Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 4, No. 2, pp. 243 – 258, May 2008

PARETO-CURVE CONTINUATION IN MULTI-OBJECTIVE OPTIMIZATION

Stephan Schmidt and Volker Schulz

Abstract: Aerodynamic shape optimization usually aims at optimizing a single design objective. This often means a drag or noise reduction given a constant lift or pitching moment. By re-formulating the constraints as additional objective functions, one can embed this scenario into a multi-objective design approach, which results in a set of indifference points between the cost functions. A potential designer can then choose from a variety of equally suitable solutions. In this paper, we explore curve continuation strategies for finding the indifference curve exemplified by an inviscid lift/drag optimization of the RAE2822 airfoil. Special attention is given to approaches that work well with a given SQP solver for the single objective problem. Different parametrizations of the curve are also studied.

Key words: *curve continuation, Pareto-optimality, multi-objective design, one-shot-optimization, shape optimization, SQP*

Mathematics Subject Classification: *90C29, 65K05, 49K20, 76N25*

1 Introduction

A wide variety of optimization problems arising in aerodynamics and engineering inherently involve optimizing multiple design objectives. The difficulty lies in the often conflicting nature of the different cost functions: A possible example in bridge construction could be the maximization of the stiffness while simultaneously minimizing the total mass of that bridge. Similarly, the drag minimization problem of an airfoil is coupled with the lift generated. A drag reduction is thus easily achievable with sacrifices in the latter. Therefore, optimization usually involves a constant lift constraint. However, the actual design goal would be to minimize drag while simultaneously maximizing lift. Since it is highly improbable that these conflicting objectives can both be extremised by the same design, the single optimum solution has to be replaced by a set of solutions that are indifferent to each other. That means, we are to find all possible designs in which the drag cannot be further reduced without also decreasing the lift. A design for which this holds true is usually called Pareto-optimal and the set of all Pareto-optimal solutions is called the indifference set.

The goal of this paper is to investigate the generalization of existing solvers for the single objective optimization into the multiobjective framework exemplified by the adjoint based One-Shot SQP drag minimization of the RAE2822 testcase governed by the Euler Equations [5, 8, 9, 10]. Special attention is given to parametrizations which are advantageous for this, i.e. a parametrization via the violation of constraints. There are numerous approaches to reach this goal and the quality of each approach can be measured by the convergence speed and the distribution of the points found. A distribution of the Pareto-optimal designs according to the curvature of the indifference set provides a preferable discretization. For more details see, e.g. [1, 17].

Several different approaches have been examined as a possible extension to the given One-Shot single objective drag minimization of the RAE2822 airfoil:

Stochastic approaches have the distinct advantage of being able to converge to the whole set of indifferent points in one optimization run. In addition to the well known Evolutionary Algorithms, there are also numerous other courses of action. For example, points of the indifference set can be found by solving certain stochastical differential equations using Brownian Motions. However, the point distribution and convergence speed of these stochastic approaches usually depend on a variety of algorithmic parameters and seem to vary from testcase to testcase, as described in [3, 17, 18].

A rather new representative of the deterministic approaches is the Normal Boundary Intersection [1, 2]. Here, one tries to project the indifference set onto a certain simplex, e.g. the convex hull of local minima of each individual objective function. Because of this projection, one can expect a rather good distribution of the Pareto-optimal designs. Some comparisons can be found in [17]. On the one hand, this provides certain advantages when the dimension of the indifference set is greater than two. On the other hand, it is not guaranteed that the whole indifference set will be found. One can also lose the global Pareto-optimality of the designs when the rim of the indifference set is not convex. Those problems, however, are present with a variety of other courses of action, also.

More traditional approaches are curve continuation strategies. It can be shown, that the minimization of the convex combination, i.e. weighted sum, of the cost functions will result in a Pareto-optimal point. One can then use these weights as a parametrization of the whole indifference set. Unfortunately, this will result in a very poor distribution of the designs [1] and solving the same optimization problem multiple times can be rather cost intensive. There are several remedies to these problems varying from reparametrizations to using curvature information for a first order approximation of the local shape of the indifference set coupled with homotopy methods. This holds true especially for the two dimensional case using homotopy curve continuation strategies like the predictor-corrector algorithm. Numerous works exist in this field [7, 11, 13, 14, 15]. Since the single objective drag minimization of the RAE2822 testcase via prSQP [16] provides some information on the Hessian - and, as seen later, this information can also be used to extract information of the local curvature of the indifference set - the predictor-corrector curve continuation seems to offer the best synergies when mitigating to the multi-objective design problem. A proper choice of the curve's parametrization can help simplify this mitigation. However, some problem remain when the Hessian is approximated using update formulas such as BFGS.

2 Mathematical Background

2.1 Problem Statement

A multi-objective optimization problem is given by

$$\begin{aligned} & \min F(x) =: \begin{pmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{pmatrix} \\ \text{s.t.} \quad & c(x) = 0 \\ & g(x) \geq 0 \end{aligned}$$

where $F : \mathbb{R}^n \to \mathbb{R}^m,\ c : \mathbb{R}^n \to \mathbb{R}^{m_1},\ g : \mathbb{R}^n \to \mathbb{R}^{m_2}$ are sufficiently differentiable. Let

$$Z := \{x \in \mathbb{R}^n |\ c(x) = 0,\ g(x) \geq 0\}$$

denote the set of all feasible solutions. Since one cannot expect a single $x^* \in Z$ to minimize all objective functions f_i at the same time, the definition of optimality must be expanded in the Pareto-sense. For $\hat{x} \in Z$ and $\bar{x} \in Z$, the vector $F(\hat{x})$ is said to dominate the vector $F(\bar{x})$ if $\hat{x}$ solves at least one cost function better than $\bar{x}$ does, e.g.:

$$\forall i \in \{1, \ldots, m\} : f_i(\hat{x}) \leq f_i(\bar{x}) \text{ and } \exists j \in \{1, \ldots, m\} : f_j(\hat{x}) < f_j(\bar{x})$$

A solution x^* is now called Pareto-optimal if it is non-dominated [2]. While this definition is quite useful for evolutionary approaches - where only the non-dominated solutions are allowed to being carried over to the next generation - it provides fewer hooks for curve continuation strategies. Far more useful for those is the following characterization:

Let $\omega \in \mathbb{R}^m$ be greater than zero component wise and let x^* be the solution of

$$\min_{x \in Z} \omega^T F(x) =: F_\omega(x). \tag{2.1}$$

It can then be easily proven by contradiction, that x^* is Pareto-optimal. However, note that the above statement merely states, that the minimization of the linear combination is sufficient for Pareto-optimality. The reverse does not generally hold. By adding the additional constraint of

$$\sum_{i=1}^{m} \omega_i = 1$$

it is possible to reduce one surplus degree of freedom in the parametrization which results in the convex combination of cost functions as described in the introduction. Applying the necessary Karush-Kuhn-Tucker optimality conditions to (2.1) results in a parameter based problem where one has to find the roots of some function $H(x, \lambda, \nu, \omega) = 0$. The implicit function theorem then provides the existence of mappings $h(\omega) = (x, \lambda, \nu)$ which is a description of the indifference set.

2.2 Existence of Indifference Sets

By using the convex combination of the individual objective functions, the problem of finding one single Pareto-optimal point has been reduced to solving a single objective constraint optimization problem for a fixed value of ω. Coupled with the implicit function theorem, the existence of a solution of this optimization problem will lead to the existence of the indifference set. Therefore, we now briefly focus on the solution theory of a single objective constraint optimization problem:

$$\begin{array}{rl} & \min f(x) \\ \text{s.t.} & c(x) = 0 \\ & g(x) \geq 0, \end{array} \tag{2.2}$$

$$f \in C_2(\mathbb{R}^n, \mathbb{R}),\ c \in C_2(\mathbb{R}^n, \mathbb{R}^{m_1}),\ g \in C_2(\mathbb{R}^n, \mathbb{R}^{m_2}).$$

This can be thought of as being equation (2.1) with ω silently omitted. Under the assumption of the linear independence constraint qualification (LICQ), that is: With $I(x) = \{i_1, \ldots, i_s\}$ as the active set,

$$\{\nabla c_1(x), \ldots, \nabla c_{m_1}(x), \nabla g_{i_1}(x), \ldots, \nabla g_{i_s}(x)\}$$

is linearly independent, one gets the following optimality conditions:
Suppose $\hat{x}$ is feasible and LICQ. Suppose the Lagrangian satisfies:

$$\begin{aligned}\mathcal{L}(x,\lambda,\mu) :=& f(x) - \langle\lambda|c(x)\rangle - \langle\mu|g(x)\rangle \\ =& f(x) - \lambda^T c(x) - \mu^T g(x)\end{aligned}$$

where $x \in \mathbb{R}^n,\ \lambda \in \mathbb{R}^{m_1},\ \mu \in \mathbb{R}^{m_2}$.

Karush-Kuhn-Tucker (KKT) Condition:

$$\nabla_x \mathcal{L}(\hat{x},\lambda,\mu) = 0 \tag{2.3}$$

Strict Complementary Condition:

$$\begin{aligned}&\mu_i \geq 0\ \forall i \in \{1,\dots,m_2\} \\ &\mu^T g(\hat{x}) = 0 \text{ rep. } g_j(\hat{x}) > 0 \Rightarrow \mu_j = 0,\ g_j(\hat{x}) = 0 \Rightarrow \mu_j > 0\end{aligned} \tag{2.4}$$

Second Order Sufficiency Condition:

$$v^T \mathrm{Hess}_x \mathcal{L}(\hat{x},\lambda,\mu) v > 0,\ \forall v \in \tilde{T}(\hat{x}) \setminus \{0\} \tag{2.5}$$

where $\tilde{T}(\hat{x}) = \{v \in \mathrm{Core}(Dc(\hat{x}))|\ (\nabla g_j(\hat{x}))^T v = 0, \forall \mu_j > 0\}$.

Condition (2.3) and (2.4) are necessary and condition (2.5) is sufficient for the existence of a solution [12]. Our goal is to apply the implicit function theorem to these necessary conditions:

$$\begin{aligned}\nabla_x f(x) - \sum_{i=1}^{m_1} \lambda_i \nabla_x c_i(x) - \sum_{j=1}^{m_2} \mu_j \nabla_x g_j(x) &= 0 \\ c(x) &= 0 \\ g(x) &\geq 0 \\ \mu^T g(x) &= 0 \\ \mu &\geq 0\end{aligned} \tag{2.6}$$

Unfortunately, the implicit function theorem does not allow any inequalities. However, it has been shown, that the above system can be rewritten without inequalities [14]. For simplicity reasons here, we can use an active set approach and assume the lift constraint always being active, thus we may omit any further considerations of inequality constraints. This is no gap in the theory, since the above system can be rewritten using equalities without losing the order of differentiability. In our special case the necessary conditions therefore are:

$$H(x,\lambda,\omega) := \begin{pmatrix} \nabla_x F_\omega(x) - \lambda \nabla c(x) \\ c(x) \end{pmatrix} = 0$$

2.3 The Implicit Function Theorem

For clarity, we define $z := (x,\lambda)^T$. Suppose $\Omega \subset \mathbb{R}^d \times \mathbb{R}^k$ is open and $H \in C_1(\Omega, \mathbb{R}^d)$. Additionally, suppose $(\tilde{z},\tilde{\omega}) \in \Omega$, so that $H(\tilde{z},\tilde{\omega}) = 0$ and $\det \frac{\partial H}{\partial z}(\tilde{z},\tilde{\omega}) \neq 0$.
Then:

(1) There exist open neighborhoods $U(\tilde{\omega}) \subset \mathbb{R}^k$ and $W(\tilde{z}, \tilde{\omega}) \subset \Omega$ as well as a function $h : U \to \mathbb{R}^d$, satisfying for $(z, \omega) \in W$:

$$H(z, \omega) = 0 \Leftrightarrow z = h(\omega).$$

(2) Additionally, $h \in C_1(U, \mathbb{R}^d)$ and

$$\frac{\partial h}{\partial \omega}(\omega) = -\left[\frac{\partial H}{\partial z}(h(\omega), \omega)\right]^{-1} \frac{\partial H}{\partial \omega}(h(\omega), \omega)$$

A proof can be found in numerous sources, e.g. [4]. Interestingly, the requirements of the implicit function theorem are automatically satisfied when the KKT Condition (2.3) and the Second Order Sufficiency Conditions (2.5) are fulfilled [15]. We now can assume the existence of the indifference set as long as the problem of minimizing the convex combination is well posed.

2.4 Example for the Importance of Sufficient Conditions

The system of necessary conditions (2.6) actually only includes the first order conditions. Condition (2.5) is not part of our function H on which we apply the implicit function theorem. From a theoretical point of view, one could simply demand (2.5) to be satisfied as part of the problem being well posed. Form a practical point of view, however, the fact that the matrix is positive definite is rarely checked and this can lead to the computation of an “indifference set” that also includes points that are not Pareto-optimal. Consider the following basic example:

$$f(x) := \begin{cases} \frac{1}{3}(x+5)^3 & \text{if } x < -5 \\ 0 & \text{if } -5 \le x \le 5 \\ \frac{1}{3}(x-5)^3 & \text{if } 5 < x \end{cases}$$

The derivatives are:

$$\frac{\partial f}{\partial x}(x) = \begin{cases} (x+5)^2 & , x < -5 \\ 0 & , -5 \le x \le 5 \\ (x-5)^2 & , 5 < x \end{cases} \text{ and } \frac{\partial^2 f}{\partial x^2}(x) = \begin{cases} 2(x+5) & , x < -5 \\ 0 & , -5 \le x \le 5 \\ 2(x-5) & , 5 < x \end{cases}$$

Consider the following multiobjective design problem:

$$\min_x \begin{pmatrix} -x \\ f(x) \end{pmatrix}$$

The convex combination and necessary conditions are as follows:

$$\min_x F_\omega(x) = \min_x -\omega x + (1-\omega) f(x) \text{ and } \frac{\partial F_\omega}{\partial x}(x) = -\omega + (1-\omega)\frac{\partial f}{\partial x}(x) \stackrel{!}{=} 0.$$

If $x \in \mathbb{R} \setminus [-5, 5]$ then $\frac{\partial F_\omega}{\partial x}(x) \neq 0$ and one gets two implicit functions $h_{1/2} : (0, 1) \to \mathbb{R}$ of the type $x = h(\omega)$ satisfying

$$h_1(\omega) = -\sqrt{\frac{\omega}{1-\omega}} - 5 \text{ and } h_2(\omega) = +\sqrt{\frac{\omega}{1-\omega}} + 5$$

Figure 1: Curve based on the implicit functions

If $x \in [-5, 5]$ then the above equation cannot be solved for x. The implicit function here $h_3 : [-5, 5] \to \{0\}$ is of the form $\omega = h_3(x)$ with $h_3(x) \equiv 0$.

Plotting the implicit functions in the f_1/f_2-space results in the curve as presented in figure 1. However, checking the sufficient conditions shows, that the curve defined by the implicit functions is actually "too large".

$$\frac{\partial^2 F_\omega}{\partial x^2}(x) = \begin{cases} 2(1-\omega)(x+5) & < 0 \ \forall x = h_1(\omega) \ \Rightarrow \ \text{maximum} \\ 0 & = 0 \ \forall \omega = h_3(x) \ \Rightarrow \ \text{saddle point} \\ 2(1-\omega)(x-5) & > 0 \ \forall x = h_2(\omega) \ \Rightarrow \ \text{minimum} \end{cases}$$

Hence, only h_2 defines the indifference set. The turning points on which the implicit functions change correspond to points where one of the eigenvalues of the Hessian changes the sign or becomes zero.

3 Curve Continuation

As mentioned earlier, one could simply solve the convex combination for a sufficient number of different values of ω. This will, however, usually result in a very poor discretization of the indifference set [1, 3]. Similar behavior was reproduced with the test example presented later on. This approach is also very cost intensive.

3.1 Predictor-Corrector Methods

Suppose $z^k := h(\omega^k)$ is the k-th point on the indifference set. Without considering singularities because of a change in the signs of the eigenvalues of the Hessian or a change of the active set, a tangential approximate $\hat{z}^{k+1}$ of the next point z^{k+1} on the indifference set is given by

$$\hat{z}^{k+1} := z^k + \frac{\partial}{\partial \omega} h(\omega^k) \left(\omega^{k+1} - \omega^k\right)$$

Hence, $\hat{z}^{k+1}$ is the prediction of z^{k+1} after a step length of $\delta^k := \|\omega^{k+1} - \omega^k\|$. In the correction step, the predicted point $\hat{z}^{k+1}$ can then be used as the starting value of any algorithm used to find z^k in the first place. This makes the predictor-corrector algorithm a very good candidate when mitigating from a single objective design problem into the multi-objective framework.

Figure 2: The Predictor-Corrector Algorithm

Again, the implicit function theorem provides a way to calculate the necessary derivative $\frac{\partial}{\partial\omega} h(\omega^k)$:

$$\frac{\partial h}{\partial \omega}(\omega^k) = -\left[\frac{\partial H}{\partial z}(z^k, \omega^k)\right]^{-1} \frac{\partial H}{\partial \omega}(z^k, \omega^k).$$

If * denotes only those components that are in the current active set, one gets in more detail:

$$\frac{\partial H^*}{\partial \omega}(x, \lambda, \nu, \omega) = \begin{bmatrix} [\nabla_x f_j(x) - \nabla_x f_m(x)]_{j=1,\dots,m-1} \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{(n+m_1+m_2^*)\times(m-1)}$$

$$\frac{\partial H^*}{\partial z}(x, \lambda, \nu, \omega) = \begin{bmatrix} \mathrm{Hess}_x \mathcal{L}(x, \lambda, \nu, \omega) & -\left[\frac{\partial c}{\partial x}(x)\right]^T & -\left[\frac{\partial g^*}{\partial x}(x)\right]^T \\ \frac{\partial c}{\partial x}(x) & 0 & 0 \\ \frac{\partial g^*}{\partial x}(x) & 0 & 0 \end{bmatrix}$$

Also note, that the expression $\frac{\partial H}{\partial z}$ corresponds to the system matrix of the quadratic subproblem when using SQP-methods for finding the original point z^k, which suggests this approach to be a very good candidate for expanding SQP based single objective solvers. For a detailed pseudo code formulation see [15].

One problem still remains: The original solver has to be adapted to now solve the parameter based convex combination of the objective function and the equality constraints have to be removed. In the case here, the existing One-Shot SQP method for solving "minimize drag, keep lift constant" must be adapted to "minimize the convex combination, no constraints". However, by a certain reparametrization, one can even further reduce the modifications to the existing solver without sacrificing the advantages of the predictor step. Basically, the weights ω can be considered the adjoint variables of a certain constraint optimization problem:

3.2 Reparametrization

Consider the following optimization problem:

$$\begin{aligned} & \min f_l(x) \\ \text{s.t.} \quad & f_k(x) = \hat{f}_k, \ \forall k \in \{1, \dots, m\} \setminus \{l\} \\ & c(x) = 0 \\ & g(x) \geq 0. \end{aligned}$$

That is, instead of moving on to the minimization of the convex combination of the individual objective functions, one stays in the original problem formulation and parametrizes the indifference set not by the value of ω but by the value of the constrains $\hat{f}_k$. A comparison of the necessary optimality conditions shows that both formulations are equivalent:

$$\begin{aligned} \sum_{j=1}^{m-1} \omega_j \nabla f_j(x) + \left(1 - \sum_{j=1}^{m-1} \omega_j\right) \nabla f_m(x) - \left[\frac{\partial c(x)}{\partial x}\right]^T \lambda - \left[\frac{\partial g(x)}{\partial x}\right]^T \mu & = 0 \\ \Leftrightarrow \nabla f_l(x) - \sum_{\substack{j=1 \\ j \neq l}}^{m-1} \frac{-\omega_j}{\omega_l} \nabla f_j(x) - \frac{-\left(1 - \sum_{j=1}^{m-1} \omega_j\right)}{\omega_l} \nabla f_m(x) - \left[\frac{\partial c(x)}{\partial x}\right]^T \lambda & \\ - \left[\frac{\partial g(x)}{\partial x}\right]^T \mu & = 0 \end{aligned}$$

As one can see, the reparametrization roughly means a division by ω_l and the original weights become - slightly transformed - the adjoint variables.

Further comparisons reveal:

- A parametrization using the convex combination depends linearly on ω. This linearity is usually lost after the reparametrization.
- When considering the convex combination, the parameter ω will enter the Hessian, i.e. the system matrix of any SQP method, thus making more changes necessary than using the reparametrization. Here, the values of $\hat{f}_k$ will only enter the right hand side, which means fewer changes. Also note, that the computation of the adjoint variables need not to be touched since the correspondence between ω and the values of $\hat{f}_k$ is irrelevant. This also simplifies the computation of the tangential in the predictor step as discussed later.
- When using the convex combination, the intervals of ω are relatively clear, whereas now, possible starting values of $\hat{f}_k$ are more difficult to find. This is less limiting from a practical point of view, where a designer can provide an approximation, but a bad choice can now lead to the problem being unsolvable.

Again, the implicit function theorem states:

$$\left[\frac{\partial H}{\partial z}(h(\hat{f}_k), \hat{f}_k)\right] \frac{\partial h}{\partial \hat{f}_k}(\hat{f}_k) = -\frac{\partial H}{\partial \hat{f}_k}(h(\hat{f}_k), \hat{f}_k)$$

where

$$\frac{\partial H}{\partial \hat{f}}(x,\lambda,\mu,\hat{f}) = \begin{bmatrix} 0 & 0 & \dots & 0 \\ -1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -1 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

Thus the gradient with respect to $\hat{f}_k$ can easily be computed by one additional solution of the SQP-system where the right hand side consists of a 1 in the $(n+k)$-th component:

$$\begin{bmatrix} \mathrm{Hess}_x \mathcal{L} & -\left[\frac{\partial f_{2,\dots,m}}{\partial x}\right]^T & -\left[\frac{\partial c}{\partial x}\right]^T & -\left[\frac{\partial g^*}{\partial x}\right]^T \\ \frac{\partial f_{2,\dots,m}}{\partial x} & 0 & 0 & 0 \\ \frac{\partial c}{\partial x} & 0 & 0 & 0 \\ \frac{\partial g^*}{\partial x} & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \frac{\partial x}{\partial \hat{f}_k} \\ \frac{\partial \lambda_{f_2,\dots,f_m}}{\partial \hat{f}_k} \\ \frac{\partial \lambda_c}{\partial \hat{f}_k} \\ \frac{\partial \mu}{\partial \hat{f}_k} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

The knowledge of this tangential can now be used twofold:

- The increments of the parametrization can be based on the slope of the Pareto-curve, which leads to a better discretization.
- The starting value of the corrector iteration can be extrapolated based on this information, hopefully accelerating convergence.

Before moving on to the Pareto-curve for the RAE2822 airfoil, the predictor-corrector method based on both parametrizations has been tested on the academic example presented in [8].

4 An Academic Testcase

The following academic testcase was chosen because the composition of the Hessian is similar to the Hessian of the RAE2822 optimization [8].

Suppose $\alpha > 0$ a regularization and $\vec{n}$ the outer normal. The following functions are defined on the region $\Omega := [0,1]^2$:

$$f_1(\varphi, q) := \int\limits_{y=1} \left(\frac{\partial \varphi}{\partial \vec{n}}(x,1) - g(x)\right)^2 dx + \alpha \int\limits_{y=1} \left(q^2(x) + \left(\frac{\partial q}{\partial x}(x)\right)^2\right) dx$$

$$f_2(\varphi, q) := \int\limits_{\Omega} \left(\varphi^2(x,y)\right) dxdy$$

Consider the multi-objective design problem:

$$\begin{array}{llll} \min\limits_{(\varphi,q)} \begin{pmatrix} f_1(\varphi,q) \\ f_2(\varphi,q) \end{pmatrix} & & & \\ \text{s.t. } \Delta\varphi(x,y) & = 0 & \text{in } \Omega & \\ \varphi(x,1) & = q(x) & \text{on } y=1 & \\ \varphi(x,0) & = \varphi_0(x) & \text{on } y=0 & \\ \varphi(0,y) & = \varphi(1,y) & & \text{(periodicy)} \end{array}$$

The necessary optimality conditions for the parametrization according to the convex combination are:

“Costate/Adjoint” $\frac{d\mathcal{L}}{d\varphi}$:

$$\begin{array}{rcll} 2\omega\frac{\partial\varphi}{\partial\vec{n}}(.,1) - 2\omega g(.) + \lambda_1(.,1) & = & 0 & \text{on } y=1 \\ 2(1-\omega)\varphi + \Delta\lambda_1 & = & 0 & \text{in } \Omega \\ \lambda_1 & = & 0 & \text{on } y=0 \\ \lambda_2 & = & \frac{\partial\lambda_1}{\partial\vec{n}} & \text{on } y=1 \\ \lambda_3 & = & \frac{\partial\lambda_1}{\partial\vec{n}} & \text{on } y=0 \end{array}$$

“Design” $\frac{d\mathcal{L}}{dq}$:

$$2\omega\alpha(q-\ddot{q}) = \lambda_2 \quad = \quad \frac{\partial\lambda_1}{\partial\vec{n}} \quad \text{on } y=1$$

“State” $\frac{d\mathcal{L}}{d\lambda}$:

$$\begin{array}{rcll} \Delta\varphi & = & 0 & \text{in } \Omega \\ \varphi(.,1) & = & q & \text{on } y=1 \\ \varphi(.,0) & = & \varphi_0 & \text{on } y=0 \end{array}$$

These equations have then been discretized using finite differences and solved for 250 equidistant values for ω on a 20x21 grid without using any information on the tangential. The resulting curve is shown in figure 3. One can clearly see the insufficient discretization

Figure 3: 20x21 grid, 250 equidistant steps in ω

of the curve. Although the increments in ω were constant, the resulting points on the curve are clustered in the upper region. For more details on this behavior see [1].

When reparametrizing the curve based on the values of the constrains, one has to decide whether f_1 or f_2 will be the parametrizing constraint. According to this decision, either the factor $\nu_2 := \omega$ or the factor $\nu_1 := (1-\omega)$ becomes the adjoint variable for the new parametrizing state constraint, e.g. a parametrization according to f_1 would add ν_1 as an additional degree of freedom to the necessary optimality condition and simultaneously would set $\nu_2 \equiv 1$. Keeping this in mind, the new optimality conditions are:

$$
\begin{array}{rcll}
\multicolumn{4}{l}{\text{“Costate/Adjoint” } \frac{d\mathcal{L}}{d\varphi}:} \\
2\nu_2 \frac{\partial\varphi}{\partial\vec{n}}(.,1) - 2\nu_2 g(.) + \lambda_1(.,1) & = & 0 & \text{on } y=1 \\
2\nu_1\varphi + \Delta\lambda_1 & = & 0 & \text{in } \Omega \\
\lambda_1 & = & 0 & \text{on } y=0 \\
\lambda_2 & = & \frac{\partial\lambda_1}{\partial\vec{n}} & \text{on } y=1 \\
\lambda_3 & = & \frac{\partial\lambda_1}{\partial\vec{n}} & \text{on } y=0 \\
\multicolumn{4}{l}{\text{“Design” } \frac{d\mathcal{L}}{dq}:} \\
2\nu_2\alpha(q-\ddot{q}) = \lambda_2 & = & \frac{\partial\lambda_1}{\partial\vec{n}} & \text{on } y=1 \\
\multicolumn{4}{l}{\text{“State” } \frac{d\mathcal{L}}{d\lambda}:} \\
\Delta\varphi & = & 0 & \text{in } \Omega \\
\varphi(.,1) & = & q & \text{on } y=1 \\
\varphi(.,0) & = & \varphi_0 & \text{on } y=0 \\
f_1(\varphi,q) = \langle\varphi|\varphi\rangle & = & \hat{f}_1 & \multirow{2}{*}{$\Big\}$ either / or} \\
f_2(\varphi,q) & = & \hat{f}_2 &
\end{array}
$$

This system has again been discretized using finite differences and was then solved via SQP. However, instead of fixed increments for the parametrization, information on the tangential has been used:

$$\hat{f}_k^{i+1} = \hat{f}_k^i + \delta \frac{\partial f_l}{\partial \hat{f}_k}(\hat{f}_k^i).$$

Here, δ denotes the step length, i is the iteration counter, k denotes any of the two functions used as parametrization, and l denotes the function used as optimization objective. The

Figure 4: 20x21 grid, 86 steps based on the tangent

resulting curve can be seen in figure 4. The chosen value for δ resulted in 86 points on the curve. Comparing these two curves, the vast improvements in the discretization become evident. Note that some possible slight discrepancies in those two curves are the result of changing the approximation of the integrals.

The other point of interest was the reduction of SQP iterations needed in the corrector step by extrapolating the starting solution for the next point on the curve. Note that the calculation of the gradient of the Pareto-curve requires solving the SQP-system matrix once. Thus, to achieve a true reduction in the numerical effort, an extrapolation based on gradient information needs to reduce the corrector steps required by at least two. Unsurprisingly, this could not be achieved in the testcase here. However, a reduction by one - from five to four SQP iterations needed - was achieved. Thus, the improvement in the discretization of the curve by using gradient information can be more or less considered numerically free of cost.

Before moving on to the Pareto-curve for the RAE2822 airfoil, the transferability of this academic testcase has to be considered: The distinct advantage here is the knowledge of the exact Hessian. In most practical applications, the exact Hessian is not known but instead approximated by some update-formulas like BFGS-updates. Although approximative SQP methods still converge to the correct solution, the usefullness of a tangential on the Pareto-Curve based on an approximate Hessian is not clear. This potential problem that was also observed by [14].

5 RAE2822 Application

The starting point here is the One-Shot optimization of the RAE2822 airfoil [6, 8, 9, 10, 16]. The problem formulation is as follows:

$$\begin{aligned} &\min_q C_D(\varphi(q), q) \\ \text{s.t.}\quad & c(\varphi(q), q) = 0 \\ & C_L(\varphi(q), q) - l_0 = 0 \end{aligned}$$

where φ denotes the states, i.e. the flow solution, and q designates the control, i.e. the parametrization of the shape of the airfoil based on Hicks-Henne functions. Additionally, $c(\varphi(q), q) = 0$ are the Euler Equations and C_D is the drag coefficient. C_L denotes the lift coefficient. The flow states φ are calculated by FLOWer, a very robust pseudo-timestepping flow solver, courtesy of the German Aerospace Center (DLR) and the gradients are computed by the adjoint approach [5].

5.1 The Pareto-Curve

The basic idea behind One-Shot is to merge the optimization with the flow solver by exploiting the separability framework. Conceptually similar to a "black-box approach", the optimization is coupled with FLOWer's pseudo-timestepping iteration, resulting in a very fast routine with fewest modifications to the original flow solver possible. Theoretical background for this procedure is the quadratic subproblem of the reduced SQP method in pseudo-time formulation:

$$\begin{aligned} &\min_{\dot q} \tfrac{1}{2}\dot q^T B \dot q + g_c^T \dot q \\ \text{s.t.}\quad & g_{C_L}^T \dot q = -C_L + \frac{\partial C_L}{\partial \varphi}\left[\frac{\partial c}{\partial \varphi}\right]^{-1} c \end{aligned}$$

where g denotes the reduced gradient and B is the reduced Hessian. For more details see any of the sources mentioned above.

The aim here is to expand the One-Shot routine further to compute the complete indifference set for the RAE2822 airfoil, i.e. to integrate the One-Shot optimization as the corrector step of a predictor-corrector curve continuation. As discussed above, a parametrization of the Pareto-curve based on the lift constraint seems most promising. Due to the "black-box approach", meaning only the control q is visible to the optimization, the calculation of the tangential to the Pareto-curve with respect to the flow states φ would not have been possible without major changes to the flow solver, thus violating one of the design goals of One-Shot. Therefore, the predictor-step was omitted in the sense of being constant. To further gauge the performance of One-Shot, the same Pareto-curve was also computed using a projected gradient based corrector step. The resulting curve is shown in figure 5 and the corresponding optimal shapes are shown in figure 6. The slightly odd inward bend of the low-lift profile is

Figure 5: Pareto-curve for the RAE2822 optimization

Figure 6: Optimal shapes

due to shortcomings of the Hicks-Henne parametrization that was not meant for such drastic deformations of the original RAE2822 profile, thus avoiding dealing with the non-smooth trailing edge of the airfoil. The slight deviations between the curves are due to the approximative nature of some values in the One-Shot approach. It should also be mentioned, that by using the Euler Equations, the resulting drag is solely inducted by the pressure shock, so one has to consider the curve being influenced by FLOWer's numerical viscosity.

5.2 Computational Effort

When considering the numerical cost for finding the whole Pareto-curve, the advantages of the One-Shot-corrector become obvious. Assuming that FLOWer needs 150 timesteppings for full convergence, the cost for the pure simulation of the curve is as follows:

- Projected Gradient:
  60 points $\times 150 \frac{\text{timesteppings}}{\text{point}} = 9,000$ timesteppings. Additionally, two adjoints (one for the gradient with respect to lift, and one for the gradient with respect to drag) must be computed, totaling to $27,000$ timesteppings.
- One-Shot:
  45 points $\times$ $150 \frac{\text{timesteppings}}{\text{point}} = 6,750$ timesteppings. With both adjoints totaling to $20,250$ timesteppings.

To judge the effectiveness of the One-Shot and the projected gradient based corrector, the cost of the optimization is now compared to the cost of the simulation alone:

Numerical Costs Projected Gradient and One-Shot

	Iterations total	Timesteppings per Iteration	Correction Timesteppings per Point, no Optimization	Relative Cost
Gradient	14.332	210	0	33,44
One-Shot	10.292	3	600	2,86

The correction timesteppings (3rd column) are needed to increase the accuracy of the approximative nature of One-Shot, thereby making the results compareable to the gradient based method. The table once again shows the effectiveness of the One-Shot approach compared to the projected gradient based method. According to [10], the relativistic cost of one One-Shot optimization is less than 7 times of the simulation cost alone. When comparing to the less than 3 times in the Pareto-case here, the effectiveness of the predictor-corrector step becomes clear, especially when one considers the constant predictor step used.

6 Conclusions

Theoretically, the Pareto-curve is the implicit function of the optimality conditions. The practicability of extending existing solvers in order to compute the Pareto-curve has been shown. The predictor-corrector approach seems best suited for this endeavor in the 2-D case and different single objective optimizer can easily be used in the corrector-step. Newton-based solvers (SQP) are preferable because the system-matrix can be reused unchanged to compute the tangent for the predictor-step when parametrizing according to the value of the constraints. Even with a naive, i.e. constant, predictor, the increase of the numerical cost efficiency when using a predictor-corrector approach is easily measurable.

References

[1] I. Das and J.E. Dennis, A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems, *Structural Optimization* 14 (1997) 63–69.

[2] I. Das and J.E. Dennis, Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems, *SIAM J. Optim.* 8 (1998) 631–657.

[3] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, Wiley, Chichester UK, 2001.

[4] O. Forster, *Analysis 2*, Friedr. Vieweg & Sohn, Braunschweig / Wiesbaden, 1982.

[5] N.R. Gauger, *Das Adjungiertenverfahren in der aerodynamischen Formoptimierung*, Forschungsbericht Deutsches Zentrum für Luft- und Raumfahrt e.V., 2003.

[6] I. Gherman and V. Schulz, Preconditioning of one-shot Pseudo-timestepping methods for shape optimization, *PAMM* 5 (2005) 741–742.

[7] J. Guddat, Parametric optimization: pivoting and predictor-corrector continuation, a survey, in *Parametric Optimization and Related Topics*, Akademie-Verlag, Berlin.

[8] S.B. Hazra and V. Schulz, Simultaneous Pseudo-timestepping for PDE-model based optimization problems, *BIT Numerical Mathematics* (2004) 457–472.

[9] S.B. Hazra and V. Schulz, How to profit from adjoints in one-shot pseudotime-stepping optimization, *Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial Problems, EUROGEN*, 2005.

[10] S.B. Hazra and V. Schulz, Simultaneous Pseudo-timestepping for aerodynamic shape optimization problems with state constraints, *SIAM J. Sci. Comput.* 2006.

[11] J. Koski, Multicriteria truss optimization, in *Multicriteria Optimization in Engineering and in the Sciences*, (ed.) W. Stadler, Plenum Press, New York, 1988

[12] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer, New York, 1999.

[13] J. Rakowska and R.T. Haftka and L.T. Watson, Tracing the efficient curve for multi-objective control-structure optimization, *Computing Systems in Engineering* 2 (1991) 461–471.

[14] J.R.J. Rao and P.Y. Papalambros, A nonlinear programming continuation strategy for one parameter design optimization problems, in *Design Automation Conference American Society of Mechanical Engineers*, 1989.

[15] J.R.J. Rao and P.Y. Papalambros, Extremal behavior of one parameter families of optimal design models, submitted to the ASME Design Automation Conf., 1989.

[16] V. Schulz, Reduced SQP methods for large-scale optimal control problems in DAE with application to path planning problems for satellite mounted robots, Universität Heidelberg, 1996.

[17] P.K. Shukla and K. Deb, *On Finding Multiple Pareto-Optimal Solutions Using Classical and Evolutionary Generating Methods*, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology, Kanpur.

[18] G. Timmel, Ein stochastisches Suchverfahren zur Bestimmung der optimalen Kompromißlösungen bei statischen polykriterialen Optimierungsaufgaben, *Wiss. Zeitschrift TH Ilmenau* 26 (1980) 159–174.

Manuscript received 11 July 2007
revised 25 September 2007
accepted for publication 27 September 2007

Stephan Schmidt
FB 4 – Department of Mathematics, University of Trier, Germany
E-mail address: Stephan.Schmidt@uni-trier.de

Volker Schulz
FB 4 – Department of Mathematics, University of Trier, Germany
E-mail address: Volker.Schulz@uni-trier.de