Efficient Methods for Aerodynamic Optimization

Ilia Gherman, Volker Schulz

University of Trier, MEGADESIGN-Project

DMV-Jahrestagung
Bonn, September 2006

Universität Trier

Outline

- One-shot approach for unconstrained drag minimization and the choice of the reduced Hessian
- Extending the method to include state constraints
- Numerical results
- Conclusions

The MEGADESIGN-Project

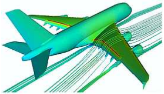
- Supported by German Federal Ministry of Economics and Technology

- Main goal of the project : fast algorithms for geometric design of an aircraft

- Partners:
  - German Aerospace Center (DLR)
  - Airbus Germany

    ⋮

    ⋮
  - University of Trier – Group of Volker Schulz



Research goal in Trier

Question

Is there a fast numerical approach for drag minimization with low relative complexity ?

Research goal in Trier

Question

Is there a fast numerical approach for drag minimization with low relative complexity ?

Difficulty

Constructing an optimization method from scratch is not viable

Universität Trier

Research goal in Trier

Question

Is there a fast numerical approach for drag minimization with low relative complexity ?

Difficulty

Constructing an optimization method from scratch is not viable

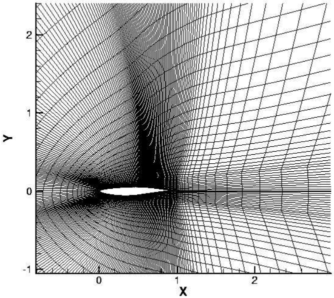
Aim

- Construct an optimization algorithm using existing simulation tools
- Overall effort: constant×simulation effort

Universität Trier

Optimization Problem without Addititional Constraints

$$\min \quad J(u,q)$$
$$\text{subject to} \quad c(u,q) = 0$$



J: the cost function (drag)
c: Euler-flow equations
u: state variable such that the Jacobian C_u is invertible
q: wing profile, parameterized by splines

Black-Box-Methods

Implicit function theorem:

$$\exists\, u : \mathbb{R}^{n_q} \to \mathbb{R}^{n_u}, q \mapsto u(q) \;:\; \forall\, q \;:\; c(u(q), q) = 0.$$

Black-Box-Methods

Implicit function theorem:

$$\exists\, u : \mathbb{R}^{n_q} \to \mathbb{R}^{n_u}, q \mapsto u(q) \;:\; \forall\, q \;:\; c(u(q), q) = 0.$$

Reduce the problem

$$\begin{aligned} \min \quad & J(u, q) \\ \text{subject to} \quad & c(u, q) = 0 \end{aligned}$$

to the unconstrained problem

$$\min \quad J(u(q), q) =: I(q)$$

Reduced Gradient via Adjoint Problem

Need to compute the reduced gradient

$$\nabla_q I = -C_q^T C_u^{-T} \nabla_u J + \nabla_q J$$

Reduced Gradient via Adjoint Problem

Need to compute the reduced gradient

$$\nabla_q I = -C_q^T C_u^{-T} \nabla_u J + \nabla_q J$$

sensitivity approach

$$C_q^T C_u^{-T} = \left(C_u^{-1} C_q\right)^T$$

Universität Trier

Reduced Gradient via Adjoint Problem

Need to compute the reduced gradient

$$\nabla_q I = -\underbrace{C_q^T C_u^{-T}}\nabla_u J + \nabla_q J$$

sensitivity approach

$$C_q^T C_u^{-T} = \left(C_u^{-1} C_q\right)^T$$

adjoint approach

$$C_u^T \lambda = \nabla_u J$$

One-Shot-Approach

Black-box steepest descent method:

- Solve the flow equations (exactly)
- Solve the adjoint equation (exactly) and compute the (exact) reduced gradient based on the adjoint approach
- Update design

One-Shot-Approach

Black-box steepest descent method:

- Solve the flow equations (exactly)
- Solve the adjoint equation (exactly) and compute the (exact) reduced gradient based on the adjoint approach
- Update design

The idea of the one-shot-method

- Based on neccessary optimality conditions
- Solve all three equations simultaneously

One-Shot-Method (1)

Define the Lagrangian functional:

$$\mathcal{L}(u, q, \lambda) = J(u, q) + \lambda^* c(u, q).$$

One-Shot-Method (1)

Define the Lagrangian functional:

$$\mathcal{L}(u, q, \lambda) = J(u, q) + \lambda^* c(u, q).$$

Neccessary optimality conditions:

$$\begin{pmatrix} \nabla_u \mathcal{L} \\ \nabla_q \mathcal{L} \\ c \end{pmatrix} = 0 \qquad \begin{array}{l} \leftarrow \text{ Adjoint equation} \\ \leftarrow \text{ Design equation} \\ \leftarrow \text{ State equation} \end{array}$$

One-Shot-Method (1)

Define the Lagrangian functional:

$$\mathcal{L}(u, q, \lambda) = J(u, q) + \lambda^* c(u, q).$$

Neccessary optimality conditions:

$$\begin{pmatrix} \nabla_u \mathcal{L} \\ \nabla_q \mathcal{L} \\ c \end{pmatrix} = 0 \qquad \begin{matrix} \leftarrow \text{ Adjoint equation} \\ \leftarrow \text{ Design equation} \\ \leftarrow \text{ State equation} \end{matrix}$$

Use Newton method to solve this system of nonlinear equations!

Universität Trier

One-Shot-Method (2)

Newton-iteration uses KKT-Matrix:

$$\begin{bmatrix} H_{uu} & H_{uq} & C_u^* \\ H_{qu} & H_{qq} & C_q^* \\ C_u & C_q & 0 \end{bmatrix} \begin{pmatrix} \Delta u \\ \Delta q \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} -\nabla_u \mathcal{L} \\ -\nabla_q \mathcal{L} \\ -c \end{pmatrix}$$

One-Shot-Method (2)

Newton-iteration uses KKT-Matrix:

$$\begin{bmatrix} H_{uu} & H_{uq} & C_u^* \\ H_{qu} & H_{qq} & C_q^* \\ C_u & C_q & 0 \end{bmatrix} \begin{pmatrix} \Delta u \\ \Delta q \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} -\nabla_u \mathcal{L} \\ -\nabla_q \mathcal{L} \\ -c \end{pmatrix}$$

KKT-Matrix approximated by rSQP-matrix:

$$\begin{bmatrix} 0 & 0 & A^* \\ 0 & \color{red}{B} & C_q^* \\ A & C_q & 0 \end{bmatrix} \begin{pmatrix} \Delta u \\ \Delta q \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} -\nabla_u \mathcal{L} \\ -\nabla_q \mathcal{L} \\ -c \end{pmatrix}$$

where A is some approximation of C_u.

Universität Trier

Appropriate choice of B

- Exact reduced Hessian

$$B_{\text{ex}} = \begin{bmatrix} -C_u^{-1} C_q \\ I \end{bmatrix}^T \begin{bmatrix} H_{uu} & H_{uq} \\ H_{qu} & H_{qq} \end{bmatrix} \begin{bmatrix} -C_u^{-1} C_q \\ I \end{bmatrix}$$

- “wrong” reduced Hessian

$$B_{\text{inex}} = \begin{bmatrix} -A^{-1} C_q \\ I \end{bmatrix}^T \begin{bmatrix} H_{uu} & H_{uq} \\ H_{qu} & H_{qq} \end{bmatrix} \begin{bmatrix} -A^{-1} C_q \\ I \end{bmatrix}$$

(similar to Bank/Welfert/Yserentant 1990)

- B according to Griewank’s piggy-back concept.

Universität Trier

Theoretical Investigations — Definition of the Quadratic Problem

Quadratic problem (QP):

$$\min_{u,q} \; \frac{1}{2}u^T H_{uu} u + \frac{1}{2}q^T H_{qq} q + f_u^T u + f_q^T q$$
$$\text{subject to} \quad C_u u + C_q q + c = 0.$$

with C_u invertible. Consider the Lagrangian:

$$\mathcal{L}(u, q, \lambda) = \frac{1}{2}u^T H_{uu} u + \frac{1}{2}q^T H_{qq} q + f_u^T u + f_q^T q + \lambda^T (C_u u + C_q q + c).$$

Universität Trier

Theoretical Investigations — Convergence Results

Theorem (Kunisch/Ito/Schulz/Gherman 2006)
There exists an $\eta > 0$, such that the iteration

$$\begin{pmatrix} u^{k+1} \\ q^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} u^{k} \\ q^{k} \\ \lambda^{k} \end{pmatrix} - \begin{bmatrix} 0 & 0 & A^T \\ 0 & B & C_q^T \\ A & C_q & 0 \end{bmatrix}^{-1} \begin{pmatrix} \nabla_u \mathcal{L} \\ \nabla_q \mathcal{L} \\ \nabla_\lambda \mathcal{L} \end{pmatrix}$$

converges to the solution of the (QP), provided

$$\max\{\rho(I - A^{-1} C_u),\ \rho(I - B^{-1} B_{inex})\} < \eta$$

and C_u symmetric.
Proof: Nilpotency of degree 3 of the iteration matrix and perturbation analysis.

Drag Minimization of an RAE 2822 Airfoil

- Flow equations: Euler flow
- Flow-Solver: FLOWer, provided by DLR in forward and adjoint mode (Gauger,. . .)
- Minimize drag (constant profile thickness is preserved in the parameterization of the airfoil)
- Technique employed per iteration:
  - State/Adjoint: single iteration-steps provided by FLOWer
  - Design: rSQP-similar step:

$$\Delta q = -B^{-1} \cdot \gamma_d^k \qquad \text{where } \gamma_d^k = C_q^* (A^*)^{-1} J_u^*,$$

γ_d^k is the current reduced gradient approximation [Hazra/Schulz/Brezillon/Gauger 2005] and B approximates the “wrong” reduced Hessian (BFGS-Updates based on γ_d).

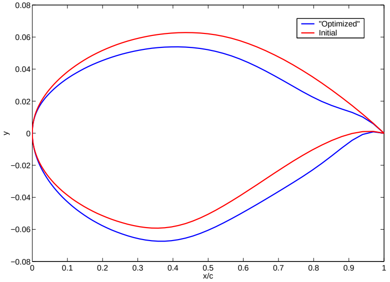
Universität Trier

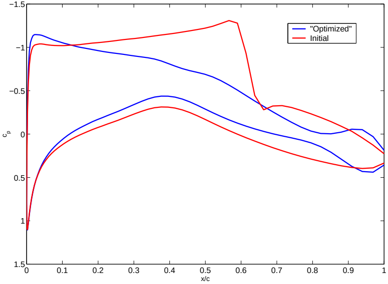
Results – Unconstrained Optimization

- Fast convergence (total effort < 4 simulations)
- Drastic reduction of the drag . . .

Results – Unconstrained Optimization

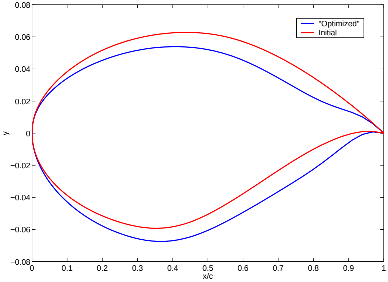
- Fast convergence (total effort < 4 simulations)
- Drastic reduction of the drag . . . but also almost total lost of lift and pitching moment

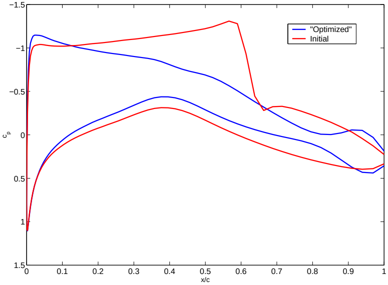




Results – Unconstrained Optimization

- Fast convergence (total effort < 4 simulations)
- Drastic reduction of the drag . . . but also almost total lost of lift and pitching moment





⟶ Neccessary: explicit formulation of aerodynamic constraints

Universität Trier

State Constraints

$$\begin{aligned} \min \quad & J(u, q) \\ \text{subject to} \quad & c(u, q) = 0 \\ & \ell(u, q) \geqslant 0 \qquad \longleftarrow \quad \text{scalar Lift constraint} \end{aligned}$$

Lift depends also on the states and design variables.
The reduced gradient w.r.t lift

$$\gamma_\ell = \frac{d\,\ell}{d\,q}$$

can be computed by the solution of yet another adjoint problem:

$$\gamma_\ell = \nabla_q \ell - C_q^* (A^*)^{-1} \nabla_u \ell$$

Universität Trier

State Constraints

$$\begin{aligned} \min \quad & J(u, q) \\ \text{subject to} \quad & c(u, q) = 0 \\ & \ell(u, q) \geqslant 0 \qquad \longleftarrow \qquad \text{scalar Lift constraint, active} \end{aligned}$$

Lift depends also on the states and design variables.
The reduced gradient w.r.t lift

$$\gamma_\ell = \frac{d\,\ell}{d\,q}$$

can be computed by the solution of yet another adjoint problem:

$$\gamma_\ell = \nabla_q \ell - C_q^* (A^*)^{-1} \nabla_u \ell$$

State Constraints

$$\begin{aligned} \text{min} \quad & J(u, q) \\ \text{subject to} \quad & c(u, q) = 0 \\ & \ell(u, q) = 0 \qquad \leftarrow \quad \text{scalar Lift constraint} \end{aligned}$$

Lift depends also on the states and design variables.
The reduced gradient w.r.t lift

$$\gamma_\ell = \frac{d\,\ell}{d\,q}$$

can be computed by the solution of yet another adjoint problem:

$$\gamma_\ell = \nabla_q \ell - C_q^* (A^*)^{-1} \nabla_u \ell$$

Universität Trier

One-Shot with Additonal State Constraints

New Lagrangian:

$$\mathcal{L}(u, q, \lambda, \mu) = J(u, q) + \lambda^* c(u, q) + \mu \ell(u, q).$$

One-Shot with Additonal State Constraints

New Lagrangian:

$$\mathcal{L}(u, q, \lambda, \mu) = J(u, q) + \lambda^* c(u, q) + \mu\ell(u, q).$$

Iterates from:

$$\begin{bmatrix} 0 & 0 & 0 & A^* \\ 0 & B & \gamma_\ell & C_q^* \\ 0 & \gamma_\ell^* & 0 & 0 \\ A & C_q & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta u \\ \Delta q \\ \Delta \mu \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} -\nabla_u \mathcal{L} \\ -\nabla_q \mathcal{L} \\ -\ell(u, q) \\ -c(u, q) \end{pmatrix}$$

Universität Trier

One-Shot with Additonal State Constraints

New Lagrangian:

$$\mathcal{L}(u, q, \lambda, \mu) = J(u, q) + \lambda^* c(u, q) + \mu \ell(u, q).$$

Iterates from:

$$\begin{bmatrix} 0 & 0 & 0 & A^* \\ 0 & B & \gamma_\ell & C_q^* \\ 0 & \gamma_\ell^* & 0 & 0 \\ A & C_q & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta u \\ \Delta q \\ \Delta \mu \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} -\nabla_u \mathcal{L} \\ -\nabla_q \mathcal{L} \\ -\ell(u, q) \\ -c(u, q) \end{pmatrix}$$

$\longrightarrow$ partially reduced SQP-method (approximate variant)

Universität Trier

Theoretical Investigations — Additional Constraints

Add to the (QP) the constraint

$$h_u^T u + h_q^T q + h = 0.$$

Convergence of the iteration can be justified analogously to the “unconstrained” theorem. Same conditions with additionally

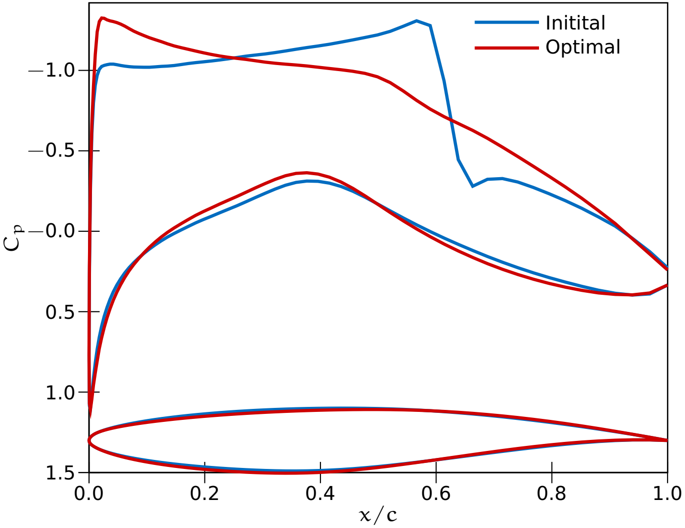
$$\gamma_\ell = h_q - C_q^T A^{-T} h_u.$$

Universität Trier

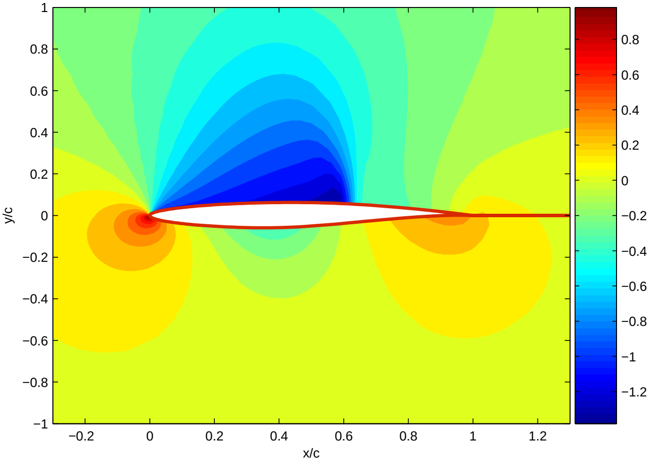
Numerical results (1)

- Minimize drag with constant lift constraint
- Same setting as for the unconstrained optimization
- Reduced gradients w.r.t. drag/lift are computet based on the adjoint solutions after single-iteration steps by `FLOWer`
- Approximations of the reduced Hessian by L-BFGS-updates based on the reduced gradients

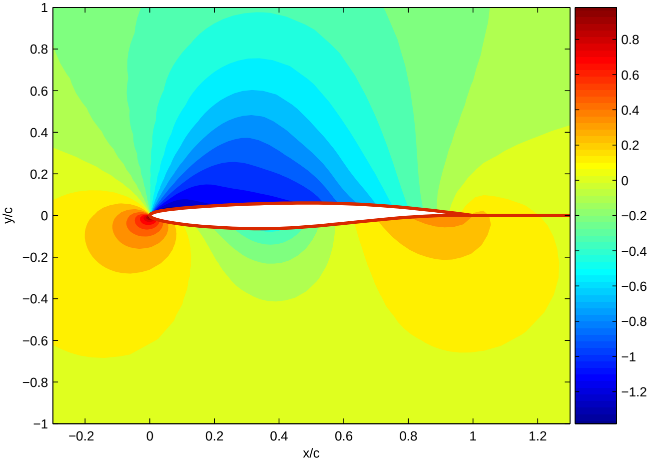
Numerical Results (2)



Numerical Results (3)

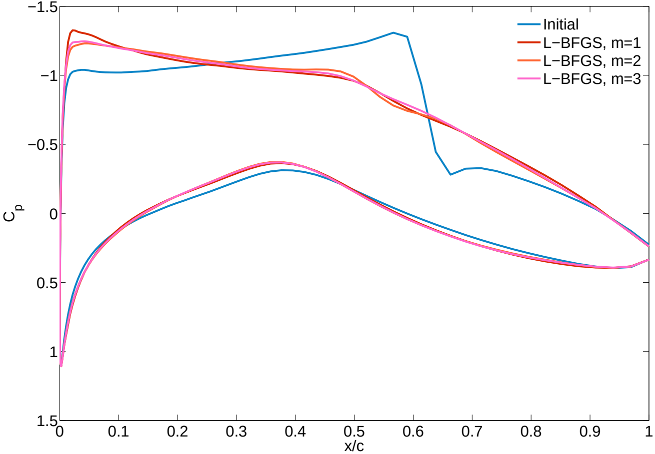


Numerical Results (3)



Numerical Results (4)

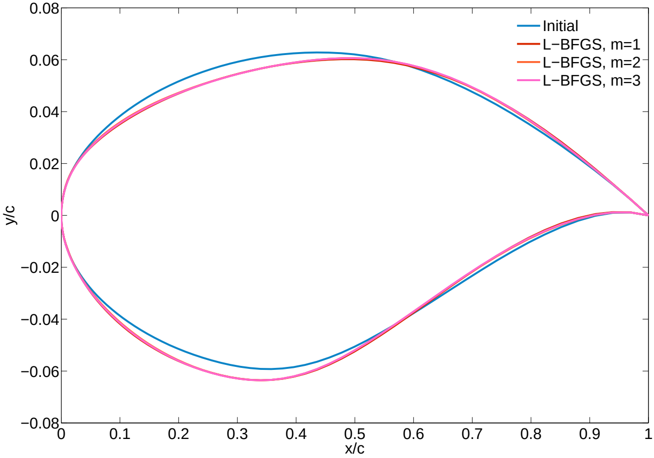
Pressure Coefficient on the surface of the airfoil



Universität Trier

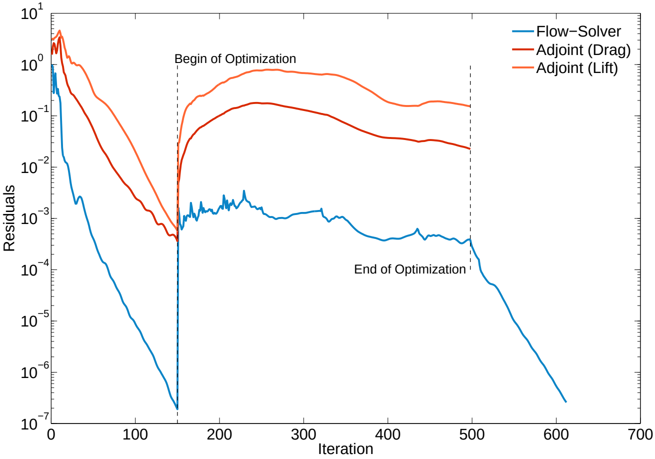
Numerical Results (5)

Airfoils



Numerical Results (6)

L-BFGS-updates with $m = 1$, result: $\frac{\text{optimization effort}}{\text{simulation effort}} < 7$



Universität Trier

Conclusions and Further Research

Conclusions and Further Research

Conclusions

- One-shot optimization by use of approximate partially reduced SQP approach
- Reduced gradient and Hessian approximations should be consistent with the state/costate solver iteration

Universität Trier

Conclusions and Further Research

Conclusions

- One-shot optimization by use of approximate partially reduced SQP approach
- Reduced gradient and Hessian approximations should be consistent with the state/costate solver iteration

Further Research

- Adding viscosity in the flow equations (Navier-Stokes)
- 3D computations
- Models contain parameters with unknown/uncertain values $\Rightarrow$ robust optimization, stochastic approach

Universität Trier