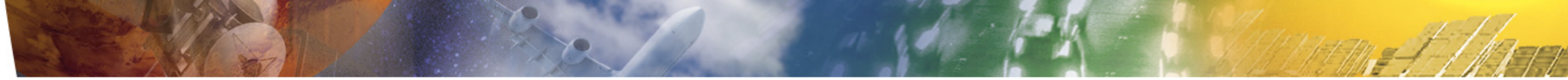
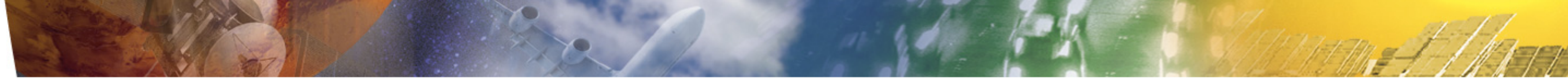


Automated extension of fixed-point PDE solvers for optimal design with bounded retardation

Nico Gauger1),2), Andreas Griewank2)

1) DLR Braunschweig
Institute of Aerodynamics and Flow Technology
Numerical Methods Branch

2) Humboldt University Berlin
Department of Mathematics



Collaborators

With contributions to this talk:

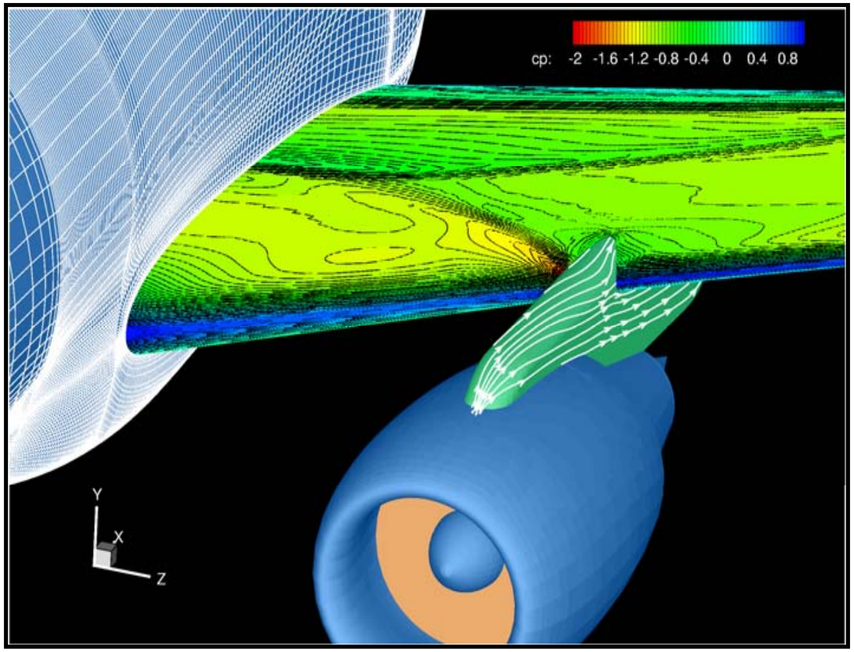
- **HU Berlin:** J. Riehme
- **Fastopt:** R. Giering, Th. Kaminski
- **TU Dresden:** C. Moldenhauer, S. Schlenkrich, A. Walther

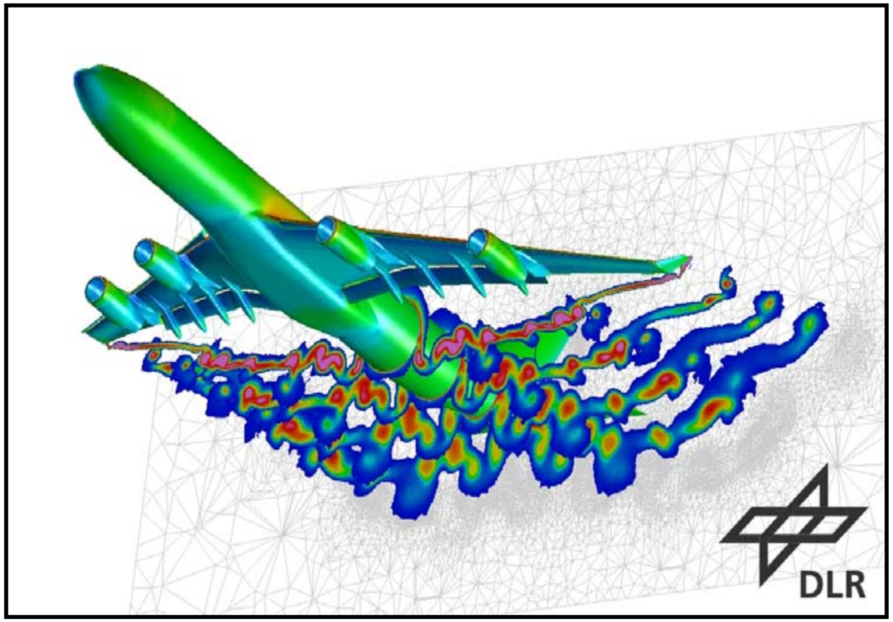
Automated extension of fixed-point PDE solvers for optimal design with bounded retardation

Outline

- MEGAFLOW Software (FLOWer, TAU)
- ADFLOWer
- Automatic differentiation of entire design chain
- All-at-Once / One-Shot / Piggy-Back
- Feasibility study
- Goals

MEGAFLOW Software





Structured RANS solver FLOWer

- block-structured grids
- moderate complex configurations
- fast algorithms (unsteady flows)
- design option
- adjoint option

Unstructured RANS solver TAU

- hybrid grids
- very complex configurations
- grid adaptation
- fully parallel software
- adjoint option

Algorithmic Differentiation (AD)

Work in progress and results

- **ADFLOWer generated with TAF (3D Navier-Stokes, k-w), first verifications and validation**
- **Adjoint version of TAUij (2D Euler) + mesh deformation and parameterization with ADOL-C, validated and tested**
- **First and second derivatives of a “FLOWer-Derivate” (2D Euler) + mesh deformation and parameterization generated with TAPENADE, used for All-at-Once (Piggy-Back)**

ADFLOWer by TAF(*Fast*Opt)

Test configuration

- 2d NACA0012
- k-omega (Wilcox) turbulence model
- cell-centred metric
- 2000 time steps on fine grid
- target sensitivity: d lift/ d alpha

Steps

- Modifications of FLOWer code (TAF Directives, slight recoding, etc...)
- tangent-linear code (verification)
- adjoint code
- efficient adjoint code

Major challenge

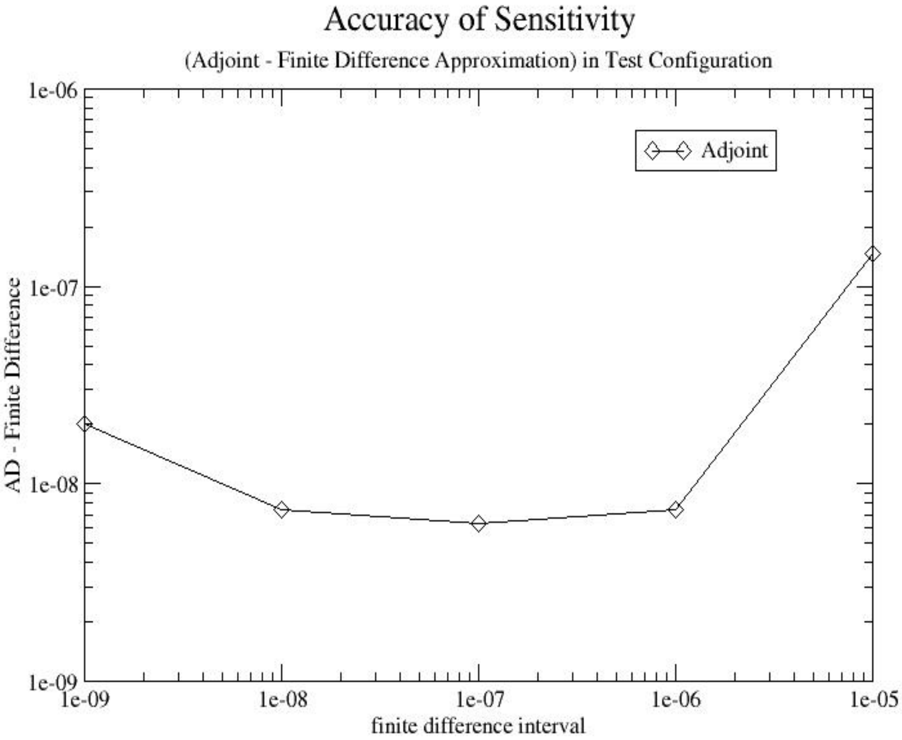
- memory management (all variables in one big field 'variab') complicates detailed analysis and handling of deallocation

(Gauger,Giering,Kaminski)

ADFLOWer

	TAF CPUs	Code lines	solve rel CPU	solve memory
Nominal		166000	1.0	57
tangent	293	268000	3.3	75
adjoint	253	310000	6.3	489

Usually better for larger configurations



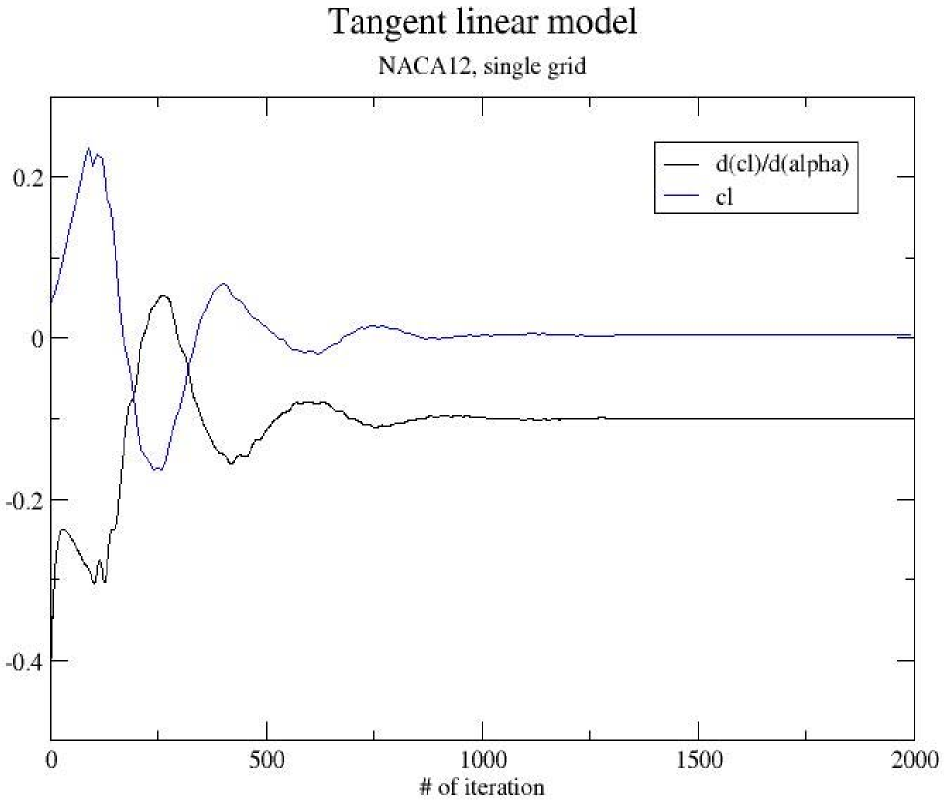
Ma = 0.734
α = 2.8°
Re = 6x10^6
kw turbulence model

(Gauger,Giering,Kaminski)

ADFLOWer

- Demonstrates convergence of discrete sensitivities including turbulence
- Same sensitivity for Navier-Stokes adjoint (Wilcox kw) and tangent linear model

Ma = 0.734
α = 2.8°
Re = 6x10^6
kw turbulence model

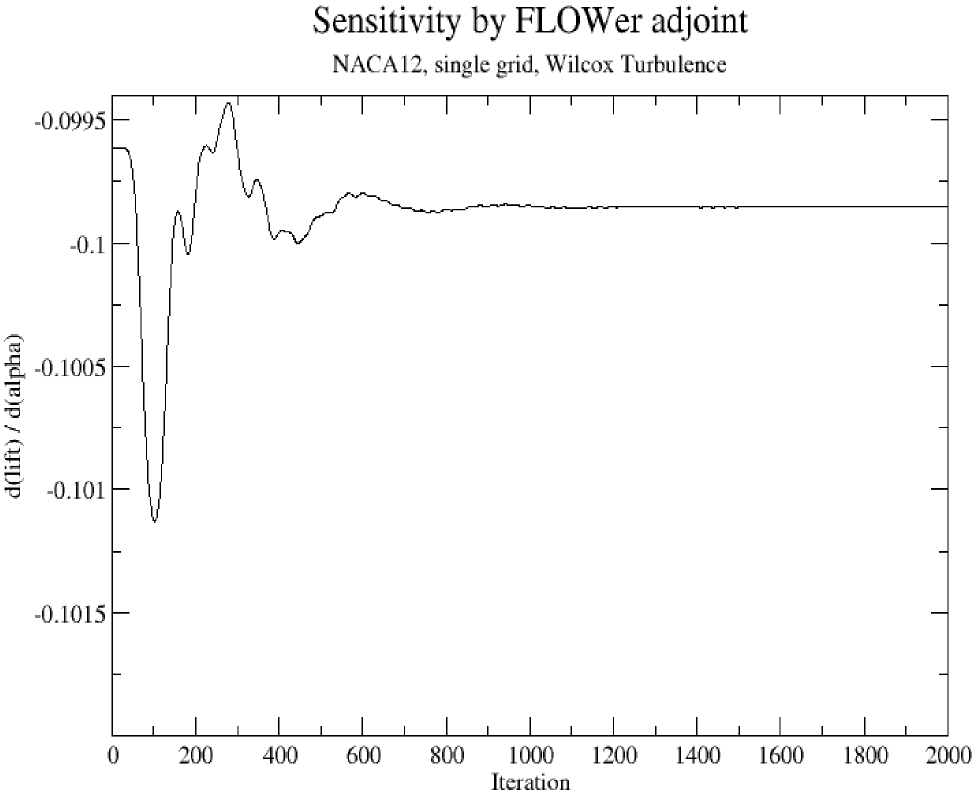


(Gauger,Giering,Kaminski)

ADFLOWer

- Demonstrates convergence of discrete sensitivities including turbulence
- Same sensitivity for Navier-Stokes adjoint (Wilcox kw) and tangent linear model

Ma = 0.734
α = 2.8°
Re = 6x10^6
kw turbulence model



(Gauger,Giering,Kaminski)

Automatic Differentiation of Entire Design Chain

$$\begin{tikzcd}
\text{design vector (P)} \arrow[r] & \text{defgeo} \arrow[r, "x_{new}"] & \text{difgeo} \arrow[r, "dx"] & \text{meshdefo} \arrow[r, "m"] & \text{flow solver} \arrow[r] & C_D \\
& \text{surface grid} \arrow[ur] & \text{grid} \arrow[ur] & & &
\end{tikzcd}$$

- Adjoint version of entire design chain by ADOL-C (TU-Dresden)
- TAUij (2D Euler) + mesh deformation + parameterization

$$\frac{dC_D}{dP} = \underset{\substack{\uparrow \\ \text{TAUij\_AD}}}{\frac{\partial C_D}{\partial m}} \cdot \underset{\substack{\nwarrow \\ \text{meshdefo\_AD}}}{\frac{\partial m}{\partial (dx)}} \cdot \frac{\partial (dx)}{\partial x_{new}} \cdot \underset{\substack{\nwarrow \\ \text{defgeo\_AD}}}{\frac{\partial x_{new}}{\partial P}} \quad \text{and} \quad \frac{\partial (dx)}{\partial x_{new}} = \frac{\partial (x_{new} - x_{old})}{\partial x_{new}} = Id$$

(Gauger,Walther)

Automatic Differentiation of TAUij (fixed-point solver)

- **Run time (2000 fixed-point iterations)**
  - **primal: 2 minutes**
  - **adjoint: 16 minutes**

- **Tape size: 340 MB (reverse accumulation approach!)**

[Christianson in 94]

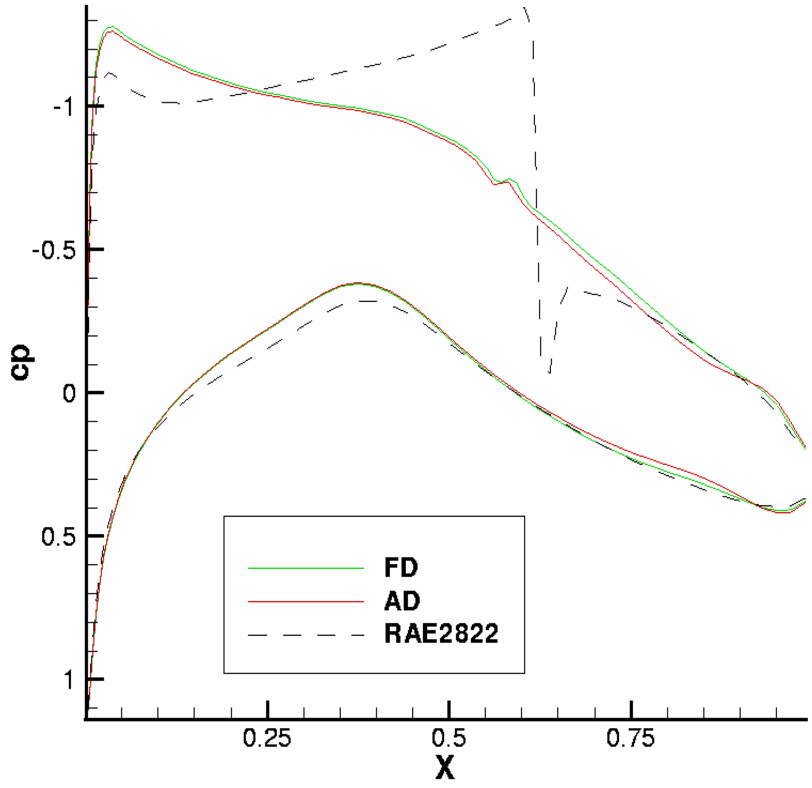
- **Run time memory**
  - **primal: 8 MB**
  - **adjoint: 45 MB**

Deutsches Zentrum
DLR für Luft- und Raumfahrt e.V.
in der Helmholtz-Gemeinschaft

(Schlenkrich,Walther)

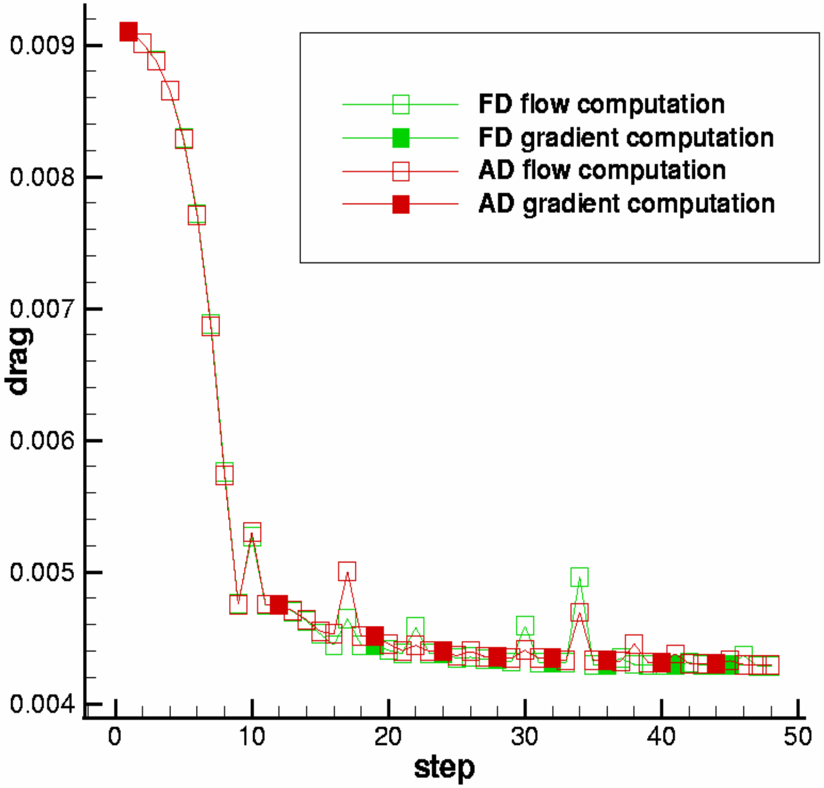
Automatic Differentiation of Entire Design Chain

First Application / Validation:



Drag reduction

- RAE 2822**,** M = 0.73, α = 2.0°
- inviscid flow, mesh 161x33 cells
- 20 design variables (Hicks-Henne)
- steepest descent

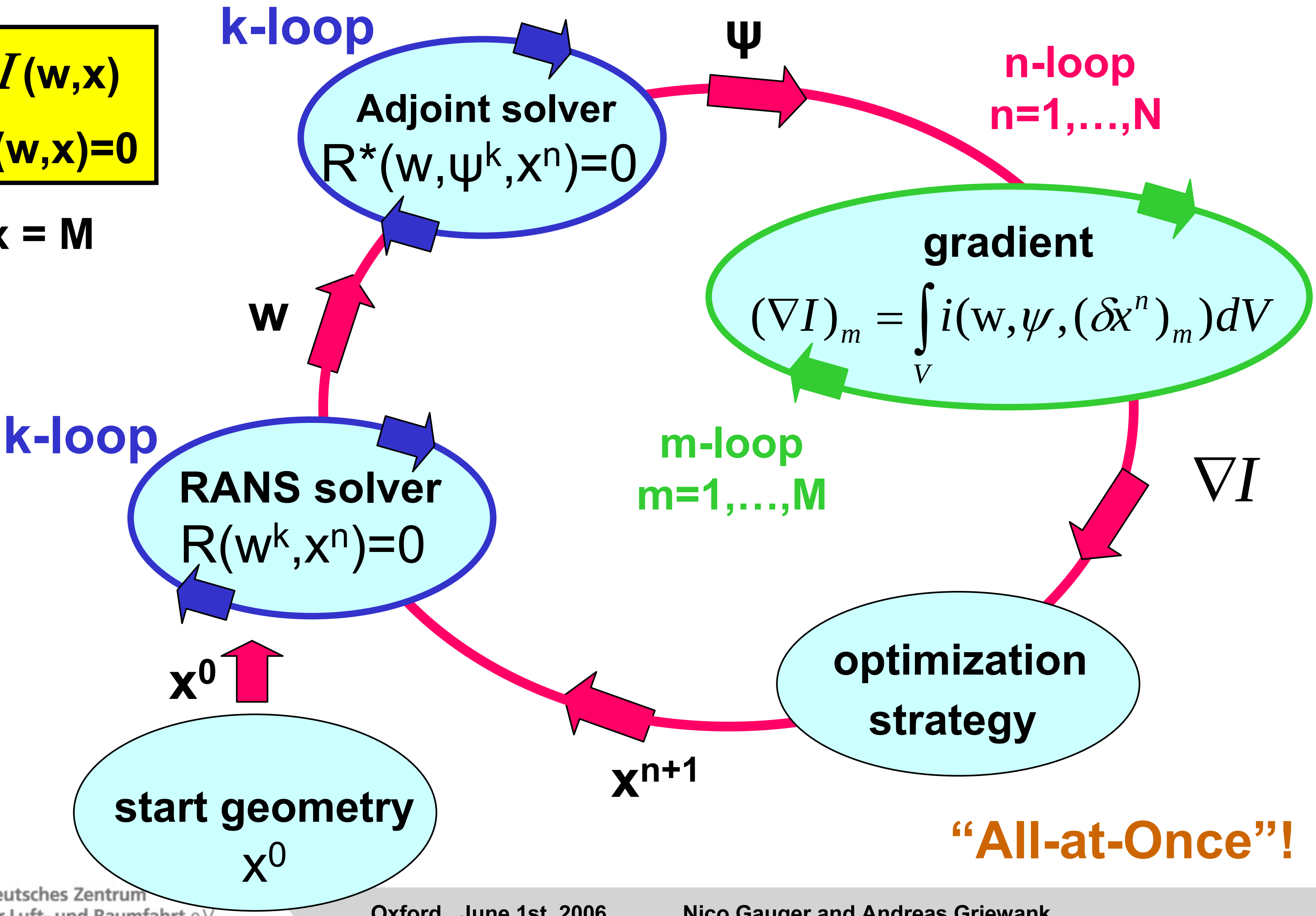


(Gauger,Moldenhauer)

Adjoint Based Optimization

min I(w,x)
s.t. R(w,x)=0

dim x = M



"All-at-Once"!

Activities on “All-at-Once”

- **MEGADESIGN (BMWi Project)** (Ends 2007)**:**

  - ***Aerodynamic shape optimization using simultaneous pseudo-timestepping (published in JCP 2005: Hazra, Schulz, Brezillon, Gauger)***

- **Optimization with PDE (DFG-SPP 1253)** (Starts this Year)**:**

  - ***Multilevel parameterizations and fast multigrid methods for shape optimization (Gauger / Schulz)***

  - ***Automated extension of fixed point PDE solvers with bounded retardation (Gauger / Griewank / Slawig)***

Automated extension of fixed-point PDE solvers for optimal design with bounded retardation

- **Problem:** $\min f(y,u)$ **s.t.** $c(y,u)=0,$

  where $y \in \Re^n$ **and** $u \in \Re^m$ **are state and design variables**

- **Available:**

  Code for $f(y,u)$ **and** $G(y,u) \approx y - \left(\frac{\partial}{\partial y} c(y,u)\right)^{-1} c(y,u)$

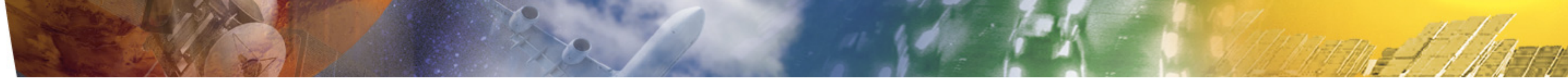
- **Contractive fixed-point iteration:**

  $G, f \in C^{2,1}(\Re^{n+m})$ **and** $\left\| \frac{\partial}{\partial y} G(y,u) \right\| \le \rho < 1$

- **Shifted Lagrangian:**

$$N(y,\bar{y},u) \equiv G(y,u)^T \bar{y} + f(y,u) \equiv Lagrangian + \bar{y}^T y,$$

Lagrangian is formed w.r.t. $c(y,u) \equiv G(y,u) - y = 0$



Automated extension of fixed-point PDE solvers for optimal design with bounded retardation

Strategy: differentiate → iterate

“Piggy-back”: automated extension to coupled iteration:

$$\begin{array}{r c}
\textbf{states} & \\
\textbf{adjoint states} & \\
\textbf{adjoint designs} & \\
\textbf{designs} &
\end{array}
\begin{bmatrix} y_{k+1} \\ \bar{y}_{k+1} \\ \bar{u}_{k+1} \\ u_{k+1} \end{bmatrix} = \begin{bmatrix} N_{\bar{y}}(y_{k,} u_k) \\ N_y(y_{k,} \bar{y}_{k,} u_k) \\ N_u(y_{k,} \bar{y}_{k,} u_k) \\ u_k - H_k^{-1} N_u(y_{k,} \bar{y}_{k,} u_k) \end{bmatrix}$$

positive definite matrix $H_k > 0$ **preconditioner**

Automated extension of fixed-point PDE solvers for optimal design with bounded retardation

Jacobian of the extended iteration:

$$J_* = \left.\frac{\partial(y_{k+1}, \bar{y}_{k+1}, u_{k+1})}{\partial(y_k, \bar{y}_k, u_k)}\right|_* = \begin{pmatrix} G_y & 0 & G_u \\ N_{yy} & G_y^T & N_{yu} \\ -H_*^{-1}N_{uy} & -H_*^{-1}G_u^T & I - H_*^{-1}N_{uu} \end{pmatrix}$$

Whenever we can define H such that

$$\frac{1-\rho(J_*)}{1-\rho(G_y)} \approx \frac{\log(\rho(J_*))}{\log(\rho(G_y))} < const$$

we have bounded retardation

Automated extension of fixed-point PDE solvers for optimal design with bounded retardation

Eigenvalues of J_* are the zeros of the equation

$$\det((\lambda - 1)H_* + H(\lambda)) = 0$$

where

$$H(\lambda) = \left(-G_u^T (G_y^T - \lambda I)^{-1}, I\right) \begin{pmatrix} N_{yy} & N_{yu} \\ N_{uy} & N_{uu} \end{pmatrix} \begin{pmatrix} -(G_y^T - \lambda I)^{-1} G_u \\ I \end{pmatrix}$$

Necessary condition for contractivity:

$H_* > H(\lambda)/(1-\lambda)$ **for** $\lambda \leq -1$

Automated extension of fixed-point PDE solvers for optimal design with bounded retardation

Necessary condition for contractivity:

$H_* > H(\lambda)/(1-\lambda)$ **for** $\lambda \leq -1$

Usually not satisfied by reduced Hessian $H(1)$

Promising alternative: $H(-1)$ [Griewank 05]

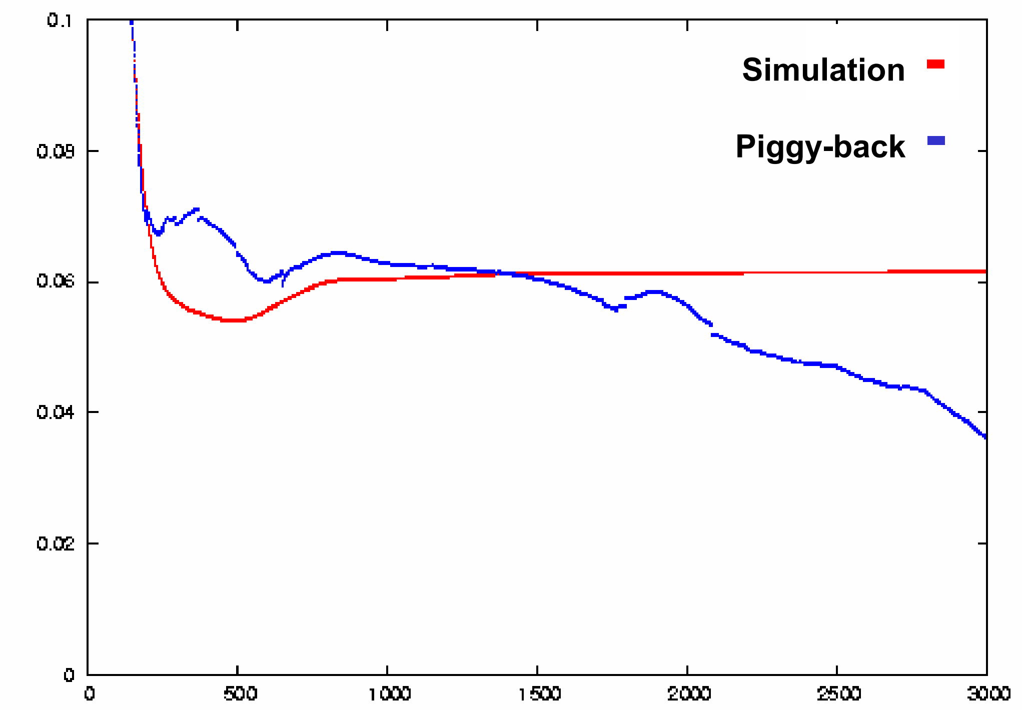
Feasibility Study

- Transonic case: NACA 0012 at Ma = 0.8 with α = 2°
- Cost function: glide ratio
- “FLOWer-Derivate” (2D Euler) + mesh deformation + parameterization
- First and second derivatives by AD tool TAPENADE
- Geometric constraint: constant thickness
- Camberline/Thickness decomposition, 20 Hicks-Henne coefficients define camberline

(Gauger,Griewank,Riehme)

Results

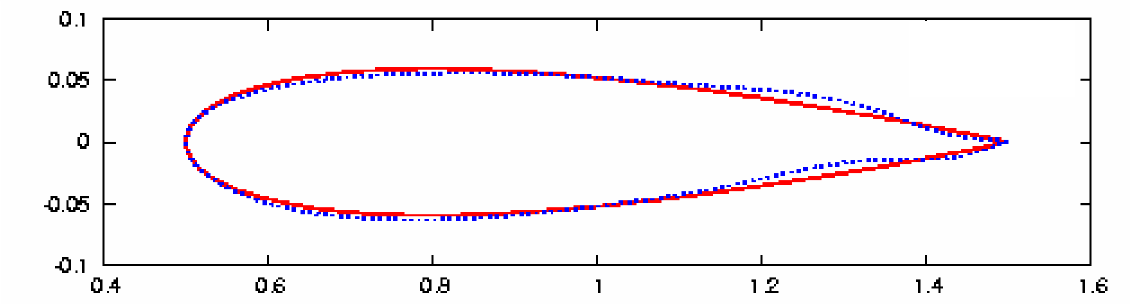
min C_D/C_L (Inverse Glide Ratio)



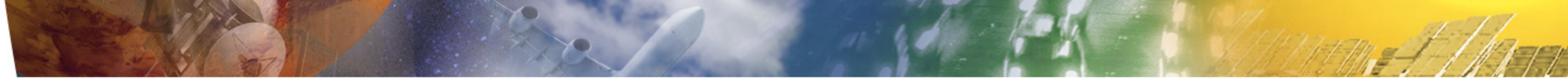
NACA 0012

Ma = 0.8 with α = 2°

Wing Shapes



(Gauger,Griewank,Riehme)



Automated extension of fixed-point PDE solvers for optimal design with bounded retardation

Globalization: enforce convergence by line search w.r.t.

augmented Lagrangian

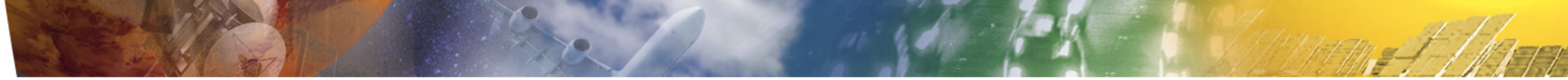
$$p(y,\bar{y},u) := (\alpha/2)\|G(y,u)-y\|_2^2 + (\beta/2)\|N_y(y,\bar{y},u)-\bar{y}\|_2^2 + N - \bar{y}^T y$$

With $\overline{G}_y = (1/2)(G_y + G_y^T)$, the directional derivative is given by

$$\begin{pmatrix}\nabla_y p\\ \nabla_{\bar{y}} p\\ \nabla_u p\end{pmatrix}^T \begin{pmatrix} G-y\\ N_y-\bar{y}\\ -H^{-1}N_u\end{pmatrix} = -\begin{pmatrix} G-y\\ N_y-\bar{y}\\ -H^{-1}N_u\end{pmatrix}^T \begin{pmatrix} \alpha(I-\overline{G}_y) & -(\beta/2)N_{yy}-I & -(\alpha/2)G_u\\ -(\beta/2)N_{yy}-I & \beta(I-\overline{G}_y) & -(\beta/2)N_{yu}\\ -(\alpha/2)G_u & -(\beta/2)N_{yu} & H \end{pmatrix} \begin{pmatrix} G-y\\ N_y-\bar{y}\\ -H^{-1}N_u\end{pmatrix}$$

which is a definite quadratic form for contractive G_y

and suitable weights $\alpha, \beta \in \Re$.



Automated extension of fixed-point PDE solvers for optimal design with bounded retardation

For suitable choices α, β and H the smallest eigenvalue of

$$\begin{pmatrix} \alpha(I-\overline{G}_y) & -(\beta/2)N_{yy}-I & -(\alpha/2)G_u \\ -(\beta/2)N_{yy}-I & \beta(I-\overline{G}_y) & -(\beta/2)N_{yu} \\ -(\alpha/2)G_u & -(\beta/2)N_{yu} & H \end{pmatrix}$$

can be bounded away from zero. Then one can design a line search procedure for determining γ_k such that the iterates

$$\left[y_{k+1}, \bar{y}_{k+1}, u_{k+1}\right] = (1-\gamma_k)\left[y_k, \bar{y}_k, u_k\right] + \gamma_k\left[G(y_k, u_k), N_y(y_k, \bar{y}_k, u_k), -H_*^{-1}N_u(y_k, \bar{y}_k, u_k)\right]$$

must converge to a KKT point.

Automated extension of fixed-point PDE solvers for optimal design with bounded retardation

Goals:

- Full fidelity optimization without model reduction
- Automated adjoint generation with verifiable results
- Bounded retardation of the original convergence rate
- Bounded increase of computational effort per step
- Bounded increase in overall storage requirement
- Efficient storage and linear algebra for the preconditioner of the combined iteration
- Selective evaluation or approximation of second derivatives
- Stability with respect to the design parameterization