role of the differentiation direction. Using the integral transformation for boundary integrals, the objective function over the perturbed domain can be brought back to the initial domain according to

$$J(\Omega_t) = \int_{\Gamma_t} g(x) dS_t(x) = \int_{\Gamma} g(T_t(x)) \|M(DT_t(x))n(x)\|_2 dS(x), \quad (3.1)$$

where $M(DT_t(x))$ is the cofactor-matrix of the Jacobian DT_t . The shape derivative is now defined as the limit

$$dJ(\Omega)[V] := \lim_{t \rightarrow 0^+} \frac{J(\Omega_t) - J(\Omega)}{t}$$

and computing this limit for (3.1) results in

$$dJ(\Omega)[V] = \int_{\Gamma} \langle \nabla g, V \rangle + g \operatorname{div}_{\Gamma} V dS, \quad (3.2)$$

where $\operatorname{div}_{\Gamma}$ is the divergence operator in the tangent space.

3.2 The Hadamard theorem

Equation (3.2) can be considered the shape optimization equivalent of a directional derivative in direction V . The Hadamard theorem or Hadamard–Zolésio structure theorem (Delfour and Zolésio 2001; Sokolowski and Zolésio 1992) states that under some regularity assumptions, the sensitivity equation (3.2) can be expressed as a scalar product of the normal component of the perturbation field V with some shape gradient $G : \Gamma \rightarrow \mathbb{R}$ on the surface of the geometric object to be optimized, i.e.

$$dJ(\Omega)[V] = \int_{\Gamma} \langle V, n \rangle G dS. \quad (3.3)$$

Thus, knowledge of the shape gradient G has some distinct advantages for application in a numerical optimization scheme, as the boundary of the fluid obstacle can easily be updated according to

$$\Gamma_{k+1} = \{x + \beta \langle V(x), n(x) \rangle n(x) G(x) : x \in \Gamma_k\}, \quad (3.4)$$

where β is the step length of the steepest descent algorithm and k is the iteration counter. It also becomes obvious that a natural choice is $V := n$ unless some parts of the geometry are to be kept fixed. Also, the Hadamard formula (3.3) and the resulting update formula (3.4) can be computed without considering field nodes, such that the need for mesh sensitivities or the consideration of how to deform the CFD mesh to match the new boundary iterate become negligible; at least insofar their derivative is no longer needed.

For the general objective function (3.2) the Hadamard form reads

$$dJ(\Omega)[V] = \int_{\Gamma} \langle V, n \rangle \left[\frac{\partial g}{\partial n} + \kappa g \right] dS, \quad (3.5)$$

where $\kappa := \operatorname{div}_{\Gamma} n$ is the additive mean curvature of Γ . Thus, the requirement to consider volume quantities and the mesh deformation is replaced by the need to compute some geometric quantities, such as curvature, for which very fast numerical schemes exist. The strategy considered here now consists of deriving the shape gradient G from (3.3) for the aerodynamic design problem and optimize the airfoil Γ according to

$$\Gamma_{k+1} = \{x + \langle V(x), n(x) \rangle n(x) \tilde{G}(x) : x \in \Gamma_k\}, \quad (3.6)$$

where \tilde{G} is an approximate Newton direction as discussed in the next section.

4 Optimization approach

4.1 One-shot method

In the following, a standard optimization problem is assumed:

$$\min_{(u,q)} f(u, q) \quad (4.1)$$

subject to

$$c(u, q) = 0, \quad (4.2)$$

$$d(u, q) = 0,$$

where q is some parametrized control and u is a PDE state, e.g. the conserved variables of the Euler equations $c(u, q) = 0$. Additionally, $d(u, q) = 0$ symbolizes additional scalar constraints such as a desired minimum lift or internal volume of the airfoil. The necessary optimality conditions are then given by

$$\nabla_u \mathcal{L} = 0,$$

$$\nabla_q \mathcal{L} = 0,$$

$$c(u, q) = \nabla_\mu \mathcal{L} = 0,$$

$$d(u, q) = \nabla_v \mathcal{L} = 0,$$

where the Lagrangian \mathcal{L} is defined by

$$\mathcal{L}(u, q, \mu, v) := f(u, q) + \langle \mu, c(u, q) \rangle + \langle v, d(u, q) \rangle.$$

Applying Newton's method on the above optimality conditions results in the SQP updates

$$\begin{bmatrix} H_{uu} & H_{uq} & (D_u c)^T & (D_u d)^T \\ H_{qu} & H_{qq} & (D_q c)^T & (D_q d)^T \\ D_u c & D_q c & 0 & 0 \\ D_u d & D_q d & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta u \\ \Delta q \\ \Delta \mu \\ \Delta v \end{pmatrix} = \begin{pmatrix} -\nabla_u \mathcal{L} \\ -\nabla_q \mathcal{L} \\ -c \\ -d \end{pmatrix}, \quad (4.3)$$

where

$$(u_{k+1}, q_{k+1}, \mu_{k+1}, v_{k+1})^T = (u_k, q_k, \mu_k, v_k)^T + (\Delta u, \Delta q, \Delta \mu, \Delta v)^T.$$

The one-shot method now assumes an approximation of the Hessian matrix in (4.3) by

$$\begin{bmatrix} H_{uu} & H_{uq} \\ H_{qu} & H_{qq} \end{bmatrix} \approx \begin{bmatrix} 0 & 0 \\ 0 & B \end{bmatrix}.$$

Assuming $(D_u c)^{-1}$ exists, a block Gauss-elimination and replacing Δv with $v_{k+1} = v_k + \Delta v$ results in the system

$$\begin{bmatrix} B & \tilde{D}_d \\ (\tilde{D}_d)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta q \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} -\nabla_q f + (D_q c)^T (D_u c)^{-T} \nabla_u f \\ -d + (D_u d)(D_u c)^{-1} c \end{pmatrix}. \quad (4.4)$$

Using the reduced derivative

$$\tilde{D}_f = \nabla_q f - (D_q c)^T (D_u c)^{-T} \nabla_u f,$$

the above system (4.4) can be transformed into

$$\tilde{D}_d^T B^{-1} \tilde{D}_d v_{k+1} = d - \lambda_d c - \tilde{D}_d^T B^{-1} \tilde{D}_f, \quad (4.5)$$

$$B \Delta q = -\tilde{D}_f - \tilde{D}_d v_{k+1}. \quad (4.6)$$

Thus, a shape one-shot method is now created by replacing the standard reduced derivatives in (4.5) and (4.6) by the respective shape gradients from (3.3). Looking at (4.6), one can see that an approximate Newton direction that is also aware of the additional scalar constraints can be computed according to

$$\tilde{G} = -B^{-1} (G_f + G_d v_{k+1}), \quad (4.7)$$

where G_f and G_d are the shape gradients in Hadamard form of the objective function and the scalar constraints. The resulting approximate Newton or Sobolev direction \tilde{G} is then applied to the airfoil boundary as in (3.6).

4.2 Hessian approximation

Crucial for fast convergence is thus the use of a proper Hessian approximation B . Here, a Hessian approximation is used based on the operator symbol of the design to state mapping using the PDE constraint only. Thus, this kind of approximation neglects the elements of the Hessian stemming from the partial derivative of the geometric quantities. For the viscous case of incompressible Stokes and Navier-Stokes flows such a symbol derivation can be found in all detail in Schmidt (2010), Schmidt and Schulz (2009). For potential flow, a similar analysis can be found in Eppler et al. (2009) and for the inviscid compressible case see Arian and Ta'asan (1996), Arian and Vatsa (1998). Thus, only a brief overview of this method is given here. Assuming an oscillatory control

$$\tilde{q}(x) = \hat{q} e^{i\omega x}, \quad (4.8)$$

where i is the imaginary unit, the symbol of an operator B can be seen by comparing the input \tilde{q} with the output $H\tilde{q}$. For example, if

$$B\tilde{q} = i\omega \hat{q} e^{i\omega x} = i\omega \tilde{q},$$

then $\Sigma(\omega) := i\omega$ is the symbol of the Hessian and this corresponds to a standard differential operator or order +1. Likewise, if one has

$$B\tilde{q} = -\omega^2\tilde{q},$$

then $\Sigma(\omega) = -\omega^2$, which corresponds to a standard second order differential operator.

A formal application of the chain rule to a PDE constraint shape optimization problem

$$\min_{(u, \Omega)} J(u, \Omega) := \int_{\Gamma} g(u) dS$$

subject to

$$L(u) = u_f \quad \text{in } \Omega,$$

$$L_b(u) = u_b \quad \text{on } \Gamma$$

immediately results in

$$dJ(u, \Omega)[V] := \int_{\Gamma} \langle V, n \rangle \left[\frac{\partial g(u)}{\partial n} + \kappa g(u) \right] dS + \int_{\Gamma} \frac{\partial g(u)}{\partial u} u'[V] dS, \quad (4.9)$$

where the first part is called the geometric part of the shape derivative and the second part is called the dynamic part of the shape derivative. Also,

$$u'[V](x) := \left. \frac{d}{dt} \right|_{t=0} u_t(x) \quad (4.10)$$

is called the local shape derivative, while

$$du[V](x) := \left. \frac{d}{dt} \right|_{t=0} u_t(x_t),$$

is the material derivative. They are related to each other by the chain rule according to

$$du[V] = u'[V] + \langle \nabla u, V \rangle. \quad (4.11)$$

Formally, the local shape derivative (4.10) is only defined on $\Omega_t \cap \Omega$. However, (4.11) can be used as a definition of the local shape derivative when $\Omega_t \cap \Omega = \emptyset$.

Using adjoint calculus, the sensitivity equation (4.9) can be transformed into

$$dJ(u, \lambda, \Omega)[V] = \int_{\Gamma} \langle V, n \rangle G(u, \lambda) dS,$$

where λ is the adjoint variable. For an approximation of the reduced Hessian B based on the dynamic part, the expression

$$\tilde{B} := \frac{\partial G}{\partial u} u'[\tilde{q}] + \frac{\partial G}{\partial \lambda} \lambda'[\tilde{q}]$$

is considered, which neglects the geometric part. Using the special structure of \tilde{q} from (4.8), the operator symbol can now be made explicit by solving

$$\frac{\partial L}{\partial u} u'[\tilde{q}] = 0, \quad L^T \lambda'[\tilde{q}] = 0$$

for the considered frequency ω . For the inviscid compressible case considered here, such an analysis is conducted in Arian and Ta'asan (1996), Arian and Vatsa (1998), showing the Hessian is a second order differential operator. For the incompressible viscous case (Schmidt 2010; Schmidt and Schulz 2009) the Hessian is a true pseudo-differential operator of order 1. Thus, we use the following reduced Hessian approximation

$$B \approx \epsilon \Delta_\Gamma + I, \quad (4.12)$$

where Δ_Γ is the Laplace-Beltrami operator, I is the identity operator and ϵ is a smoothing parameter.

4.3 Implementation

Usually, one has an iterative process, e.g. a fix-point iteration, to solve the flow and adjoint state. Embedding this in a shape optimization procedure, one automatically arrives at a two loop approach: One outer optimization loop and inner flow solver loops, which are needed to compute flow and adjoint state for each optimization iteration. In order to overcome this costly approach, the one-shot method from Sect. 4.1 is integrated into the flow solver according to the following algorithm: Given an initial airfoil shape Γ_i , and an initial flow state u_i and an initial adjoint state λ_i do

- (1) Perform some flow solver iterations to compute u_{k+1} .
- (2) Perform some adjoint flow solver iterations to compute λ_{k+1} using the new values u_{k+1} .
- (3) Compute the shape gradients of the objective functions and constraints.
- (4) Compute the one-shot descent direction according to (4.7)

$$\tilde{G} = -B^{-1}(G_f + G_d v_{k+1}).$$

- (5) Update the airfoil geometry by changing each node position according to (3.6)

$$\Gamma_{k+1} = \{x + \tau \tilde{G}(x)n(x) : x \in \Gamma_k\}.$$

- (6) Adapt the volume mesh and loop.

Here, τ is the step-length of the optimization procedure. Thus, in this approach intermediate results have to be treated with care, as the objective and constraints are evaluated from state values for which the state equation residual, i.e. the flow solver residual, is comparatively large. The respective values become exact again only at the end of the iteration, when optimization and flow residual haven been driven to zero simultaneously.

The next section will now derive the shape gradients for the usual Euler drag reduction problem. Afterwards numerical results are shown.

5 Euler drag reduction: shape derivatives

5.1 Problem setup

In this section, the shape derivative for the Euler drag minimization problem will be derived. For a given incident vector a the inviscid force component

$$F_a := \int_{\Gamma} p \langle a, n \rangle dS \quad (5.1)$$

is defined, where p is the fluid pressure. Thus, for a given angle of attack α we define

$$a_D := (\cos \alpha, \sin \alpha)^T$$

and the respective force F_D will be the drag force. Analogously, when using

$$a_L := (-\sin \alpha, \cos \alpha)^T,$$

the respective force F_L will be lift. Since the incident vector a is independent of the shape, it suffices to consider the general form (5.1) when deriving the gradient. For the inviscid airfoil optimization problem using the compressible Euler equations, the following problem is thus considered:

$$\min_{(\rho, u, E, \Omega)} F_D(\rho, u, E, \Omega) \quad (5.2)$$

subject to

$$\begin{aligned} \int_{\Omega} \operatorname{div}(\rho u) dA &= 0, \\ \int_{\Omega} \sum_{j=1}^2 \left[\frac{\partial(\rho u_i u_j)}{\partial x_j} \right] + \frac{\partial p}{\partial x_i} dA &= 0, \quad \text{for } i = 1, 2, \\ \int_{\Omega} \operatorname{div}(\rho H u) dA &= 0, \\ \langle u, n \rangle &= 0 \quad \text{on } \Gamma, \\ F_L &= F_{L_0}, \\ \operatorname{Vol} &= V_0. \end{aligned} \quad (5.3)$$

Here, the first three constraints are the Euler equations, and the pressure p is linked to the conserved variables

$$U := (\rho, \rho u, \rho E)^T$$

by the perfect gas law

$$p = (\gamma - 1) \rho \left(E - \frac{1}{2} \|u\|^2 \right),$$

where ρ is the fluid density, u is the fluid velocity vector, E is the total energy, and H is the enthalpy of the fluid. Also, γ is the adiabatic exponent. The condition $\langle u, n \rangle = 0$ is the Euler slip boundary condition on the aircraft surface Γ . The farfield boundary conditions are usually more complex involving characteristics and are thought of as treated by the flow solver discretely. Condition $F_L = F_{L_0}$ means the optimal aircraft must maintain lift and the volume constraint is needed to prevent a degeneration of the airfoil. Additionally, there is also the constraint of the leading edge being fixed at $(0, 0)^T$ and the trailing edge being fixed at $(1, 0)^T$. Otherwise, the optimization changes the reference length C_{ref} of the airfoil, which would lead to a wrong non-dimensionalization of the flow quantities during the operation of the flow solver.

5.2 Shape derivative

Assuming sufficient regularity on the domain boundaries and the fluid state such that the chain rule is applicable, a formal differentiation according to (3.5) immediately results in

$$dF_a(U, \Omega)[V] = \int_{\Gamma} \langle V, n \rangle \left[\frac{\partial p}{\partial n} \langle a, n \rangle + \kappa p \langle a, n \rangle \right] + p \langle a, dn[V] \rangle + p'[V] \langle a, n \rangle dS. \quad (5.4)$$

Next, the local shape derivative of the pressure $p'[V]$ is removed via adjoint calculus similar to Gauger (2003), Giles and Pierce (1997). Almost all compressible flow solvers operate on the conserved variables U . To be in-line with such flow solver, the Euler equations are therefore linearized with respect to

$$U'[V] = (\rho'[V], (\rho u)'[V], (\rho E)'[V])^T.$$

According to Gauger (2003), Giles and Pierce (1997), the vector C defined by

$$C := \left(\frac{\gamma-1}{2} (u_1^2 + u_2^2 + u_3^2), (1-\gamma)u_1, (1-\gamma)u_2, (1-\gamma)u_3, \gamma-1 \right)$$

satisfies

$$p'[V] = CU'[V], \quad (5.5)$$

thus translating the perturbation in the pressure to a perturbation in the conserved state variables. We use this property again in (5.8). The compressible Euler equations from (5.3) can also be written in non-conservative form

$$A_1 \frac{\partial U}{\partial x_1} + A_2 \frac{\partial U}{\partial x_2} + A_3 \frac{\partial U}{\partial x_3} = 0,$$

where A_k are the Euler Flux Jacobian matrices. For a definition of them see Gauger (2003). This form of the Euler equations makes the derivation of the linearized state equation rather convenient. It can be directly read as

$$\frac{\partial}{\partial x_1} (A_1 U'[V]) + \frac{\partial}{\partial x_2} (A_2 U'[V]) + \frac{\partial}{\partial x_3} (A_3 U'[V]) = 0. \quad (5.6)$$

Multiplication by an arbitrary $\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)$ and integration by parts gives

$$\begin{aligned} 0 &= \int_{\Omega} \sum_{k=1}^3 \lambda \frac{\partial}{\partial x_k} (A_k U'[V]) dA \\ &= - \int_{\Omega} \sum_{k=1}^3 \left(\frac{\partial}{\partial x_k} \lambda \right) A_k U'[V] dA + \int_{\partial\Omega} \sum_{k=1}^3 \lambda n_k A_k U'[V] dS \\ &= - \int_{\Omega} \sum_{k=1}^3 A_k^T \left(\frac{\partial \lambda}{\partial x_k} \right) U'[V] dA + \int_{\partial\Omega} \sum_{k=1}^3 \lambda n_k A_k U'[V] dS. \end{aligned} \quad (5.7)$$

Looking at (5.7) we see that if λ solves the adjoint equation

$$-A_1^T \frac{\partial}{\partial x_1} \lambda - A_2^T \frac{\partial}{\partial x_2} \lambda - A_3^T \frac{\partial}{\partial x_3} \lambda = 0 \quad \text{in } \Omega$$

then the volume integrals will vanish.

According to Gauger (2003), Giles and Pierce (1997), the matrices

$$T := \begin{bmatrix} 0 & \frac{n_1}{\rho} & \frac{n_2}{\rho} & \frac{n_3}{\rho} & 0 \\ \frac{\gamma-1}{2}(u_1^2 + u_2^2 + u_3^2) & (1-\gamma)u_1 & (1-\gamma)u_2 & (1-\gamma)u_3 & \gamma-1 \end{bmatrix}$$

and

$$T^* := \begin{bmatrix} \rho & \rho u_1 & \rho u_2 & \rho u_3 & \rho H \\ 0 & n_1 & n_2 & n_3 & 0 \end{bmatrix}$$

satisfy the condition

$$\sum_{k=1}^3 n_k A_k = T^{*T} T$$

on the boundary Γ of the airfoil. The enthalpy H of the fluid is defined by $\rho H = \rho E + p$. Recalling (5.5) we now see that

$$T U'[V] = \left(\frac{\sum_{k=1}^3 n_k u'_k[V]}{p'[V]} \right) \quad (5.8)$$

and

$$\lambda T^{*T} = (\lambda U_H, (\lambda_2, \lambda_3, \lambda_4, \lambda_5)n),$$

where $u'[V]$ is the perturbation in the speed of the fluid and

$$U_H := (\rho, \rho u_1, \rho u_2, \rho u_3, \rho H)^T$$

are the conserved variables with the last component replaced by ρH . Finally, we can simplify (5.7) with the above to

$$\begin{aligned} 0 &= \int_{\Gamma} \lambda \sum_{k=1}^3 n_k A_k U'[V] dS = \int_{\Gamma} \lambda T^{*T} T U'[V] dS \\ &= \int_{\Gamma} \lambda U_H \langle u'[V], n \rangle + (\lambda_2, \lambda_3, \lambda_4) n p'[V] dS. \end{aligned}$$

The linearized Euler slip boundary condition on the airfoil is given by

$$\langle du[V], n \rangle + \langle u, dn[V] \rangle = 0,$$

where $du[V]$ is the material derivative of the fluid velocity. According to (4.11), the corresponding boundary condition for the local shape derivative of the velocity is given by

$$\begin{aligned} \langle u'[V], n \rangle &= -\langle DuV, n \rangle - \langle u, dn'[V] \rangle \\ &= -\langle V, n \rangle \left\langle \frac{\partial u}{\partial n}, n \right\rangle - \langle u, dn'[V] \rangle, \end{aligned}$$

where the second part holds for normal perturbations only. However, this suffices according to the Hadamard theorem as tangential perturbations do not change the shape. When inserted in the above equation, one arrives at

$$\begin{aligned} 0 &= \int_{\Gamma} \lambda \sum_{k=1}^3 n_k A_k U'[V] dS = \int_{\Gamma} -\langle V, n \rangle \lambda U_H \left\langle \frac{\partial u}{\partial n}, n \right\rangle - \lambda U_H \langle u, dn[V] \rangle \\ &\quad + (\lambda_2, \lambda_3, \lambda_4) n p'[V] dS. \end{aligned}$$

Adding the above to the preliminary gradient (5.4), we see that

$$\begin{aligned} dF_a(U, \Omega)[V] &= \int_{\Gamma} \langle V, n \rangle \left[\frac{\partial p}{\partial n} \langle a, n \rangle + \kappa p \langle a, n \rangle - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n \right\rangle \right] \\ &\quad + \langle pa - \lambda U_H u, dn[V] \rangle + p'[V] [\langle a, n \rangle + (\lambda_2, \lambda_3, \lambda_4) n] dS. \end{aligned}$$

If the adjoint boundary condition

$$(\lambda_2, \lambda_3, \lambda_4) n + \langle a, n \rangle = 0$$

is satisfied on the wing, the gradient will further simplify to

$$\begin{aligned} dF_a(U, \Omega)[V] &= \int_{\Gamma} \langle V, n \rangle \left[\frac{\partial p}{\partial n} \langle a, n \rangle + \kappa p \langle a, n \rangle - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n \right\rangle \right] \\ &\quad + \langle pa - \lambda U_H u, dn[V] \rangle dS. \end{aligned} \quad (5.9)$$

The resulting gradient expression (5.9) is already very well applicable for numerical computations. The required material derivative of the normal $dn[V]$ can be computed

quite efficiently using finite differences. However, shape calculus can also be used to remove the material derivative of the normal making the above gradient expression fully in-line with the Hadamard theorem.

5.3 Shape derivative of the normal

The parts of the gradient that involve the perturbation of the normals $dn[V]$ are not yet sensitivity free and thus are not in Hadamard form. Therefore we now focus on equations of the type

$$\int_{\Gamma} \langle \varphi, dn[V] \rangle dS.$$

According to Delfour and Zolésio (2001), Sokolowski and Zolésio (1992), the variation of the normal is given by

$$dn[V] = \langle DVn, n \rangle n - DV^T n - D^2 bV,$$

where b is some extension of the normal field into the domain and DV is the Jacobian of the perturbation field V . Recall the property

$$DV^T n = D_{\Gamma} V^T n + \langle DVn, n \rangle n$$

with D_{Γ} being the tangential Jacobian. Thus, the normal variation can also be expressed as

$$\langle DVn, n \rangle n - DV^T n = -D_{\Gamma} V^T n.$$

Using the above equations one sees that

$$\begin{aligned} \langle \varphi, dn[V] \rangle &= \langle \varphi, \langle DVn, n \rangle n - DV^T n - D^2 bV \rangle \\ &= \langle \varphi, -D_{\Gamma} V^T n - D^2 bV \rangle \\ &= -\langle \varphi, D_{\Gamma} V^T n + D^2 bV \rangle = -\langle \varphi, \nabla_{\Gamma} \langle V, n \rangle \rangle. \end{aligned}$$

Thus, the gradient can further be refined to

$$\begin{aligned} dF_a(U, \Omega)[V] &= \int_{\Gamma} \langle V, n \rangle \left[\frac{\partial p}{\partial n} \langle a, n \rangle + \kappa p \langle a, n \rangle - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n \right\rangle \right] \\ &\quad - \langle pa - \lambda U_H u, \nabla_{\Gamma} \langle V, n \rangle \rangle dS. \end{aligned} \quad (5.10)$$

Using the tangential Stokes formula from tangential calculus,

$$\int_{\Gamma} g \operatorname{div}_{\Gamma} V + \langle \nabla_{\Gamma} g, V \rangle dS = \int_{\Gamma} \kappa g \langle V, n \rangle dS,$$

(5.10) can be transformed into Hadamard form

$$dF_a(U, \Omega)[V] = \int_{\Gamma} \langle V, n \rangle \left[\frac{\partial p}{\partial n} \langle a, n \rangle - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n \right\rangle + \operatorname{div}_{\Gamma}(pa - \lambda U_H u) \right] dS. \quad (5.11)$$

Thus, the need to compute curvature and normal variation is replaced by one computation of the tangential or surface divergence operator.

6 Euler drag reduction: application

6.1 Numerical procedure

The gradient of the objective F_D and the lift constraint F_L is computed either using the adjoint flow solution with finite differencing for mesh sensitivities (Gauger 2003) as a reference solution, or by the shape derivative (5.11), which removes the need for mesh sensitivities. The gradient of the volume constraint is not flow dependent, and can be easily derived analytically as

$$d \operatorname{Vol}(\Omega)[V] = \int_{\Gamma} \langle V, n \rangle 1 dS$$

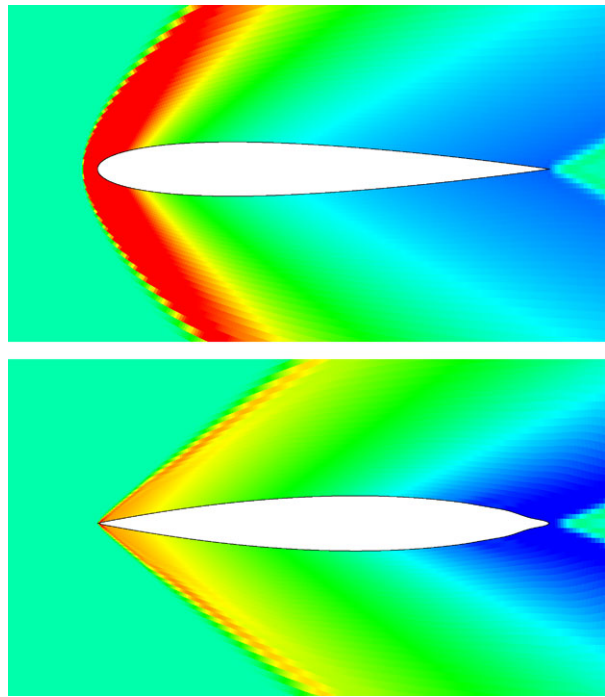
or computed on the discrete level. The Euler equations are solved using the DLR TAU code, with the continuous adjoint formulation for the co-state (Widhalm et al. 2007). This flow solver is based on hybrid finite volume multigrid pseudo-time stepping. As flow discretization, both upwind and central schemes with stabilization can be selected. It also comes with corresponding mesh deformation utility programs, which are used to fit the field mesh to the new airfoil surface in each iteration. It is worth noting that the shape derivative approach does not need the partial derivative of these mesh adaptation subroutines for a correct gradient computation. Any other approach to fit the field mesh (including re-meshing) is equally usable.

In order to avoid stability issues with the deformation of the volume mesh to match the new airfoil boundary, the maximum step size is determined either as a fraction of the local grid size on the boundary, or a small multiple of the previous step size, and the line search backtracks from that value. This is a practical measure to allow mesh-deformation tools to perform well, but we have observed that after a few initial optimization steps, the step sizes anyway tend to be less than the grid-fraction maximum.

6.2 Supersonic non-lifting case

A simple 2D problem that demonstrates the features of Euler flow drag reduction is that of a symmetric airfoil in supersonic flow (e.g. the cross-section of an airplane's vertical fin). The supersonic setup, as opposed to subsonic, is advantageous for testing purposes for several reasons. Physical drag is always present and substantially higher than numerical drag, i.e. effects from artificial viscosity stemming from the dissipation/stabilization terms of central finite volume schemes (Gauger 2003;

Fig. 1 Initial NACA0012 airfoil and resulting optimized ogive body at Mach 2.0. Pressure field shown



Jameson et al. 1981), optimal shapes for the different constraints are known beforehand (Haack 1941), and it is easier to capture the drag correctly using coarser grids. In subsonic flow, physical drag may disappear, leading to many possible local optima, and good grid resolution is needed in the area of the shock, the position of which changes during the optimization run. Also, because one can expect a sharp nose body to be optimal, this supersonic setup shows what large deformations are possible when using the shape derivative and nodal movement.

The initial NACA0012 airfoil and the optimized ogive shape are shown in Fig. 1. The mesh consists of 400 surface mesh nodes, of which 398 are design parameters due to the fixing of the reference length. The computed initial drag value is $C_D = 9.430 \times 10^{-2}$ for the supersonic NACA0012 airfoil and the optimized ogive shape has a drag value of $C_D = 4.721 \times 10^{-2}$, a relative improvement by 49.9%. Due to the nodal movement and the volume constraint instead of a constant thickness constraint, we observe the automatic formation of the sharp nose without using prior knowledge about what optimal shape to expect. In fact, the resulting optimal shape of a sharp nosed body with an attached shock wave matches the analytic predictions from Haack (1941) very well. The presented shapes are found after 80 iterations using the one-shot method with inexact gradients based on 3 flow solver and 3 adjoint solver iterations. The smoothing parameter ϵ in (4.12) was 0.5. Before starting the optimization, the flow solution was initialized by 278 flow solver iterations and the adjoint solution was initialized by 404 iterations.

In this supersonic flow, the flow state is actually continuous at the airfoil surface due to the detached bow shock of the blunt nose NACA0012 body. Hence, the

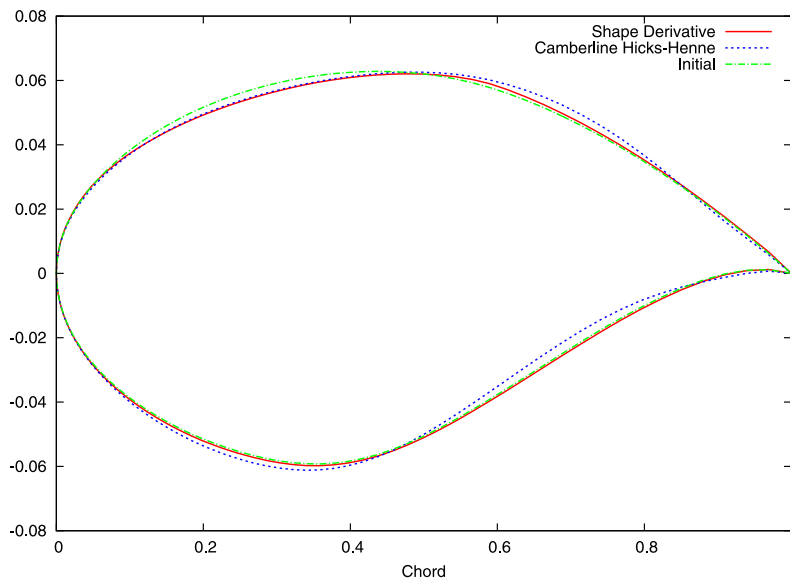


Fig. 2 Initial and optimized RAE2822 airfoils

required derivatives for the correct evaluation of the shape derivative are easily computable. For transonic flow there usually is a weak local shock wave directly on the airfoil surface, which requires the correct evaluation of the surface quantities needed for the shape derivative across jumps. We found this to not produce any kind of problems when using shape derivatives for the transonic case considered next. However, there is also some very minor smearing of the shock wave due to numerical stabilization terms in the flow solver.

6.3 Transonic lifting case

While the supersonic case is used to present the large deformation possibilities of this approach, here we focus on the transonic RAE2822 airfoil, where rather minuscule changes are usually needed to achieve optimal performance. The reference Mach number is Mach 0.73 and the airfoil is inclined by 2° angle of attack. An initial drag of $C_D = 6.55 \times 10^{-3}$ and an initial lift of $C_L = 7.82 \times 10^{-1}$ are computed. Also, the initial volume is $\text{Vol} = 7.79 \times 10^{-2}$. The airfoil surface consists of 128 nodes and the whole mesh is a structured C-type mesh with 6144 hexahedra. Achieving a flow residual of 10^{-7} on this mesh takes 9.5 seconds. However, the adjoints for drag and lift converge slower. We use this setup to compare the shape one-shot method with a classic SQP method based on (2.2). The initial and optimized airfoils are shown in Fig. 2.

The shape one-shot method uses 126 design parameters and a perturbation direction of $V = (0, 1)^T$, which ensures an airfoil reference length of 1 is maintained during optimization by fixing the leading and trailing edge node. To prevent a degeneration of the shape, the initial volume is used as a constraint in addition to maintaining lift. The computation of the gradient is based on a discretization of (5.9),

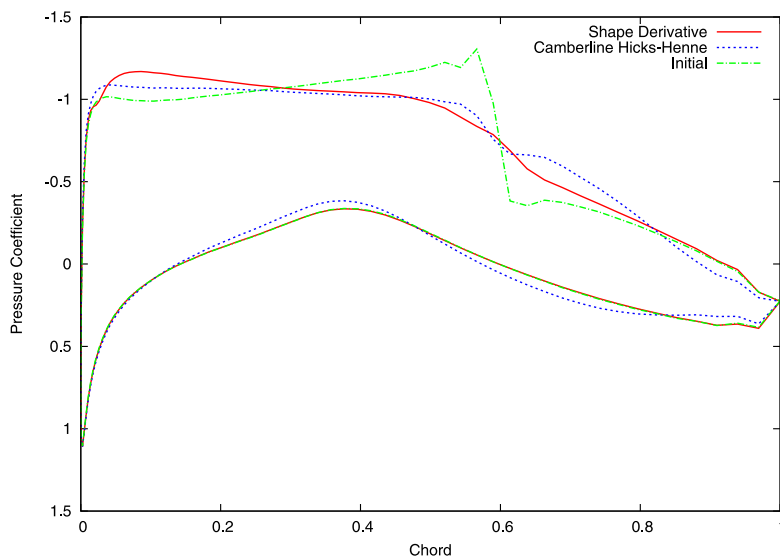


Fig. 3 Surface pressure coefficient C_P of the initial and optimized RAE2822 airfoils

where the variation of the normal $dn[V]$ is computed discretely. To filter the integral of (5.9) into discrete gradient components, a discrete perturbation field V^h is chosen such that for node i the perturbation $V_i^h = (0, 1)^T$ and zero for each other node with linear interpolation. The flow solution was initialized by 223 primal iterations and 380 iterations for the drag adjoint as well as 479 iterations for the lift adjoint. During optimization, we use 40 iterations each in the primal and dual flow solver per optimization iteration. Thereby, the flow solver residual is kept at $\approx 10^{-4}$ during the one-shot optimization. The reduced Hessian is approximated according to (4.12) with a smoothing parameter of $\epsilon = 4.0$. The optimum is found after 10 one-shot iterations or approximately 35 seconds. Afterwards, the flow solution is driven to full convergence with a residual of 9.904×10^{-8} . The computed optimal drag value is $C_D = 3.089 \times 10^{-3}$ with a lift coefficient of $C_L = 7.820 \times 10^{-2}$. Thus, drag is reduced by 52.82% while lift is maintained precisely.

For the classical reference solution based on (2.2) we use a standard SQP method with a BFGS-approximation of the Hessian. The airfoil shape is parametrized by eight Hicks–Henne bump functions applied to the camber line of the airfoil as shown in (2.1). Thus, the volume constraint is discarded because using such a parametrization, the thickness distribution of the airfoil is already fixed. The required partial derivatives with respect to the Hicks–Henne parameters $\frac{\partial c}{\partial q}$ are computed by one sided finite differences. Thus, for each optimization iteration and design parameter, the evaluation of the flow solution sensitivity $\frac{\partial c}{\partial q}$ requires making a perturbed volume mesh. In return, a one-shot approach is no longer the most efficient in terms of CPU time to solution, which we want to use as measure of performance. We therefore switch to a classical two loop approach, where each optimization step uses fairly well converged flow residuals. Before starting the optimization, the flow solution is initialized with 389 flow solver iterations. The adjoint for drag is initialized with 644

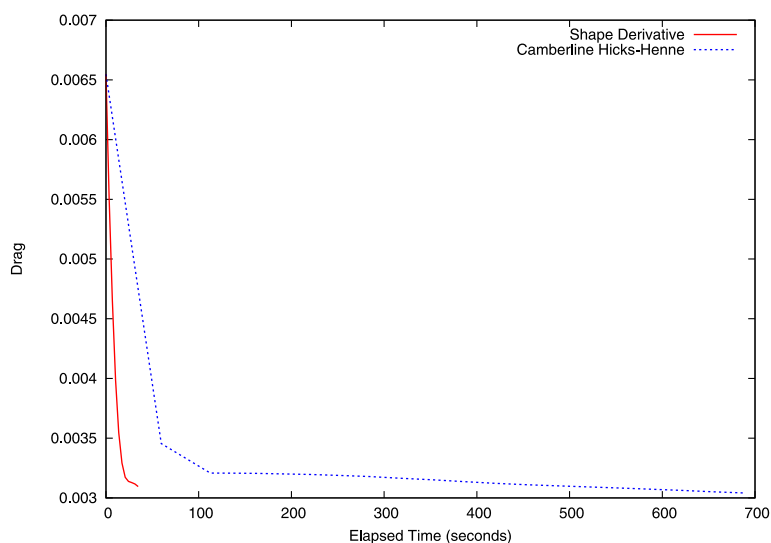


Fig. 4 Invested CPU time versus drag decrease

iterations and the adjoint for lift is initialized with 804 iterations. During each optimization step, the flow solver is set to iterate until the flow residual is again below 10^{-7} . In this setting, the optimum is found after 20 SQP iterations. The reported drag value is 3.042×10^{-3} and the reported lift value is 7.762×10^{-1} , an improvement by 53.59% with 0.9% loss of lift. Here, the optimization requires 688 seconds or roughly 10.5 minutes, making the shape one-shot approach approximately 95% faster in comparison. From these 688 seconds, about 45 seconds are needed to make the deformed meshes and compute the mesh sensitivities.

Comparing the resulting airfoil shapes in Fig. 2, one can see that the higher resolution of the 126 surface node positions versus eight Hicks–Henne parameters tends to produce optimal shapes that are closer to the initial design. Often, this can be advantageous as smaller changes can more easily be incorporated in existing or almost finalized designs. For the RAE2822 case considered here, this results in a flattening of the first half of the airfoil’s upper surface, which in return leads to a more rear-loading airfoil. Both shape derivative and the Hicks–Henne parametrization produce shock-free shapes as shown in the surface pressure coefficient plot in Fig. 3. Looking at the time comparison in Fig. 4 of the invested CPU time on a single core of an Intel Core 2 Duo E6600, one can see that both approaches start fairly steep, but the Hicks–Henne SQP approach requires significant more time to arrive at a comparable solution. Even when accepting the intermediate solution after 120 seconds, the shape derivative approach would still be roughly four times faster. It is also worth noting that this gap will widen with an increase in Hicks–Henne parameters due to the time and memory intensive computation of the partial derivatives in (2.2), which makes the shape derivative approach very promising for large scale three dimensional applications.

7 Conclusions

The aim of this paper was to prepare the one-shot optimization method for large 3D application. The key ingredient is the use of shape derivatives for the objective functions, which can be computed without so called “mesh sensitivities”. Previously, the computation of these sensitivities consumed the theoretical speedup of the one-shot approach when the number of design parameters is increased, which made the application to industrial size 3D problems impractical. The idea of one-shot is to couple the optimization with the flow solver. Thus, the one-shot approach tends to trade few long optimization steps for many smaller steps and the increased cost to compute the mesh sensitivities for a gradient evaluation becomes prohibitive.

We apply the shape derivatives and Hessian approximation to supersonic and transonic drag reduction problems using the compressible Euler equations, which is a first step before including viscosity. While the presented method should be applicable analogously to compressible viscous laminar flows, some turbulence models such as the popular Spalart–Allmaras model cause difficulties for turbulent flows as the continuous adjoint of these models is problematic to derive. Using the shape derivatives and the Hessian approximation, we are able to speed up the optimization significantly. As the shape derivative is indeed independent of the number of design parameters, the speedup will increase even further when the problem size is increased.

References

- Arian E, Ta’asan S (1996) Analysis of the Hessian for aerodynamic optimization: Inviscid flow. Technical Report 96-28, Institute for Computer Applications in Science and Engineering (ICASE)
- Arian E, Vatsa VN (1998) A preconditioning method for shape optimization governed by the Euler equations. Technical Report 98-14, Institute for Computer Applications in Science and Engineering (ICASE)
- Delfour MC, Zolésio J-P (2001) Shapes and geometries: analysis, differential calculus, and optimization. In: *Advances in design and control*. SIAM, Philadelphia
- Eppler K, Schmidt S, Schulz V, Ilic C (2009) Preconditioning the pressure tracking in fluid dynamics by shape Hessian information. *J Optim Theory Appl* 141(3):513–531
- Gauger NR (2003) Das Adjungiertenverfahren in der aerodynamischen Formoptimierung. Forschungsbericht DLR-FB–2003-05, Deutsches Zentrum für Luft- und Raumfahrt eV
- Gherman I (2008) Approximate partially reduced SQP approaches for aerodynamic shape optimization problems. PhD thesis, University of Trier, Trier, Germany
- Giles MB, Pierce NA (1997) Adjoint equations in CFD: duality, boundary conditions and solution behaviour. *AIAA*, 97-1850
- Haack W (1941) Geschößformen kleinsten Wellenwiderstandes. *Ber Lilienthal-Gesellschaft* 136(1):14–28
- Hazra SB, Schulz V (2004) Simultaneous pseudo-timestepping for PDE-model based optimization problems. *BIT Numer Math* 44(3):457–472
- Hazra SB, Schulz V (2005) How to profit from adjoints in one-shot pseudotime-stepping optimization. In: *Evolutionary and deterministic methods for design, optimization and control with applications to industrial problems*, EUROGEN
- Hazra SB, Schulz V (2006) Simultaneous pseudo-timestepping for aerodynamic shape optimization problems with state constraints. *SIAM J Sci Comput* 28(3):1078–1099
- Hicks RM, Henne PA (1978) Wing design by numerical optimization. *J Aircr* 15(7):407–412
- Jameson A (1988) Aerodynamic design via control theory. *J Sci Comput* 3(3):233–260
- Jameson A (1990) Automatic design of transonic airfoils to reduce the shock induced pressure drag. In: *Proceedings of the 31st Israel annual conference on aviation and aeronautics*, Tel Aviv, pp 5–17

- Jameson A (1994) Optimum aerodynamic design via boundary control. In: AGARD-VKI lecture series, optimum design methods in aerodynamics. von Karman Institute for Fluid Dynamics, Rhode Saint Genese
- Jameson A, Schmidt W, Turkel E (1981) Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. In: AIAA 14th fluid and plasma dynamic conference, 23–25 June 1981, pp 1981–1259
- Kroll N, Schwamborn D, Becker K, Rieger H, Thiele F (eds) (2009) MEGADESIGN and MegaOpt—German initiatives for aerodynamic simulation and optimization in aircraft design. Springer, Berlin
- Nemec M, Zingg DW (2002) A Newton–Krylov algorithm for aerodynamic design using the Navier–Stokes equations. *AIAA J* 40(6):1146–1154
- Nemec M, Zingg DW, Pulliam TH (2003) Multipoint and multi-objective aerodynamic shape optimization. *AIAA J* 42(6):1057–1065
- Schmidt S (2010) Efficient large scale aerodynamic design based on shape calculus. PhD thesis, University of Trier, Germany
- Schmidt S, Schulz V (2009) Impulse response approximations of discrete shape Hessians with application in CFD. *SIAM J Control Optim* 48(4):2562–2580
- Schmidt S, Schulz V (2010) Shape derivatives for general objective functions and the incompressible Navier–Stokes equations. *Control Cybern* 39(3):677–713
- Sokolowski J, Zolésio J-P (1992) Introduction to shape optimization: shape sensitivity analysis. Springer, Berlin
- Ta’asan S (1991) One-shot methods for optimal control of distributed parameter systems I: Finite dimensional control. *ICASE*, 91-2
- Ta’asan S, Kuruvila G, Salas MD (1991) Aerodynamic design and optimization in one shot. *AIAA*, 92-0025
- Widhalm M, Ronzheimer A, Hepperle M (2007) Comparison between gradient-free and adjoint based aerodynamic optimization of a flying wing transport aircraft in the preliminary design. In: AIAA 25th applied aerodynamics conference, 25–28 June 2007. American Institute of Aeronautics and Astronautics, New York