AIAA JOURNAL
Vol. 51, No. 11, November 2013

Three-Dimensional Large-Scale Aerodynamic Shape Optimization Based on Shape Calculus

Stephan Schmidt*
Imperial College, London, England SW7 2AZ, United Kingdom
Caslav Ilic†
DLR, German Aerospace Center, 38108 Braunschweig, Germany
Volker Schulz‡
University of Trier, 54296 Trier, Germany
and
Nicolas R. Gauger§
RWTH Aachen University, 52062 Aachen, Germany

DOI: 10.2514/1.J052245

Large-scale three-dimensional aerodynamic shape optimization based on the compressible Euler equations is considered. Shape calculus is used to derive an exact surface formulation of the gradients, enabling the computation of shape gradient information for each surface mesh node without having to calculate further mesh sensitivities. Special attention is paid to the applicability to large-scale three dimensional problems like the optimization of an Onera M6 wing or a complete blended-wing–body aircraft. The actual optimization is conducted in a one-shot fashion, in which the tangential Laplace operator is used as a Hessian approximation, thereby also preserving the regularity of the shape.

Nomenclature

a	=	rotation of the coordinate system
A_i	=	Euler flux Jacobian matrices
B	=	reduced Hessian
c	=	vector constraints, e.g., flow solver residual
C_D	=	drag coefficient
C_L	=	lift coefficient
C_P	=	pressure coefficient
$D_x z$	=	Jacobian of quantity z with respect to x
$\tilde{D}_x z$	=	reduced Jacobian of quantity z with respect to x
$\mathrm{d}A$	=	general integration measure for volume integrals
$\mathrm{d}S$	=	general integration measure for surface integrals
$\mathrm{d}z[V]$	=	material derivative of quantity z in direction V
E	=	total energy
f	=	objective function, standard optimization problem
g	=	shape gradient
H	=	enthalpy
$H_{z_1 z_2}$	=	second partial derivative of the objective with respect to z_1 and z_2
I	=	identity matrix
J	=	objective function, shape optimization problem
$\mathcal{L}$	=	Lagrangian
ℓ	=	scalar constraints, e.g., lift or structural constraints
n	=	outer normal
p	=	pressure
q	=	design vector
t	=	degree of deformation applied by T_t, baseline design corresponds to t equals 0
T_t	=	bijective family of mappings applying the shape deformation
U	=	vector of conserved variables
u	=	velocity vector
U_H	=	conserved variables with enthalpy as last component
V	=	smooth vector field prescribing the deformation direction
x_1	=	coordinate axis, chord direction
x_2	=	coordinate axis, span direction
x_3	=	coordinate axis, thickness direction
y	=	state vector
$z'[V]$	=	shape derivative of quantity z in direction V
α	=	angle of attack
Γ	=	unknown boundary to be optimized
γ	=	adiabatic exponent
Γ_0	=	Euler slip wall, i.e., the aircraft surface
Δ_Γ	=	Laplace–Beltrami operator
$\delta_{i,k}$	=	Kronecker symbol
ϵ	=	smoothing parameter
κ	=	additive mean curvature
λ	=	adjoint variable
μ	=	adjoint variable for vector constraints
ν	=	adjoint variable for scalar constraints
ρ	=	density
τ	=	one-shot dampening factor
Ω	=	domain occupied by the fluid

Presented as Paper 2011-3718 at the 41st AIAA Fluid Dynamics Conference and Exhibit, Honolulu, HI, 27–30 June 2011; received 30 July 2012; revision received 9 December 2012; accepted for publication 5 March 2013; published online 10 September 2013.

*Lecturer, Department of Aeronautics, South Kensington Campus; s.schmidt@imperial.ac.uk.

†Research Scientist, Institute of Aerodynamics and Flow Technology, Lilienthalplatz 7; caslav.ilic@dlr.de.

‡Professor, Department of Mathematics, Universitätsring 15; Volker.Schulz@uni-trier.de.

§Professor, Computational Mathematics Group, Schinkelstraße 2; gauger@mathcces.rwth-aachen.de. AIAA Senior Member.

I. Introduction

ADJOINT-BASED aerodynamic shape optimization, especially for industry-sized problems, has in the past almost always followed a parametric approach, meaning that parts of the aircraft like the wing cross sections are deformed by adding smooth ansatz functions, such as the popular Hicks–Henne functions [1], to the geometry. Other approaches frequently encountered for CAD-free fully three-dimensional (3-D) parameterizations are, for example, perturbing the control points of spline surfaces or free-form deformations. All of these approaches have in common that the actual optimization problem is considered postparameterization, meaning

the gradient is computed according to the standard Lagrange formula:

$$\frac{\mathrm{d}J}{\mathrm{d}q} = \frac{\partial J}{\partial q} - \lambda^T \frac{\partial c}{\partial q} \tag{1}$$

where λ is the corresponding adjoint variable solving

$$\left[\frac{\partial c}{\partial y}\right]^T \lambda = \frac{\partial J}{\partial y} \tag{2}$$

The adjoint flow solver is therefore independent of the shape optimization nature of the problem as only the derivatives with respect to the flow states are needed. However, in order to construct the gradient out of the primal and adjoint states, the parameterization of the shape must be considered. Especially the term $\frac{\partial c}{\partial q}$ requires knowledge of the sensitivity of the flow solver residual with respect to mesh nodes positions effected by the parameterization q. Although this approach is known to be applicable and well working, one is often forced into finite differencing for these terms [2,3]. This often makes very fine parameterizations, such as using the mesh node positions itself, rather impractical if not prohibitive as the time to compute the adjoint flow solution is indeed independent of the number of design parameters, but the gradient computation actually is not. Although it is possible to counter this problem by introducing another adjoint for the mesh deformation, using, e.g., algorithmic differentiation on the mesh deformation tools in reverse mode [4] or a continuous approach [5], special care must be taken not to run into memory limitations by considering the entire design chain at once. These problems arise in part due to the fact that in order to maintain mesh quality, usually the whole volume mesh, or at least significant parts of it, need to be moved and adapted, even for small changes such as moving a single surface node. Thereby, any locality that might otherwise be present due to the local finite element or finite volume stencils is destroyed. Furthermore, most flow solvers, although they have an adjoint capability, cannot efficiently compute the derivative of the flow residual with respect to the input mesh. Finally, the mapping of any other set of design parameters, e.g., CAD parameters, to the actual mesh nodes is frequently hidden within closed source software environments, and no derivative is available either.

The alternative is to treat the problem in a nonparametric fashion, while still staying within the framework of the continuous adjoint approach. This has the additional advantage that some of the standard infrastructure necessary for adjoint-based optimization can be reused easily. In the past, nonparametric approaches have been used to derive optimal shapes for certain flow situations on a theoretical level. For example, in [6,7], a rugby ball–like shape is shown to be optimal for creeping Stokes flows, whereas in [8], optimality of the so-called Sears–Haack body for inviscid compressible flow is shown. Nonparametric approaches can also be found in [9,10], but they are hardly used for any actual computations. The idea considered in the present work is to use shape calculus to differentiate the aerodynamic forces directly with respect to the input geometry, thereby arriving at a form of Eq. (1), which is specific for shape optimization problems and does not need explicit knowledge of the problematic partial derivatives. Shape calculus or shape sensitivity analysis describes the mathematical topic when the shape of a domain is the unknown. The methodology can be used to arrive at exact surface formulations of the gradients for shape optimization problems, which is often termed the Hadamard form [11,12]:

$$\mathrm{d}J(\Gamma) = \int_\Gamma \langle V, n\rangle g \,\mathrm{d}S \tag{3}$$

Once g is known, a steepest descent algorithm can easily be conducted according to

$$\Gamma^{k+1} := \{x + \tau g(x) n(x) : x \in \Gamma^k\}$$

where τ is the step length of the algorithm. Therefore, using the surface mesh node positions is a natural choice, and furthermore, the deformation of the volume mesh is completely removed from the derivative chain. Although the volume mesh nodes must of course still be somehow adapted to the new surface geometry, the derivative of the mesh deformation and the variation of the flow residual with respect to the design are not required for an exact gradient evaluation because they are included in g on an analytic level. Thus, an advantage of the approach discussed here is that knowledge of the volume mesh deformation scheme is not required, one can use any volume mesh generator while conducting the actual optimization. This is due to the analytic nature of the formulation, and so a one-to-one correspondence of the surface tessellation and the volume mesh is not necessary throughout the optimization process, hence the name nonparametric optimization. Using the surface mesh node positions as design parameters encompasses any other form of parameterization, as any parameterization can be interpreted as a restriction of the design space, i.e., a projection of the mesh node movement into a coarser space. Although there might be frequent occasions in which engineering restrictions necessitate a restriction of the design space, being able to efficiently optimize all surface node positions is a strong indication that any other parameterization is also going to work well, unless the design space becomes too restrictive. There are previous works in aerodynamic shape optimization that use all surface mesh node positions [13], but usually, the considerable overhead in computing the gradient based on Eq. (1) has made this approach very inefficient [14].

To summarize the preceding, we present the applicability of a novel adjoint-based shape optimization algorithm to very-large-scale 3-D problems. The algorithm is very fast because it exploits the mathematical problem structure and operates on surfaces alone. Furthermore, it does not require any partial derivatives of the mesh deformation. A possible downside could be discrepancies between the continuous and discrete models of the flow solver and applicability to shapes of a low regularity class.

Here, the shape gradient g is used in a one-shot optimization strategy similar to [15,16]. Being a reduced Sequential Quadratic Programming method, one-shot depends on a proper approximation of the reduced Hessian, for which the surface or tangential Laplace operator is used. Pseudodifferential operator symbol calculus conducted in [17,18] suggests that using a Hessian approximation based on an anisotropic operator with anisotropies in the chord and span directions would be best, but we found isotropic diffusion to be working very well also. Sometimes called gradient smoothing or the Sobolev gradient method, similar techniques have been used in [19,20] as a means to preserve the regularity of the aircraft shape.

Although the applicability to two-dimensional (2-D) airfoil optimizations using the compressible Euler equations has been previously considered in [21,22], the aim of this paper is to study the applicability to large-scale 3-D problems. To this end, the optimization of both the Onera M6 wing as well as the optimization of a complete blended-wing–body aircraft is shown. Special emphasis also lies on the correct computation of the respective surface quantities needed for evaluating the shape derivative on triangulated unstructured surface meshes. We found the combination of the one-shot approach paired with surface gradients based on shape calculus extremely efficient, such that a blended-wing–body aircraft could be optimized conveniently using up to 460,517 individual mesh node positions as design unknowns. Further considerations for the incompressible Navier–Stokes equations can be found in [21,23,24]. Potential flow inverse design is considered in [25].

II. Shape Calculus

A. Problem Introduction: Aerodynamic Forces

A very brief review of shape calculus is given next. More details on shape sensitivity analysis in general can be found in [11,12]. The inviscid fluid forces acting on the aircraft surface Γ_0 are given by

$$J(U, \Omega) = \int_{\Gamma_0} \langle p \cdot a, n\rangle \,\mathrm{d}S \tag{4}$$

where $U := (\rho, \rho u, \rho E)$ are the conserved Euler state variables with ρ being the density, $u = (u_1, u_2, u_3)^T$ is the velocity vector, and E is the

total energy of the fluid. Furthermore, the pressure p is linked to the conserved state variables by the perfect gas law:

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

with $\gamma \approx 1.4$ being the adiabatic exponent of air. The normal to the aircraft surface is denoted by n, and a is the rotation of the coordinate system, meaning for an angle of attack α, choosing

$$a_D := (\cos\,\alpha, 0, \sin\,\alpha)^T$$

leads to J being the aerodynamic inviscid pressure drag force. Similarly, choosing a as

$$a_L := (-\sin\,\alpha, 0, \cos\,\alpha)^T$$

will result in J being the lift force. In the following, it is thus sufficient to consider surface functionals only. Also, the lift and drag forces do not need to be treated separately.

B. Shape Calculus for Surface Functionals

A finite deformation of the aircraft surface is thought to be given by

$$\Gamma_0^t := T_t(\Gamma_0) = \{T_t(x) : x \in \Gamma_0\} \tag{5}$$

where T_t is a family of bijective mappings usually given by either the perturbation of identity

$$T_t(x) = x + tV(x) \tag{6}$$

or the speed method

$$\frac{\partial x}{\partial t} = V(t, x), \qquad x(0) = x_0 \in \Gamma_0 \tag{7}$$

Thus, the actual perturbation direction is given by the vector field V, which is supposed to be Lipschitz continuous. Sometimes, V is also called the "design velocity." For first-order calculus, both the perturbation of identity and the speed method are known to be equivalent [11,12]. Assuming enough regularity such that the chain rule holds, the preliminary shape derivative of Eq. (4) is given by

$$\mathrm{d}J(U)[V] = \left[\frac{\mathrm{d}}{\mathrm{d}t}\bigg|_{t=0} \int_{\Gamma_0^t} \langle p \cdot a, n\rangle \,\mathrm{d}S_t\right] + \int_{\Gamma_0} \langle p \cdot a, \mathrm{d}n[V]\rangle \,\mathrm{d}S + \int_{\Gamma_0} \langle p'[V] \cdot a, n\rangle \,\mathrm{d}S \tag{8}$$

Using standard shape differentiation techniques and tangential calculus [11,12,21,22], one arrives at

$$\frac{\mathrm{d}}{\mathrm{d}t}\bigg|_{t=0} \int_{\Gamma_0^t} \langle p \cdot a, n\rangle \,\mathrm{d}S_t = \int_{\Gamma_0} \langle V, n\rangle \left[\left\langle \frac{\partial p}{\partial n} \cdot a, n\right\rangle + \kappa\langle p \cdot a, n\rangle\right] \mathrm{d}S \tag{9}$$

for the first term in Eq. (8). Using the same techniques, one can also arrive at

$$\int_{\Gamma_0} \langle p \cdot a, \mathrm{d}n[V]\rangle \,\mathrm{d}S = \int_{\Gamma_0} \langle V, n\rangle [\mathrm{div}_\Gamma(p \cdot a) - \kappa\langle p \cdot a, n\rangle] \,\mathrm{d}S \tag{10}$$

where div_Γ is the surface or tangential divergence operator.

C. Shape Calculus for the Local Shape Derivative of the State Equation

Adjoint calculus must now be used to remove the local shape derivative of the pressure $p'[V]$ in Eq. (8), which will be conducted analogously to [26,27]. Let the local shape derivatives of the conserved variables be given by

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

They satisfy the linearized Euler equations given by

$$\frac{\partial}{\partial x_1}(A_1 U'[V]) + \frac{\partial}{\partial x_2}(A_2 U'[V]) + \frac{\partial}{\partial x_3}(A_3 U'[V]) = 0 \tag{11}$$

inside the flow domain. Letting λ solve the adjoint compressible Euler equations,

$$-A_1^T \frac{\partial}{\partial x_1}\lambda - A_2^T \frac{\partial}{\partial x_2}\lambda - A_3^T \frac{\partial}{\partial x_3}\lambda = 0 \text{ in } \Omega \tag{12}$$

integration by parts in Eq. (11) shows that

$$0 = \int_{\partial\Omega} \sum_{k=1}^{3} \lambda n_k A_k U'[V] \,\mathrm{d}S$$

As discussed in [26,27] and given proper far-field adjoint boundary conditions, the relation

$$\begin{aligned} 0 &= \int_{\Gamma_0} \lambda \sum_{k=1}^{3} n_k A_k U'[V] \,\mathrm{d}S \\ &= \int_{\Gamma_0} \lambda U_H \langle u'[V], n\rangle + (\lambda_2, \lambda_3, \lambda_4) n p'[V] \,\mathrm{d}S \end{aligned} \tag{13}$$

holds, where U_H is given by

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

Because of the fluid velocity satisfying the Euler slip boundary condition

$$\langle u, n\rangle = 0$$

on the aircraft surface, the local shape derivatives of the velocities are then correspondingly given by

$$\langle u'[V], n\rangle = -\langle V, n\rangle \left\langle \frac{\partial u}{\partial n}, n\right\rangle - \langle u, \mathrm{d}n'[V]\rangle$$

For more details, see [21,22]. Inserting this into Eq. (13), one arrives at

$$\begin{aligned} 0 = \int_{\Gamma_0} &-\langle V, n\rangle \lambda U_H \left\langle \frac{\partial u}{\partial n}, n\right\rangle - \lambda U_H \langle u, \mathrm{d}n[V]\rangle \\ &+ (\lambda_2, \lambda_3, \lambda_4) n p'[V] \,\mathrm{d}S \end{aligned}$$

Adding the preceding to the preliminary gradient Eq. (8) and using Eq. (9) but not yet Eq. (10), one can see that

$$\begin{aligned} \mathrm{d}J(U)[V] = &\int_{\Gamma_0} \langle V, n\rangle \left[\frac{\partial p}{\partial n}\langle a, n\rangle + \kappa p\langle a, n\rangle - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n\right\rangle\right] \\ &+ \langle pa - \lambda U_H u, \mathrm{d}n[V]\rangle + p'[V][\langle a, n\rangle + (\lambda_2, \lambda_3, \lambda_4) n] \,\mathrm{d}S \end{aligned}$$

If the adjoint boundary condition

$$(\lambda_2, \lambda_3, \lambda_4) n + \langle a, n\rangle = 0 \tag{14}$$

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

$$\begin{aligned} \mathrm{d}J(U)[V] = &\int_{\Gamma_0} \langle V, n\rangle \left[\frac{\partial p}{\partial n}\langle a, n\rangle + \kappa p\langle a, n\rangle - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n\right\rangle\right] \\ &+ \langle pa - \lambda U_H u, \mathrm{d}n[V]\rangle \,\mathrm{d}S \end{aligned} \tag{15}$$

This especially means that existing adjoint flow solvers need not be modified to be useable for the computation of nonparametric shape

derivatives, because both the adjoint field equation and the boundary conditions stay the same as in the classical approach. Using now also Eq. (10), one arrives at

$$\mathrm{d}J(U)[V] = \int_{\Gamma_0} \langle V, n\rangle \left[\frac{\partial p}{\partial n}\langle a, n\rangle - \lambda U_H \left\langle \frac{\partial u}{\partial n}, n\right\rangle + \mathrm{div}_\Gamma(pa - \lambda U_H u)\right] \mathrm{d}S \tag{16}$$

which is the final form of the gradient.

Comparing Eq. (15) with Eq. (16), one can see that the final Hadamard form of the gradient requires the evaluation of the tangential divergence operator on the unstructured surface mesh of the aircraft, but this is equivalent to the computation of the mean curvature κ and the variation of the normal $\mathrm{d}n[V]$. This transformation is based on the tangential Stokes theorem, which states for a real valued differentiable function g and a vector valued function v on Γ_0

$$\int_{\Gamma_0} g\,\mathrm{div}_\Gamma v + \langle \nabla_\Gamma g, v\rangle\,\mathrm{d}S = \int_{\Gamma_0} \kappa g \langle v, n\rangle\,\mathrm{d}S$$

For more information on the required tangential calculus, please see [11,21]. Because of more literature being available concerning mesh curvature of unstructured triangulated surfaces and the normal variation $\mathrm{d}n[V]$ being quite easily computable on a discrete level, the latter approach was chosen.

The following methodology is used for evaluating Eq. (15) within the discretized framework of flow and adjoint solver. We assume that V is given by a linear hat function on the curved surface mesh, that is,

$$V_k(x_i) = n(x_k)\delta_{i,k}$$

with linear interpolation in between. Thus, index k here denotes the kth component of the discretized gradient vector and at the same time also the node number of the kth surface node. Index i corresponds to all nodes adjacent to node k. At a point x that is on the surface but not part of the mesh, we assume a linear interpolation in between. When computing the kth gradient component, which is determined by V_k, the discretized integral (15) has therefore a compact support on the surface, and the integration reduces to the patch of surface triangles of which node k is the center. Finally, this integration is conducted using a standard quadrature rule for triangles. The process is visualized in Fig. 1. As discussed in [21], the variation $\mathrm{d}n_T[V_k](x_k)$ of the face normal n_T is given by

$$\mathrm{d}n_T[V_k](x_k) = \frac{n_k \times (x_i - x_{i+1})}{|T|}$$

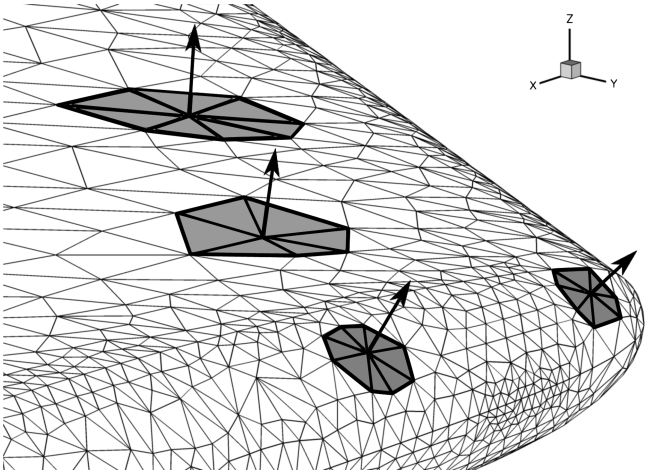

Fig. 1 Several surface integration patches for evaluating four components of the discrete gradient vector.

where T is the surface triangle patch centered around node x_k with vertices x_i. The mean curvature κ of the surface mesh is computed as described in [28] and the necessary normal derivatives are computed automatically within the flow solver software environment. Although not the main focus of this work, we also found the preceding methodology well applicable in less smooth situations. For example, in [22], we see the automatic formation of sharp leading edges in supersonic flow. Similarly, the automatic formation of sharp-nosed bodies of shiplike shapes in incompressible viscous fluids can be seen in [23,24].

III. One-Shot Optimization and Hessian Approximation

A. Overview of the One-Shot Method

To motivate the one-shot method, a standard minimization problem is considered:

$$\min_{(y,q)} f(y, q)$$

subject to

$$c(y, q) = 0 \quad \ell(y, q) = 0$$

where $c(y, q) = 0$ refers to the flow solution residual being zero and $\ell(y, q) = 0$ means that additional constraints such as lift, volume, or bending stiffness are kept. The Lagrangian of the preceding is given by

$$\mathcal{L}(y, q, \mu, \nu) = f(y, q) - \langle \mu, c\rangle - \langle \nu, \ell\rangle$$

and using Newton's method to solve the necessary optimality conditions of the preceding problem, the system

$$\begin{bmatrix} H_{yy} & H_{yq} & (D_y c)^T & (D_y \ell)^T \\ H_{qy} & H_{qq} & (D_q c)^T & (D_q \ell)^T \\ D_y c & D_q c & 0 & 0 \\ D_y \ell & D_q \ell & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta y \\ \Delta q \\ \Delta \mu \\ \Delta \nu \end{pmatrix} = \begin{pmatrix} -\nabla_y \mathcal{L} \\ -\nabla_q \mathcal{L} \\ -c \\ -\ell \end{pmatrix}$$

needs to be solved with the actual design update given by

$$(y_{k+1}, q_{k+1}, \mu_{k+1}, \nu_{k+1})^T = (y_k, q_k, \mu_k, \nu_k)^T + (\Delta y, \Delta q, \Delta \mu, \Delta \nu)^T$$

Here, $\mathcal{L}$ denotes the Lagrangian. Assuming there exists an approximation of the Hessian of the form

$$\begin{bmatrix} H_{yy} & H_{yq} \\ H_{qy} & H_{qq} \end{bmatrix} \approx \begin{bmatrix} 0 & 0 \\ 0 & B \end{bmatrix}$$

and further assuming $(D_y c)^{-1}$ exists, a block Gauss elimination and replacing $\Delta\nu$ with $\nu_{k+1} = \nu_k + \Delta\nu$ results in the system

$$\begin{bmatrix} B & \tilde{D}_\ell \\ (\tilde{D}_\ell)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta q \\ \nu_{k+1} \end{pmatrix} = \begin{pmatrix} -\tilde{D}_f \\ -\ell + \lambda_\ell c \end{pmatrix} \tag{17}$$

where $\lambda_\ell = (D_y \ell)(D_y c)^{-1}$, which is given by the adjoint flow solver. More details can be found in [15,21]. In the context of a standard minimization problem, the reduced gradient $\tilde{D}_f$ of the objective function is given by

$$\tilde{D}_f = \nabla_q f - (D_q c)^T (D_y c)^{-T} \nabla_y f$$

with an analogous definition of the reduced gradient $\tilde{D}_\ell$ of the scalar constraints. Here, however, the respective shape derivatives will be used directly for $\tilde{D}$, resulting in a shape one-shot method.

An additional aspect of the one-shot method, not directly visible in Eq. (17), is the fact that the state and adjoint flow variables are usually computed by an iterative flow solver. This usually results in any optimization procedures essentially becoming a two-loop approach: an outer optimization loop with several inner loops for the primal and

respective adjoint iterative flow solvers. For the problems considered here, this two-loop approach is broken up by only performing a limited number of solver and adjoint iterations to compute the derivatives needed in Eq. (17). Thus, optimality of the design and feasibility of the flow state is computed simultaneously, thereby greatly reducing the wall-clock runtime. Please note that a discussion of the convergence properties of the one-shot method as well as a comparison of the discrete and continuous adjoint approaches is outside the scope of this paper. However, the method presented here can be both employed within the continuous and discrete adjoint approaches and also without a one-shot optimization scheme. More details on the convergence properties of one shot can be found in [29–31].

Assuming an exact Hessian approximation B, an update based on Newton's method as in Eq. (17) requires a step length restriction or line search only for convergence globalization with respect to the initial guess q_0. However, this does no longer hold when using inexact quantities within the one-shot context. Therefore, a dampening factor τ is introduced in Eq. (17) that acts on the direction of optimality only but not feasibility. This leads to an update of the design being computed according to

$$q_{k+1} = q_k - \tau B^{-1}\tilde{D}_f - B^{-1}\tilde{D}_\ell \nu_{k+1} \tag{18}$$

where τ was chosen manually for each of the applications. More details can be found in [15].

B. Summary of the Algorithm

The following flow chart is a summary of the optimization algorithm. In the following, y is used to denote all physical unknowns of the flow solution, whereas c is the residual of the flow solver.

1) Initialization: $k \leftarrow 0$, start with initial guess y^0, Γ_0^0, λ^0, λ_ℓ^0. Set V as family of admissible deformations. Initialize dampening τ and Hessian approximation ϵ.
2) **repeat**
3) Perform n_a adjoint iterations to approximate λ^{k+1} the solution to Eqs. (12) and (14).
4) Compute the shape derivative according to Eq. (15)
$\tilde{D}_f = \langle V, n\rangle\left(\frac{\partial p}{\partial n}\langle a, n\rangle + \kappa p\langle a, n\rangle - \lambda U_H\left\langle\frac{\partial u}{\partial n}, n\right\rangle\right) + \langle pa - \lambda U_H u, \mathrm{d}n[V]\rangle$
5) Perform n_ℓ adjoint iterations for all additional (scalar) constraints ℓ. Obtain λ_ℓ^{k+1}.
6) Use λ_ℓ^{k+1} to compute the Jacobian $\tilde{D}_\ell$ of all additional (scalar) constraints.
7) Discretize $B = (\epsilon\Delta_\Gamma + I)$ as an approximation of the reduced Hessian.
8) Solve Eq. (17) for ν_{k+1}
$\tilde{D}_\ell^T B^{-1}\tilde{D}_\ell \nu_{k+1} = \ell - \lambda_\ell^{k+1} c - \tilde{D}_\ell^T B^{-1}\tilde{D}_f$
9) Solve Eq. (18) and update boundary
$\Gamma_0^{k+1} = \{x - \tau B^{-1}\tilde{D}_f - B^{-1}\tilde{D}_\ell \nu_{k+1} : x \in \Gamma_0^k\}$
10) Adapt volume mesh or remesh by any preferable means.
11) Update state y^k by n_f steps in the forward solver. Obtain y^{k+1}.
12) **until** "convergence"

Note that, in our special case here, we set the family of admissible deformations V to contain a movement of all nodes except for those that define the planform. The stopping criteria of the algorithm is adapted from one of the stopping criteria of the flow solver and can be seen as a form of a Cauchy criterion: the optimization is considered to have terminated when there is no longer any progress within the objective.

C. Hessian Approximation

Crucial for the performance of one shot is having a good approximation of the reduced Hessian operator B. A natural choice would be the shape Hessian of the problem. However, shape Hessians are fairly complex objects even for moderate problems. Although they have been successfully used in solving shape optimization problems numerically [32,33], it is often much more convenient to use a suitable approximation, especially in the cases in which the Hessian is not positive definite away from the optimum.

An analysis of the operator symbol of the Hessian for the Euler shape optimization problem conducted in [17,18] suggests it is best to approximate the Hessian by an anisotropic operator in the chord and span directions, in which chordwise, the Hessian closely resembles a diffusive operator like the Laplacian. Because of this fact and the previous successes of gradient smoothing techniques [19,20], we approximate the Hessian according to

$$B \approx -\epsilon\Delta_\Gamma + I \tag{19}$$

where Δ_Γ is the tangential Laplace operator on the curved 2-D aircraft surface mesh and I is the identity. Further studies of shape Hessians for a variety of other fluid dynamics problems can be found in [23,25]. During computation, the tangential Laplacian is computed as described in [34]. The effects of this preconditioning on the drag gradient of the Onera M6 wing can be seen in Fig. 2. The following test cases were chosen to demonstrate the applicability of the methodology discussed in the preceding sections to large-scale complex 3-D problems. Because a direct comparison to more classical approaches in such a high-dimensional search space is quite cumbersome, we would like to refer to [22] for more details on how this methodology compares to Broyden–Fletcher–Goldfarb–Shanno (BFGS) update formulas and classical Hicks–Henne ansatz functions. Likewise, the approach discussed here is also very well applicable in robust or multipoint design problems, and more details can be found in [35].

IV. Onera M6

A. Overview of the Test Case

The first problem under consideration is the shape optimization of an Onera M6 wing at a cruise condition of Mach 0.83 and a 3.01 deg angle of attack, which are being held constant during the optimization. In this configuration, the computed lift coefficient is $C_L = 0.2762$, which is to be kept constant. The initial drag coefficient is computed as $C_D = 0.01058$. The primal and adjoint

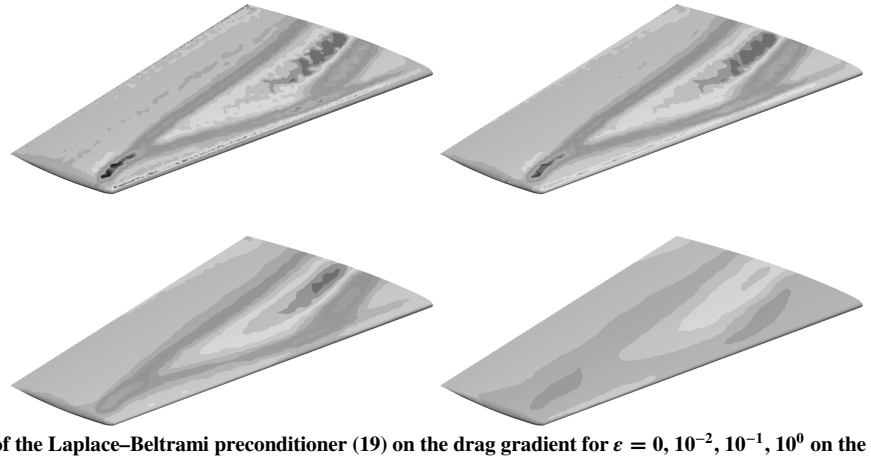

Fig. 2 Effects of the Laplace–Beltrami preconditioner (19) on the drag gradient for $\epsilon = 0, 10^{-2}, 10^{-1}, 10^{0}$ on the Onera M6 wing.

flow states are computed using vertex-centered finite volumes with the DLR flow solver TAU. More details and validations of TAU can be found in [36,37]. The mesh consists of 18,285 surface mesh nodes. Because the volume mesh is perturbed using the algebraic mesh deformation tool that is part of the TAU suite, the planform had to be fixed as otherwise the deformation tool was very often unable to make volume meshes of satisfying quality. Because of this reason, the surface mesh nodes were also moved in the x_3 direction only, meaning the gradient was evaluated according to Eq. (15) for a movement of each node in direction of the normal in the current optimization iteration at that node. However, before any actual mesh deformation is applied, there is a projection of this gradient with respect to a movement in the x_3 direction only. We therefore expect a better performance of Eq. (15) when more sophisticated mesh deformation tools are available. Fixing the planform reduces the effective number of unknowns for the shape to 16,792. Because most inviscid meshes feature a numerically sharp trailing edge with potentially infinite curvature, any possible problems stemming from this point are therefore also circumvented. Counting the field nodes also, there are 541,980 unknowns for the Euler fluid state.

B. Gradient Validation

Before the actual nonparametric wing optimization is conducted, we first confirm the accuracy of the surface gradient by the means of a comparison with the classical gradient Eq. (1) evaluated using TAU adjoint and finite differences for the respective partial derivatives. Because all 16,792 nonplanform relevant nodes need to be perturbed in the x_3 axis direction, an equal number of deformed meshes and flow solver residual calculations must be made. The whole process required 2 days and 7 h on a 3 GHz Pentium IV processor. To ease the computational burden, one-sided finite differences were used with a perturbation of $\epsilon = 10^{-3}$, multiplied with an approximation of inscribed circle diameter of the associated dual volume face.

Cuts of the drag gradient for different spanwise positions are shown in Fig. 3, which uses the same convention as the C_P plots do, meaning the gradients for the upper and lower sides are plotted within the same figure. We found the results obtained using the surface shape derivative to be surprisingly accurate, especially given the fact that the derivation and benefits of the surface shape derivatives heavily rely on the exploitation of the nature of shape optimization problems in the continuous setting. Although these analytic considerations are completely independent of the discrete mesh deformation chain, there are next to no discrepancies between the surface gradient formulation and the finite differences for all areas of the wing, except maybe for the singularity of the leading-edge stagnation point in this inviscid flow.

C. Results and Optimized Wing

Although the lift constraint itself already implies a certain internal volume, the total volume is to be kept constant as a very rough

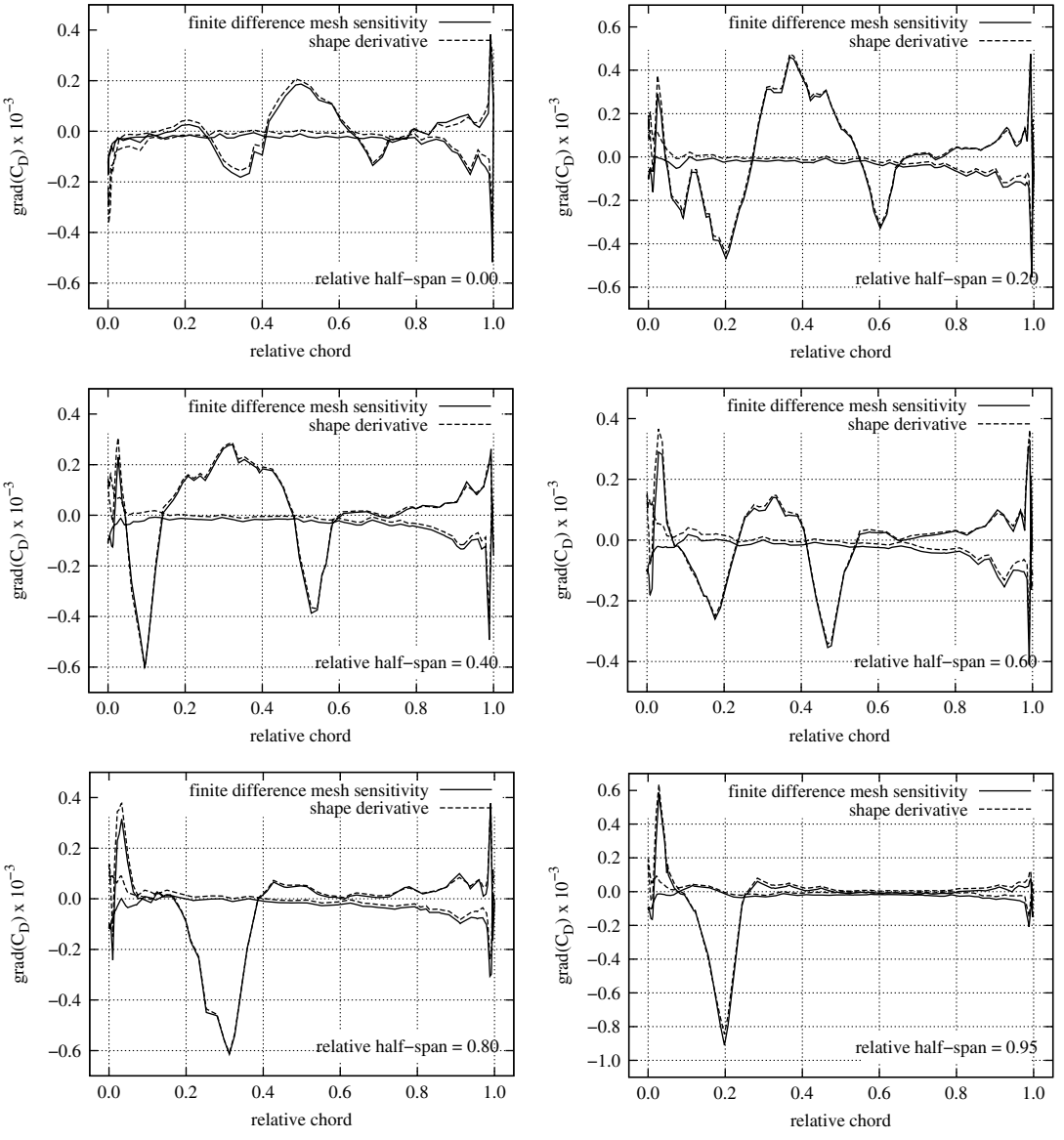

Fig. 3 Comparison between the surface shape derivative and the derivative obtained via finite differences for different cuts through the Onera M6 wing.

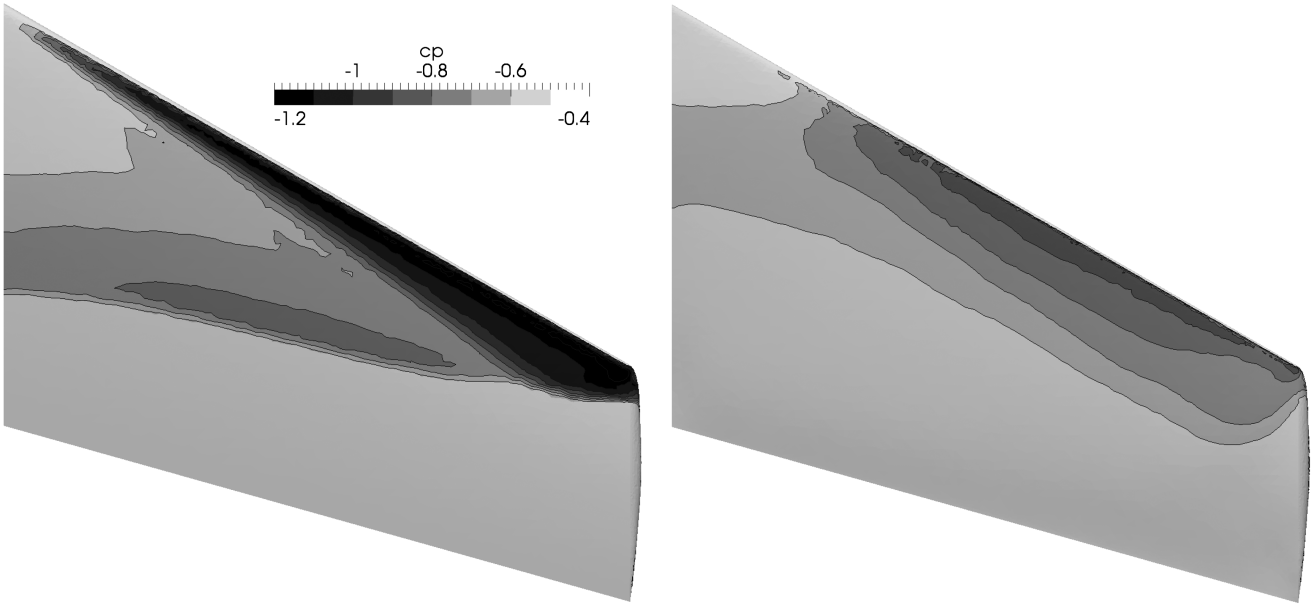

Fig. 4 Initial and optimized Onera M6 wings.

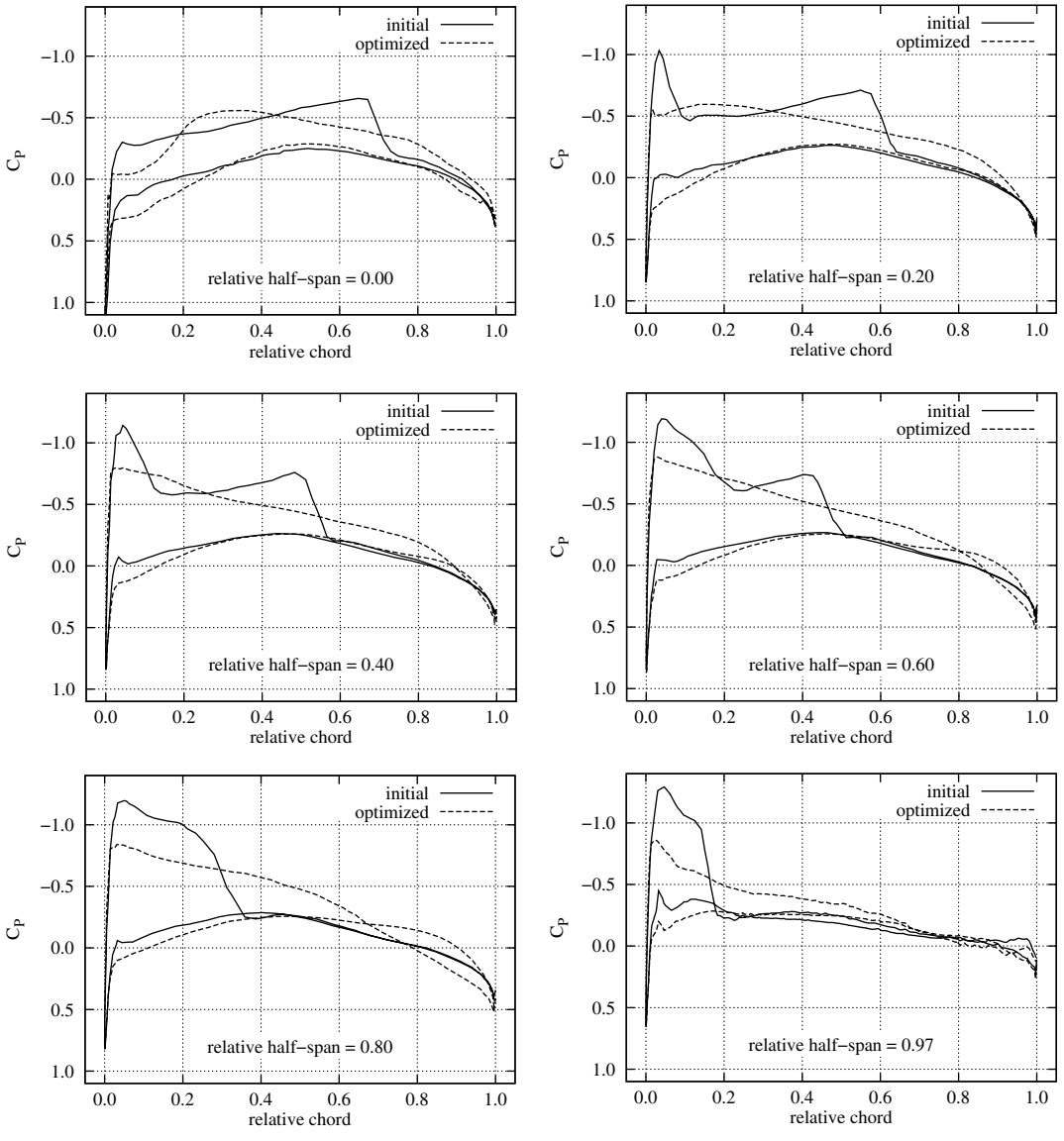

Fig. 5 Pressure distributions across airfoil cuts for the initial and optimized Onera M6 wings.

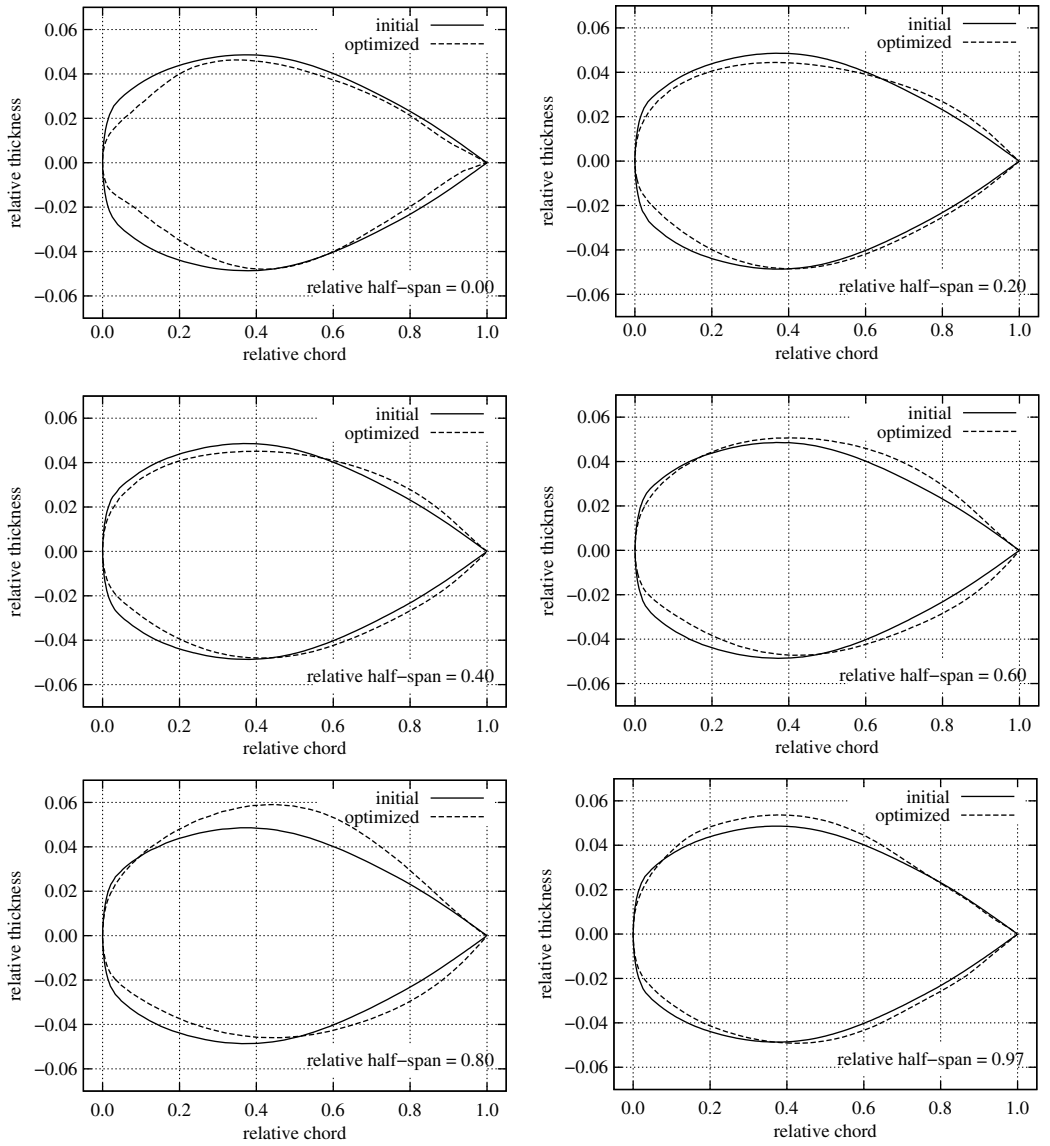

Fig. 6 Airfoil cuts for the initial and optimized Onera M6 wings.

measure of preserving structural strength in case the shock is strong enough that the optimizer attempts thinning the wing excessively. The volume constraint is treated explicitly by the optimizer, and its gradient has been derived exactly on the discrete level. For the Laplace–Beltrami Hessian approximation, a value of $\epsilon = 1$ is used.

The planform pressure distribution on the upper surfaces of the initial and optimized wings is shown in Fig. 4. Respective CP distributions are shown in Fig. 5 and the initial and optimized airfoil cuts are shown in Fig. 6. The optimized Onera M6 wing has a drag coefficient of $C_D = 0.007567$, which corresponds to an improvement of 28.47%. Also, the optimized wing has a lift coefficient of $C_L = 0.2723$, which means the lift was preserved up to 1.41%. The optimal solutions were found after 70 one-shot iterations with 10 inner iterations in each adjoint flow solver and 20 iterations in the primal. The actual design update was conducted as given by Eq. (18), which also contains the typical one-shot step dampening. Looking at the pressure distributions in Fig. 5, one can see that the optimized wing is indeed shock free over the complete span. Because the cross-sectional thickness of each airfoil was not fixed but only the total volume of the wing, one can see that the optimized wing has become thinner at the root and thicker towards the tip, which is less than prefect from a structural point of view. However, there was no mathematical constraint to account for structural requirements, making this acceptable for the purpose of the present work.

The convergence of the drag objective during the optimization can be seen in Fig. 7.

V. Very Efficient Large Aircraft Blended-Wing–Body

A. Standard Mesh

The second test is the optimization of the "Very Efficient Large Aircraft (VELA)," a blended-wing–body concept [38]. A traditional optimization of this aircraft can also be found in [39]. The tetrahedral mesh consists of 115,673 surface mesh points, of which 113,956 remain as design unknowns after fixing the planform. The mesh has a total number of 1,061,433 nodes in the field. The flow and both the

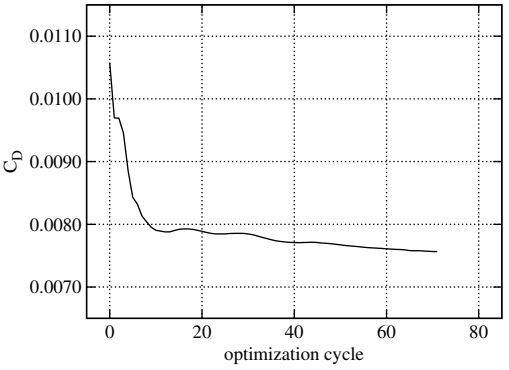

Fig. 7 Convergence of the objective function for the M6 test case.

Fig. 8 Initial and optimized VELA aircraft.

adjoints for lift and drag are again computed using the DLR flow solver TAU. As in the Onera M6 case, the gradient is again computed according to Eq. (15). After one update of the aircraft surface, the new volume mesh is created by deforming the mesh from the previous iteration using the algebraic mesh deformation tool that is part of the TAU software. Because of this tool having difficulties in deforming the volume mesh for perturbations of the type $V_k(x_i) = n(x_k)\delta_{i,k}$ with linear interpolation in between points, the gradient is again

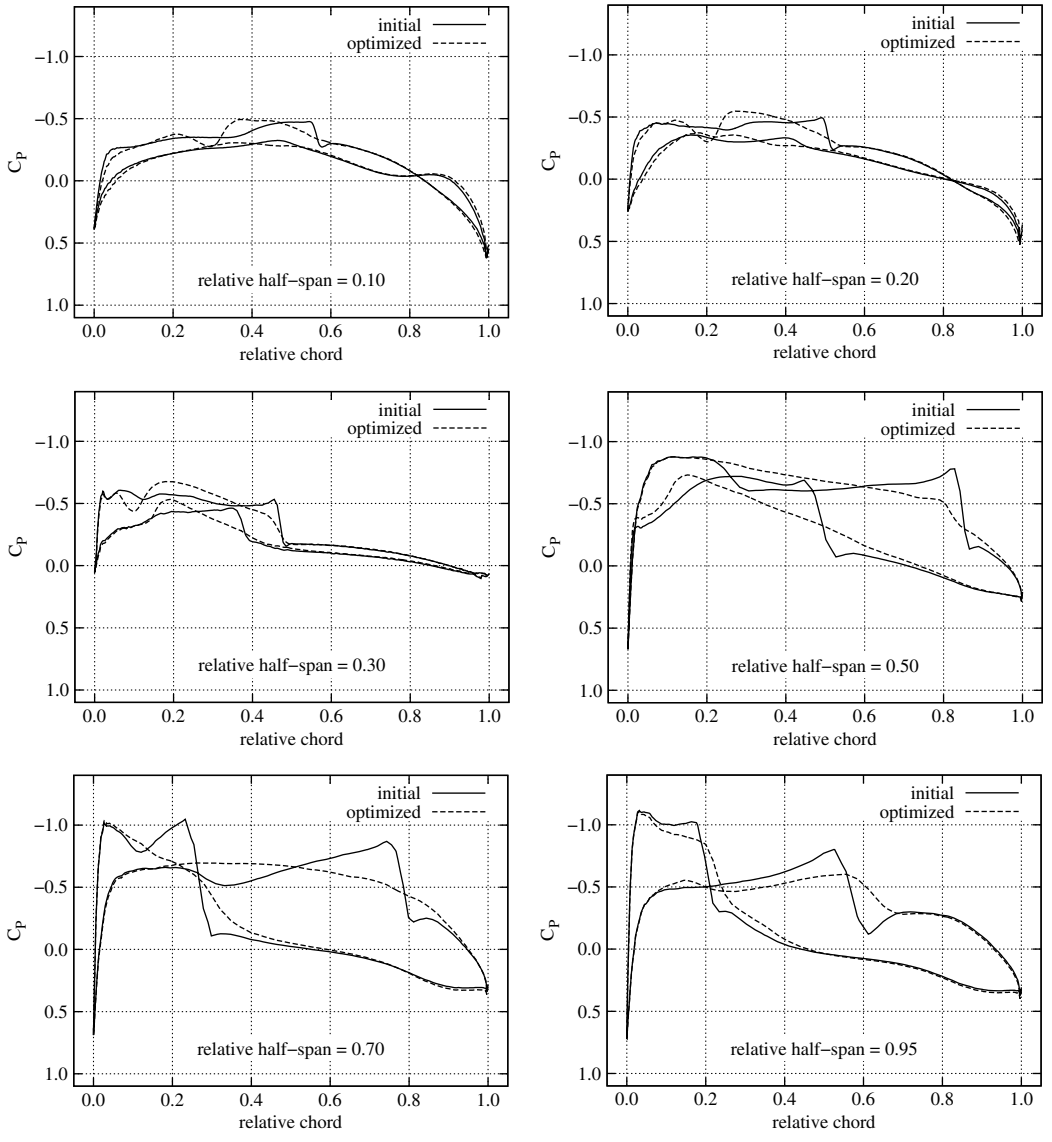

Fig. 9 Pressure distributions across airfoil cuts for the initial and optimized VELA aircraft.

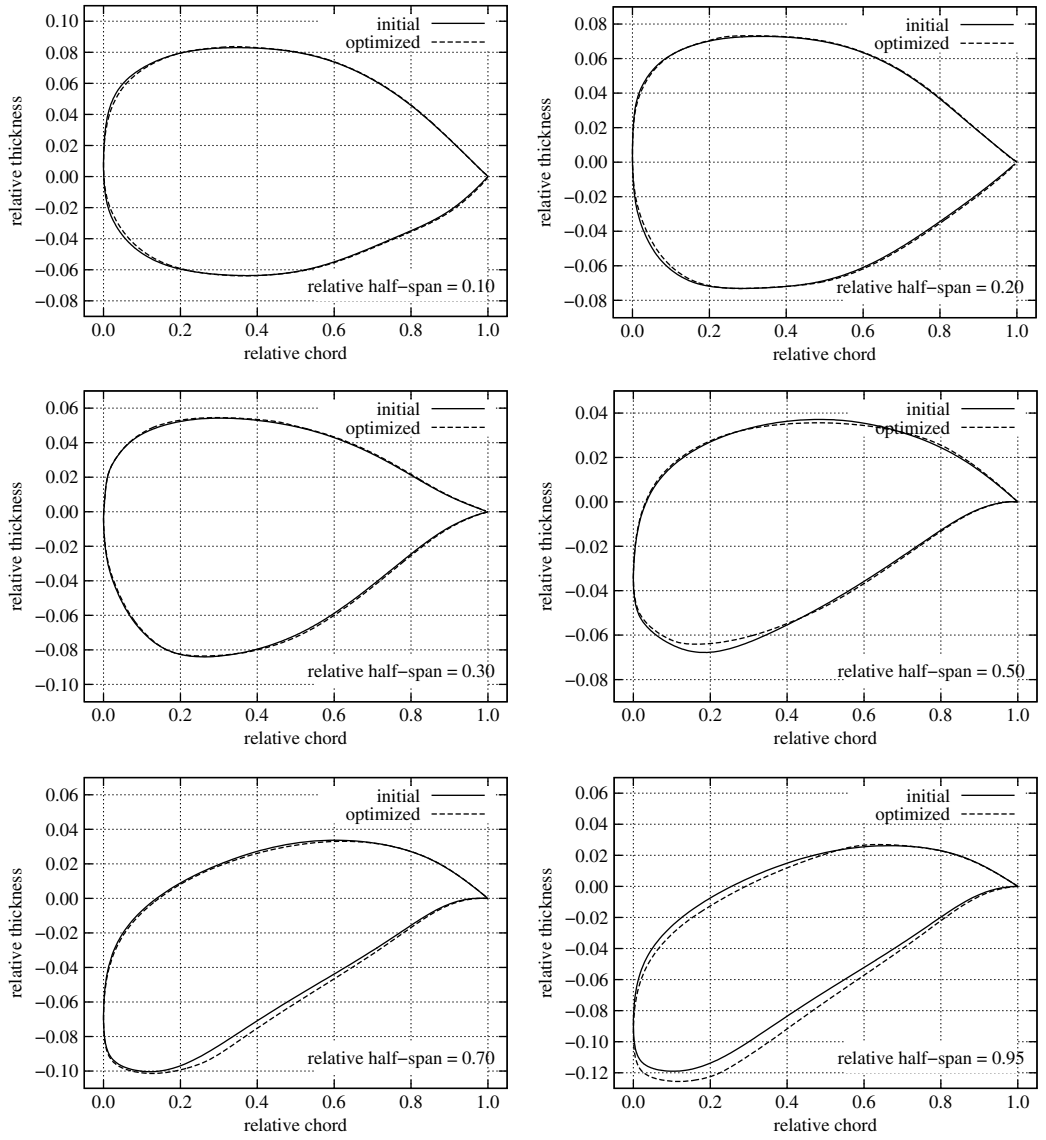

Fig. 10 Airfoil cuts for the initial and optimized VELA aircraft.

projected for a movement in the x_3 direction only. For the smoothing preconditioner, $\epsilon = 10$ is used.

The initial and optimized aircraft are shown in Fig. 8. Some C_P plots are shown in Fig. 9, whereas the respective airfoil cross sections are presented in Fig. 10. At a 1.8 deg angle of attack and a cruise condition of Mach 0.85, the initial configuration has a drag coefficient of $C_D = 0.004770$ and a lift coefficient of $C_L = 0.1787$ (these values are unusually low due to the high reference area of the wing–body planform). Both the angle of attack and the Mach number are being held constant during the optimization. The optimal solution is found after 151 one-shot iteration steps with 20 inner iterations for each of the two adjoint solvers and 40 inner iterations for the primal flow solver. The optimized design has a drag coefficient of $C_D = 0.003342$ and a lift coefficient of $C_L = 0.1775$. In total, drag was reduced by 29.93%, whereas lift was almost precisely kept with a relative loss of only 0.67%. The total amount of time needed for each shape update is around 390 s including the evaluation of the shape derivative for all 113,956 design unknowns and one solution of the surface Laplace gradient smoothing operator. The precise timings are shown in Table 1. Note that the timings do not exactly add up to 390, as some servicing steps and the solve with the surface Laplacian are not accounted for. The flow and adjoint solvers were running on four cores of an AMD Phenom II 2.8 GHz PC, whereas the other steps were computed on one core only. A full convergence to a TAU residual of 10^{-9} of the primal solution without optimization requires 1552 iterations or roughly 66 min. Thus, given 151 optimization steps, one can see that the optimal shape for 113,956 design parameters is found in as little as 15 equivalent flow solutions.

B. Analysis of the Optimized VELA

Looking at Fig. 8 and the C_P plots in Fig. 9, one can see that the shock wave on the upper and lower sides of the wing could be removed for almost the whole span, whereas the pressure distribution of the fuselage is also somewhat improved. Observing the airfoil cuts in Fig. 10, one can see that during optimization the twist of the wing fuselage near the root has slightly decreased, whereas the twist of the wing near the tip as increased, even though there were no design

Table 1 Time spent during each VELA optimization step

Operation	Time in seconds
Volume mesh deformation	36
Dual mesh construction and partitioning	49
Curvature computation	4
Primal flow solver (40 iterations)	101
Adjoint flow solver (drag, 20 iterations)	57
Adjoint flow solver (lift, 20 iterations)	57
Shape derivative evaluation	26
Derivative of volume constraint	4

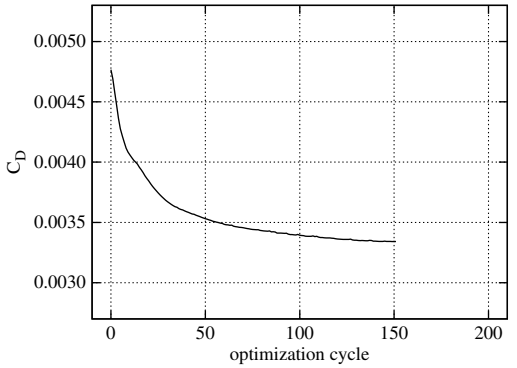

Fig. 11 Convergence of the VELA optimization.

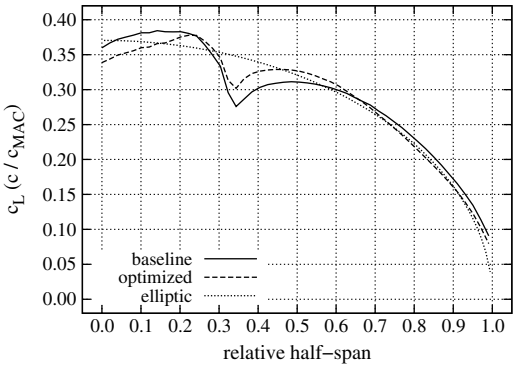

Fig. 12 Spanwise lift distribution of the initial and optimized VELA aircraft. Elliptic distribution plotted for reference.

parameters specifically controlling the twist. However, with such a fine parameterization available, the optimizer can achieve a shock-free or almost shock-free aircraft geometry that is very close to the original layout, which appears to be very beneficial for the actual design process, because, usually, larger deformations for improving aerodynamics are often problematic from a structural point of view. This can also be seen when comparing this optimal solution with the one from [39], in which the optimized wing has a substantially higher sweep and aspect ratio, whereas here, much less dramatic changes have resulted in an optimal shape with a comparable performance increase. Thus, there are potential benefits if the actual design process of the aircraft is already in a more advanced state and major changes can no longer be incorporated. The actual convergence of the objective during the optimization can be seen in Fig. 11.

Furthermore, the spanwise lift distribution is shown in Fig. 12. One can see that the optimized distribution gets closer to the elliptical,

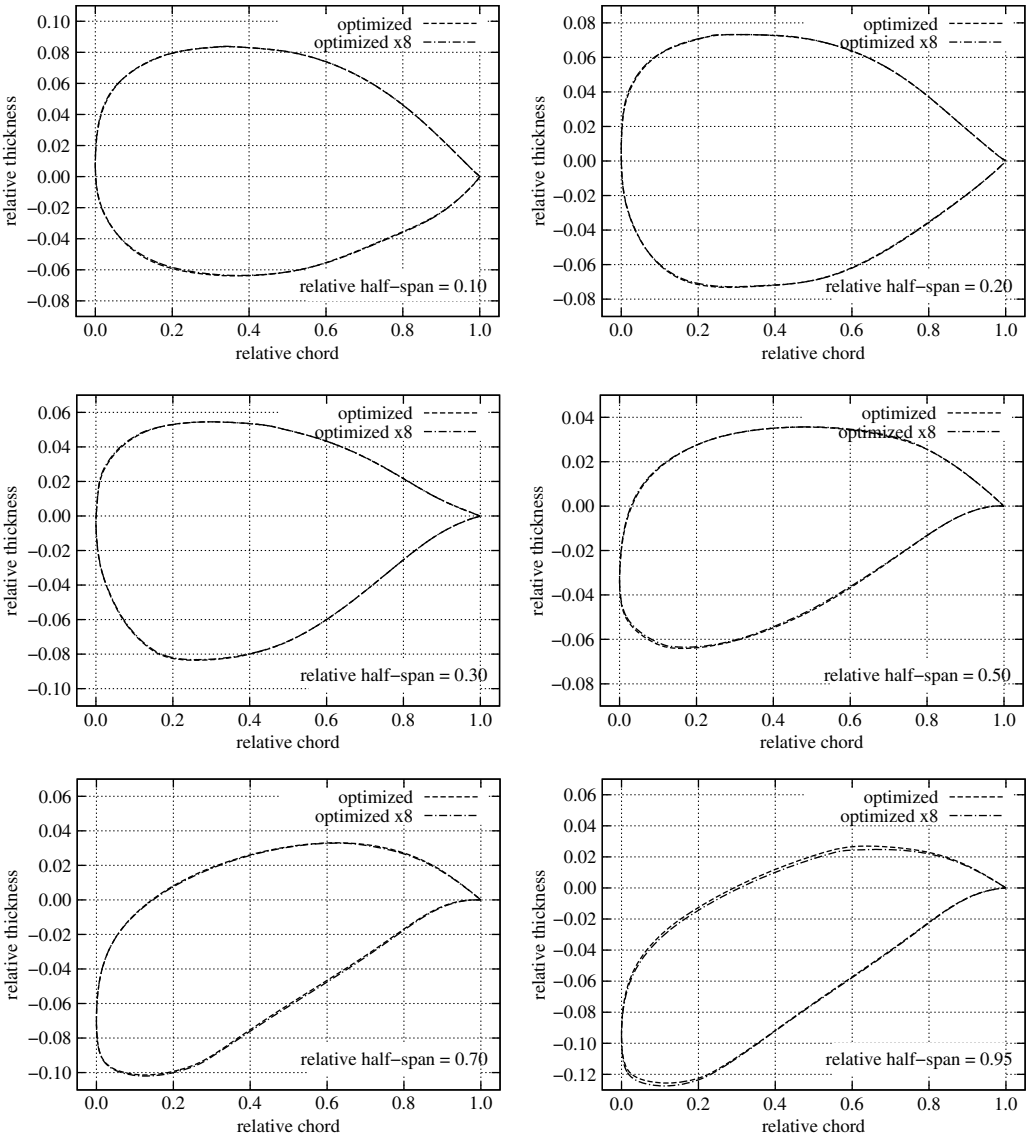

Fig. 13 Airfoil cuts for the optimized VELA aircraft based on original and eight-times-adapted meshes.

which means that part of the drag reduction is also due to reducing the induced drag. The graph y axis is the local lift coefficient multiplied by the local chord length divided by the mean aerodynamic chord. Because the VELA is a blended-wing–body aircraft, the mean aerodynamic chord is based on the outer wing trapezoidal, as if it were extended to the root.

C. Mesh Refinement

Next, we want to check how much the optimized shape depends on the mesh fineness, for the same initial geometry. To this end, we perform the same optimization on a mesh with eight times the number of points, derived by pressure-based adaptation from the original mesh. This adapted mesh contains 462,238 surface mesh points, giving 460,517 design unknowns, and 8,110,568 points in total. At the same angle of attack, the initial drag coefficient on this mesh is $C_D = 0.004378$, which is 8.22% less than on the original mesh; the optimized value is $C_D = 0.002862$ or 14.36% less than that of the optimized original mesh. In spite of these significant differences, the optimized shapes on the original and refined mesh are very close (Fig. 13), much closer than either is to the original shape.

The flow and adjoint solvers for optimization on the refined mesh were running on 4 AMD Opteron 2384 2.7 GHz cluster nodes, each with 8 cores, for 32 processes in total. Two hundred one-shot iteration steps were performed, with each taking about 2550 s. Twice the amount of iterations, 80 primal and 40 for each of the adjoint solvers, were used compared to the original mesh. The smoothing factor was increased to $\epsilon = 40$.

VI. Outlook: Viscous Fluids

The extension of the shape optimization technique presented here to also include viscosity is straightforward. and preliminary theoretical studies for the compressible laminar Navier–Stokes equations can be found in [21]. The actual application to viscous compressible fluids is part of current research. The situation becomes somewhat more delicate when turbulent flows are considered. Most of the standard turbulence models have elements for which a formal derivation of the continuous adjoint equation or the partial shape derivative is not straightforward. A good example would be the wall-distance functions of the Spalart–Allmaras turbulence model or some of the boundary conditions in the k-ϵ and the k-ω model. Although these difficulties can easily be circumvented by considering a frozen the eddy viscosity, there are also reports of successful uses of analytically adjointed turbulence models [40]. Given the fact that, for example, the partial derivative of the wall-distance functions or even the complete turbulent flow solver could also be treated efficiently on a discrete level using, e.g., algorithmic differentiation [41], the shape optimization method presented here appears to be also applicable to turbulent flows, although the derivation is probably not straightforward and might require some form of hybridization.

VII. Conclusions

Large-scale aerodynamic shape optimization for the compressible Euler equations in three dimensions is considered. By using the Hadamard form of the shape gradient, a sensitivity information for the aerodynamic forces can be computed extremely efficiently, such that each surface mesh node position can be used as a design parameter. Being an analytic exact surface expression, the partial derivatives of the mesh deformation tool and the mesh sensitivity Jacobians are not required. Using these shape gradients as the reduced gradients in a one-shot optimization strategy creates a shape one-shot method for which the Hessian is approximated using the surface or tangential Laplace operator. Feasibility of the method for large-scale aerodynamic problems is shown through the optimization of an Onera M6 wing with 16,792 unknowns of the shape and the optimization of the Very Efficient Large Aircraft blended-wing–body concept aircraft using 113,956 coarse mesh and 460,517 fine mesh surface node positions as design parameters.

Acknowledgments

The authors wish to thank Markus Widhalm at the DLR, German Aerospace Center, Braunschweig, for the discussions and the VELA test case.

References

[1] Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412. doi:10.2514/3.58379

[2] Nemec, M., and Zingg, D. W., "A Newton–Krylov Algorithm for Aerodynamic Design Using the Navier–Stokes Equations," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1146–1154. doi:10.2514/2.1764

[3] Nemec, M., Zingg, D. W., and Pulliam, T. H., "Multipoint and Multi-Objective Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 42, No. 6, 2003, pp. 1057–1065. doi:10.2514/1.10415

[4] Gauger, N., Walther, A., Moldenhauer, C., and Widhalm, M., "Automatic Differentiation of an Entire Design Chain for Aerodynamic Shape Optimization," *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, Vol. 96, 2007, pp. 454–461. doi:10.1007/978-3-540-74460-3_56

[5] Nielsen, E. J., and Park, M. A., "Using an Adjoint Approach to Eliminate Mesh Sensitivities in Computational Design," *AIAA Journal*, Vol. 44, No. 5, 2006, pp. 948–953. doi:10.2514/1.16052

[6] Pironneau, O., "On Optimum Profiles in Stokes Flow," *Journal of Fluid Mechanics*, Vol. 59, No. 1, 1972, pp. 117–128. doi:10.1017/S002211207300145X

[7] Pironneau, O., "On Optimum Design in Fluid Mechanics," *Journal of Fluid Mechanics*, Vol. 64, No. 1, 1974, pp. 97–110. doi:10.1017/S0022112074002023

[8] Haack, W., "Geschoßformen Kleinsten Wellenwiderstandes," *Bericht der Lilienthal-Gesellschaft*, Vol. 136, No. 1, 1941, pp. 14–28.

[9] Castro, C., Lozano, C., Palacios, F., and Zuazua, E., "Systematic Continuous Adjoint Approach to Viscous Aerodynamic Design on Unstructured Grids," *AIAA Journal*, Vol. 45, No. 9, 2007, pp. 2125–2139. doi:10.2514/1.24859

[10] Mohammadi, B., and Pironneau, O., "Numerical Mathematics and Scientific Computation," *Applied Shape Optimization for Fluids*, Clarendon, Oxford, England, U.K., 2001.

[11] Delfour, M. C., and Zolésio, J.-P., "Advances in Design and Control," *Shapes and Geometries: Analysis, Differential Calculus, and Optimization*, SIAM, Philadelphia, 2001.

[12] Sokolowski, J., and Zolésio, J.-P., *Introduction to Shape Optimization: Shape Sensitivity Analysis*, Springer–Verlag, Berlin, Heidelberg, 1992.

[13] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260. doi:10.1007/BF01061285

[14] Kim, S., "Design Optimization of High-Lift Configurations Using a Viscous Adjoint-Based Method," Ph.D. Dissertation, Department of Aeronautics & Astronautics, Stanford Univ., Stanford, CA, 2001.

[15] Gherman, I., "Approximate Partially Reduced SQP Approaches for Aerodynamic Shape Optimization Problems," Ph.D. Dissertation, Department IV: Mathematics, Univ. of Trier, Trier, Germany, 2008.

[16] Schulz, V., and Gherman, I., "One-Shot Methods for Aerodynamic Shape Optimization," *MEGADESIGN and MegaOpt — German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design*, edited by Kroll, N., Schwamborn, D., Becker, K., Rieger, H., and Thiele, F., Vol. 107, Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Springer–Verlag, Berlin Heidelberg, 2009, pp. 207–220.

[17] Arian, E., and Ta'asan, S., "Analysis of the Hessian for Aerodynamic Optimization: Inviscid Flow," Institute for Computer Applications in Science and Engineering TR-96-28, Hampton, VA, 1996.

[18] Arian, E., and Vatsa, V. N., "A Preconditioning Method for Shape Optimization Governed by the Euler Equations," Institute for Computer Applications in Science and Engineering TR-98-14, Hampton, VA, 1998.

[19] Jameson, A., Martinelli, L., and Pierce, N. A., "Optimum Aerodynamic Design Using the Navier–Stokes Equations," *Theoretical and Computational Fluid Dynamics*, Vol. 10, No. 1–4, 1998, pp. 213–237. doi:10.1007/s001620050060

[20] Kim, S., Hosseini, K., Leoviriyakit, K., and Jameson, A., "Enhancement of Adjoint Design Methods via Optimization of Adjoint Parameters," *43rd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2005-0448, 2005.

[21] Schmidt, S., "Efficient Large Scale Aerodynamic Design Based on Shape Calculus," Ph.D. Dissertation, Department IV: Mathematics, Univ. of Trier, Trier, Germany, 2010.

[22] Schmidt, S., Ilic, C., Gauger, N., and Schulz, V., "Airfoil Design for Compressible Inviscid Flow Based on Shape Calculus," *Optimization and Engineering*, Vol. 12, No. 3, 2011, pp. 349–369. doi:10.1007/s11081-011-9145-3

[23] Schmidt, S., and Schulz, V., "Impulse Response Approximations of Discrete Shape Hessians with Application in CFD," *SIAM Journal on Control and Optimization*, Vol. 48, No. 4, 2009, pp. 2562–2580. doi:10.1137/080719844

[24] Schmidt, S., and Schulz, V., "Shape Derivatives for General Objective Functions and the Incompressible Navier-Stokes Equations," *Control and Cybernetics*, Vol. 39, No. 3, 2010, pp. 677–713.

[25] Eppler, K., Schmidt, S., Schulz, V., and Ilic, C., "Preconditioning the Pressure Tracking in Fluid Dynamics by Shape Hessian Information," *Journal of Optimization Theory and Applications*, Vol. 141, No. 3, 2009, pp. 513–531. doi:10.1007/s10957-008-9507-y

[26] Gauger, N., "Das Adjungiertenverfahren in der Aerodynamischen Formoptimierung," DLR Forschungs-Bericht, Rept. 2003-05, Braunschweig, Germany, 2003.

[27] Giles, M. B., and Pierce, N. A., "Adjoint Equations in CFD: Duality, Boundary Conditions and Solution Behaviour," AIAA Paper 1997-1850, 1997.

[28] Rusinkiewicz, S., "Estimating Curvatures and Their Derivatives on Triangle Meshes," *Symposium on 3D Data Processing, Visualization, and Transmission*, IEEE Computer Society, Los Alamitos, CA, 2004, pp. 486–493.

[29] Özkaya, E., and Gauger, N., "Single-Step One-Shot Aerodynamic Shape Optimization," *International Series of Numerical Mathematics*, Vol. 158, 2009, pp. 191–204. doi:10.1007/978-3-7643-8923-9_11

[30] Özkaya, E., and Gauger, N., "Automatic Transition from Simulation to One-Shot Shape Optimization with Navier–Stokes Equations," *GAMM-Mitteilungen*, Vol. 33, No. 2, 2010, pp. 133–147. doi:10.1002/gamm.201010011

[31] Ito, K., Kunisch, K., Schulz, V., and Gherman, I., "Approximate Nullspace Iterations for KKT Systems," *SIAM Journal on Matrix Analysis and Applications*, Vol. 31, No. 4, 2010, pp. 1835–1847. doi:10.1137/080724952

[32] Eppler, K., and Harbrecht, H., "A Regularized Newton Method in Electrical Impedance Tomography Using Shape Hessian Information," *Control and Cybernetics*, Vol. 34, No. 1, 2005, pp. 203–225.

[33] Hintermüller, M., and Ring, W., "A Second Order Shape Optimization Approach for Image Segmentation," *SIAM Journal on Applied Mathematics*, Vol. 64, No. 2, 2004, pp. 442–467. doi:10.1137/S0036139902403901

[34] López Pérez, L. D., "Régularisation d'Images sur des Surfaces non Planes," Ph.D. Dissertation, Ecole Doctorale STIC, Université de Nice–Sophia Antipolis, 2006.

[35] Schillings, C., Schmidt, S., and Schulz, V., "Efficient Shape Optimization for Certain and Uncertain Aerodynamic Design," *Computers and Fluids Journal*, Vol. 46, No. 1, 2011, pp. 78–87. doi:10.1016/j.compfluid.2010.12.007

[36] Schwamborn, D., and Gerhold, T., "The DLR TAU-Code: Recent Application in Research and Industry," *ECCOMAS CFD 2006*, edited by Wessling, P., Oñate, E., and Périaux, J., Egmond aan Zee, The Netherlands, 5–8 Sept. 2006.

[37] Gerhold, T., "Overview of the Hybrid RANS TAU-Code," *MEGA-FLOW — Numerical Flow Simulation for Aircraft Design*, edited by Kroll, N., and Fassbender, J. K., Vol. 89, Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Springer–Verlag, Berlin Heidelberg, 2005, pp. 81–92.

[38] Strüber, H., and Hepperle, M., "Optimization of Flying Wing Transport Aircraft Configurations in the Vela Project," DLR, German Aerospace Center TR-DLR-IB-124-2005/905, Braunschweig, Germany, 2005.

[39] Brezillon, J., Dwight, R., and Widhalm, M., "Aerodynamic Optimization for Cruise and High-Lift Configurations," *MEGADESIGN and MegaOpt — German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design*, edited by Kroll, N., Schwamborn, D., Becker, K., Rieger, H., and Thiele, F., Vol. 107, Springer–Verlag, Berlin, Heidelberg, 2009, pp. 249–262.

[40] Zymaris, A. S., Papadimitriou, D. I., Giannakoglou, K. C., and Othmer, C., "Continuous Adjoint Approach to the Spalart–Allmaras Turbulence Model for Incompressible Flows," *Computers and Fluids Journal*, Vol. 38, No. 8, 2009, pp. 1528–1538. doi:10.1016/j.compfluid.2008.12.006

[41] Carnarius, A., Thiele, F., Özkaya, E., and Gauger, N., "Adjoint Approaches for Optimal Flow Control," AIAA Paper 2010-5088, 2010.

J. Martins
Associate Editor