Weierstraß–Institut für Angewandte Analysis und Stochastik

im Forschungsverbund Berlin e.V.

Preprint ISSN 0946 – 8633

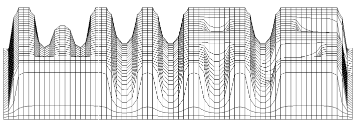
Practical shape optimization for turbine and compressor blades

Hans Georg Bock[1], Wolfgang Egartner[1], Wolfgang Kappis[2], Volker Schulz[3]

submitted: 5th December 2000

Preprint No. 621

Berlin 2000



2000 Mathematics Subject Classification. 90C55, 65H05, 65Y20, 76D55.

Key words and phrases. Turbomachinery design, shape optimization, partially reduced SQP methods, working range optimization.

[1]Interdisciplinary Center for Scientific Computing, University of Heidelberg
Im Neuenheimer Feld 368, D–69120 Heidelberg, Germany

[2]ABB, CH-8021 Zürich/Baden, Switzerland

[3]Weierstrass Institute for Applied Analysis and Stochastics
Mohrenstrasse 39, D–10117 Berlin, Germany
email: schulzv@wias-berlin.de

Abstract. The shape optimization of blades is a crucial step within the design cycle of a whole turbomachine. This paper is a report on a joint project between academia and industry leading to an efficient solution software for this problem to be used in the daily work of concerned engineers. The problem description and solution method, characterized as a partially reduced SQP method, as well as numerical results are presented.

1 Introduction

In this paper we report on a joint effort to alleviate the labor in designing good blade profiles for turbine and compressor blades between industry and academia within a research project[4] funded by the German ministry for research and education (BMBF).

The interest in this research is focused on the determination of blade cross-sections, which minimize the overall pressure loss. This problem is similar to the optimization of airfoil cross-sections, for which there exists a vast body of literature. Concerning the numerical flow modeling, we relied on well evaluated existing technology, which is frequently used in the design process of blades. The aim of the research is to develop a fast and reliable optimization method based on this flow model. In an earlier project stage we have used a multigrid solver for a potential flow model in [20, 19].

The requirement of a fast solution forbids the use of so-called "black box" approaches, where an outer optimization loop iterates over the design variables only and an inner simulation loop iterates over the state variables describing the flow. The alternative, a simultaneous approach, typically requires a close coupling of the optimization aspect of the overall algorithm with the flow computation—essentially by incorporating the optimization within the flow computation, which leads to high investments in terms of manpower for the implementation.

Here we pursue a simultaneous approach to the optimization problem but nevertheless maintain a high level of modularity within the implementation of the optimization algorithm. This is achieved by the use of a partially reduced SQP approach, which reduces computational complexity while still being able to cope with the geometry conditions necessary for practical implementation.

Shape optimization for airfoils has been investigated in many publications, see e.g. [12, 17]. In contrast, the optimization of blades in turbines has received much less attention (e.g. [2]). CFD-based aerodynamic design methods can be divided into two basic categories: inverse methods and numerical optimization methods. Inverse methods derive their name from the fact that they directly invert the goal of the flow analysis algorithm (typically the goal is reformulated by a boundary condition). On the one hand this leads to rather fast algorithms which require the equivalent of 2-10 complete flow solutions in order to render a complete design. On the other the range of objectives treatable by this approach is rather small and the user must be highly experienced in order to be able to prescribe surface distributions or choose initial geometries which lead to the desired

[4] Project number 032741A within the funding program Hochtemperaturgasturbine/TURBOTECH, industrial partners: ABB Mannheim, Germany and MTU Munich, Germany

aerodynamic properties. Furthermore it is difficult to formulate inverse methods that can satisfy desired aerodynamic and geometric constraints.

The alternative approach of numerical optimization methods, which avoids some of the difficulties of inverse methds, is usually believed to be computationally expensive (see e.g. [11]). Here, we present a method which is competitive concerning computation time with inverse methods, but on the other hand is flexible enough to easily incorporate geometric constraints for the blade design. The partially reduced SQP approach proposed here reveals several enhancements when compared with the reduced SQP methods suggested in [9, 13]. Reduced SQP techniques are simultaneous optimization approaches—or methods within the optimization boundary value problem framework [3, 5, 4], i.e. iterating over all variables (state and influence) in one loop—, but share with "black box" approaches the property that only reduced Hessians are constructed and used in the algorithm - which results in computational gains. But other reduced approaches (as mentioned above) are limited to optimization problems with the flow equation as the only constraint and thus are not able to consider additional constraints resulting, e.g., from geometrical design restrictions. Partially reduced methods, as established in [18, 19], overcome this limitation by incorporating the geometrical constraints into small quadratic subproblems to be solved in each iteration. Theoretical convergence proofs for this approach have been presented in [18]. A similar methodology has been applied to process optimization problems in chemical engineering in [1, 22, 14].

A striking feature of the method developed in this paper is that it can be easily generalized to working range optimization tasks, modeled as multiple setpoint optimization problems, which are much more important for the practical use of the optimal shapes computed than the single setpoint results, which are usually computed. The approach provides a very natural parallelization technique, as well.

The paper is organized in the following way. Section 2 is devoted to the flow model used. Section 3 describes the geometry of the problem and in particular geometric constraints. The optimization problem under investigation is formulated in section 4 together with its multiple set point variant. A PRSQP approach adapted to the solution of the various formulations is presented in section 5. Numerical results are given in section 6. Finally, concluding remarks are given in section 7.

2 The flow model

The numerical modeling of the flow is performed by the use of the solver MISES (Multiple blade Interacting Streamtube Euler Solver) [6, 23]. Considering the state of the art in computational methods, the most appropriate flow description would be by three-dimensional Navier-Stokes equations. However, their solution is still computationally highly complex and the results are generally considered as differing from practical flow measurements by approximately the same amount as the results obtained from the computations as described briefly below. Although the flow in axisymmetrical turbomachinery is three dimensional, a useful and often necessary simplification for design purposes is to approximate the flow through a stage as a set of two-dimensional blade-to-blade problems defined on axisymmetrical stream surfaces—examples of them are shown in figure

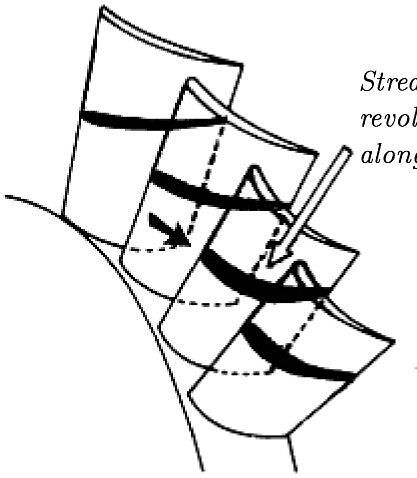


Figure 1: *Blade-to-blade flow on a stream surface of revolution*

1. Axisymmetrical through-flow codes are used early in the preliminary design process to define circumferentially averaged conditions in one or more stages of the machine based on initial estimates of work and loss. At the next level of design refinement (pertaining to our situation), the stream surface radius and spacing can be used to define a set of quasi-3D blade-to blade design problems for each stage. These allow the designer to select or design blade profiles at several radial stations to define the complete three-dimensional rotor or stator blade. The blade-to-blade technique works very well for most design applications, limited in effectiveness largely by the estimates for boundary layer effects on the inner and outer walls and by three-dimensional effects not accounted for with the axisymmetric assumptions.

Instead of solving the viscous flow directly, a zonal approach is used, where an equivalent inviscid flow in the interior of the computational domain is postulated using a displacement surface to represent the viscous layer. The inner boundary is displaced outward from the wall by the boundary layer displacement thickness δ. The inviscid flow is modeled by the steady state Euler equations (here in integral form over a control volume A)

$$\begin{aligned} \int_{\partial A} \rho \cdot v^\top n \,\mathrm{ds} &= 0, \quad \text{(mass)} \\ \int_{\partial A} \rho \cdot v^\top n \cdot v + p \cdot n \,\mathrm{ds} &= -\int_A \rho \cdot f \,\mathrm{dA}, \quad \text{(momentum)} \\ \int_{\partial A} \rho \cdot v^\top n \cdot R \,\mathrm{ds} &= 0, \quad \text{(energy)} \end{aligned}$$

while the boundary layer flow is modeled by integral boundary layer equations of the form

$$\begin{aligned} \frac{d\theta}{ds} &= F_1(\theta, \delta, u_e), \\ \frac{dH(\theta, \delta, u_e)}{ds} &= F_2(\theta, \delta, u_e, C_\tau), \\ \frac{dC_\tau}{ds} &= F_3(\theta, \delta, u_e, C_\tau). \end{aligned}$$

Here, ρ denotes the density, v the velocity of the flow and p its pressure, n the unit normal and R the stagnation enthalpy. θ denotes the momentum thickness, δ the displacement thickness, u_e the velocity at the boundary layer, C_τ the shear stress coefficient and H the energy thickness shape parameter.

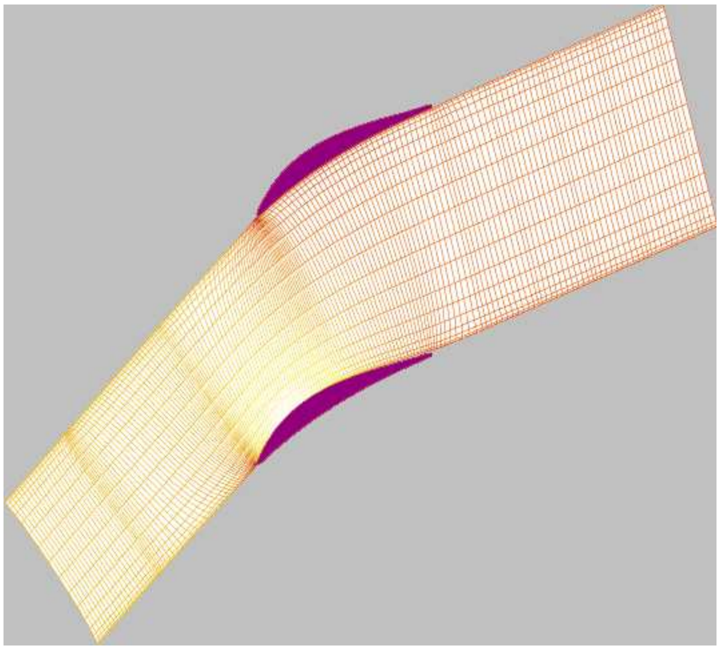


Figure 2: *Example for the computational grid for compressor blades.*

These basic equations have to be complemented by additional coupling equations and inflow/outflow boundary conditions. A complete and detailed description of the flow model and its discretization can be found in [23]. The discretization is performed on a stream-line aligned structured grid as depicted in figure 2. The grid sizes one needs for blade design is case dependent and ranges from about 150×20 to 300×40.

3 Blade description and geometric constraints

The blade profiles are defined in (m, θ) streamsurface coordinates. From cone-coordinates (r, z, ϕ) for a given streamline $r(z)$ computed by an axisymmetrical through-flow code these can be obtained as

$$m = \int_{z_0}^{z} \frac{\sqrt{(dr(\zeta))^2 + (d\zeta)^2}}{r(\zeta)}, \quad \theta = \phi - \theta_0.$$

The offset θ_0 is chosen so that θ vanishes for the smallest z-coordinate. The streamline is typically approximated by a straight line

$$r(z) = a + bz$$

with appropriately chosen parameters a and b. The blade profiles themselves are represented by quintic B-Splines in both coordinates:

$$\begin{aligned} m(t) &:= \sum_{i=1}^{12} p_i B_i(t), \\ \theta(t) &:= \sum_{i=1}^{12} p_{i+12} B_i(t), \end{aligned} \qquad 0 \le t \le 1.$$

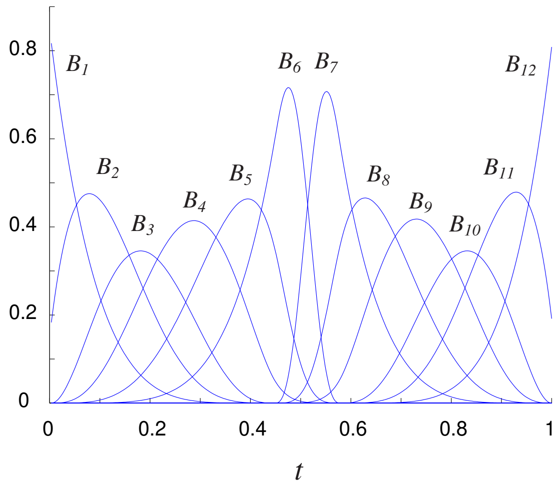


Figure 3: *B-spline basis functions*

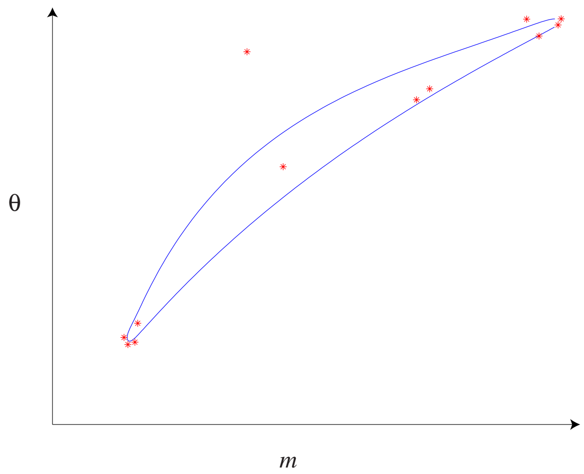


Figure 4: *B-spline example profile and control points*

The basis functions $B_i(t)$ are plotted in Fig. 3, while Fig. 4 shows an example profile $(m(t), \theta(t)), 0 \leq t \leq 1$, with its control points $(p_i, p_{i+12}), i = 1, \ldots, 12$. The basis functions are chosen so that the whole spline profile is two times continuously differentiable everywhere.

These spline profiles are subjected to geometric constraints for two reasons:

- On the one hand there are certain geometric requirements resulting, e.g., from the actual construction process of the blades and the necessity to cool the blades by the use of inner air pipes.
- On the other hand geometric constraints are used to stabilize the optimization algorithm, because otherwise intermediate blades could result, which cannot be treated by MISES, thus aborting the overall program. Fig. 5 shows the results of missing geometrical constraints.

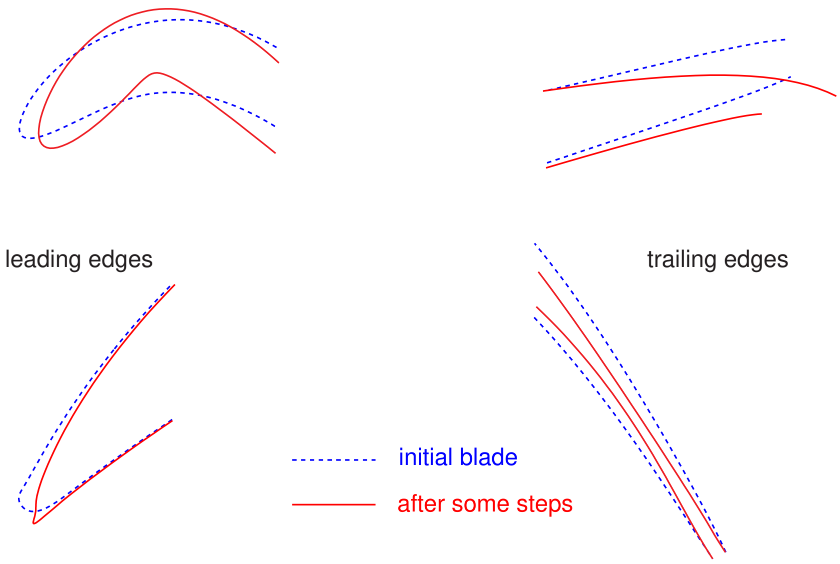


Figure 5: *Results of missing geometric constraints.*

Some of these geometric constraints are explained in more detail to give a rough impression on how many and what type of constraints these are.

Curvature

Curvature constraints are imposed over some regions at the leading edge and the suction side of the blades. The extent of these regions can be chosen by the user of the optimization program, as well as the lower and upper bounds imposed. Curvature restrictions at the trailing edge are imposed to counteract undesirable side effects of angle conditions.

Trailing edge thickness

Since the blade profile computationally exists only as a set of discrete points, the term "thickness" itself has to be defined. A practical definition is to define it as the projection of a straight line between opposite points onto the normal of the sceleton curve (cf. Fig. 6). It has turned out to be of special importance to restrict the thickness not only at the trailing edge but also overall in some region towards the trailing edge.

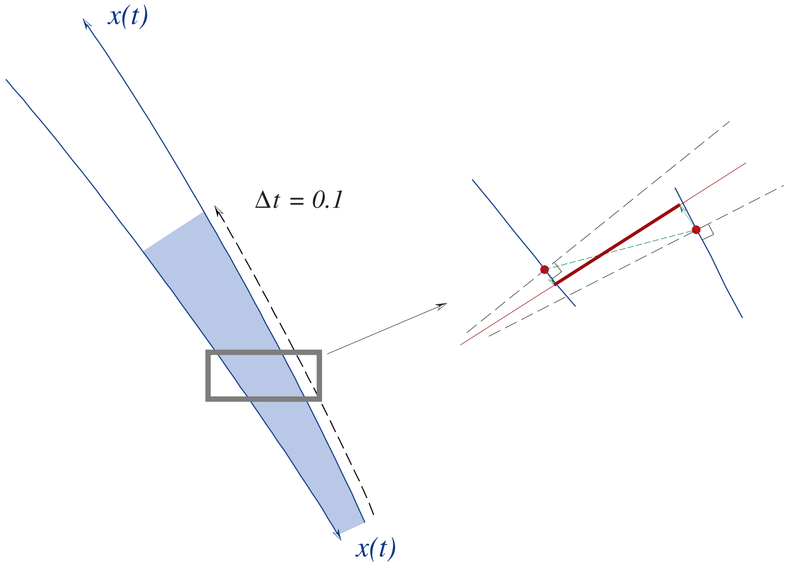


Figure 6: *Trailing edge thickness.*

Area

The area enclosed by the blade profile is a measure for the mechanical stiffness of the profile.

Shift of trailing edge points

In principle the endpoints of the open blade at the trailing edge can be shifted freely. But this may lead to unrealistic results. Therefore the angle between the line connecting these points and the normal to the sceleton line is restricted (cf. Fig. 7).

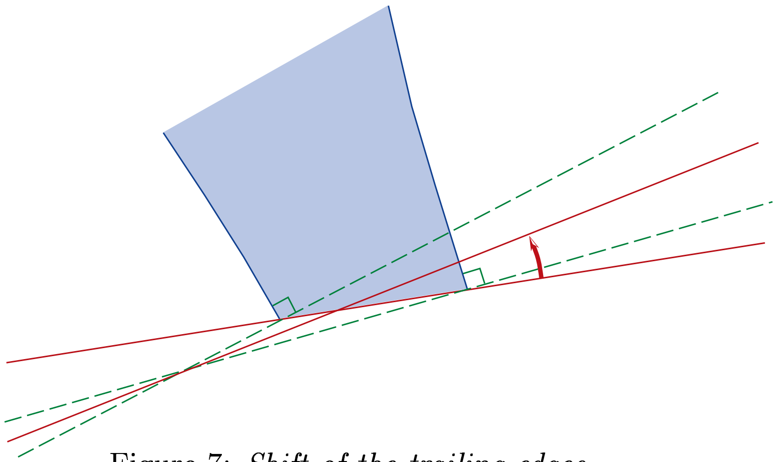


Figure 7: *Shift of the trailing edges.*

Leading edge thickness

The leading edge thickness is also not a well defined quantity. We consider instead of this the distance between the two control points at the leading edge. In combination with area and curvature constraints this has been used with good results.

Blade width

The blade width is defined as the distance between minimal and maximal m-coordinate (cf. Fig. 8).

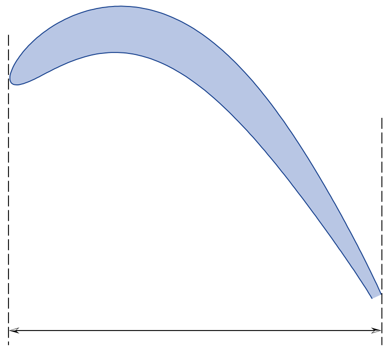


Figure 8: *Blade width.*

Aperture angle

The angle of the aperture of the trailing edge is another means to control the shape of the trailing edge. The quantity restricted is the difference between the trailing edge tangent angle.

There are some more types of geometric restrictions mainly related to criteria which are rather specific for this branch of engineering. They are considered in the optimization procedure but a detailed discussion of them is beyond the scope of this paper. All of the above geometric constraints are collected in the inequality

$$g(p) \geq 0, \quad g \in C^2(\mathbb{R}^{24} \to \mathbb{R}^m).$$

4 The optimization problems

We start out with the problem formulation

$$\begin{array}{rrcll} & \min\limits_{x,p} f(x,p), & & & f: \ \mathbb{R}^{n_x+n_p} \to \mathbb{R} \\ \text{s.t.} & c(x,p) & = & 0, & c: \ \mathbb{R}^{n_x+n_p} \to \mathbb{R}^{n_x}, \ c_x \text{ nonsingular} \\ & g(p) & \geq & 0, & g: \ \mathbb{R}^{n_p} \to \mathbb{R}^{n_g}, \end{array} \tag{4.1}$$

where $x \in \mathbb{R}^{n_x}, p \in \mathbb{R}^{n_p}$. The cost functional f denotes the total pressure loss coefficient. For a detailed definition see [23]. The equality constraint c summarizes the discretized flow equations as discussed in section 2 for the flow state variables x and the profile parameters p. The condition, c_x nonsingular, reflects the fact that we assume that the flow equations can always be solved—which actually is an assumption, for which one has to take care of in order to be justified, as we have indicated in section 2.

That is the problem formulation for given working conditions and for one stream-surface-wise cross section of the blade. In general turbomachine industry is much more interested in having optimal blades for a whole range of working conditions (e.g. varying inflow angle) and also cross sectional shapes for various "heights" of the blade are searched for, which are not undulating too much when put together—i.e. which possess a certain smoothness in the third dimension. Fig. 9 shows a typical scalar curve of importance $\omega(\alpha) > 0$ for

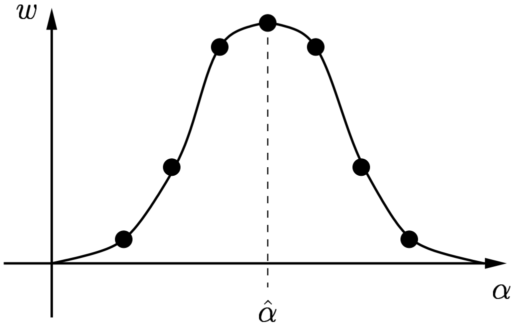


Figure 9: *Working range weight distribution.*

working conditions characterized by a vector (or a scalar) $\alpha \in A$ within a range A. The point $\hat{\alpha}$ represents the working condition in the design optimization like above. This curve arises from empirical observations. Thus an infinite form of the working range optimization problem can be derived for the one-parameter family $x(\alpha)$ as

$$\begin{array}{rcl} \min\limits_{x(\alpha),p} \displaystyle\int\limits_A \omega(\alpha) f(x(\alpha), p; \alpha) d\alpha, & & \\ \text{s.t.} \quad c(x(\alpha), p; \alpha) & = & 0, \quad \forall \alpha \in A, \\ g(p) & \geq & 0. \end{array} \tag{4.2}$$

After choosing an appropriate quadrature formula for the integral in equation (4.2) with wights $\{w_i\}_{i=1}^N$ at locations $\{\alpha_i\}_{i=1}^N$ indicated by the bullets in fig. 9, we formulate a corrsponding multiple setpoint problem of the form

$$\begin{array}{rcl} \min\limits_{x_1,\ldots,x_N,p} \displaystyle\sum_{i=1}^N w_i f(x_i, p; \alpha_i), & & \\ \text{s.t.} \quad c(x_1, p; \alpha_1) & = & 0, \\ & \vdots & \\ c(x_N, p; \alpha_N) & = & 0, \\ g(p) & \geq & 0, \end{array} \tag{4.3}$$

where now appear several state vectors x_i corresponding to the working conditions α_i, but only one profile vector p. Note that the α_i are not part of the optimization variables.

If in addition we are interested in a stack of blade cross sections $\{p_j\}_{j=1}^M$, where each $p_j \in \mathbb{R}^{24}$, corresponding to different stream surfaces and thus stacked ontop of each other and have a mathematical description of 3D-smoothness in the form of inequalities

$$h(p_1, \dots, p_M) \geq 0, \quad h : \mathbb{R}^{M \times n_p} \to \mathbb{R}^{n_h}$$

then we come to an optimization problem formulation reflecting the complexity and structure of the overall optimization task:

$$\begin{array}{rcl} \min\limits_{\{\{x_{ij}\}_{i=1}^N, p_j\}_{j=1}^M} & \displaystyle\sum_{j=1}^M \sum_{i=1}^N w_{ij} f(x_{ij}, p_j; \alpha_{ij}), & \\ \text{s.t.} \quad c(x_{ij}, p_j; \alpha_{ij}) & = & 0, \ \forall i = 1, \dots, N, \ \forall j = 1, \dots, M \\ g_j(p_j) & \geq & 0, \ \forall j = 1, \dots, M \\ h(p_1, \dots, p_M) & \geq & 0, \end{array} \tag{4.4}$$

where everything has got an additional index j for the stream surface level.

5 The PRSQP approach

In the presentation of the partially reduced SQP method (PRSQP) used for the solution of the optimization problems first we focus on problem formulation (4.1). There is a huge difference in the number of state variables x correlated to the flow and the number of design variables p. Therefore, without the constraints g, the solution method of choice would be a reduced SQP method within the separability framework [9, 13]. The idea of reduced SQP methods in contrast to usual SQP methods is to use only an approximation of the projected Hessian of the Lagrangian onto the kernel of the linearized constraint c, instead of an approximation of the full Hessian of the Lagrangian. In order to apply these methods, one must have a global parameterization of the kernel of all active constraints. In the presence of additional equality and inequality constraints, g, such a global parameterization is hard to determine, possibly resulting in instabilities.

The partial reduction concept used here and introduced in [18, 19] is meant to overcome these difficulties without sacrificing the advantages of the reduction of the Hessian of the Lagrangian. On the one hand, one uses the possible reduction in complexity by exploiting the Null space structure of some equality constraints, but on the other hand one allows for a convenient treatment of inequality constraints and other equality constraints. The reduced SQP method is formulated only w.r.t. those constraints which allow for a straight forward parameterization (c). The remaining constraints (g) are treated in the same way as in usual SQP, but reduced on the kernel of the above constraints.

In order to clarify the presentation we first consider the steps of a (full) SQP method. In each iteration of an SQP method the following QP subproblems have to be solved:

$$\begin{array}{rcl} \min\limits_{\Delta x^k, \Delta p^k} \ \frac{1}{2} \left(\Delta x^k, \Delta p^k\right)^\top H^k \left(\Delta x^k, \Delta p^k\right) + \nabla(f^k)^\top \left(\Delta x^k, \Delta p^k\right) & & \\ \text{s.t.} \quad C^k \left(\Delta x^k, \Delta p^k\right) + c^k & = & 0 \\ G^k \Delta p^k + g^k & \geq & 0, \end{array} \tag{5.1}$$

where $H^k := \partial^2\mathcal{L}^k/\partial(x,p)^2$ denotes the Hessian of the Lagrangian of the optimization problem (4.1)

$$\mathcal{L} := f(x,p) - c(x,p)^\top\lambda - g(p)^\top\mu$$

and capital letters C, G denote the derivatives of the constraints c, g at the k-th iterate.

We define

$$T := \begin{bmatrix} -C_x^{-1}C_p \\ I \end{bmatrix} \quad \text{as a basis of the kernel of} \quad C = \begin{bmatrix} C_x \ C_p \end{bmatrix}$$

so that each solution of the first constraint equation in (5.1) satisfies

$$\Delta x^k = \Delta x^k_{\mathcal{R}} + T^k \Delta x^k, \tag{5.2}$$

where $\Delta x^k_{\mathcal{R}} = -(C^k_x)^{-1}c^k$. The relationship (5.2) is now used to formally eliminate Δx^k from (5.1). In the manner of typical RSQP methods we set the cross-term $H_k\Delta y^{\mathcal{R}}_k$ to zero, thus leaving $(T^k)^\top\left(\partial^2\mathcal{L}^k/\partial(x,p)^2\right)T^k$ as the only part of the Hessian to be considered in the algorithm. This is so-called reduced Hessian is approximated by a matrix $B^k \in \mathbb{R}^{n_p \times n_p}$.

A conceptual PRSQP method is sketched in Fig. 10. As before, indices k mean evaluation

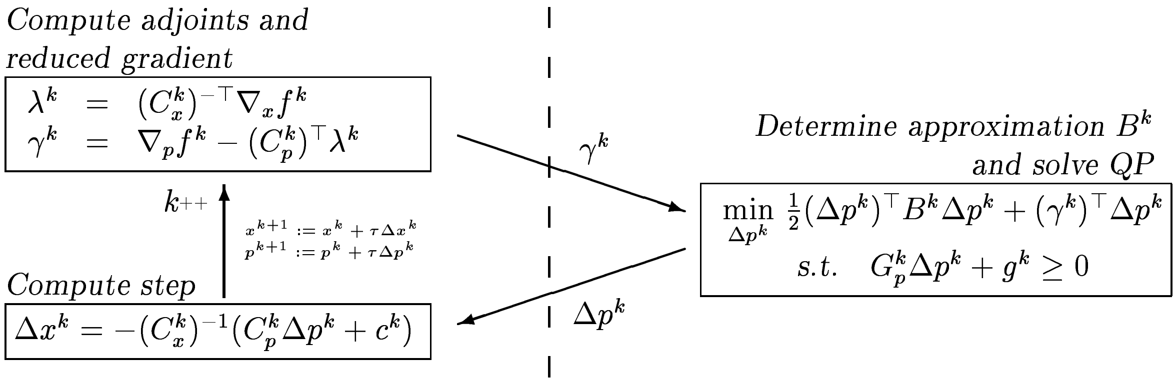


Figure 10: *Sketch of a PRSQP method.*

at the k-th iterate.

Remark: The PRSQP algorithm requires the solution of a QP-subproblem. During this whole presentation we assume that it has a nonempty feasible set. Otherwise we might apply well known remedies (cf. e.g. [10]) for quadratic subproblems of SQP methods. For a nonempty feasible set the solvability of the QP is guaranteed, if $v^\top B_k v > 0$ for all v in the kernel of all active constraints. As we will see below, we always use reduced Hessian approximations B^k which are positive definite and thus satisfy this requirement.

In Fig. 10 there appears a vertical dashed line. It indicates the basic interface between the flow computation modul MISES and the pure optimization part of the algorithm. Thus, although the optimization method is one of the simultaneous type with all its benefits, the modularity is rather high and there is no need to update the optimization part of the algorithm, every time there comes up a new release of the flow computation modul.

The strength of the PRSQP method arises from the fact that the reduced Hessian can be approximated by B^k and thereby computationally expensive applications of T and $T^\top$ can be avoided. The approximation of the reduced Hessian B^k has to be performed in a sufficiently accurate way in order to provide local convergence properties which are better than linear. For this approximation, one can employ, e.g., the BFGS update formula (cf. [8]):

$$\begin{aligned} B^{k+1} &:= B^k + U^{BFGS}(B^k, z, w), \\ \text{where} \quad U^{BFGS}(B, z, w) &:= \tfrac{ww^\top}{z^\top w} - \tfrac{(Bz)(Bz)^\top}{z^\top Bz}. \end{aligned} \tag{5.3}$$

The key property of this formula (which it shares with all other members of the convex Broyden class) is the secant condition $B^k z = w$. The intention of this kind of update is to collect second order information from first order magnitudes available in each iteration. Therefore z is formed by the differences of p-variable values, $z := \bar{p}^k - p^k$ and w is formed by the resulting difference of reduced gradients of the Lagrangian

$$w := \bar{\gamma}^k - \gamma^k - \left[(\bar{G}^k)^\top - (G^k)^\top\right] \mu^k, \tag{5.4}$$

where a bar over a symbol means evaluation at an intermediate point. The intermediate point may be chosen to be (x^{k+1}, p^{k+1}), which defines an update strategy in the spirit of [15], or

$$(\bar{y}_k, \bar{w}_k) = (y_k, w_k) + (T_{1,k}\Delta w_k, \Delta w_k), \tag{5.5}$$

which defines an asymptotically correct update strategy. A proof for the resulting local 2-step-superlinear convergence of the algorithm can be found in [18]. For implementational ease we chose the former alternative together with a Han-Powell-modification [16] of the update formula in order to maintain positive definiteness and with a limited memory strategy to avoid blow up of the condition number.

The line-search parameter τ in the algorithm is determined by the use of a weighted sum of the objective function and the norm of the flow residual as merit function. In addition there is a safeguard strategy implemented projecting the iterate back in the direction of the feasible manifold (i.e. setting $\Delta p^k := 0$), when necessary. The termination of the algorithm is controled by the following criteria:

- either the norm of the reduced gradient is below a certain tolerance
- or the actual change of the blade, i.e.

  $$\int\limits_{\text{blade}} \left| \left(p^k(t) - p^{k-1}(t)\right)^\top n_{p^k}(t) \right| dt,$$

  where $p(t) \in \mathbb{R}^2$ means the whole spline curve determined by the spline coefficients $p \in \mathbb{R}^{24}$ and $n_{p^k}(t) \in \mathbb{R}^2$ means the unit normal to this curve. So the program stops also, if the spline curve is only more or less changed tangentially without changing the shape itself. This may happen due to the non-uniqueness of the B-spline parameterization.

The solution of the linear systems arising in multiplications with $(C_x^k)^{-1}$ and $(C_x^k)^{-\top}$ are performed by a block-sparse factorization. In a previous pilot project, reported on in

[20, 19, 7], we had to use a multigrid iterative technique for that purpose. The details of the resulting multigrid technique are described in these references.

The algorithmic variants of the basic algorithm above for the problem formulations (4.3) and (4.4) now reveal the full modularity potential of the PRSQP approach. First we focus on problem (4.3). If we identify x in the algorithm 10 with the collection of all x_i from problem (4.3) and identify $c(.,.) := (c(.,.;\alpha_1),\dots,c(.,.;\alpha_N))^\top$, we immediately see that C_x possesses the block diagonal structure

$$C_x(x,p) = \begin{bmatrix} \frac{\partial c(x_1,p;\alpha_1)}{\partial x_1} & & 0 \\ & \ddots & \\ 0 & & \frac{\partial c(x_N,p;\alpha_N)}{\partial x_N} \end{bmatrix}$$

By defining

$$\begin{aligned}
\lambda_i^k &:= \left(\frac{\partial c(x_i^k,p^k;\alpha_i)}{\partial x_i}\right)^{-\top} \nabla_{x_i} f^k(x_i^k,p^k;\alpha_i) \\
\gamma_i^k &:= \nabla_p f^k(x_i^k,p^k;\alpha_i) - \left(\frac{\partial c(x_i^k,p^k;\alpha_i)}{\partial p}\right)^{\top} \lambda_i^k \\
\Delta x_i^k &:= -\left(\frac{\partial c(x_i^k,p^k;\alpha_i)}{\partial x_i}\right)^{-1} \left(\frac{\partial c(x_i^k,p^k;\alpha_i)}{\partial p}\Delta p^k + c(x_i^k,p^k;\alpha_i)\right)
\end{aligned}$$

we obtain the following observation formulated in lemma 5.1.

Lemma 5.1

$$\gamma^k = \sum_{i=1}^N w_i \gamma_i^k.$$

□

Therefore, we obtain the following algorithmic variant depicted in fig. 11. In this algorithm the linear forward and adjoint problems can be solved completely independently (indicated by the dashed lines) for the different setpoints of the working range. This supports a parallel implementation very well.

For problem formulation (4.4), also the linear quadratic subproblems reveal additional structure. The linear quadratic subproblems to be solved in an analogous algorithm are of the form

$$\begin{aligned}
&\min_{\{\Delta p_j^k\}_{j=1}^M} \frac{1}{2}\begin{pmatrix}\Delta p_1^k \\ \vdots \\ \Delta p_M^k\end{pmatrix}^\top B^k \begin{pmatrix}\Delta p_1^k \\ \vdots \\ \Delta p_M^k\end{pmatrix} + \begin{pmatrix}\gamma_1^k \\ \vdots \\ \gamma_M^k\end{pmatrix}^\top \begin{pmatrix}\Delta p_1^k \\ \vdots \\ \Delta p_M^k\end{pmatrix} \\
&\text{s.t.} \qquad \frac{\partial g_j^k}{\partial p_j}\Delta p_j^k + g_j^k \geq 0, \quad \forall j = 1,\dots,M \qquad\qquad (5.6) \\
&\qquad \sum_{j=1}^M \frac{\partial h^k}{\partial p_j}\Delta p_j^k + h^k \geq 0,
\end{aligned}$$

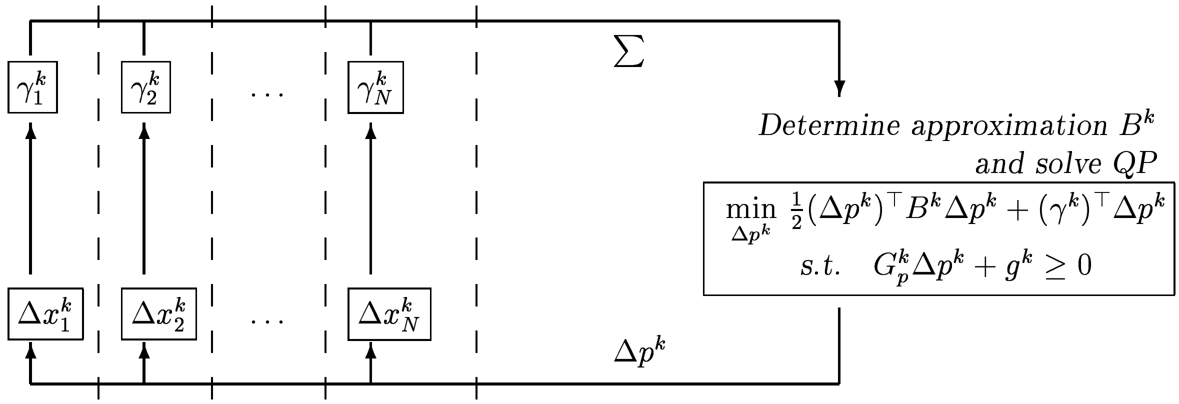


Figure 11: *Sketch of the PRSQP method for problem (4.3).*

where

$$\gamma_j^k := \sum_{i=1}^{N} \nabla_p f^k(x_{ij}^k, p_j^k; \alpha_{ij}) - \left(\frac{\partial c(x_{ij}^k, p_j^k; \alpha_{ij})}{\partial p} \right)^\top \lambda_{ij}^k, \ \forall j,$$

and

$$\lambda_{ij}^k := \left(\frac{\partial c(x_{ij}^k, p_j^k; \alpha_{ij})}{\partial x_{ij}} \right)^{-\top} \nabla_{x_{ij}} f^k(x_{ij}^k, p_j^k; \alpha_{ij}).$$

Consequently, the state variable increments are to be computed as in lemma 5.2.

Lemma 5.2

$$\Delta x_{ij}^k = - \left(\frac{\partial c(x_{ij}^k, p_j^k; \alpha_{ij})}{\partial x_{ij}} \right)^{-1} \left(\frac{\partial c(x_{ij}^k, p_j^k; \alpha_{ij})}{\partial p_j} \Delta p_j^k + c(x_{ij}^k, p_j^k; \alpha_{ij}) \right).$$

□

The reduced Hessian in QP (5.6)—to be approximated by B^k—obviously possesses block-diagonal form, which can be exploited by the application of partitioned block-updates as presented in [5] for full SQP methods and in [18, 14] for the reduced case.

6 Numerical results

Here we show examples for the efficacy of the optimization algorithm. In Fig. 12 the result of the optimization for a turbine blade at a specific working range set point—indicated by the downward arrow on the right hand side of this figure—is shown. This single set point problem corresponds to the formulation (4.1). Although the effect of the optimization might seem not much when comparing the initial blade with the optimized, the gain in the objective criterion (pressure loss) is approximately 14% and thus considerable. Here

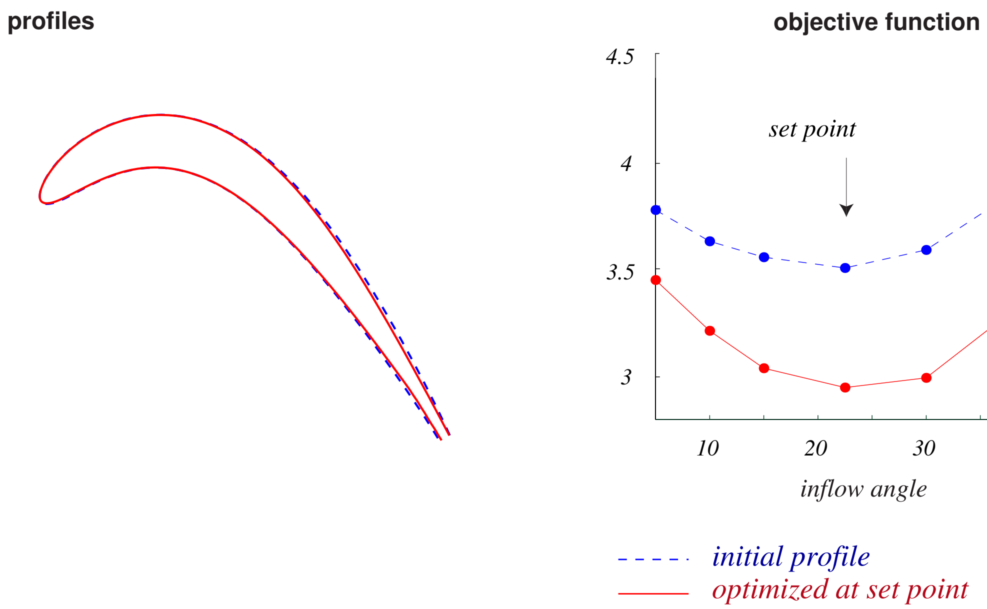


Figure 12: *Turbine blade, optimized at set point and over the working range, respectively.*

13000 state variables are involved in the discretized flow description and 15 PRSQP iterations are needed for convergence. The overall computing time, however, has been on an IBM workstation of type RS6000/900 only 3 cpu minutes, which corresponds to 4 forward flow equation solutions (which takes each 45 seconds). Thus, here the simultaneous optimization approach really pays off, although modularity of the implementation is still maintained. The lower curve on the right hand side shows the objective functional over the whole working range (nevertheless the blade is optimal only for the one set point). One can see the optimized blade leads to good results all over in the working range (Fig. 12).

However, the situation is completely different in the case of compressor blades, as it is shown in figure 13. On the left hand side two pairs of blades are plotted, where in each pair the initial blade before optimization and the results of the optimization is shown. The upper pair is for a single set point optimization for the set point (inflow angle) indicated on the right hand side by an arrow. This single set point optimization took 15 cpu minutes on Pentium II/400 based Linux system, which again corresponds to 4 forward solution sweeps.

Looking at the whole working range for the inflow angle, one recognizes that near the boundary of the working range the pressure loss is increasing dramatically for the blade optimized for a single set point. That surely is not what turbomachine engineers think of as an "optimal" blade. On the other hand, the working range formulation (4.3) leads to blades, which are better all over in the working range, where the pressure loss shows reasonable behavior. But, as it can be expected, the blade optimized over the whole working range behaves slightly worse at the specific set point considered above.

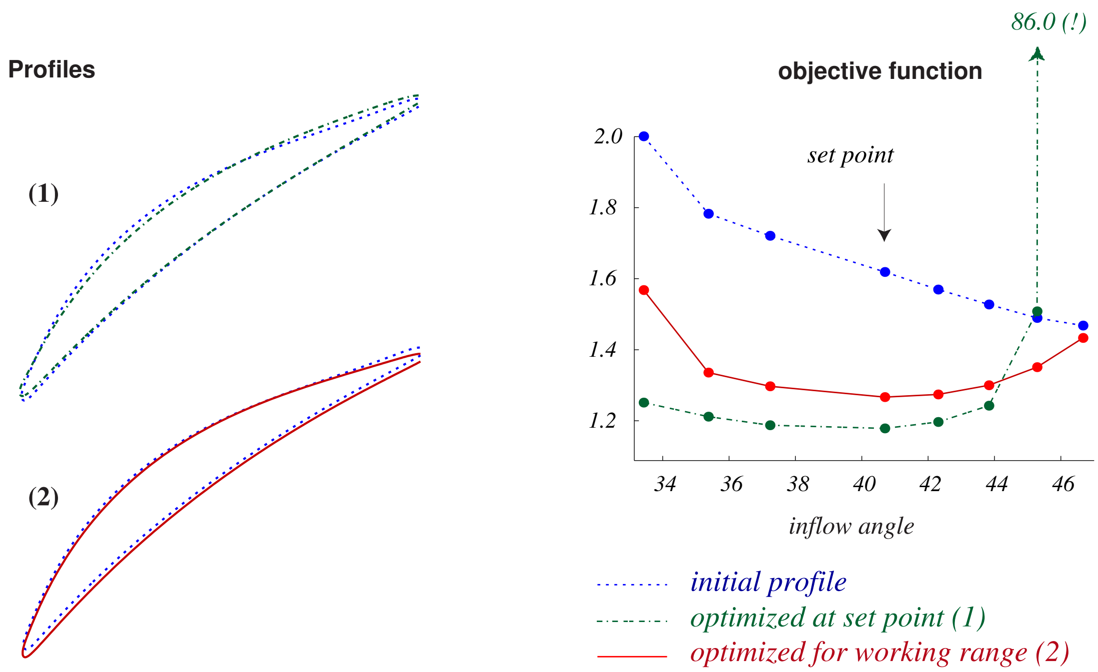


Figure 13: *Compressor blade, optimized at set point and over the working range, respectively.*

7 Conclusions

The determination of optimal blade shapes is an important and usually time-consuming stage within the design-cycles of turbines and compressors for aircrafts and power plants. The large amount of time necessary in practice up to now is related to the fact, that is task is primarily performed manually by interactive control of several simulation runs. To perform this task automatically, two ingredients are necessary: a well posed problem description, so that the optimization routine does not lead to physically irrelevant solutions and, on the other hand, a fast optimization algorithm, which is able to incorporate simulation strategies which are well tested.

In this paper both important steps are explained. We have formulated basic geometric constraints, which limit the configuration space, so that blades are not determined which go beyond the capabilities of the flow model. Also, we have constructed a new and fast optimization method which consumes computation time of the same order as a pure simulation run. In addition, we have shown, how working range formulations, which are of high practical importance, can be treated efficiently.

References

[1] L.T. Biegler, J. Nocedal, and C. Schmid. A reduced Hessian method for large-scale constrained optimization. *SIAM Journal on Optimization*, 5(2):314–347, 1995.

[2] J. Blazek. Aerodynamic shape optimization of turbomachinery cascades. Technical Report Paper 97-2131 (A97-32475), AIAA, 1997.

[3] H.G. Bock. Numerical treatment of inverse problems in chemical reaction kinetics. In Ebert, Deuflhard, and Jäger, editors, *Modelling of Chemical Reaction Systems*, volume 18 of *Springer Series Chemical Physics*, pages 102–125, Heidelberg, 1981.

[4] H.G. Bock. Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen. *Bonner Mathematische Schriften 183*, Bonn, 1987.

[5] H.G. Bock and K.J. Plitt. A multiple shooting algorithm for direct solution of constrained optimal control problems. In *Proceedings 9th IFAC World Congress Automatic Control*. Pergamon Press, 1984.

[6] M. Drela. *Two-dimensional transonic aerodynamic design and analysis using the Euler equations*. PhD thesis, MIT, 1985.

[7] Th. Dreyer, B. Maar, and V. Schulz. Multigrid optimization in applications. *J. Comput. Appl. Math.*, 120 (cf. [21]):67–84, 2000.

[8] R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, New York, 1987.

[9] D. Gabay. Reduced quasi-Newton methods with feasibility improvement for nonlinear constrained optimization. *Mathematical Programming Study*, 16:18–44, 1982.

[10] P.E. Gill, W. Murray, and M.H. Wright. *Practical optimization*. Academic Press, London, 1981.

[11] A. Jameson and J. Reuther. Aerodynamic shape optimization of wing and wing-body configurations using control theory. Technical Report Paper 95-0123 (A95-21625), AIAA, 1995.

[12] A. Jameson, J. Reuther, L. Martinelli, and J. Vassberg. Aerodynamic shape optimization techniques based on control theory. Technical Report Paper 98-2538 (A98-32805), AIAA, 1998.

[13] F.-S. Kupfer. An infinite-dimensional convergence theory for reduced sqp methods in hilbert space. *SIAM Journal on Optimization*, 6(1):126–163, 1996.

[14] D. Leineweber. *Efficient reduced SQP methods for the optimization of chemical processes described by large sparse DAE models*. PhD thesis, IWR, University of Heidelberg, 1999. (also published in Forschritt-Berichte VDI, Reihe 3, Nr. 613).

[15] J. Nocedal and M.L. Overton. Projected Hessian updating algorithms for nonlinearly constrained optimization. *SIAM Journal on Numerical Analysis*, 22(5):821–850, 1985.

[16] M.J.D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In G.A. Watson, editor, *Numerical Analysis Proceedings Dundee 1977*, pages 144–157. Springer-Verlag, 1978.

[17] J. Reuther, A. Jameson, L. Martinelli, and D. Saunders. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. Technical Report Paper 96-0094 (A96-18067), AIAA, 1996.

[18] V.H. Schulz. *Reduced SQP methods for large-scale optimal control problems in DAE with application to path planning problems for satellite mounted robots.* PhD thesis, University of Heidelberg, 1996.

[19] V.H. Schulz. Solving discretized optimization problems by partially reduced SQP methods. *Computing and Visualization in Science*, 1(2):83–96, 1998.

[20] V.H. Schulz, Th. Dreyer, Th. Speer, and H.G. Bock. Optimum shape design of turbine blades. In P. Kleinschmidt, A. Bachem, U. Derigs, D. Fischer, U. Leopold-Wildburger, and R. Möhring, editors, *Operations Research Proceedings 1995*, pages 190–195. Springer, Heidelberg, 1996.

[21] V.H. Schulz (guest-editor). Special issue on SQP-based direct discretization methods for practical optimal control problems. *Journal of Computational and Applied Mathematics*, 120, 2000.

[22] M. von Schwerin, O. Deutschmann, and V. Schulz. Process optimization of reactive systems by partially reduced sqp methods. *Computers and Chemical Engineering*, 24:89–97, 2000.

[23] H. Youngren. *Analysis and Design of transonic cascades with splitter vanes.* PhD thesis, MIT, 1991.