Jacobian-free Optimization

Andreas Griewank

Humboldt-Universität zu Berlin
DFG Research Center Matheon

19th December 2006

DMV Jahrestagung 2006 Bonn

with thanks to
U. Naumann(RWTH) and A. Walter(TUD)
O. Vogel (Klingelnberg) and S. Schlenkrich(TUD)
N. Gauger (DLR) and V. Schulz (Uni Trier)
R. Giering and Th. Kaminski (FastOpt)
CAMEL (Carbon Assimilation and Modelling)

Outline

1. Introduction and Background
   - From Simulation to Optimization
   - "Direct" Optimization in Aerodynamics
   - Difference Quotients unstable and costly
   - (Meta) Cost of Algorithmic Derivatives
2. Approximation of Jacobians/Hessians
   - Classical Low–Rank Updating
   - Domain Invariance via Adjoint Vectors
   - Applications to Least Squares
   - Total quasi–Newton for NLP
3. (Almost) Matrix–free Design Optimization
   - Implicit and Iterative Differentiation
   - Two phase method on TAUij Code (Walter)
   - Preconditioning Task in One–Shot Approach

Questions and Worries — Are there any hard results?

Recent answers regarding global convergence proofs in NLP:

- M.J.D. Powell : "What for?"
- Nick Gould : "Useless !"
- Andreas Wächter : "Irrelevant !"

NP complete ⇒ (?) Exponential complexity:

- Nonconvex constraints feasibility
- Coloring for sparse matrix (pre-)compression
- Optimal Jacobian accumulation

Algorithmic quality measures:

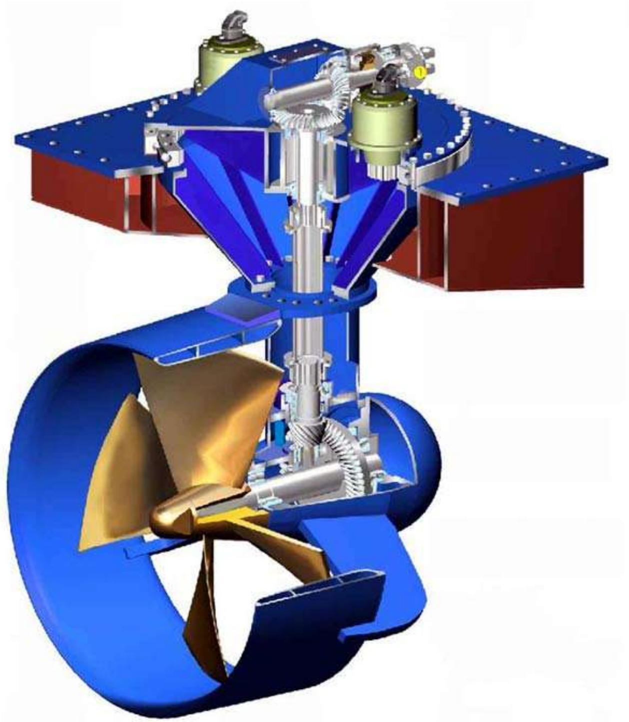
- Complexity of major/minor iteration
- Local convergence rates/orders
- Linear domain and/or range invariance

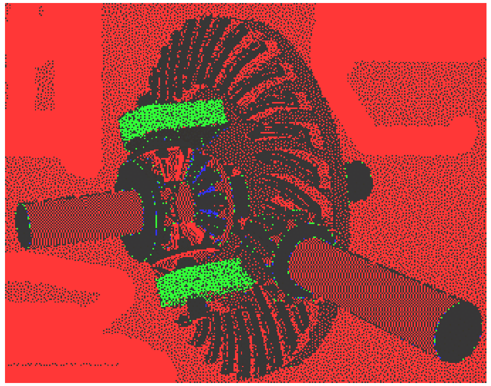
Introduction and Background

1. Introduction and Background
   - From Simulation to Optimization
   - "Direct" Optimization in Aerodynamics
   - Difference Quotients unstable and costly
   - (Meta) Cost of Algorithmic Derivatives

From Simulation to Optimization

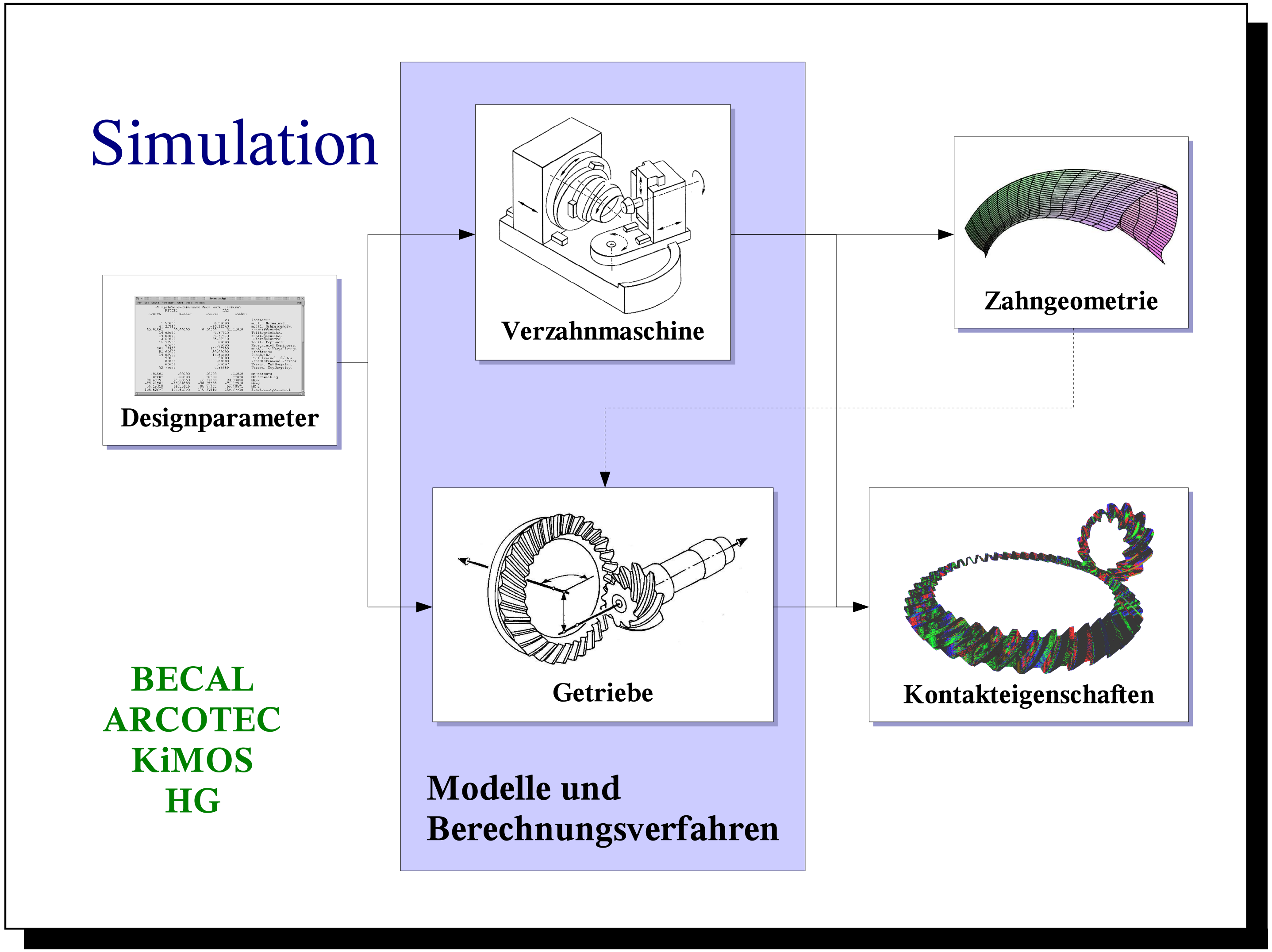
Kegelradgetriebe





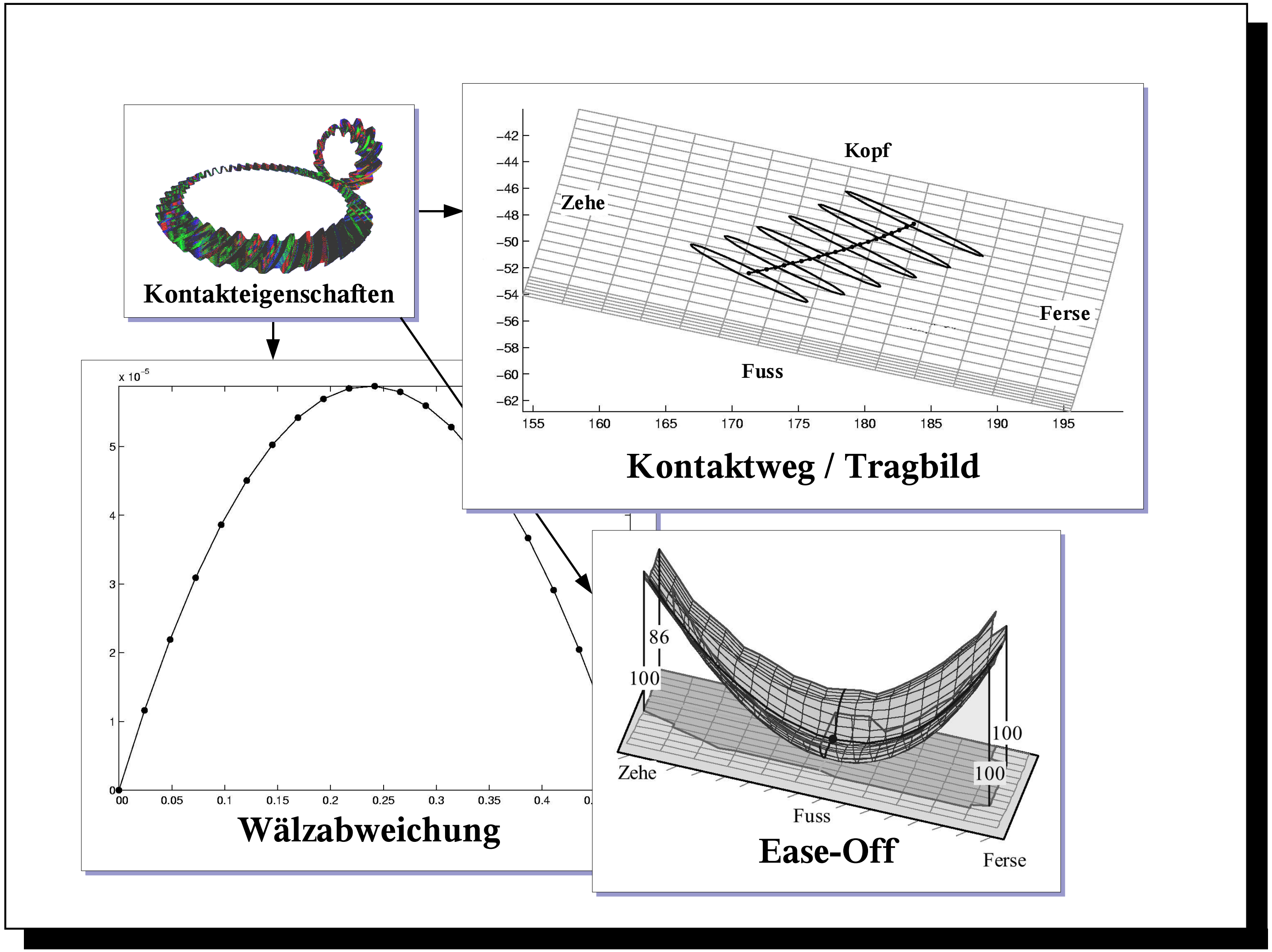
Thanks to Olaf Vogel, Klingelnberg

From Simulation to Optimization



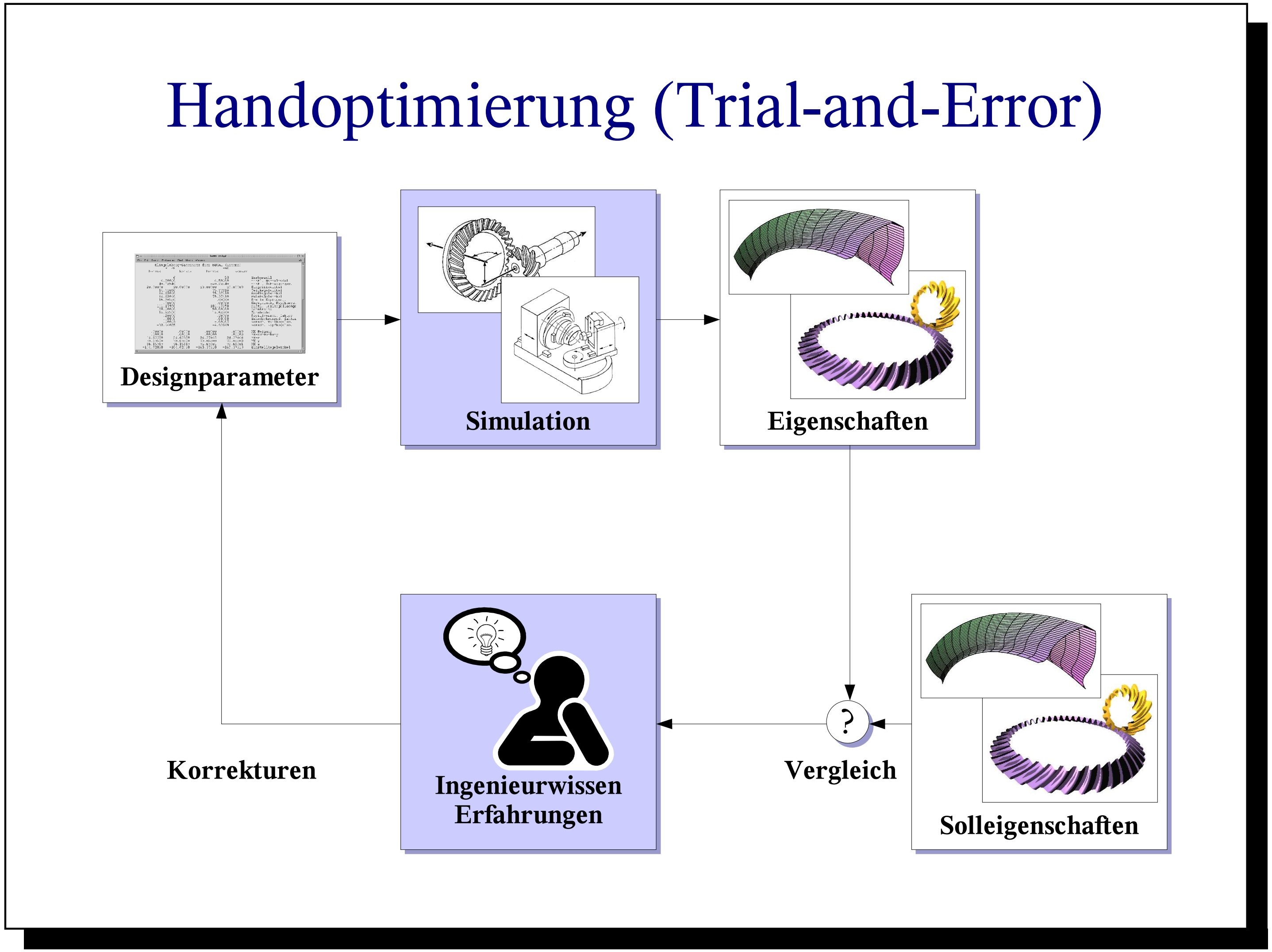
Thanks to Olaf Vogel, Klingelnberg

From Simulation to Optimization



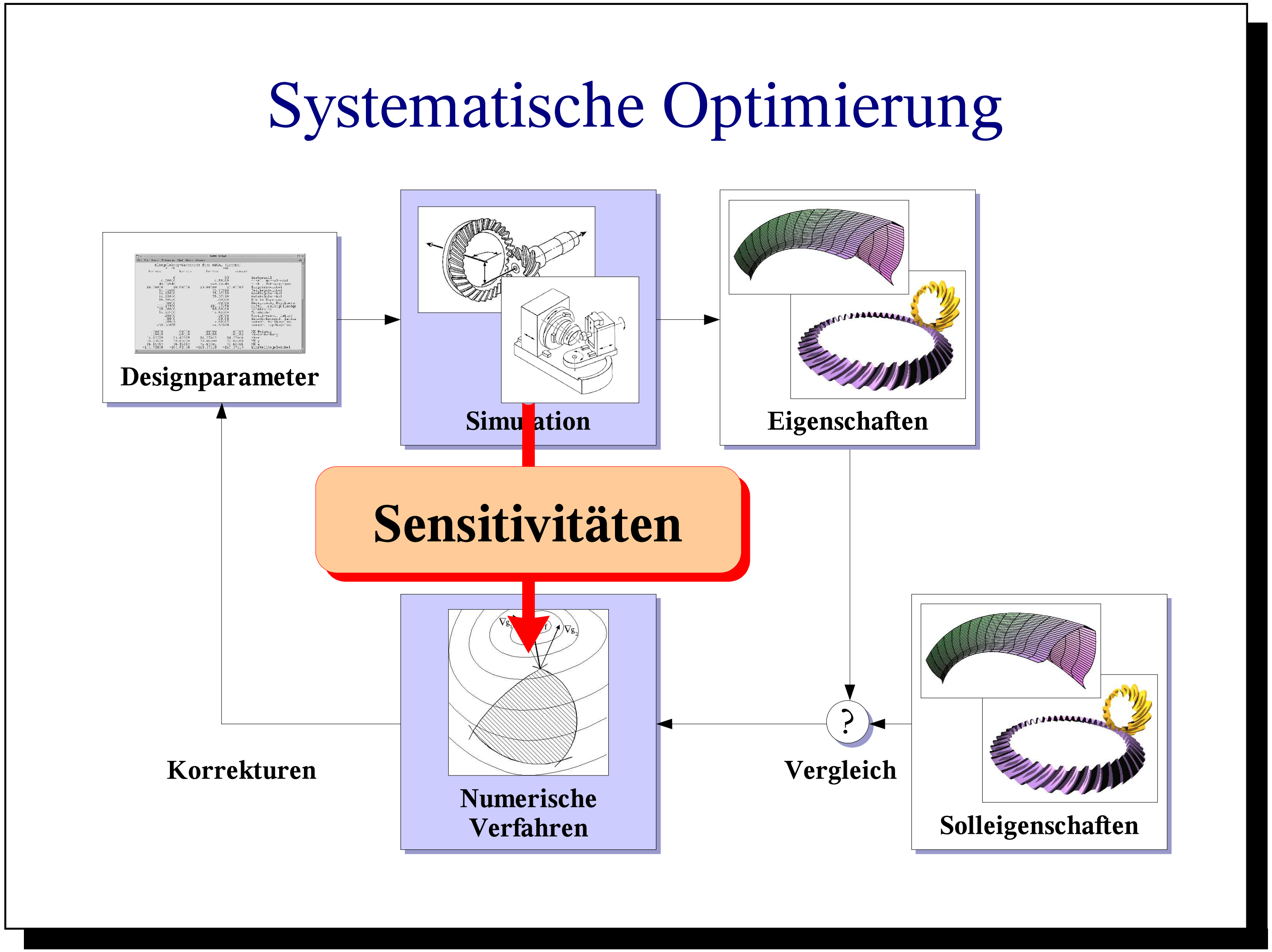
Thanks to Olaf Vogel, Klingelnberg

From Simulation to Optimization



Thanks to Olaf Vogel, Klingelnberg

From Simulation to Optimization



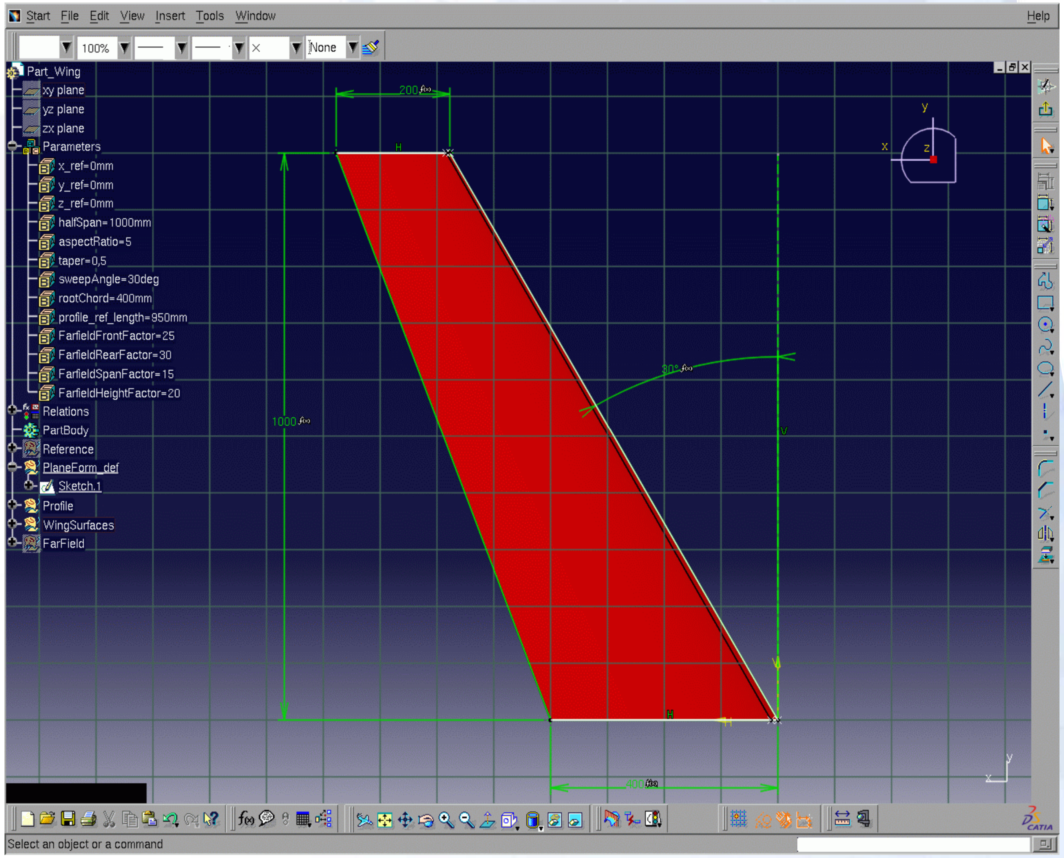
Thanks to Olaf Vogel, Klingelnberg

"Direct" Optimization in Aerodynamics

Military Aircraft

EADS DEFENCE & SECURITY

Inviscid shape optimisation of a wing planform – using CATIA_v5



39 Optimisation in Aerodynamics, Humboldt University Berlin, 9 May 2005

Thanks to Werner Haase, EADS

"Direct" Optimization in Aerodynamics

Military Aircraft

Inviscid shape optimisation of a wing planform – using CATIA_v5

Flow conditions:
Mach= 0.85, angle of attack = 1°

Design parameters:

sweep angle	(range: -60° to +60°)
halfspan	(range: 0.750 m to 1.250 m)
aspect ratio	**(defined by const. wing plan area constraint)**
taper ratio	(range: 0.2 to 0.8)

Design constraints:
Pitching moment restricted to range –0.025 to +0.0001

40 Optimisation in Aerodynamics, Humboldt University Berlin, 9 May 2005

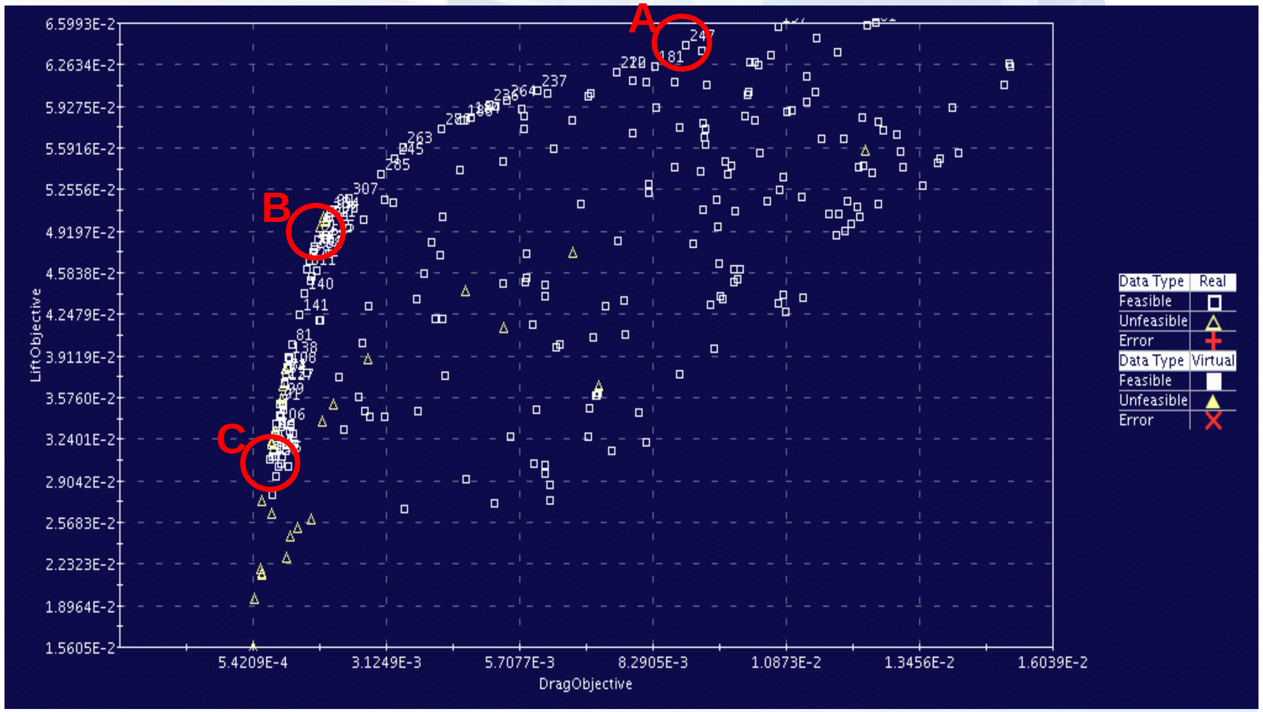
Thanks to Werner Haase, EADS

"Direct" Optimization in Aerodynamics

Military Aircraft

Inviscid shape optimisation of a wing planform – using CATIA_v5

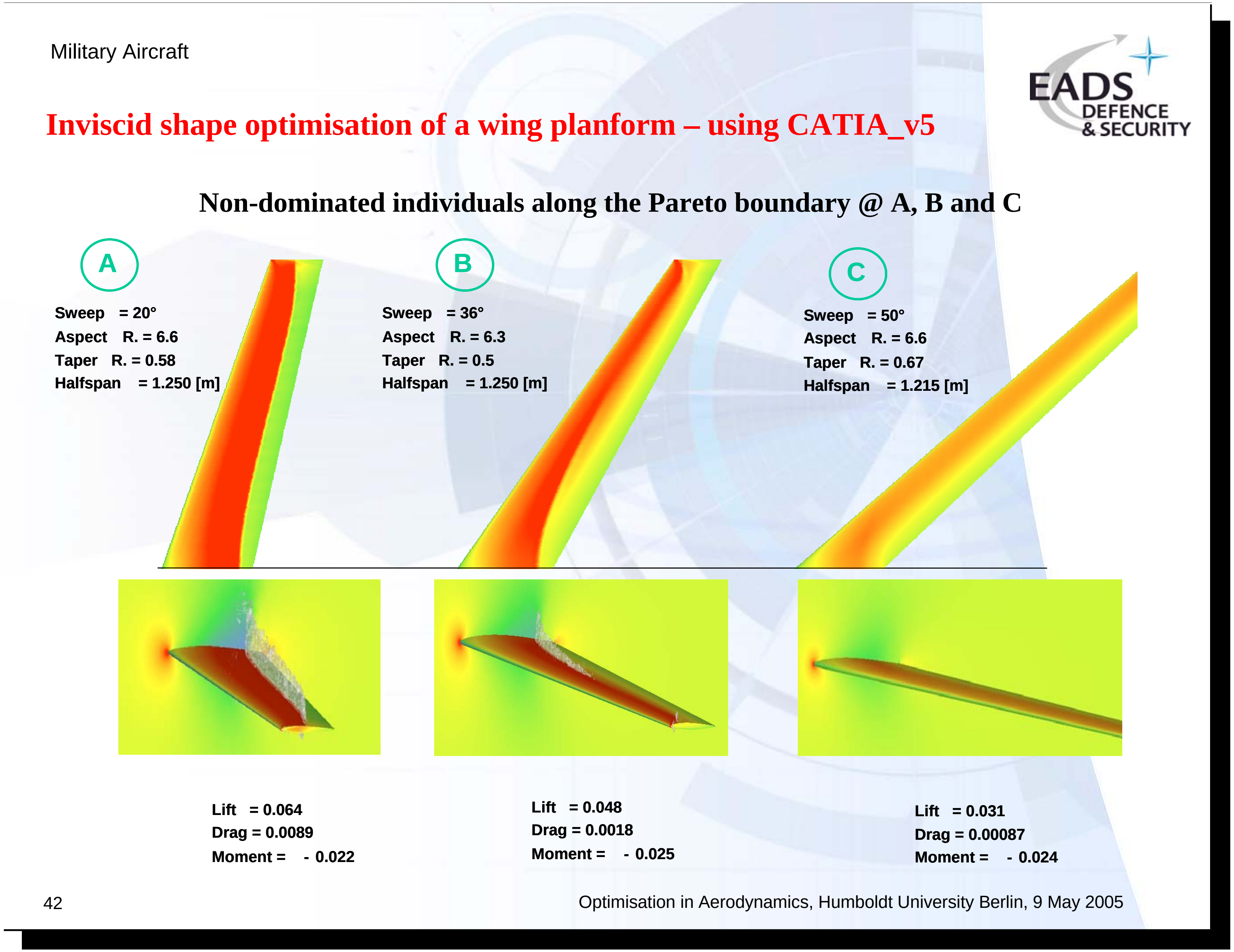
The correct approach: Wing area kept constant



41 Optimisation in Aerodynamics, Humboldt University Berlin, 9 May 2005

Thanks to Werner Haase, EADS

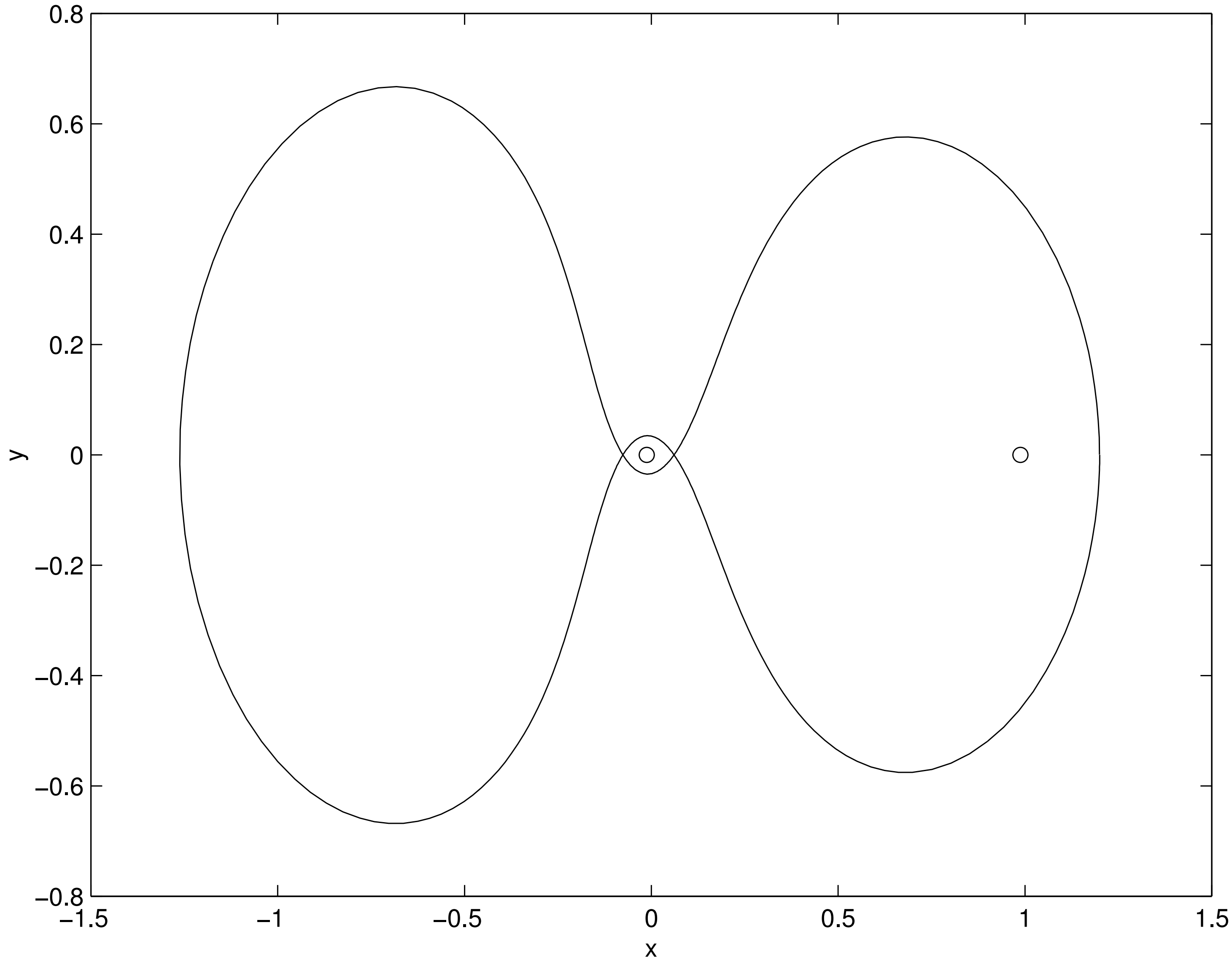
"Direct" Optimization in Aerodynamics



Thanks to Werner Haase, EADS

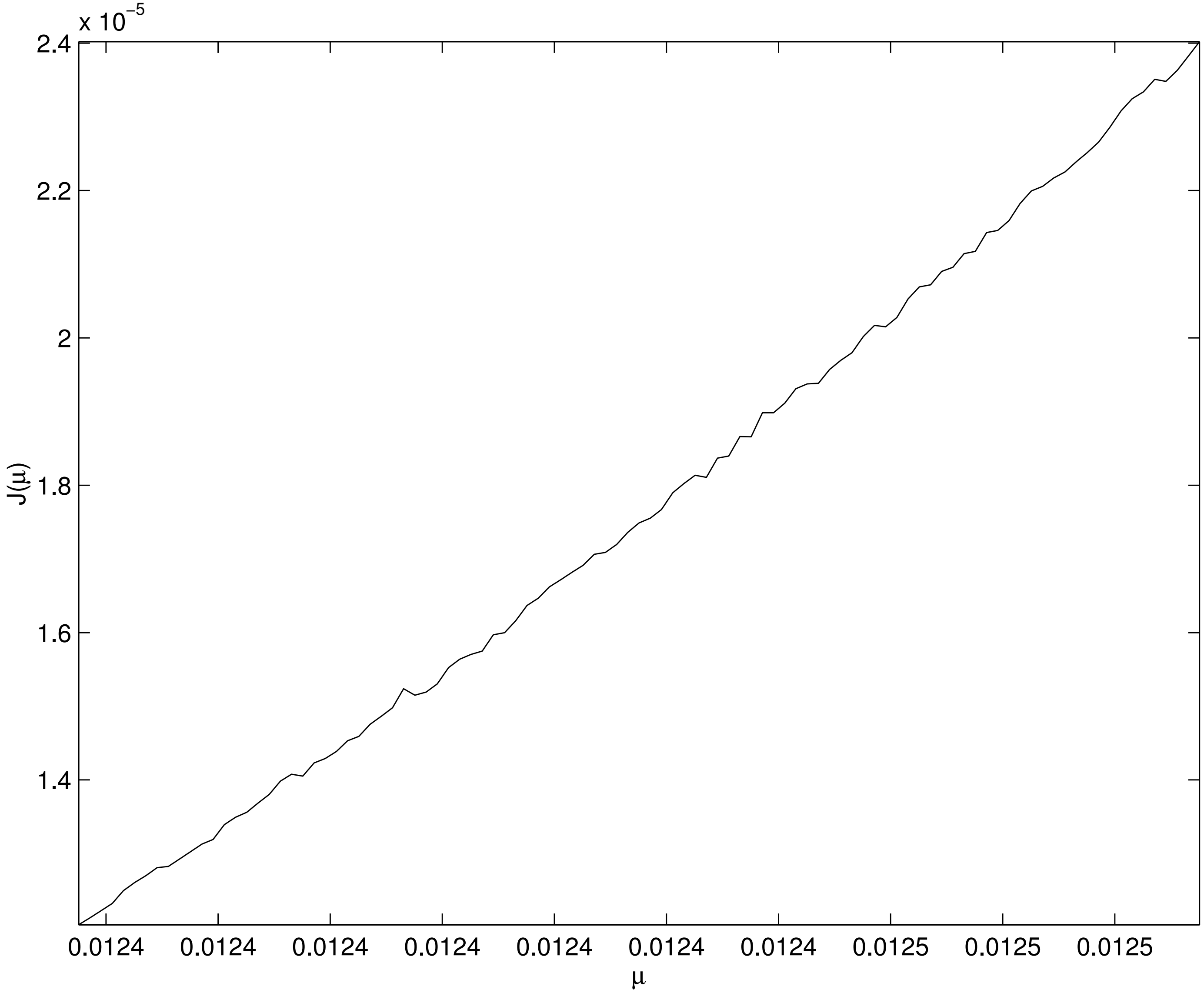
Difference Quotients unstable and costly

Celestial Motion (Gockenbach et Symes):



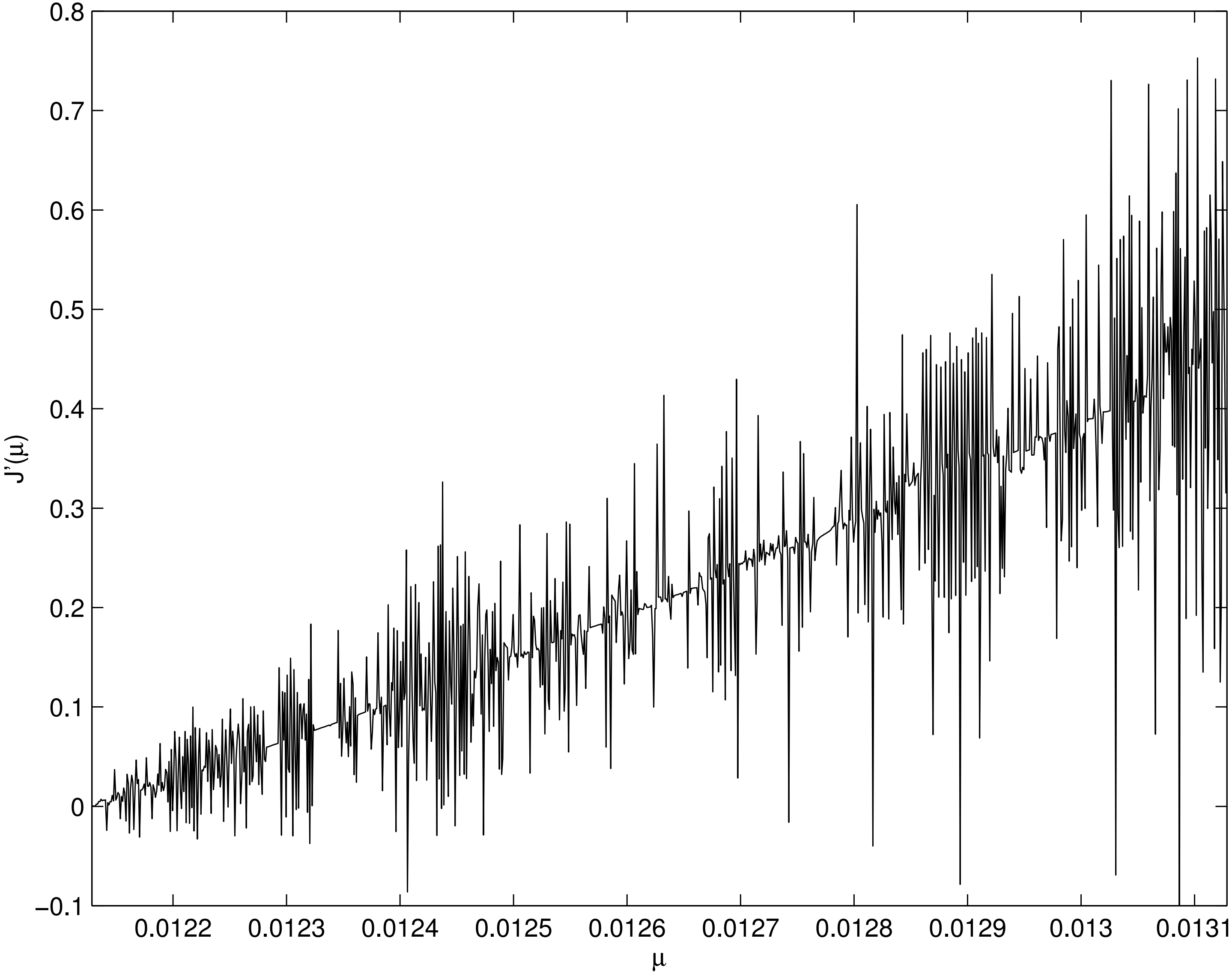
Difference Quotients unstable and costly

Distance Results by RK45



Difference Quotients unstable and costly

Difference Approximation to Derivative:



(Meta) Cost of Algorithmic Derivatives

Realistic assumption:

$$y = F(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

defined by *long* evaluation loop

$$\begin{aligned} \text{input}: \quad v_{i-n} &= x_i & \text{for } i = 1 \ldots n \\ \text{evaluation}: \quad v_i &= \varphi_i\big(v_j\big)_{j \prec i} & \text{for } i = 1 \ldots \ell \\ \text{output}: \quad y_{m-i} &= v_{\ell-i} & \text{for } i = 0 \ldots m-1 \end{aligned}$$

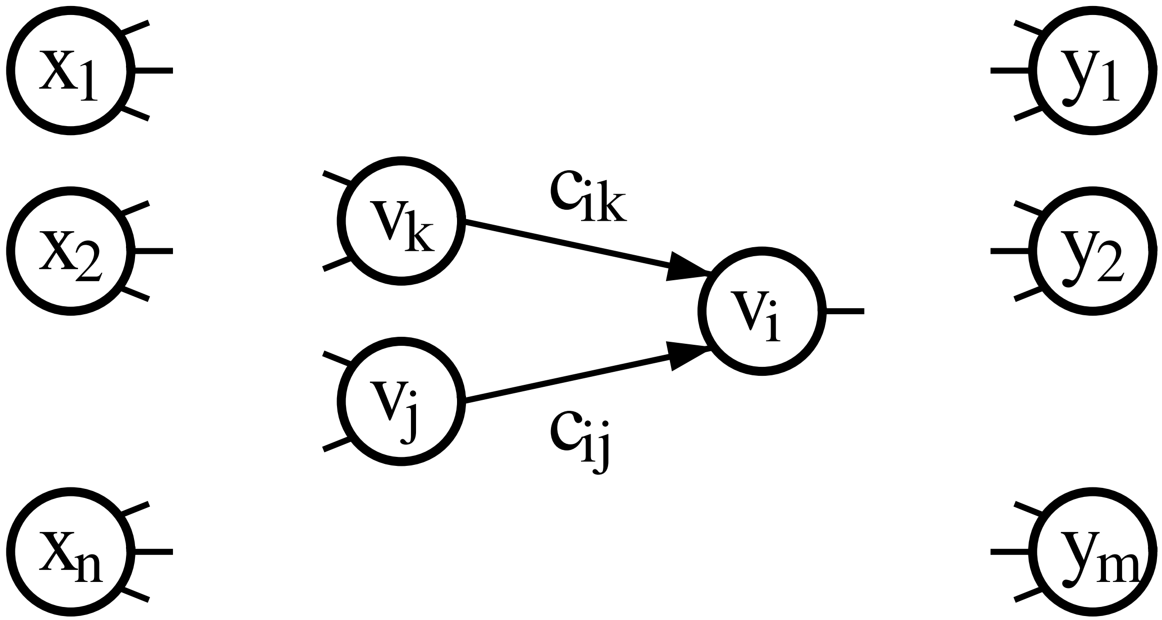
where $v_i \in \mathbb{R}$ for $i = 1-n \ldots \ell$ and

$$\varphi_i \in \{+, -, *, /, \exp, \log, \sin, \cos, \ldots\}$$

Partial pre-ordering

$$j \prec i \iff c_{ij} \equiv \frac{\partial \varphi_i}{\partial v_j} \not\equiv 0 \, .$$

Computational (Directed) Acyclic Graph



Bauer's Accumulation Formula

$$\frac{\partial y_i}{\partial x_j} = \sum_{\mathcal{P} \in [x_j \to y_i]} \prod_{(\tilde{\jmath}, \tilde{\imath}) \in \mathcal{P}} c_{\tilde{\imath}\tilde{\jmath}}$$

over all paths $\mathcal{P}$. 'Explicit' complexity exponential in depth of graph .

Common Subexpressions

Search for optimal usage of common subexpressions is NP hard (Naumann 2005) By reduction from:

Ensemble computation:

Given generic commuting factors

$$c_j \quad \text{for} \quad j = 1, \ldots, \tilde{n}$$

and index subsets

$$J_i \subset \{1, 2, \ldots, \tilde{n}\} \quad \text{for} \quad i = 1, \ldots, \tilde{m}$$

compute the family of products

$$a_i = \prod_{j \in J_i} c_j \quad \text{for} \quad i = 1, \ldots, \tilde{m}$$

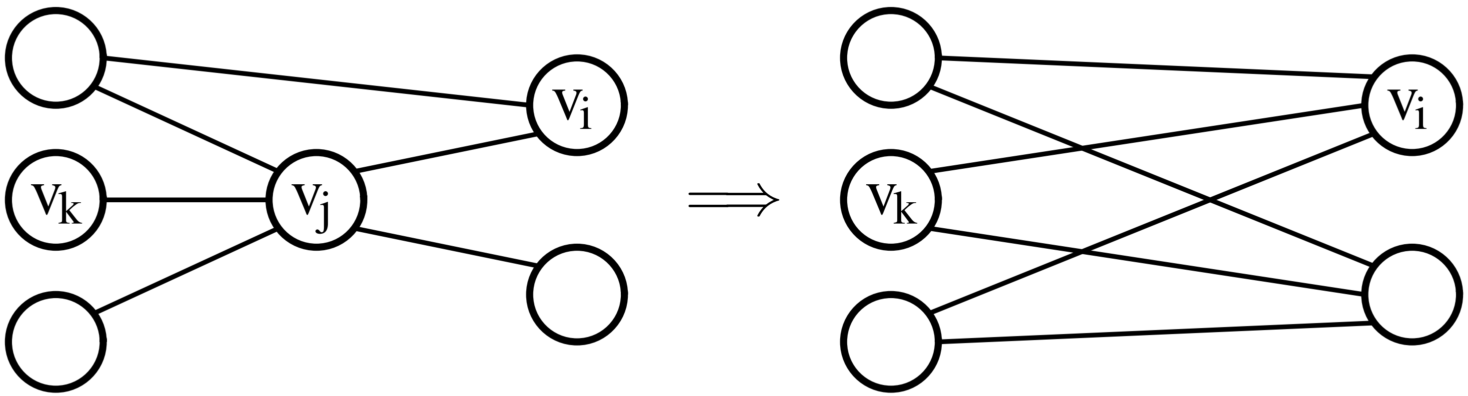
using minimal number of binary multiplications.

Elimination of Vertices (or edges, or facets)

Gauss-like elimination of vertex j

$$
\begin{aligned}
c_{ik} \mathrel{+}= \; c_{ij} \cdot c_{jk} \qquad & \text{for } k \prec j \prec i; \\
c_{ij} = 0 \wedge c_{jk} = 0 \quad & \text{for } k \prec j \wedge j \prec i.
\end{aligned}
$$

Elimination of all intermediates in any order yields bipartite graph whose edge values are the nonzero Jacobian entries.



$$
\begin{aligned}
\text{Forward Mode} \quad &: \quad j = 1, 2, \ldots\ldots, \ell - m - 1, \ell - m \\
\text{Reverse Mode} \quad &: \quad j = \ell - m, \ell - m - 1, \ldots\ldots, 2, 1
\end{aligned}
$$

Resulting Operation Count=OPS

For $\dot{x} \in \mathbb{R}^n$ and $\bar{y} \in \mathbb{R}^m$ forward on univariate $F(x + t\dot{x})$ and reverse on scalar-valued $\bar{y}^\top F(x)$ yields

$$\frac{\mathrm{OPS}\{\dot{y} \equiv F'(x)\dot{x}\}}{\mathrm{OPS}\{y \equiv F(x)\}} \quad \leq \quad 3 \quad \geq \quad \frac{\mathrm{OPS}\{\bar{x}^\top \equiv \bar{y}^\top F'(x)\}}{\mathrm{OPS}\{y \equiv F(x)\}}$$

and

$$\mathrm{OPS}\{\dot{\bar{x}}^\top \equiv \bar{y}^\top F''(x)\,\dot{x}\} \quad \leq \quad 9 \cdot \mathrm{OPS}\{y \equiv F(x)\}$$

BUT!!!

$$\frac{\mathrm{OPS}\{F'(x)\}}{\mathrm{OPS}\{F(x)\}} \quad = \quad ??? \quad \leq 3 \cdot \min(m, n)$$

and

$$\frac{\mathrm{OPS}\{\nabla_x^2(\bar{y}^\top F(x))\}}{\mathrm{OPS}\{F(x)\}} \quad = \quad ??? \quad \leq \quad 9\ n$$

Real World Example

Periodic Adsoption Processes

Real World Example

Periodic Adsoption Processes:

- separation of components, e.g. O_2 from air
- operate at steady-state $\Rightarrow$ CSS
- maximize overall recovery at desired purity, ...
- can be modeled by:

$$
\begin{aligned}
\min \quad & \varphi(y(t_f)) \\
s.t. \quad \dot{z}_1 &= f_1(y_1, p, t), y_1(t_0) = y_0, \ t \in [t_0, t_1] \\
\dot{z}_i &= f_i(y_i, p, t), y_i(t_{i-1}) = y_{i-1}(t_{i-1}), \\
& \quad t \in [t_{i-1}, t_i], \ i \in [2, N] \\
C(y_0) &= y_0 - y_N(t_N) = 0 \\
0 &= W(y(t, y_0, p)) + s \\
s &\geq 0
\end{aligned}
$$

Thanks to Larry Biegler, CMU

Real World Example

Optimization Problem (Periodic Adsoption Processes):

$$\text{Min} \;\; f(x) \qquad \text{subject to} \;\; c(x) = 0,$$

with nonlinear and sufficiently smooth

- $f : \mathbb{R}^{644} \rightarrow \mathbb{R}$ objective,
- $c : \mathbb{R}^{644} \rightarrow \mathbb{R}^{640}$ constraints: CSS and design constraints

Cost of dense and 'hand-optimized' Jacobian :

$$\approx 100 \; \textbf{TIME}(f, c)$$

Thanks to Andrea Walter, TU Dresden

Intermediate Summary & Conclusion

- Efficient and accurate optimization requires sensitivities.
- Gradients and other derivative vectors computable at little extra cost.
- Jacobians and Hessians may be costly to evaluate and/or factorize.
- Derivatives should only be evaluated and used selectively !!!.

Approximation of Jacobians/Hessians

2. Approximation of Jacobians/Hessians
   - Classical Low–Rank Updating
   - Domain Invariance via Adjoint Vectors
   - Applications to Least Squares
   - Total quasi–Newton for NLP

Classical Low–Rank Updating

- Nonlinear Equation System:

$$F(x) = 0 \quad \text{with} \quad F \in C^{1,1}(\mathbb{R}^n, \mathbb{R}^n)$$

where in the symmetric case

$$F(x) = \nabla f(x) \quad \text{with} \quad f \in C^{2,1}(\mathbb{R}^n, \mathbb{R}).$$

- Approximating Jacobians $A_k \simeq F'(x_k) \in \mathbb{R}^{n \times n}$ define quasi-Newton iteration

$$x_{k+1} = x_k + s_k \quad \text{by} \quad -A_k s_k = F_k \equiv F(x_k).$$

- Secant condition

$$A_{k+1} s_k = y_k \equiv F_{k+1} - F_k = F'_{k+1} s_k + O\left(\|s_k\|^2\right)$$

combined with least change criterion yields

$$A_{k+1} - A_k = \Delta A_k(A_k, s_k, y_k) \;\text{ of rank } \leq 2.$$

Characteristic Properties

- Numerical linear algebra effort of $O(n^2)$ compared to $O(n^3)$ for Newton via update of inverse A_k^{-1} or factorization $A_k = Q_k R_k$.
- Local and superlinear convergence, i.e.

$$x_0 \approx x_* \wedge A_0 \approx F'(x_*) \quad \text{with} \quad x_* \in F^{-1}(0) \quad \text{'regular'}$$

$$\Longrightarrow \; x_k \underset{k}{\rightarrow} x_* \quad \text{with} \quad \|x_{k+1} - x_*\| / \|x_k - x_*\| \underset{k}{\rightarrow} 0$$

- As A_k depends on (s_{k-j}, y_{k-j}) for $j < n$, the maximal $\rho > 0$ s.t.

$$\|x_{k+1} - x_*\| \sim \|x_k - x_*\|^\rho$$

is given by $\rho = \rho_n$ where

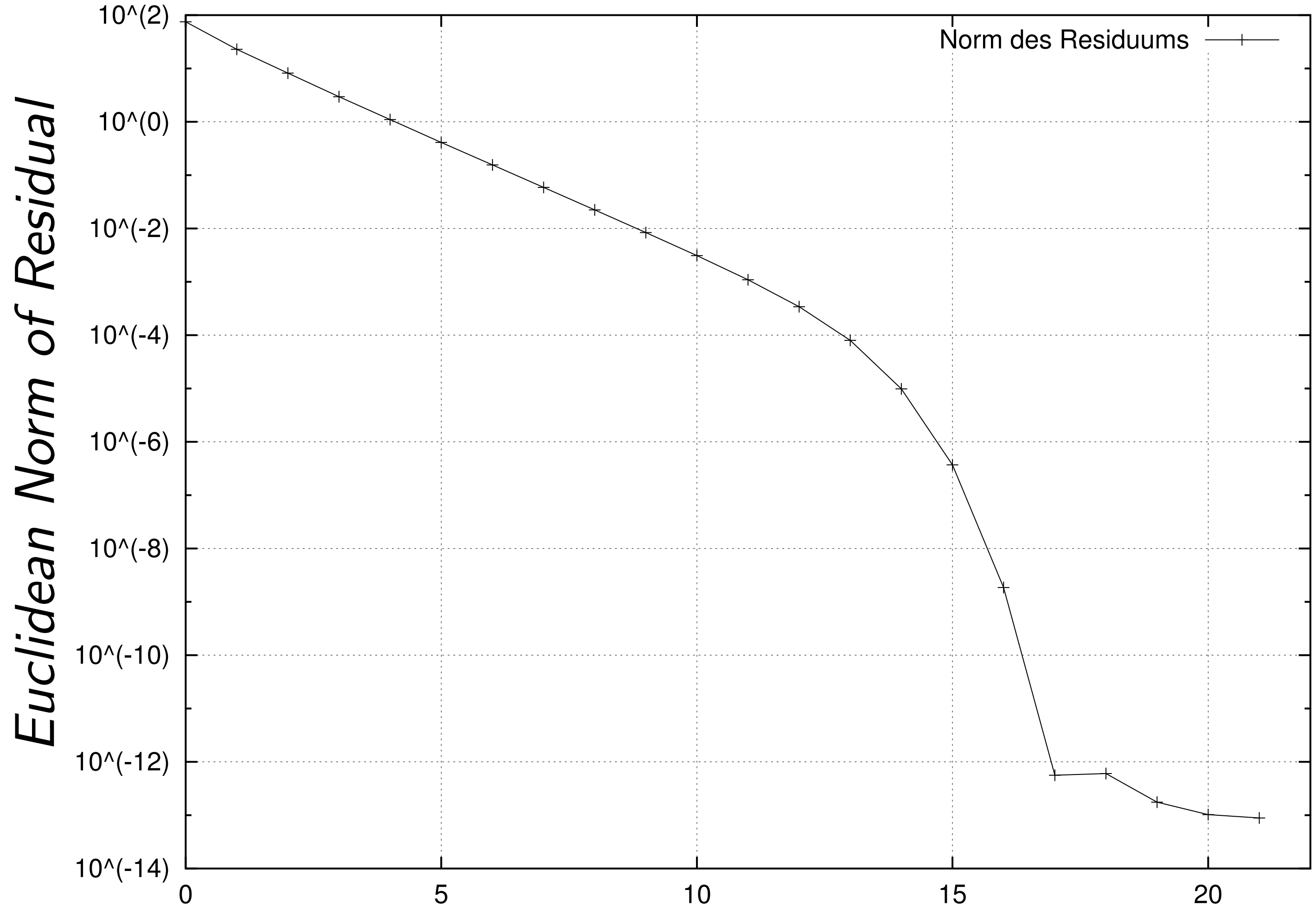
$$\rho_n^n(\rho_n - 1) = 1 \quad \Longrightarrow \quad \rho_n \approx 1 + \ln(n)/n \approx \sqrt[n]{n}.$$

- In Hilbert space case $n = \infty$ superlinear rate maintained iff

$$I - A_0^{-1} F'(x_*) : \ell_2 \rightarrow \ell_2 \quad \text{compact.}$$

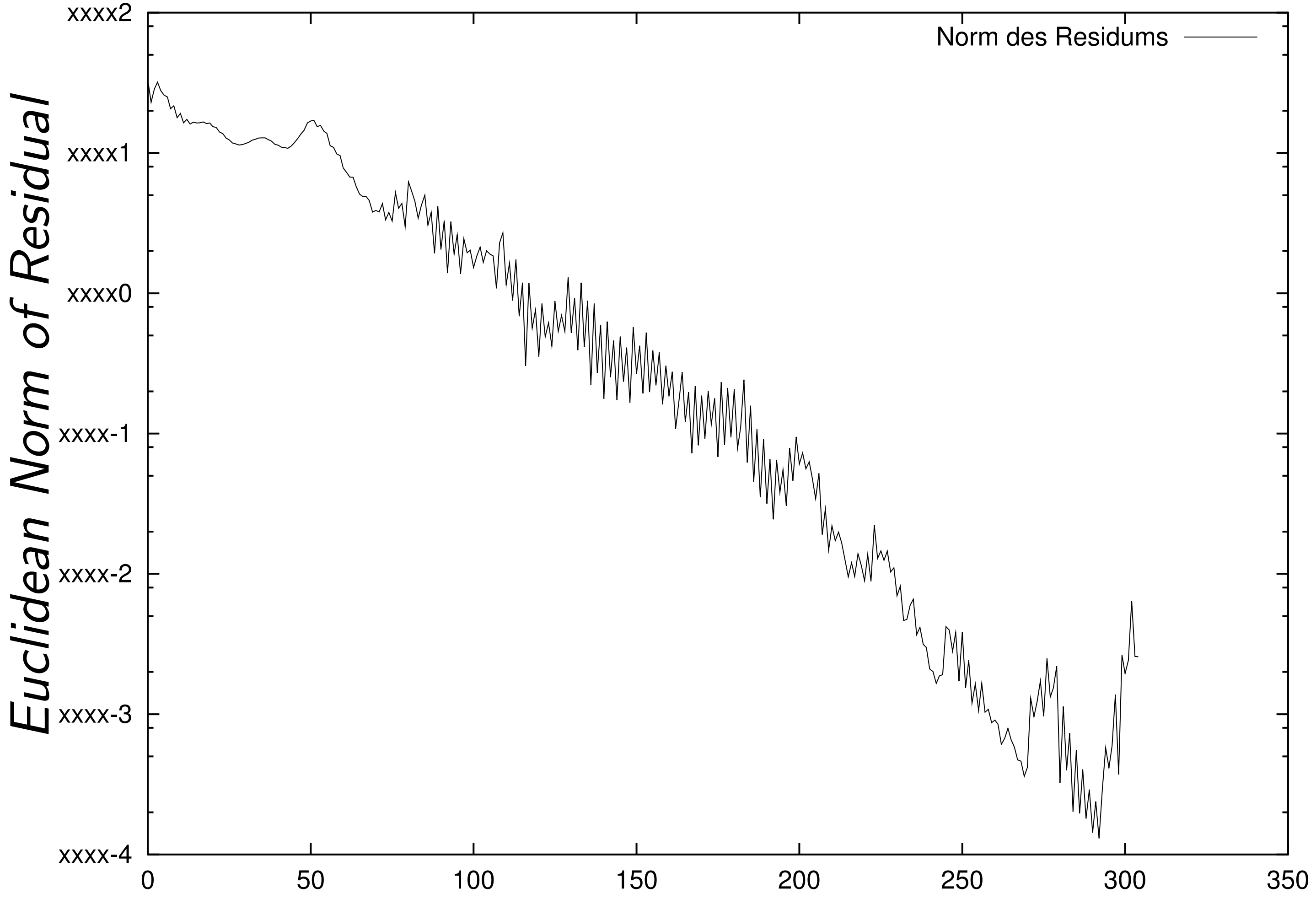
Results on Chandrasekhar

$$F(x)(\mu) = x(\mu) - \left(1 - \frac{c}{2}\int_0^1 \frac{\mu x(v)dv}{\mu + v}\right)^{-1} = 0$$



Results on Convection-Diffusion Equation

$$0 = -\Delta u + (v, w)\nabla u = f \quad \text{on} \quad \Omega = \Box$$



Domain Invariance via Adjoint Vectors

Problem:

Broyden and other updates for unsymmetric case are dependent on:
norms $\longrightarrow$ scaling $\longrightarrow$ coordinate transformations

Remedy:

Include adjoint secant condition

$$A_{k+1}^\top \sigma_k = \mu_k \equiv F'(x_{k+1})^\top \sigma_k \in \mathbb{R}^n$$

for suitable dual steps $\sigma_k \in \mathbb{R}^m$.

Result is Two-Sided-Rank-One (TR1):

$$A_{k+1} = A_k + \frac{(y_k - A_k s_k)(\mu_k^\top - \sigma_k^\top A_k)}{(\mu_k^\top - \sigma_k^\top A_k) s_k}$$

which (almost) satisfies both secant conditions.

Key Properties of TR1

- Reduces to SR1 formula in the symmetric case $F = \nabla f$ with $\sigma_k = s_k$.
- Invariant with respect to linear transformation on $x \in \mathbb{R}^n$.
- Heredity for affine $F(x) = a + A_* x$ in that

$$A_k s_j = y_j \quad \text{and} \quad A_k^\top \sigma_j = \mu_j \quad \text{for all} \quad j < k$$

so that generally

$$A_k = A_* = F' \quad \text{for} \quad k \geq \min(m, n).$$

- (Griewank et al 2006) Maximal R-order $\rho = \rho_n$ achieved by

$$\sigma_k = r_k = y_k - A_k s_k \quad \text{for} \quad k \geq 0$$

in nonlinear equation case $m = n$.

Applications to Least Squares

Minimize $\varphi(x) \equiv \frac{1}{2}\|F(x) - y\|_2^2$

by quasi-Gauss-Newton iteration

$$x_{k+1} = x_k + \alpha_k s_k \quad \text{with} \quad 0 < \alpha_k \quad \text{by line-search}$$

where

$$-(A_k^\top A_k)s_k = (F_k'^\top)F_k = \nabla\varphi(x_k)$$

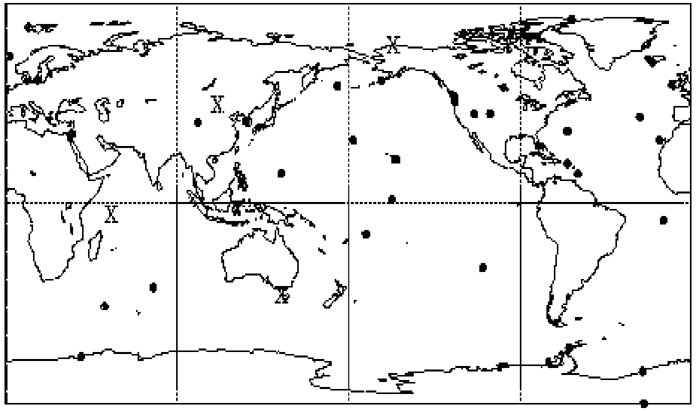
and $A_k \to A_{k+1}$ by TR1 with $\sigma_k = r_k$.

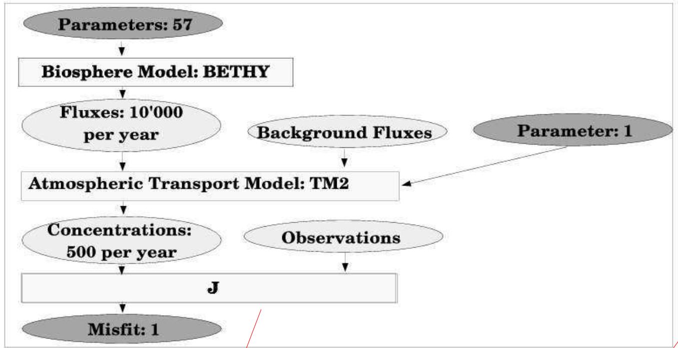
In comparison to Gauss-Newton ($A_k = F_k'$)

- Reduction in evaluation effort by roughly $OPS(F')/OPS(F)$.
- Reduction in linear algebra from mn^2 to $O(mn)$ via $A_k = Q_k R_k$.
- Essentially identical invariance and local convergence properties/rates.

CCDAS approach

- Terrestrial biosphere model BETHY (Knorr 97) delivers CO_2 fluxes to atmosphere
- Uncertainty in process parameters from laboratory measurements
- Global atmospheric network provides additional constraint





covariance of uncertainty in priors for parameters

covariance of uncertainty in measurements + model

priors for parameters

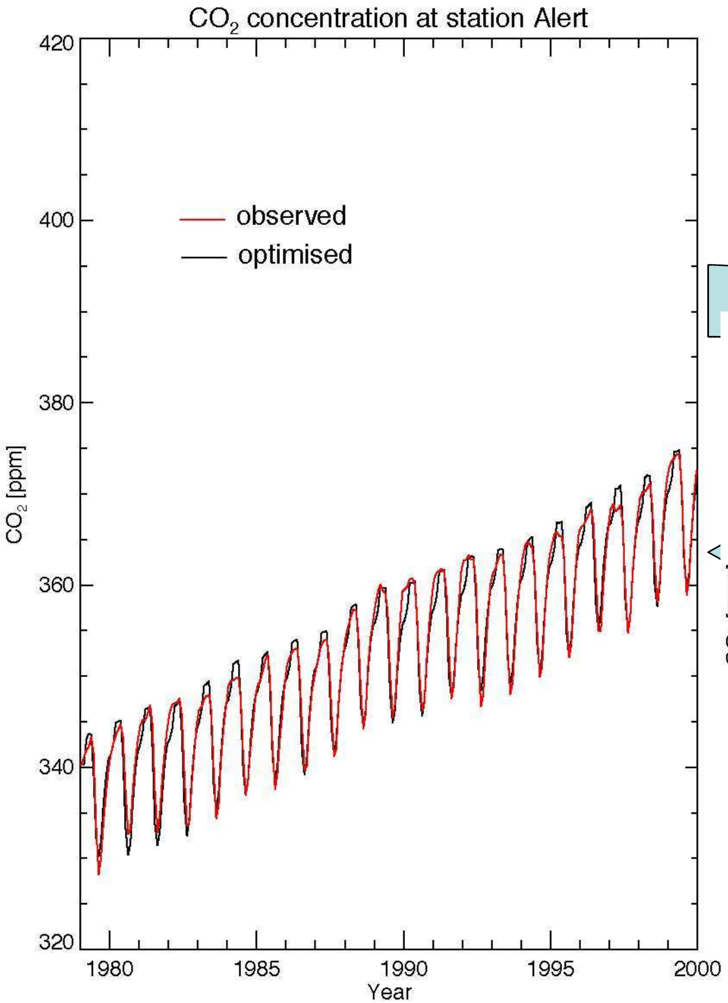
observed concentrations

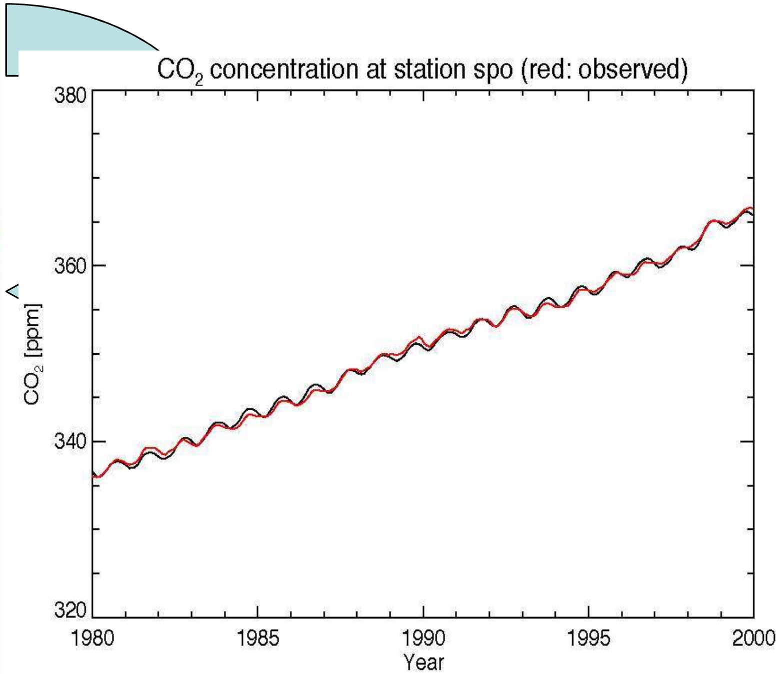
$$J(m) = \frac{1}{2}(m - m_0)^T C_m^{-1}(m - m_0) + \frac{1}{2}(c(m) - d)^T C_d^{-1}(c(m) - d)$$

FastOpt Max-Planck-Institut für Biogeochemie CSIRO

Thanks to Thomas Kaminski, FastOpt

Optimisation
(BFGS+ adjoint gradient)

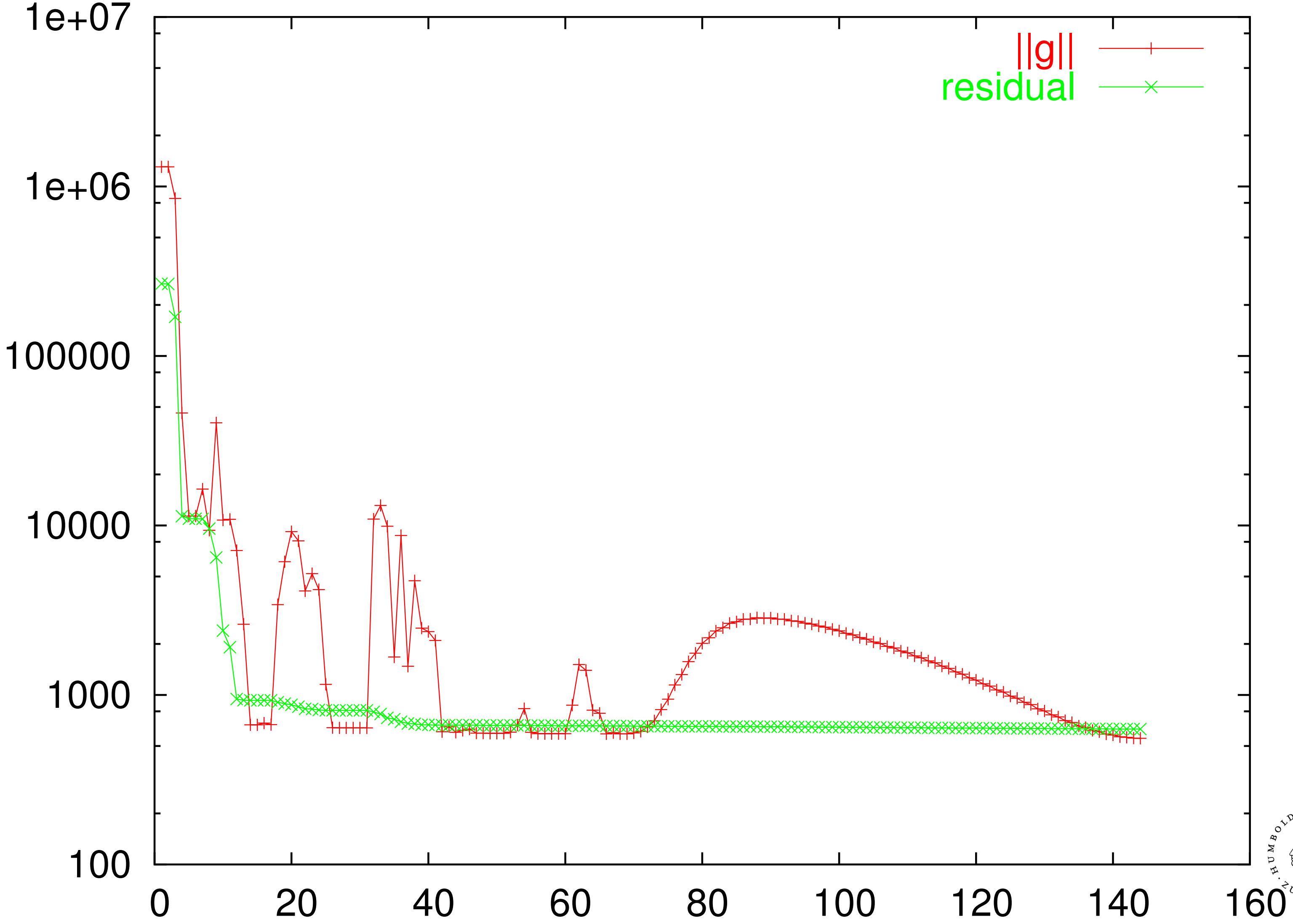




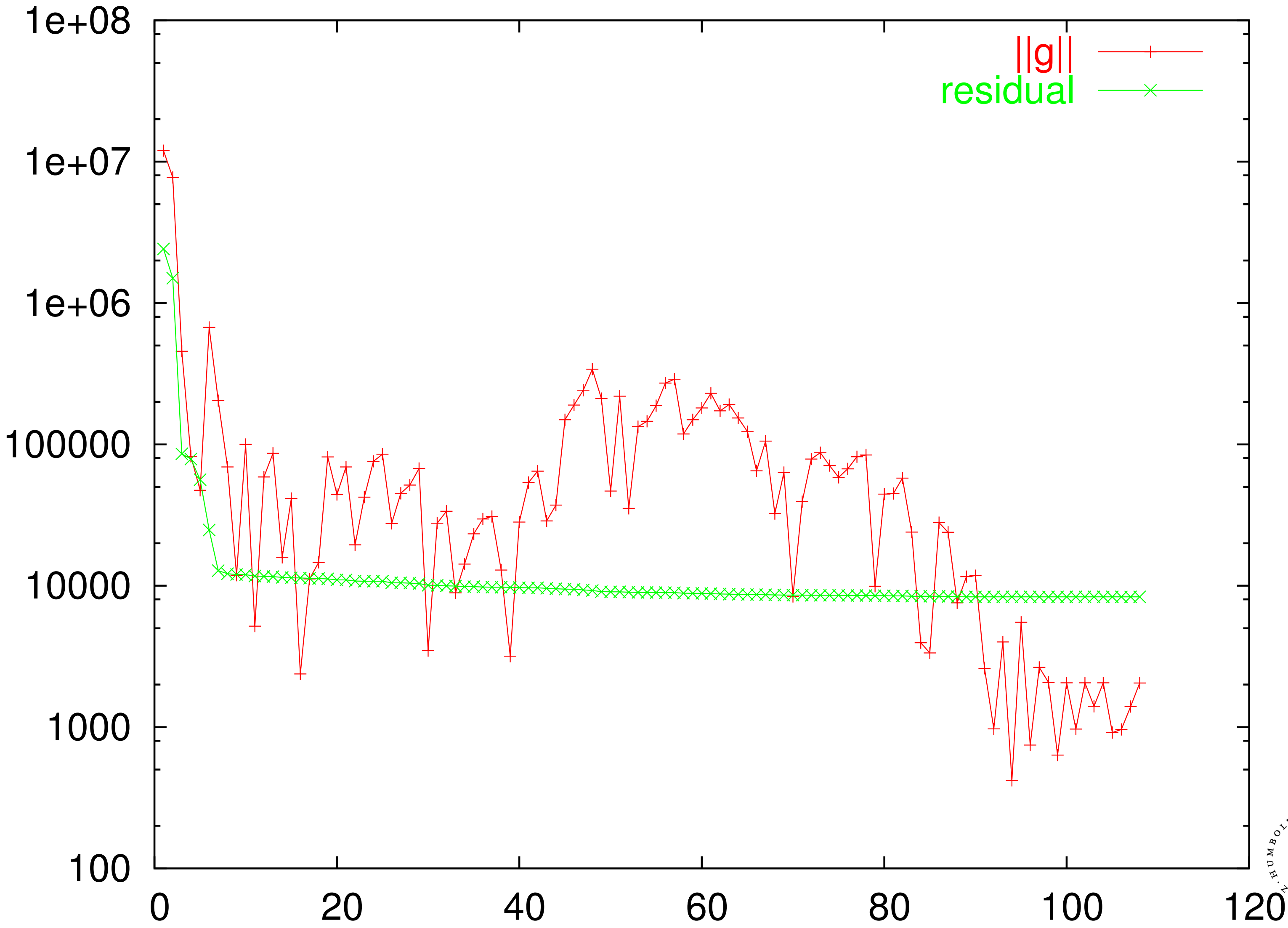
FastOpt Max-Planck-Institut für Biogeochemie CSIRO MIT ECCO

Thanks to Thomas Kaminski, FastOpt

quasi-Gauss-Newton - $n = 53, \ m = 500$ from scaled identity



quasi-Gauss-Newton - $n = 53$, $m = 10000$ from scaled identity



Total quasi–Newton for NLP

- Min $f(x)$ s.t. $c(x) = 0 \in \mathbb{R}^m$

  locally equivalent to solving KKT system

$$0 = \nabla_{x,\lambda} L(x, \lambda) \quad \text{with} \quad L(x, \lambda) \equiv f(x) + \lambda^T c(x).$$

- Approximations $A_k \approx c'(x_k)$ and $B_k \approx \nabla_k^2 L(x_k, \lambda_k)$ yield total-quasi-Newton steps (s_k, σ_k) by

$$\begin{bmatrix} B_k & A_k^\top \\ A_k & 0 \end{bmatrix} \begin{bmatrix} s_k \\ \sigma_k \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) + c'(x_k)^\top \lambda \\ c(x_k) \end{bmatrix}$$

  where RHS is evaluated exactly but cheaply.
- Implementation via nullspace factorization

$$A_k = [L_k \, , \, 0] \cdot [Y_k \, , \, Z_k]^\top, \quad Z_k^\top B_k Z_k = C_k C_k^\top$$

  with $Y_k^\top Z_k = 0$ or more economical LU variant.

$A_k \to A_{k+1}$ by TR1 and $B_k \to B_{k+1}$ by SR1 yield

- Reduction in linear algebra from $O(mn^2)$ to $O(m+n)^2$

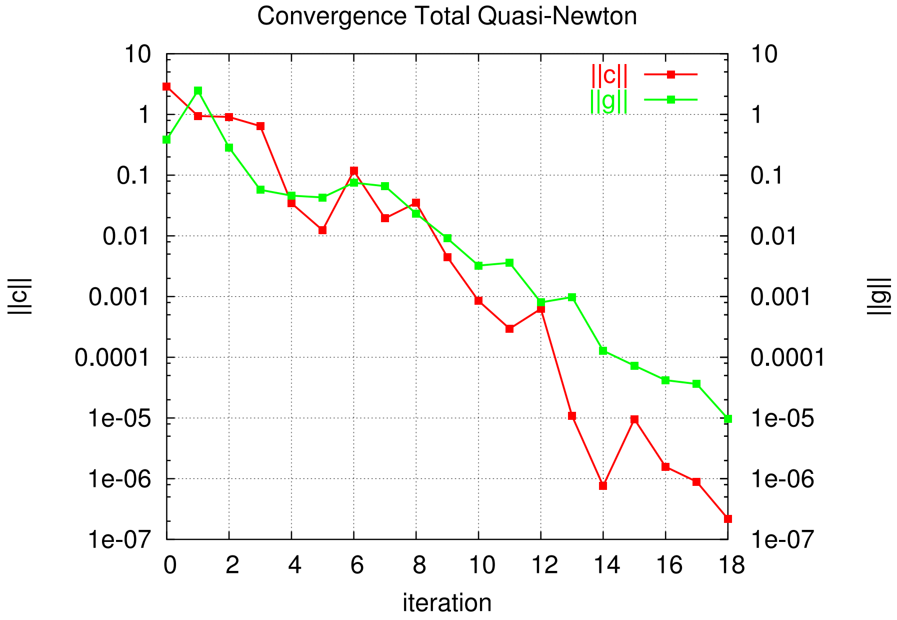
- Invariance to linear transformations on domain and range of $c(x) = 0$.

- Heredity and thus finite termination on quadratic programs.

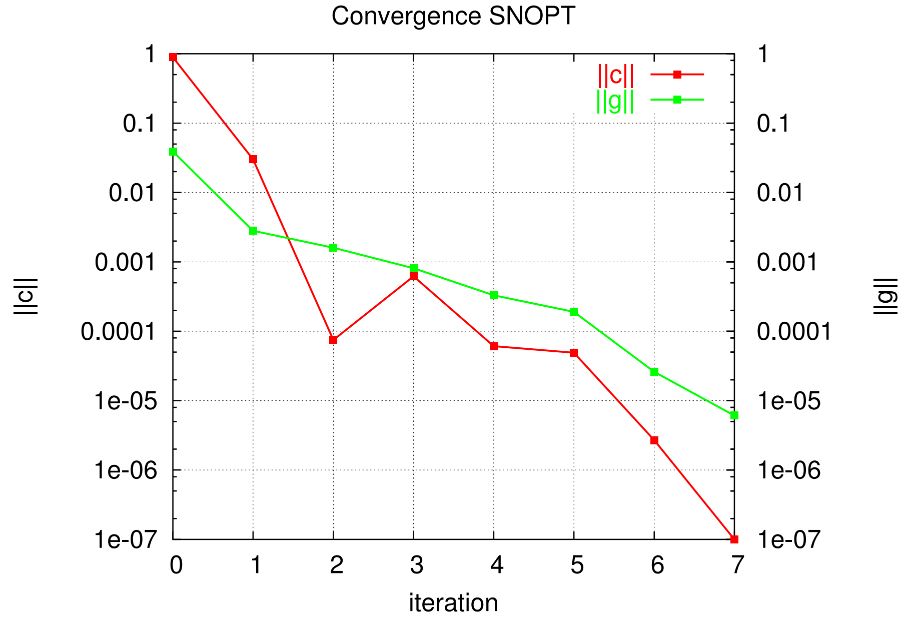
- Local and superlinear convergence under some additional assumptions.

- Challenges to the algorithm designer with respect to inequality handling and forcing of global convergence.

HS99 problem from CUTEr test set
(NLP, 7 variables, 16 constraints)



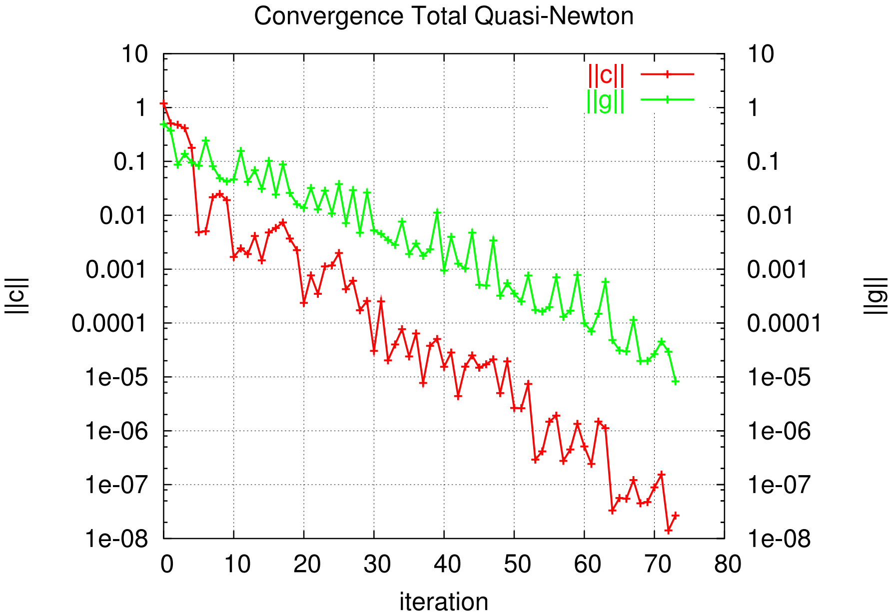
18 iterations

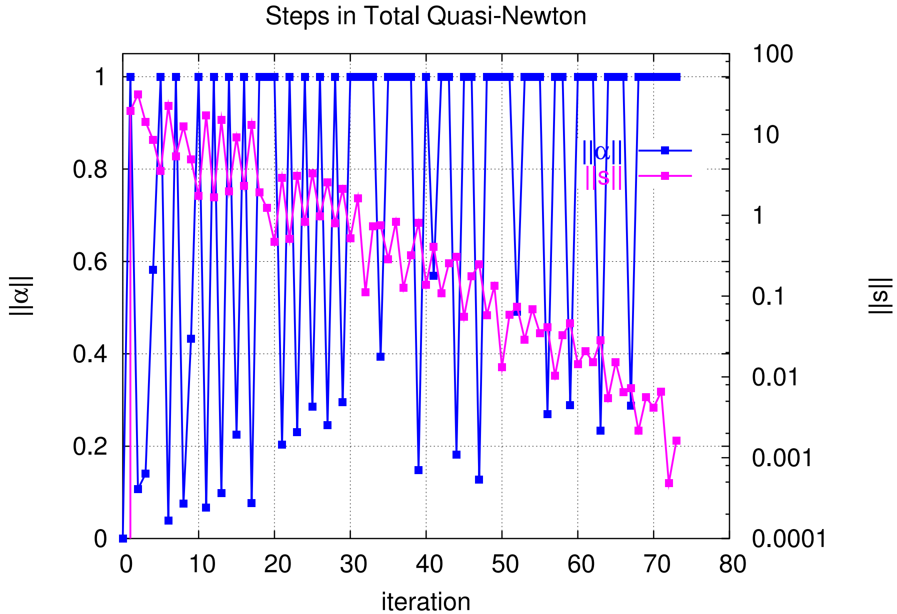


7 iterations

Convergence Behaviour of Total Quasi-Newton

CHAIN problem from CUTEr test set
(NLP, 102 variables, 53 constraints)

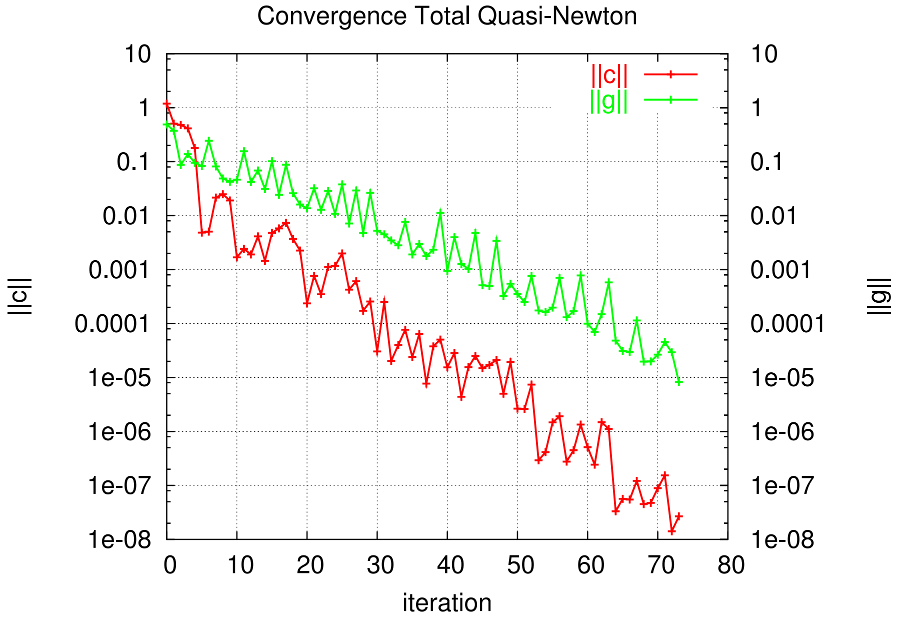




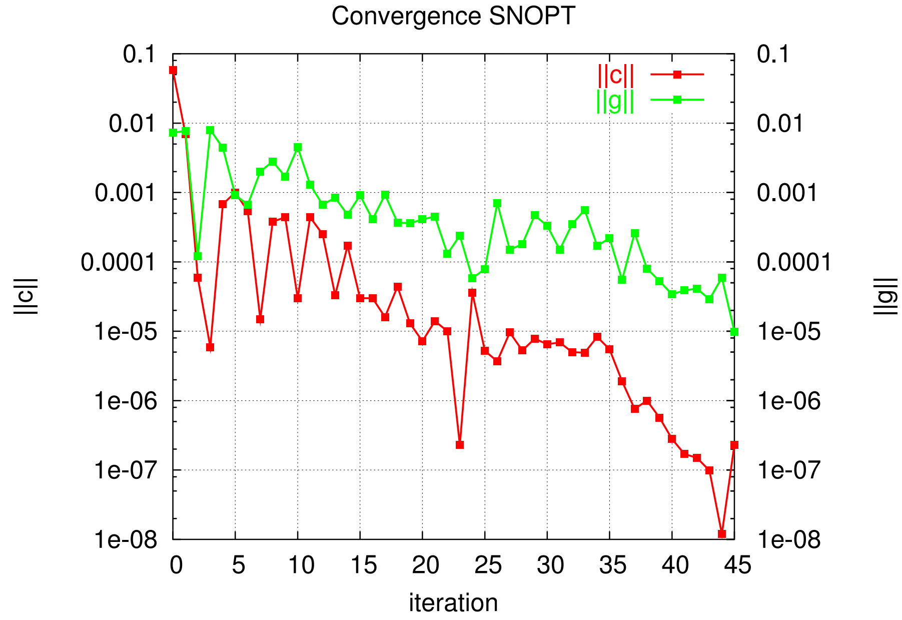
73 iterations

Comparison with SNOPT

CHAIN problem from CUTEr test set
(NLP, 102 variables, 53 constraints)



73 iterations



45 iterations

Intermediate Summary and Conclusion

- Adjoint information eliminates dependence on norms and scaling.
- Adjoint information yields exact right hand sides without Jacobians.
- For NLLS and NLP somewhat more but much cheaper steps.
- Storage and effort of $O((n+m)^2)$ reducable to $O((n+m)k)$ by limited memory variants under development.
- On large scale problems preconditioning must ensure relative compactness of initial discrepancy

(Almost) Matrix–free Design Optimization

3. (Almost) Matrix–free Design Optimization
   - Implicit and Iterative Differentiation
   - Two phase method on TAUij Code (Walter)
   - Preconditioning Task in One–Shot Approach

Implicit and Iterative Differentiation

- Optimal Design Scenario

$$\text{Min } f(y,u) \quad \text{such that} \quad c(y,u) = 0$$

where $\dim(c) = \ell = \dim(y) \gg \dim(u) = n$.

- Feasibility restored by user povided slow solver

$$y_{k+1} = G(y_k, u) : \mathbb{R}^\ell \times \mathbb{R}^n \to \mathbb{R}^\ell$$

with $G(y,u) = y \iff c(y,u) = 0.$

- Contractivity assumption on $G_y \equiv \frac{\partial G}{\partial y}$

$$\|G_y(y,u)\| \le \rho < 1 \quad \text{for some} \quad \|\cdot\|$$

implies by BFT

$$y_k \underset{k}{\rightarrow} y_* = y_*(u) \quad \text{with} \quad c(y_*, u) = 0.$$

Shifted Lagrangian Function

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

yields by algorithmic differentiation at similar cost the adjoint iteration

$$\bar{y}_{k+1} = N_y(\tilde{y}_k, \bar{y}_k, u) \quad \text{with} \quad \partial \bar{y}_{k+1} / \partial \bar{y}_k = G_y^\top$$

Contraction to adjoint solution

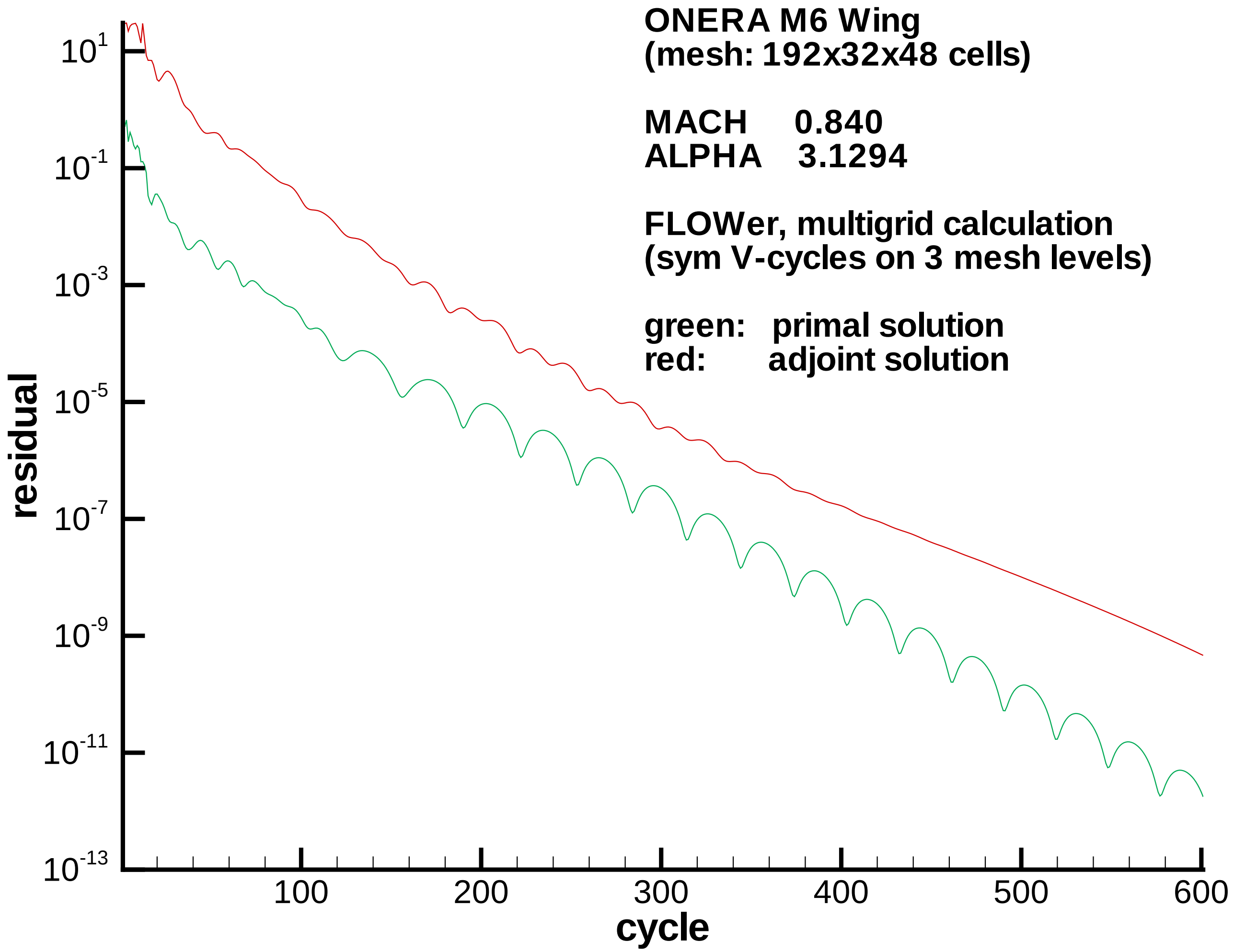
$$\bar{y}_* = \bar{y}_*(u) \quad \text{with} \quad \left[I - G_y^\top(y_*, u)\right] \bar{y}_* = f_y(y_*, u)$$

Single-Phase: $\tilde{y}_k = y_k$ current iterate
Two-Phase: $\tilde{y}_k = y_\infty$ final iterate

By IFT one obtains as reduced gradient

$$\bar{u}_{k+1} = N_u(\tilde{y}_k, \bar{y}_k, u) \quad \underset{k}{\longrightarrow} \quad \frac{d}{du} f(y_*(u), u)$$

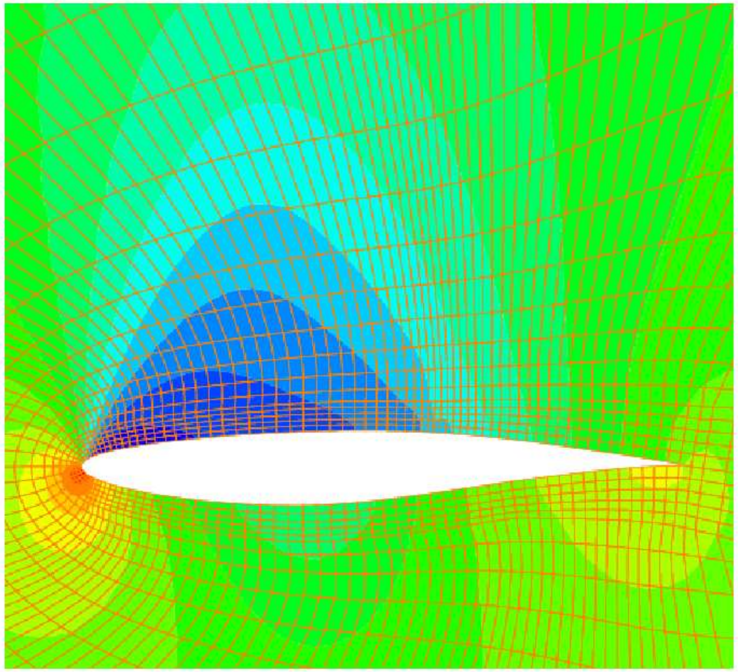
Implicit and Iterative Differentiation



Feasible Point Method on TAUij Code (Walter)

CFD code

- provided by DLR (N. Kroll, N. Gauger)
- 2D Euler Equation
- Finite Volume discretization with multi-grid acceleration
- Runge-Kutta scheme for pseudo time integration
- derivative computation with ADOL-C



Thanks to Andrea Walter, TU Dresden

TAUij – Direct model

Code structure

1. grid generation and initialization
2. time step integration as fixed point iteration
3. evaluation of objective function

Properties

- approximately 5.000 lines of C/C++ code
- model requires at least 2000 iterations to reach quasi-steady state

Task

Optimize drag with respect to 20 shape parameters

Thanks to Andrea Walter, TU Dresden

TAUij – Adjoint model

Two configurations for discretization

- small: 161×33 grid points
- medium: 321×65 grid points

Complexity (one time step)

- $70 \times 10^6 / 285 \times 10^6$ active variables (small/medium)
- $20 \times 10^6 / 116 \times 10^6$ operations (small/medium)

Memory requirements

- 344 MB/2.3 GB memory per step (small/medium)

Thanks to Andrea Walter, TU Dresden

TAUij – Run times

small configuration

	PC	Cluster
2000 direct steps	2 min 39 sec	1 min 44 sec
1000 adjoint steps	1 h 21 min 1 sec	15 min 8 sec

medium configuration

	PC	Cluster
2000 direct steps	11 min 36 sec	7 min 40 sec
1000 adjoint steps	≈5 days	1 h 5 min 3 sec

Cluster: RUNTIME(Derivative) $\leq 9 \times$ RUNTIME(Function)

PC: AMD Athlon 3200, 1 GB RAM

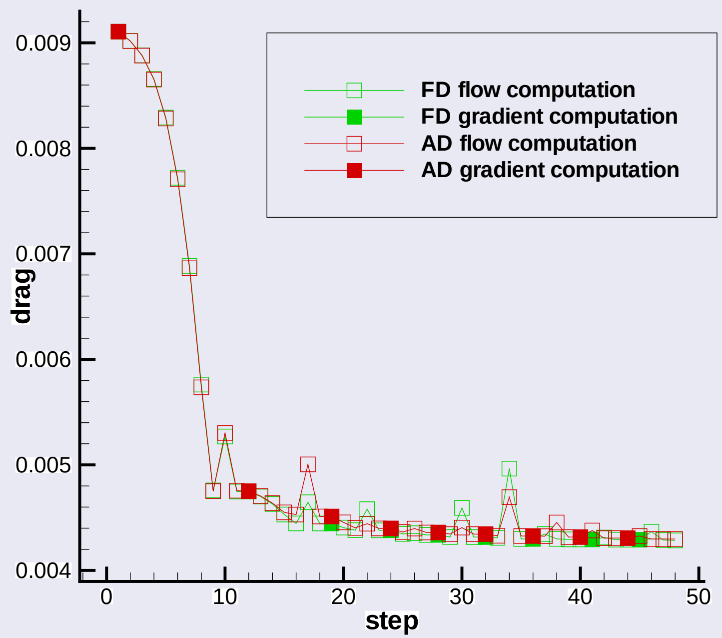
Cluster: single node, Dual AMD Opteron 240 (1.4GHz), 12 GB RAM, fast memory access

A.Walter, TU Dresden

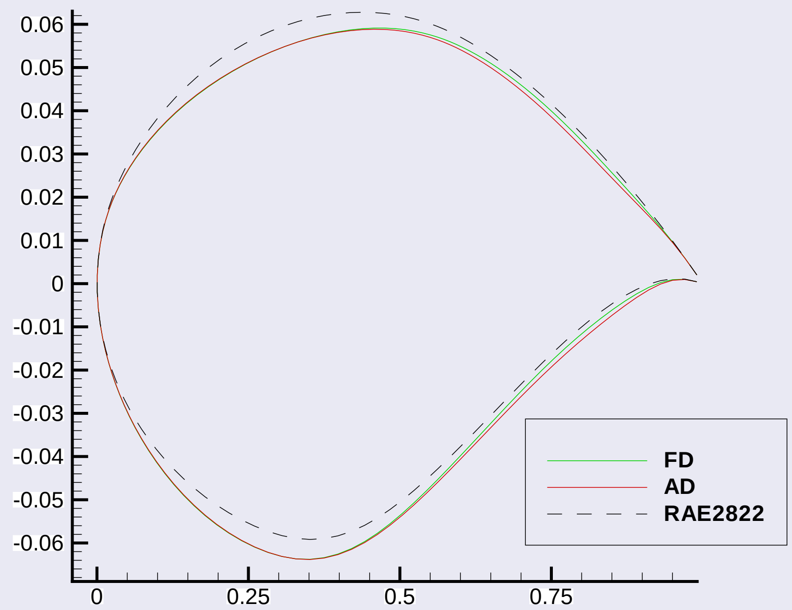
TAUij – Numerical results (small configuration)

- Comparing Finite Differences (FD) to Automatic Differentiation (AD)

Optimization history



Shape deformation



Thanks to Andrea Walter, TU Dresden

Preconditioning Task in One–Shot Approach

$$y_{k+1} = G(y_k, u_k) \qquad \Longrightarrow \text{ primal feasibility}$$

$$\bar{y}_{k+1} = N_y(y_k, \bar{y}_k, u_k) \qquad \Longrightarrow \text{ adjoint feasibility}$$

$$\bar{u}_{k+1} = N_u(y_k, \bar{y}_k, u_k)$$

$$u_{k+1} = u_k - B_k^{-1}\bar{u}_{k+1} \qquad \Longrightarrow \text{ optimality}$$

where $0 \prec B_k = B_k^T \in \mathbb{R}^{n\times n}$

"Preconditioner" B_k should ensure:

- Asymptotic convergence rate in $[\rho, 1)$ (bounded retardation)
- Descent of $\left(G - y, N_y - \bar{y}, -B^{-1}N_u\right)$ w.r.t. augmented Lagrangian

$$\mathcal{L} \equiv \frac{\alpha}{2}\|G - y\|_2^2 + \frac{\beta}{2}\|N_y - \bar{y}\|_2^2 + N - \bar{y}^\top y$$

Preliminary Observations

- Suitable B exist but depend on second derivatives of N.
- Reduced Hessian $\nabla^2_u f\left(y_*(u), u\right) \in \mathbb{R}^{n \times n}$ leads to divergence.
- Invariance w.r.t. linear transformations on design $u \in \mathbb{R}^n$ achievable.
- Good preconditioning involves as yet $\dim(y) \cdot \dim(u) = \ell \times n$ matrices.

Conclusion and Summary

- Sensitivities for complex, iterative codes obtainable.
- Calculus based optimization remains feasible and promising.
- 'Right hand side' vectors should be evaluated exactly.
- 'Left hand side' matrices should be approximated or avoided.
- Transition from simulation to optimization still too laborious.

Acknowledgements/Pointers

Thanks to HU Group:

J. Kerger, S. Körkel, L. Lehmann, J. Riehme, Cl. Tutsch

Funding Sources

- MATHEON C12
- SPP 1253

Software Pointers

- C/C++: ADOL-C TU Dreden
- Fortran: FastOpt Hamburg
- General AD stuff: autodiff.org
- ORMS: Oberwolfach Repository of Mathematical Software