One-Shot Optimization with Steady and Unsteady PDEs

Prof. Dr. Nicolas R. Gauger

Chair for Scientific Computing
University of Kaiserslautern
67663 Kaiserslautern, Germany

nicolas.gauger@scicomp.uni-kl.de

Outline

- Consistent and Robust Discrete Adjoints
- Application to Separation Control on the 3D High-Lift Configuration HIREX
- One-Shot Approach for Optimization with Steady PDEs
- Adjustments for Optimization with Unsteady PDEs
- One-Shot Optimization with Unsteady RANS
- Improving the Efficiency
- Numerical Results for Van der Pol Oscillator and Advection-Diffusion Equation

Outline

- Consistent and Robust Discrete Adjoints
- Application to Separation Control on the 3D High-Lift Configuration HIREX
- One-Shot Approach for Optimization with Steady PDEs
- Adjustments for Optimization with Unsteady PDEs
- One-Shot Optimization with Unsteady RANS
- Improving the Efficiency
- Numerical Results for Van der Pol Oscillator and Advection-Diffusion Equation

Optimality System

- **Optimization Problem:**

$$\min_{\phi\in\Phi} J(W,\phi) \quad s.t. \quad R(W,\phi)=0$$

- **Lagrangian:**

$$L = J + \Lambda^T R$$

- **Optimality condition (KKT system, 1. order necessary cond.):**

$$\frac{\partial L}{\partial \Lambda} = R \overset{!}{=} 0$$ State equation

$$\frac{\partial L}{\partial W} = \frac{\partial J}{\partial W} + \Lambda^T \frac{\partial R}{\partial W} \overset{!}{=} 0$$ Adjoint state equation

$$\frac{\partial L}{\partial \phi} = \frac{\partial J}{\partial \phi} + \Lambda^T \frac{\partial R}{\partial \phi} \overset{!}{=} 0$$ Design equation

Optimality System

- **Optimization Problem:**

$$\min_{\phi \in \Phi} J(W,\phi) \quad s.t. \quad R(W,\phi) = 0$$

- **Lagrangian: instead** $L = J + \Lambda^T R$**, continuous** L**:**

$$L = J + \langle \Lambda, R \rangle_{H^*,H}$$

- **Optimality condition (KKT system, 1. order necessary cond.):**

$$\frac{\partial L}{\partial \Lambda} = R \overset{!}{=} 0$$ State equation

$$\frac{\partial L}{\partial W} = \frac{\partial J}{\partial W} + \Lambda^T \frac{\partial R}{\partial W} \overset{!}{=} 0$$ Adjoint state equation

$$\frac{\partial L}{\partial \phi} = \frac{\partial J}{\partial \phi} + \Lambda^T \frac{\partial R}{\partial \phi} \overset{!}{=} 0$$ Design equation

Optimality System

- **Optimization Problem:**

$$\min_{\phi\in\Phi} J(W,\phi) \quad s.t. \quad R(W,\phi)=0 \quad \leftrightarrow$$

Fixed point iteration:

$$W = G(W,\phi)$$

- **Lagrangian: instead $L = J + \Lambda^T R$, continuous or discrete L:**

$$L = J + \langle \Lambda, R \rangle_{H^*,H} \quad \leftrightarrow \quad L(W,\Lambda,\phi) = J(W,\phi) + \Lambda^T (G(W,\phi) - W)$$

- **Optimality condition (KKT system, 1. order necessary cond.):**

$$\frac{\partial L}{\partial \Lambda} = R \overset{!}{=} 0$$ State equation

$$\frac{\partial L}{\partial W} = \frac{\partial J}{\partial W} + \Lambda^T \frac{\partial R}{\partial W} \overset{!}{=} 0$$ Adjoint state equation

$$\frac{\partial L}{\partial \phi} = \frac{\partial J}{\partial \phi} + \Lambda^T \frac{\partial R}{\partial \phi} \overset{!}{=} 0$$ Design equation

Optimality System

- **Optimization Problem:**

$$\min_{\phi\in\Phi} J(W,\phi) \quad s.t. \quad R(W,\phi)=0 \quad \leftrightarrow$$

Fixed point iteration:
$$W = G(W,\phi)$$

- **Lagrangian: instead** $L = J + \Lambda^T R$**, continuous or discrete** L**:**

$$L = J + \langle \Lambda, R \rangle_{H^*,H} \quad \leftrightarrow \quad L(W,\Lambda,\phi) = J(W,\phi) + \Lambda^T (G(W,\phi) - W)$$

- **Optimality condition (KKT system, 1. order necessary cond.):**

Black-box differentiation:

$$\frac{\partial L}{\partial \Lambda} = G(W,\phi) - W \overset{!}{=} 0$$

State equation

$$\frac{\partial L}{\partial W} = \frac{\partial J}{\partial W} + \Lambda^T \left(\frac{\partial G}{\partial W} - I \right) \overset{!}{=} 0 \Leftrightarrow N_W^T(W,\Lambda,\phi) = \Lambda$$

Adjoint state equation

$$\frac{\partial L}{\partial \phi} = \frac{\partial J}{\partial \phi} + \Lambda^T \frac{\partial G}{\partial \phi} \overset{!}{=} 0$$

Design equation

Optimality System

- **Optimization Problem:**

$$\min_{\phi\in\Phi} J(W,\phi) \quad s.t. \quad R(W,\phi)=0 \quad \Leftrightarrow$$

Fixed point iteration:
$$W = G(W,\phi)$$

- **Lagrangian: instead** $L = J + \Lambda^T R$**, continuous or discrete** L**:**

$$L = J + \langle \Lambda, R\rangle_{H^*,H} \quad \Leftrightarrow \quad L(W,\Lambda,\phi) = J(W,\phi) + \Lambda^T (G(W,\phi) - W)$$

Black-box differentiation:

$$G(W,\phi) - W \overset{!}{=} 0$$

$$\frac{\partial J}{\partial W} + \Lambda^T \left(\frac{\partial G}{\partial W} - I \right) \overset{!}{=} 0 \Leftrightarrow N_W^T(W,\Lambda,\phi) = \Lambda$$

Primal contractivity: $\|G_W\| = \|G_W^T\| \le \rho < 1 \;\Rightarrow$ Adjoint contractivity:

$$\left\| \frac{\partial N_W^T}{\partial \Lambda} \right\| = \|G_W^T\| \le \rho < 1$$

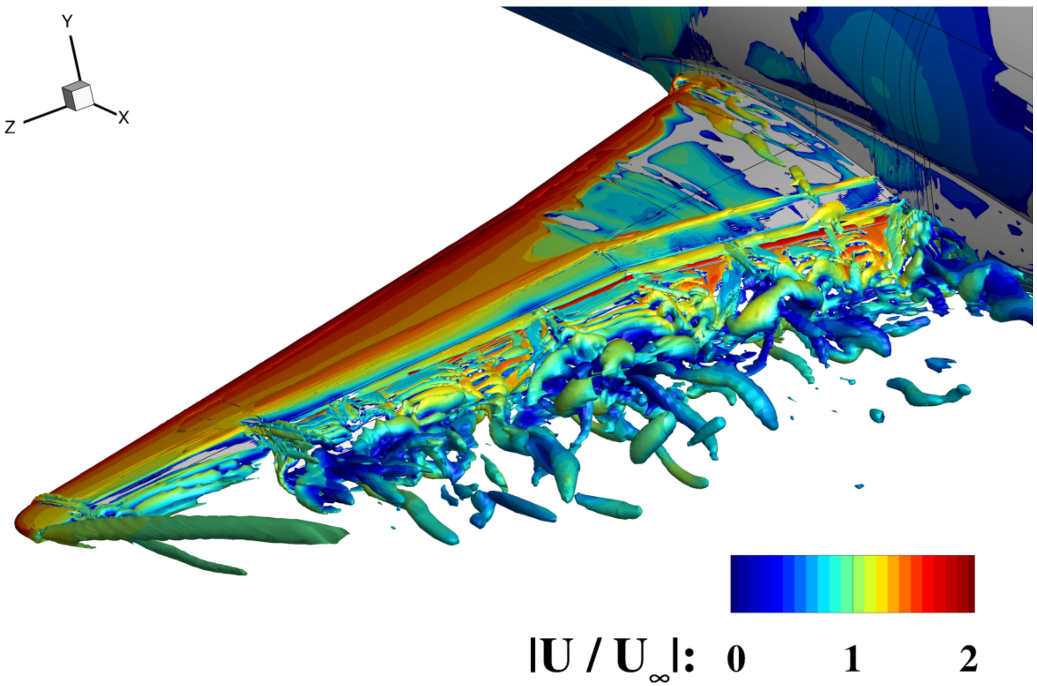
Adjoint code inherits convergence properties of primal code

Outline

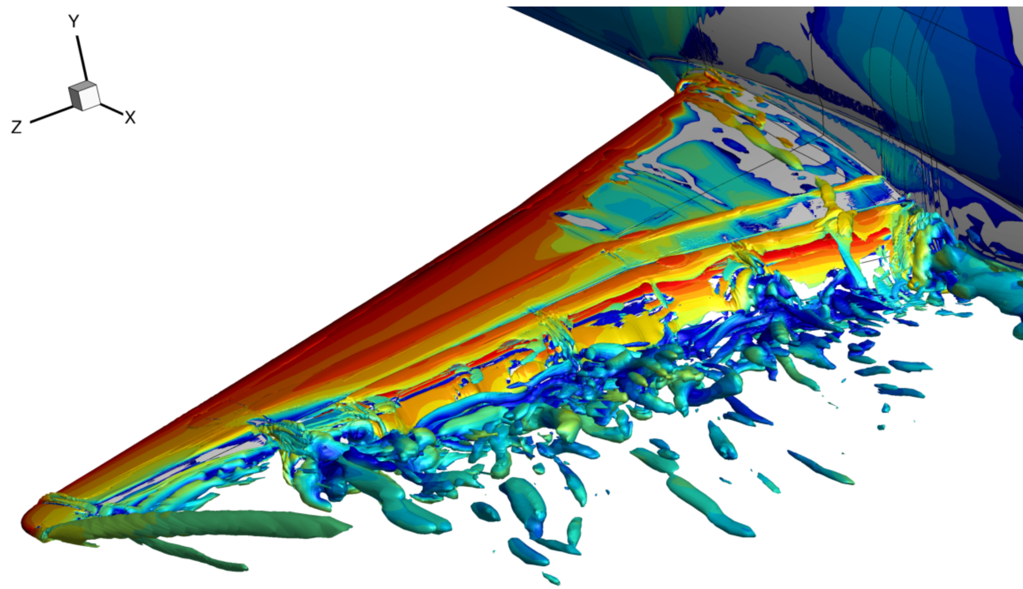
- Consistent and Robust Discrete Adjoints
- Application to Separation Control on the 3D High-Lift Configuration HIREX
- One-Shot Approach for Optimization with Steady PDEs
- Adjustments for Optimization with Unsteady PDEs
- One-Shot Optimization with Unsteady RANS
- Improving the Efficiency
- Numerical Results for Van der Pol Oscillator and Advection-Diffusion Equation

Optimal Separation Control on Airbus HIREX Configuration

(Primal) Flow Simulation



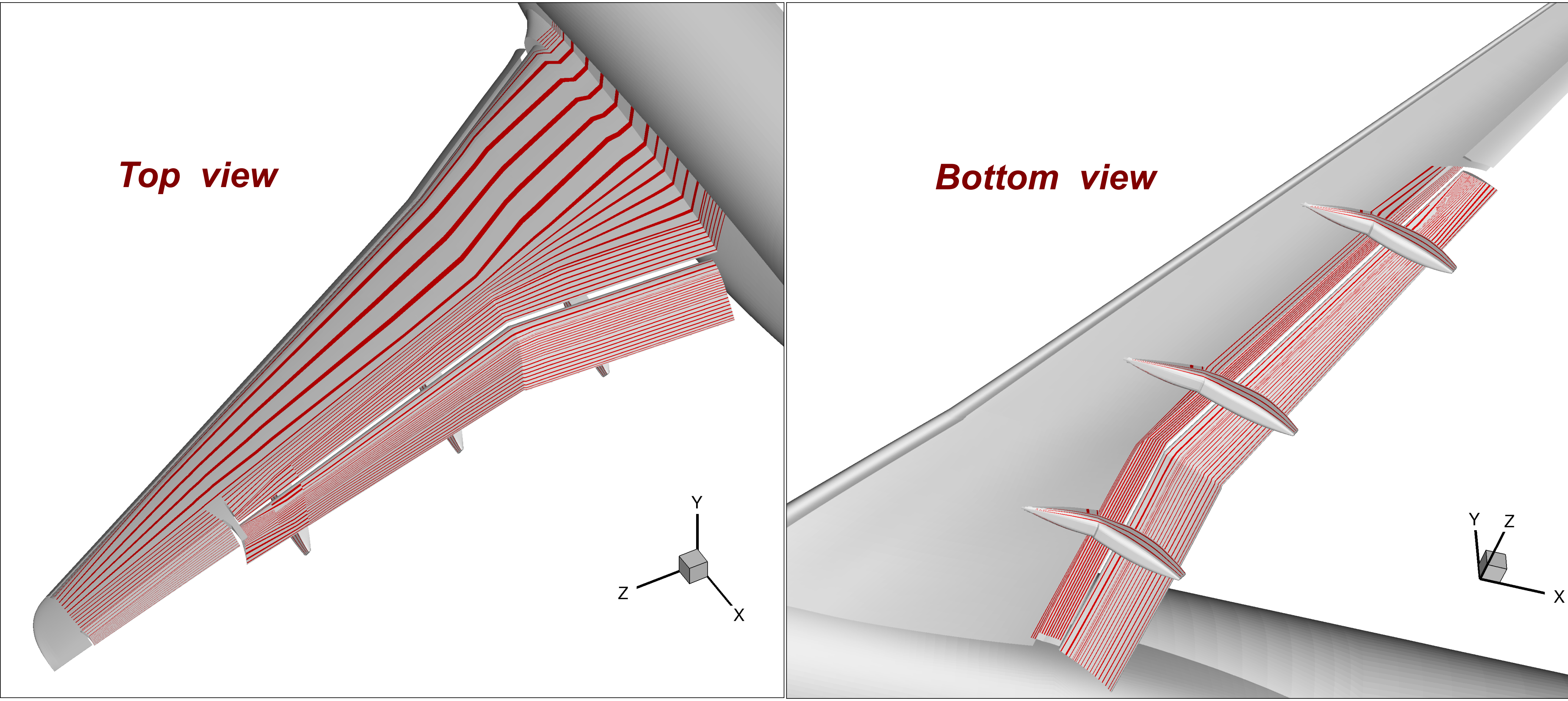
Un-actuated flow



Actuated flow

- *Grid size : 48 Million cells*
- *Re= : 1.5 x 10^6 , AoA=7^o*
- *ELAN Code (TU Berlin)*

Actuation slot distribution on HIREX



- **33, 441 synthetic jet actuators on wing and flap**
- **No. of control variables : 167, 205**

Actuation boundary condition:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = A\left[\vec{n} + \frac{1}{\tan\beta_1}\vec{t}_1 + \frac{1}{\tan\beta_2}\vec{t}_2\right]\cos\left(2\pi f\left(t - t_0\right)\right)$$

Control variables:

$$\begin{pmatrix} A & - \text{Amplitude} \\ \beta_{1,2} & - \text{Angles} \\ t_0 & - \text{Phase shift} \\ f & - \text{Frequency} \end{pmatrix}$$

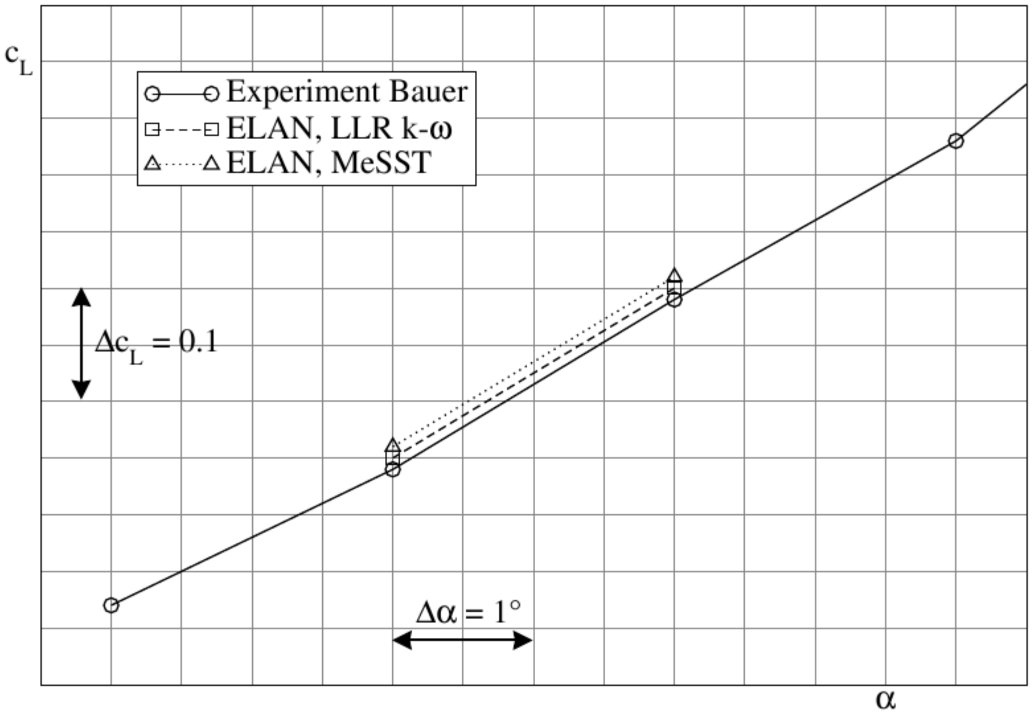
Flow Solver and Techniques for Algorithmic Differentiation

- *RANS flow solver: ELAN (TU Berlin)*
  - *Block-structured, FVM, incompressible, SIMPL*
  - *Fully implicit, MPI based parallelisation*
  - *Turbulence model : SST k-ω, LLR k-ω, …*
  - *Coded in Fortran*
- *AD tool for adjoint : TAPENADE (INRIA Sophia – Antipolis)*
- *Optional: Reverse Accumulation for SIMPL loops* [Christianson]
- *Checkpointing by REVOLVE* [Griewank, Walther]
  - *Usable for Fortran and C*

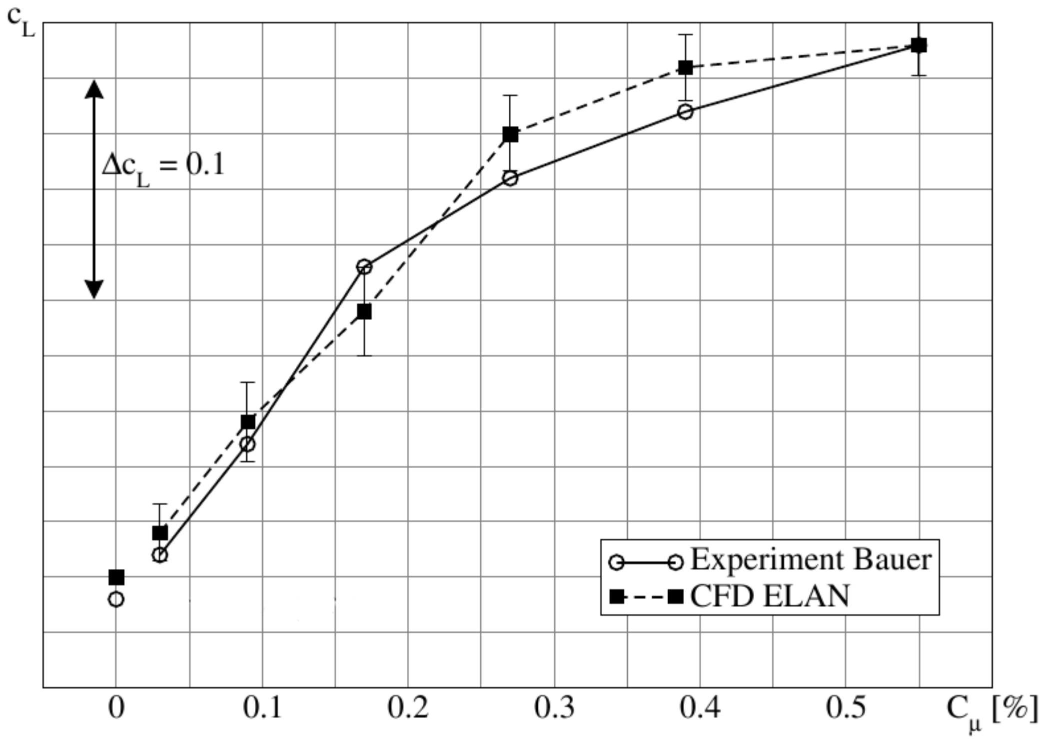
Optimal Separation Control on Airbus HIREX Configuration

(Primal) Flow Simulation – Validation (Team: RWTH/TU KL, TU Berlin, Airbus)

TFB / SFB 557, DFG GA 857/5-1



Variation of mean lift with angle of attack



Variation of mean lift with actuation intensity

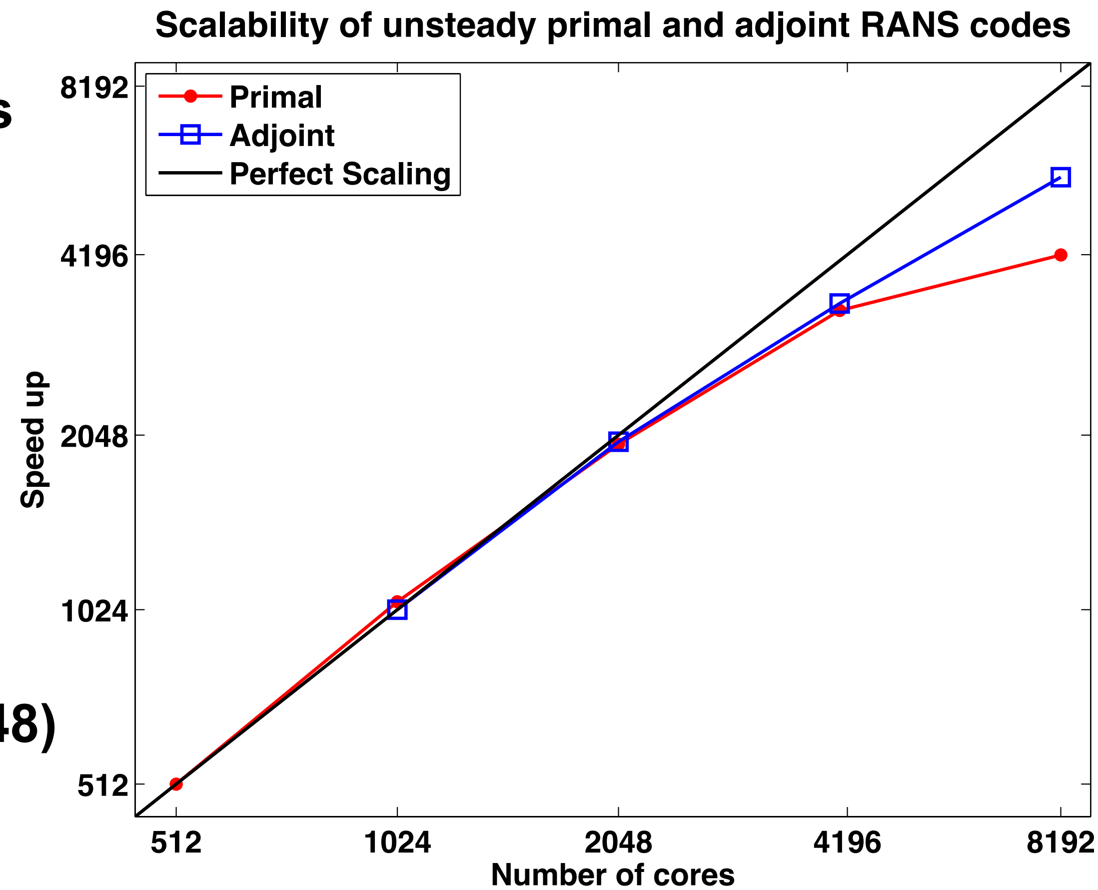
- *Grid size : 48 Million cells*
- *Re= : 1.5×10^6 , AoA=7^o*

Scalability of ELAN Code (Primal / Adjoint)

- Simulations are performed on JUQUEEN (IBM Blue Gene/Q) at Jülich, Germany
- IBM PowerPC A2, 16 cores/node, 1.6 GHz, 1GB RAM/core
- Number of cores : 458,752
- Peak performance : ~6 Peta Flops

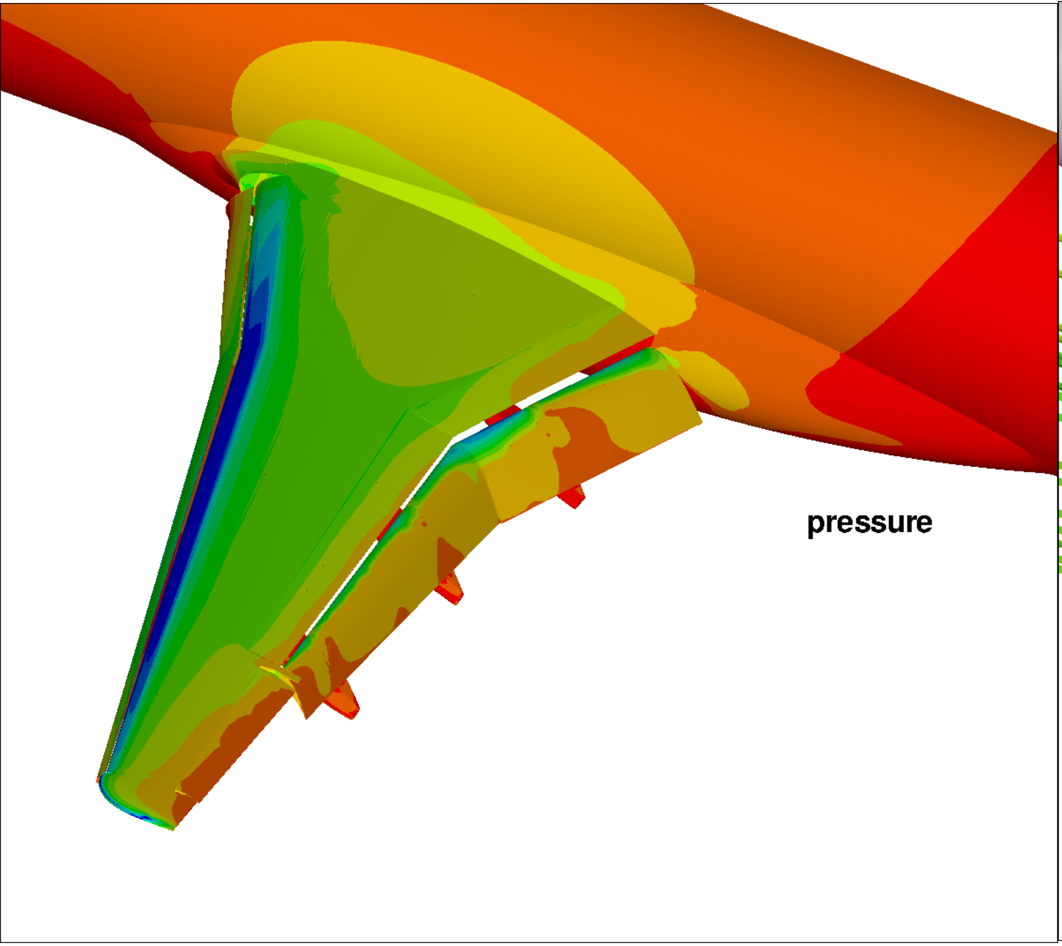
- No. of cells : 33,554,432
- No. of blocks : 8,192
- Cells/core on 8,192 cores : 4,096
- Cells/core on 512 cores : 65,536

- (Cells/core on 16,384 cores : 2,048)

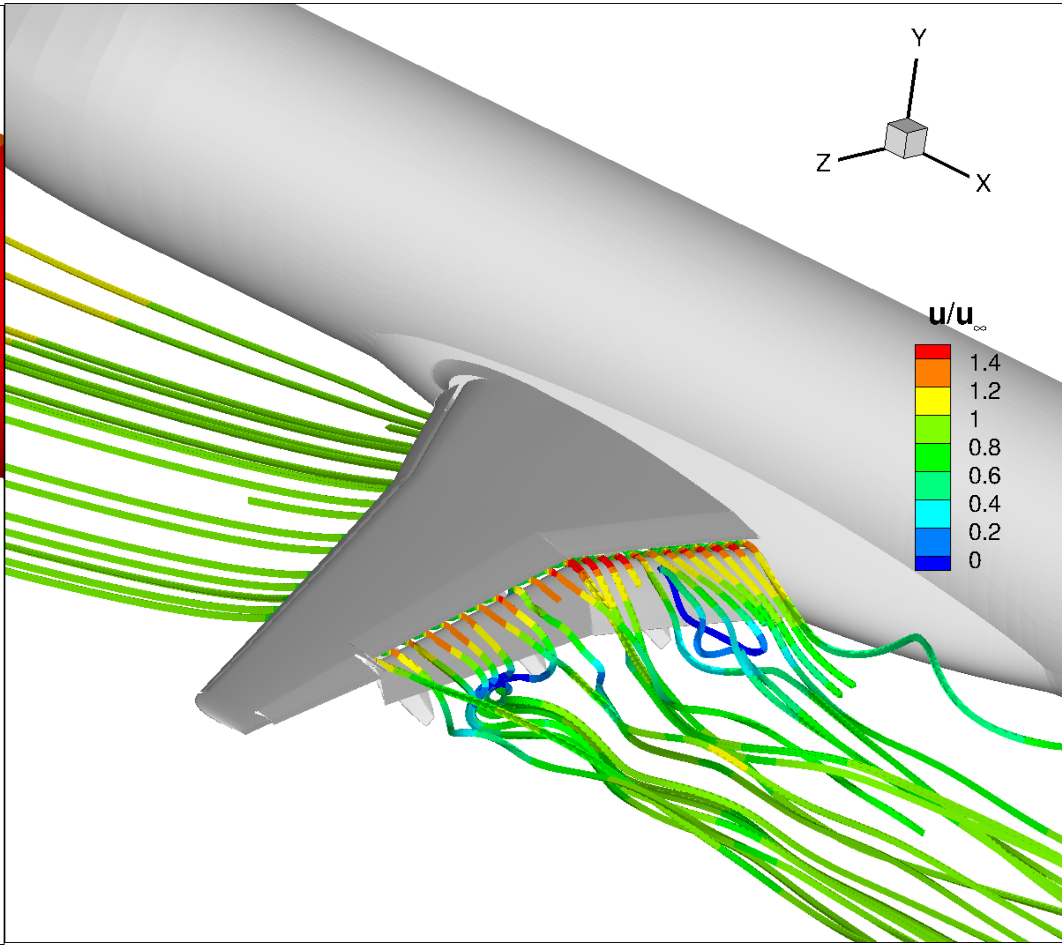


Numerical Results

Initial actuated flow: $A / u_\infty = 0.00736,\ f = 200Hz,\ t_0 = 0,\ \beta_{1,2} = 90^o$



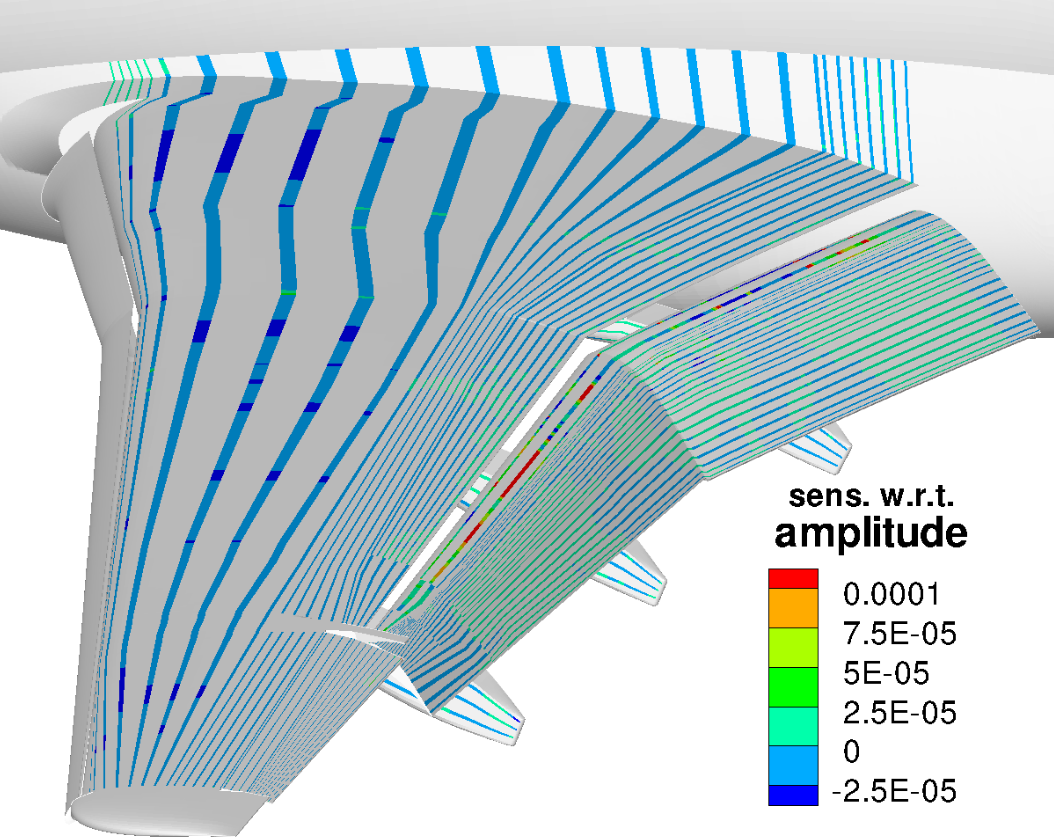
Pressure contours



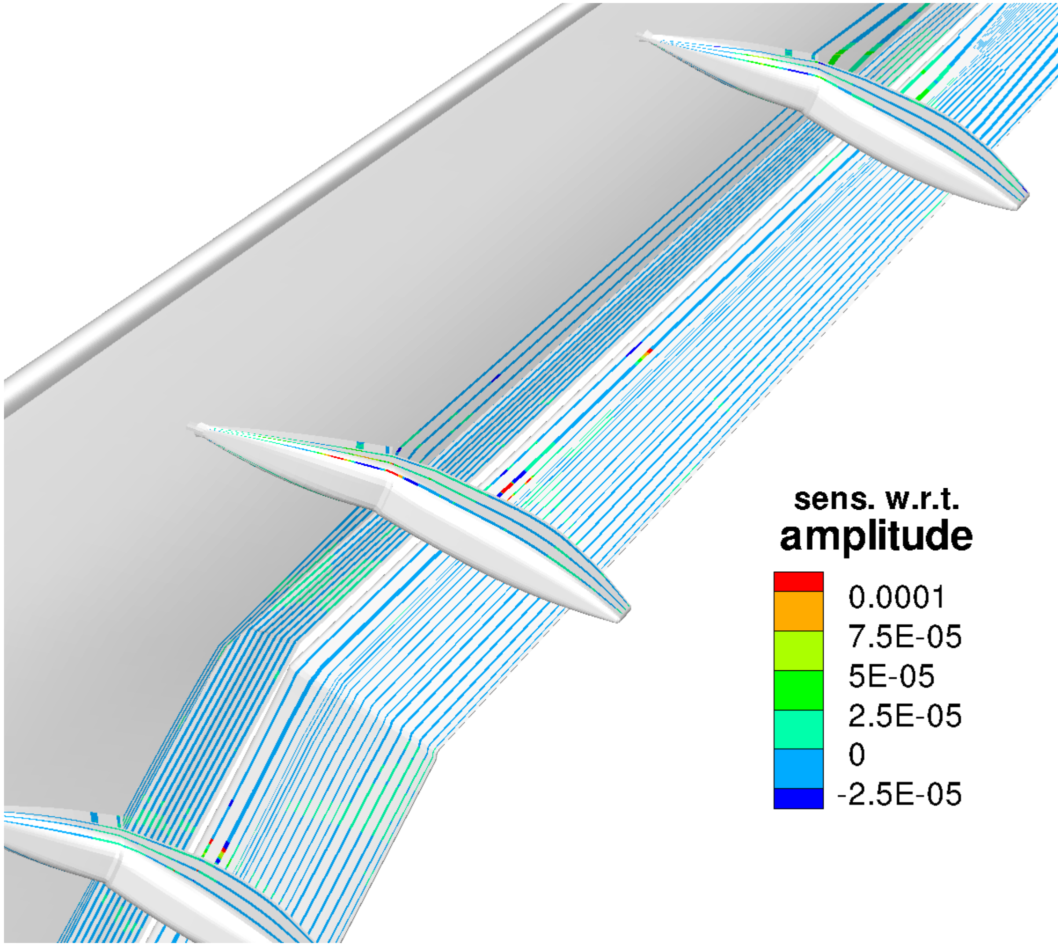
Streamlines

Numerical Results

Amplitude sensitivities for the initial actuated flow



Top view



Bottom view

[Nemili, Özkaya, Gauger, Kramer, Höll, Thiele, 2014]

Numerical Results

Validation of the 3D discrete adjoint URANS solver

Control parameter	Forward code	Adjoint code
Amplitude	-1.172328410564145E-05	-1.172328410564145E-05
Frequency	-2.753956866987843E-05	-2.753956866987779E-05
Phase shift	2.101437468727700E-03	2.101437468727652E-03
Blowing angle β_1	-5.469583093515221E-12	-5.469583093515383E-12
Blowing angle β_2	3.607784441402103E-12	3.607784441402158E-12

Comparison of sensitivities at a randomly selected actuation slot

[Nemili, Özkaya, Gauger, Kramer, Höll, Thiele, 2014]

Numerical Results

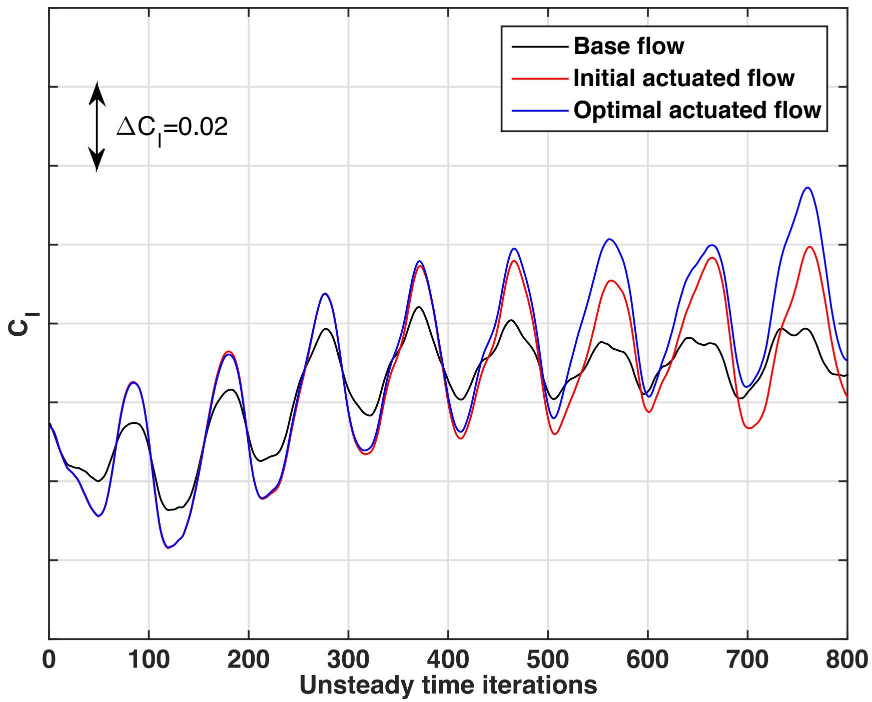
Optimal active flow control on HIREX

Objective function

$$J = \bar{C}_l = \frac{1}{N - N^*} \sum_{n=N^*+1}^{N} C_l^n$$

$$N = 800 \quad \text{and} \quad N^* = 250$$

- **Optimizer: Method of steepest ascent**
- **Number of cycles 11**
- **Optimal actuation increased the mean lift**
  by 90 counts over the base flow
  by 60 counts over the initial actuation



Comparison of lift for base flow, Initial and optimal actuated flow

[Nemili, Özkaya, Gauger, Kramer, Höll, Thiele, 2014]

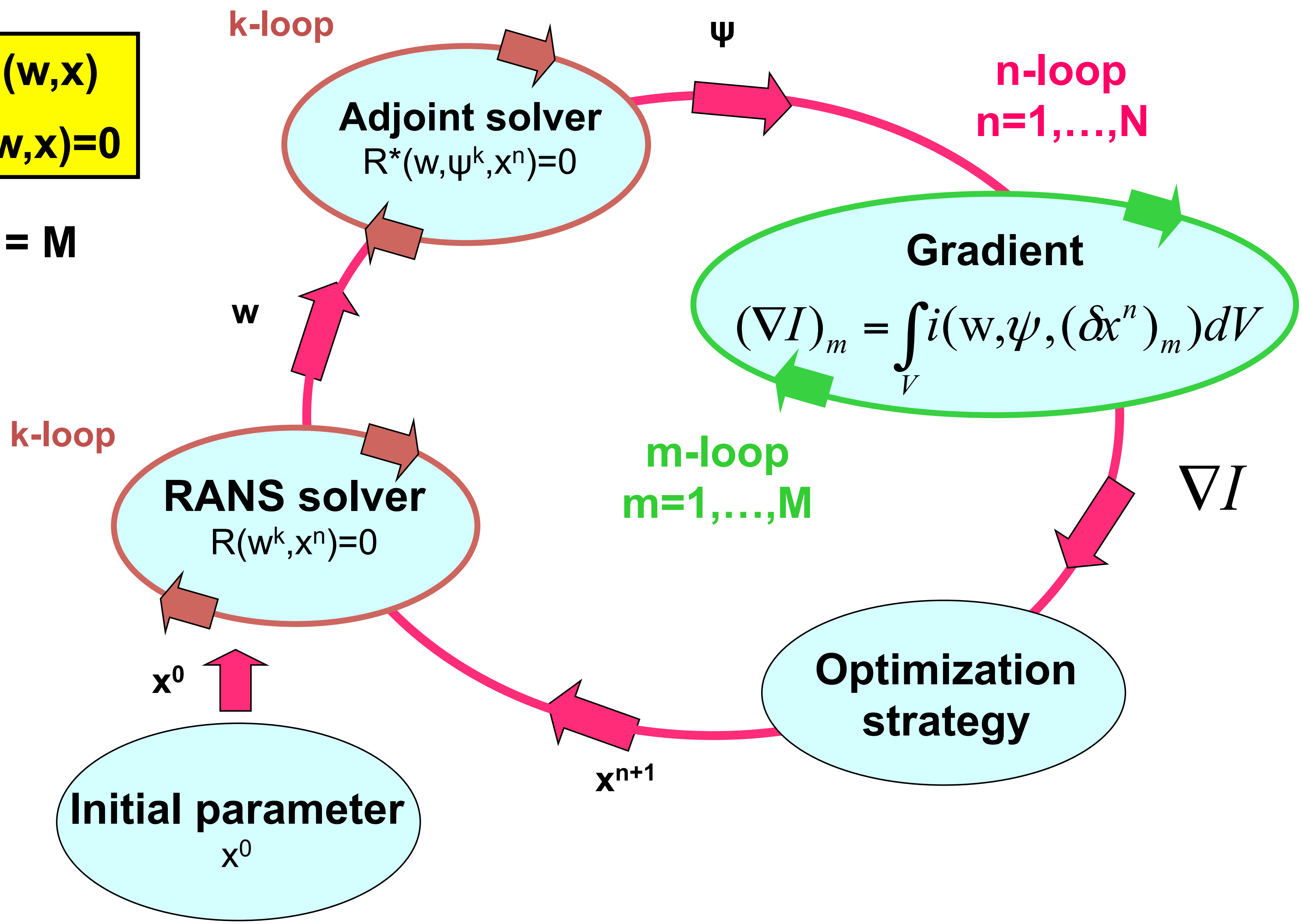
Outline

- Consistent and Robust Discrete Adjoints
- Application to Separation Control on the 3D High-Lift Configuration HIREX
- One-Shot Approach for Optimization with Steady PDEs
- Adjustments for Optimization with Unsteady PDEs
- One-Shot Optimization with Unsteady RANS
- Improving the Efficiency
- Numerical Results for Van der Pol Oscillator and Advection-Diffusion Equation

Nested Approach

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

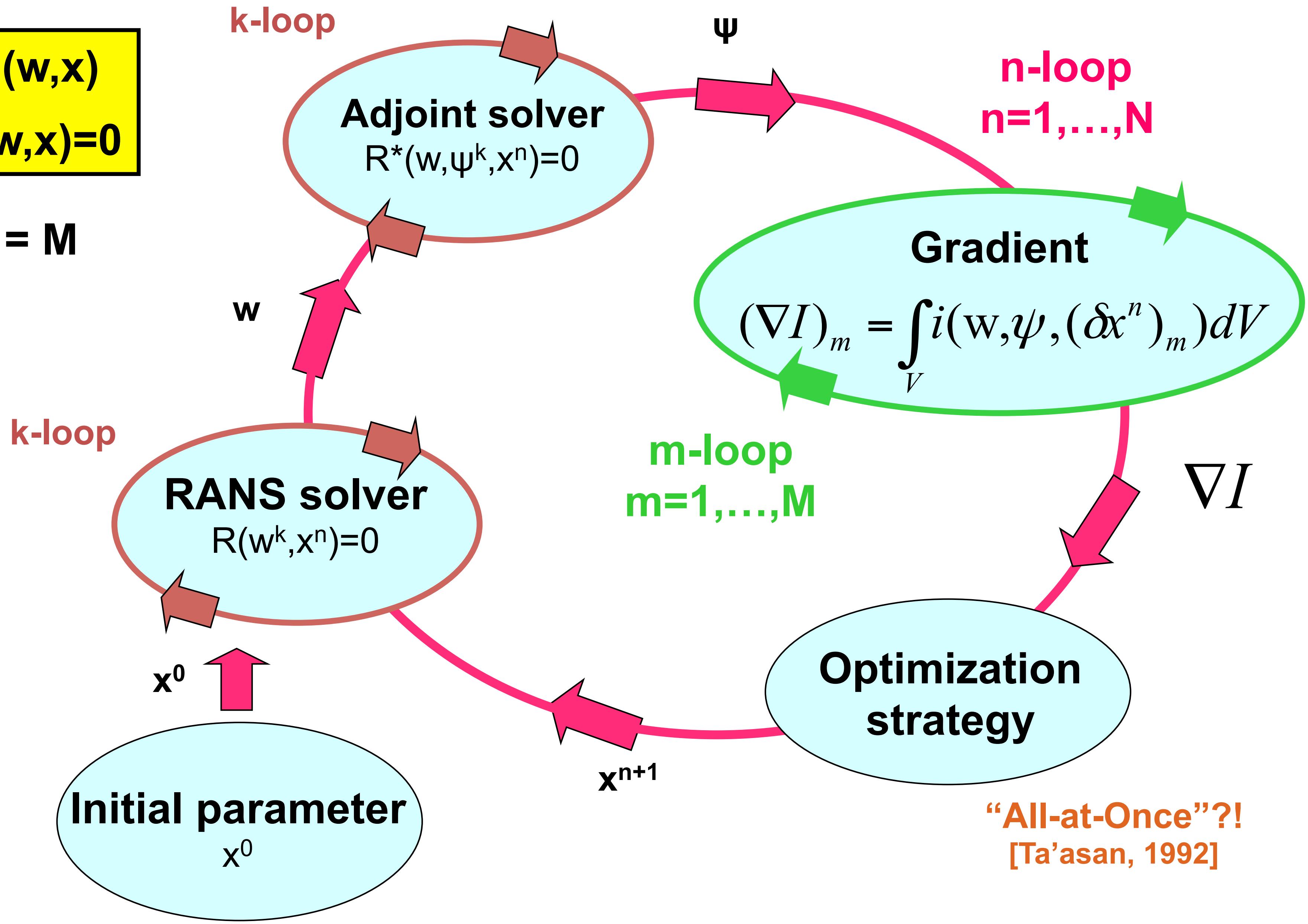
dim x = M



Nested Approach

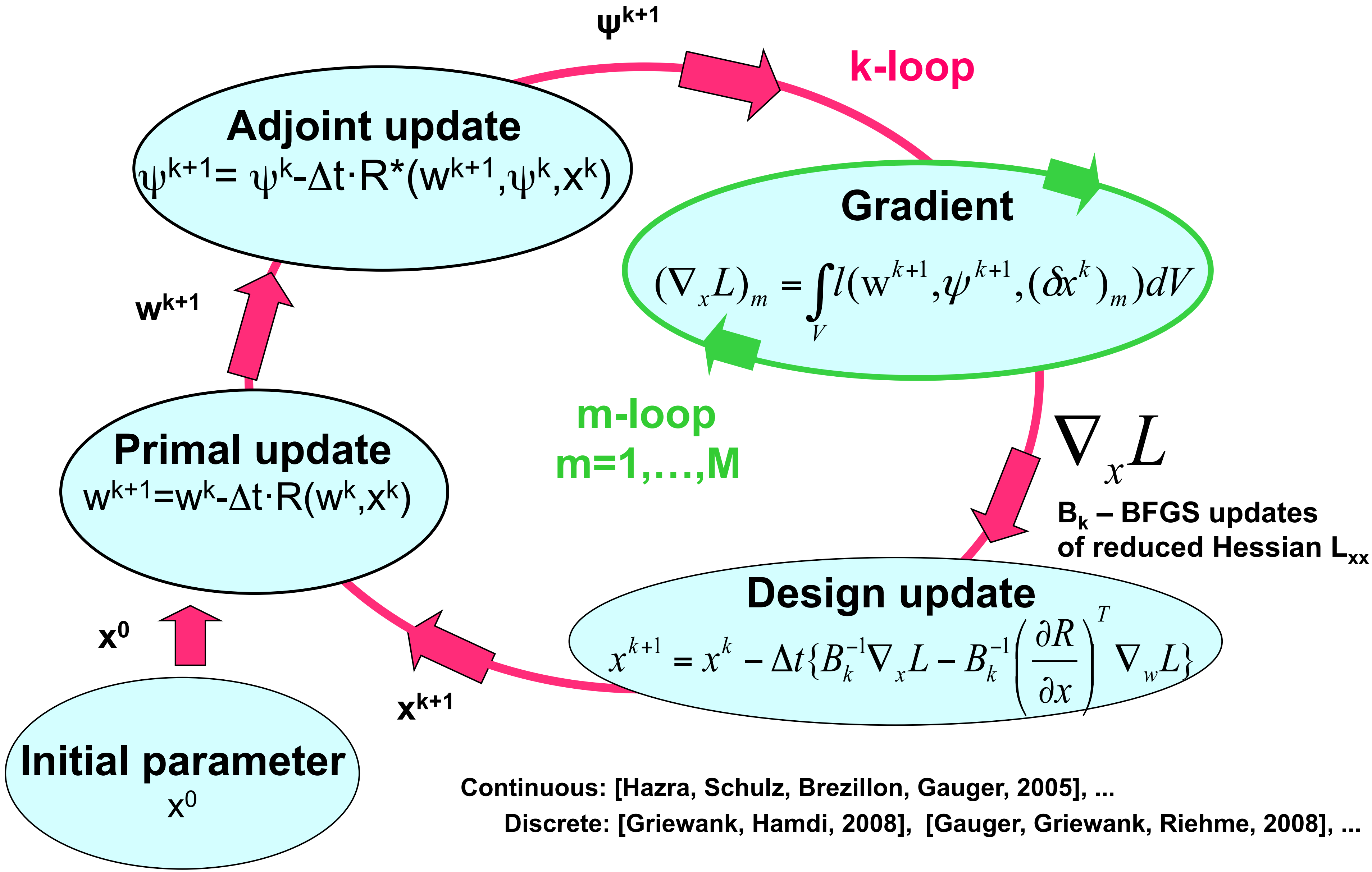
min I (w,x)
s.t. R(w,x)=0

dim x = M



"All-at-Once"?!
[Ta'asan, 1992]

One-Shot Approach



Continuous: [Hazra, Schulz, Brezillon, Gauger, 2005], ...

Discrete: [Griewank, Hamdi, 2008], [Gauger, Griewank, Riehme, 2008], ...

Problem Setup (Steady PDE)

Goal: $$\min_{y,u} f(y,u) \quad s.t. \quad c(y,u) = 0$$

- $y \in Y$ state variables, $u \in U$ design variables
- PDE is solved by an iterative fixed-point solver G:

$$y_{k+1} = G(y_k, u) \overset{k\to\infty}{\longrightarrow} y_* = G(y_*, u)$$

Contractivity: $\left\| \frac{\partial G}{\partial y} \right\| \leq \rho < 1$

- KKT-system for $L(y, \bar{y}, u) := f(y,u) + (G(y,u) - y)^T \bar{y}$:

$y = G(y,u)$	**State Equation**
$\bar{y} = \nabla_y f(y,u) + G_y(y,u)^T \bar{y}$	**Adjoint Equation**
$0 = \nabla_u f(y,u) + G_u(y,u)^T \bar{y}$	**Design Equation**

Classical Nested vs. One-Shot Optimization Approach

- Classical nested optimization approach

  Repeat for $m = 1, \dots$

  - Solve for state $y_{k+1} = G(y_k, u_m) \stackrel{k\to\infty}{\longrightarrow} y_*$
  - Solve adjoint $\bar{y}_{l+1} = \nabla_y f(y_*, u_m) + G_y(y_*, u_m)^T \bar{y}_l \stackrel{l\to\infty}{\longrightarrow} \bar{y}_*$
  - Update design $u_{m+1} = u_m - B_m^{-1}\left(\nabla_u f(y_*, u_m) + G_u(y_*, u_m)^T \bar{y}_*\right)$

Classical Nested vs. One-Shot Optimization Approach

- Classical nested optimization approach

  Repeat for $m = 1, \ldots$

  - Solve for state $y_{k+1} = G(y_k, u_m) \xrightarrow{k\to\infty} y_*$
  - Solve adjoint $\bar{y}_{l+1} = \nabla_y f(y_*, u_m) + G_y(y_*, u_m)^T \bar{y}_l \xrightarrow{l\to\infty} \bar{y}_*$
  - Update design $u_{m+1} = u_m - B_m^{-1}\left(\nabla_u f(y_*, u_m) + G_u(y_*, u_m)^T \bar{y}_*\right)$

- One-Shot optimization approach

  Repeat for $k = 1, \ldots$

  - Update state $y_{k+1} = G(y_k, u_k)$
  - Update adjoint $\bar{y}_{k+1} = \nabla_y f(y_k, u_k) + G_y(y_k, u_k)^T \bar{y}_k$
  - Update design $u_{k+1} = u_k - B_k^{-1}\left(\nabla_u f(y_k, u_k) + G_u(y_k, u_k)^T \bar{y}_k\right)$

- Choice of B ensures convergence of the One-Shot method.

  [Gauger, Griewank, Riehme, 2008], [Hamdi, Griewank, 2008], ...

One-Shot Approach

$$L(y,\bar{y},u) = f(y,u) + (G(y,u) - y)^T \bar{y}$$
$$= \underbrace{N(y,\bar{y},u)}_{\text{shifted Lagrangian}} - y^T \bar{y}$$

Stationary point:

$$\begin{cases} L_{\bar{y}} = G(y,u) - y = 0 \\ L_y = N_y(y,\bar{y},u)^T - \bar{y} = 0 \\ L_u = N_u(y,\bar{y},u)^T = 0 \end{cases}$$

One-step one-shot (step k+1):

$$(OS)\begin{cases} y_{k+1} = G(y_k, u_k) & \text{primal update} \\ \bar{y}_{k+1} = N_y(y_k, \bar{y}_k, u_k)^T & \text{adjoint update} \\ u_{k+1} = u_k - B_k^{-1} N_u(y_k, \bar{y}_k, u_k)^T & \text{design update} \end{cases}$$

One-Shot Approach

$$L(y,\bar{y},u) = f(y,u) + (G(y,u) - y)^T \bar{y}$$

$$= \underbrace{N(y,\bar{y},u)}_{\text{shifted Lagrangian}} - y^T \bar{y}$$

Stationary point:

$$\begin{cases} L_{\bar{y}} = G(y,u) - y = 0 \\ L_y = N_y(y,\bar{y},u)^T - \bar{y} = 0 \\ L_u = N_u(y,\bar{y},u)^T = 0 \end{cases}$$

One-step one-shot (step k+1):

Piggy-Back

$$\begin{cases} y_{k+1} = G(y_k, u_k) & \text{primal update} \\ \bar{y}_{k+1} = N_y(y_k, \bar{y}_k, u_k)^T & \text{adjoint update} \\ u_{k+1} = u_k - B_k^{-1} N_u(y_k, \bar{y}_k, u_k)^T & \text{design update} \end{cases}$$

[Griewank,Faure 2002]

One-Shot Approach

One-step one-shot (step k+1):

$$(OS)\begin{cases} y_{k+1} = G(y_k, u_k) & \text{primal update} \\ \bar{y}_{k+1} = N_y(y_k, \bar{y}_k, u_k)^T & \text{adjoint update} \\ u_{k+1} = u_k - B_k^{-1} N_u(y_k, \bar{y}_k, u_k)^T & \text{design update} \end{cases}$$

Aims: **Choose** B **such that:**

- **Convergence of** (OS)
- **Bounded retardation**
  i.e. O(opt) / O(sim) < const

Jacobian:

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

Doubly Augmented Lagrangian

- Deriving (sufficient) conditions on B for J_* to have a spectral radius smaller than 1 has proven difficult.

- Instead, we look for descent on the **augmented Lagrangian**

$$L^a(y,\bar{y},u) := \frac{\alpha}{2}\underbrace{\|G(y,u)-y\|^2}_{\text{primal residual}} + \frac{\beta}{2}\underbrace{\|N_y(y,\bar{y},u)^T-\bar{y}\|^2}_{\text{adjoint residual}} + \underbrace{N-\bar{y}^T y}_{\text{Lagrangian}},$$

where $\alpha > 0$ and $\beta > 0$.

- Gradient of L^a:

$$\begin{bmatrix} \nabla_y L^a \\ \nabla_{\bar{y}} L^a \\ \nabla_u L^a \end{bmatrix} = -\underbrace{\begin{bmatrix} \alpha(I-G_y)^T & -I-\beta N_{yy} & 0 \\ -I & \beta(I-G_y) & 0 \\ -\alpha G_u^T & -\beta N_{yu}^T & B \end{bmatrix}}_{=:\,M} \underbrace{\begin{bmatrix} G(y,u)-y \\ N_y(y,\bar{y},u)^T-\bar{y} \\ -B^{-1}N_u(y,\bar{y},u)^T \end{bmatrix}}_{=:s\ \text{one-shot increment}}$$

Descent Direction for L^a

Theorem (Correspondence condition):

L^a **is an exact penalty function, if** $\alpha\beta(I-G_y)^T(I-G_y) \succ I+\beta N_{yy}.$

Theorem (Descent condition):

The One-Shot increment vector $s(y,\bar{y},u) := \begin{bmatrix} G(y,u)-y \\ N_y(y,\bar{y},u)^T-\bar{y} \\ -B^{-1}N_u(y,\bar{y},u)^T \end{bmatrix}$

is a descent direction for all large positive B **if and only if**

$$\alpha\beta(I-\frac{1}{2}(G_y+G_y^T)) \succ (I+\frac{\beta}{2}N_{yy})(I-\frac{1}{2}(G_y+G_y^T))^{-1}(I+\frac{\beta}{2}N_{yy}).$$

- **Both conditions are implied by** $\sqrt{\alpha\beta}(1-\rho) > 1+\frac{\beta}{2}\|N_{yy}\|.$

[Hamdi, Griewank, 2008], …

[Gauger, Griewank, Hamdi, Kratzenstein, Özkaya, Slawig, 2012]

Convergence of One-Shot

- Choose $\beta = \frac{2}{c}, \quad \alpha = \frac{2c}{(1-\rho)^2}$ with $c = \|N_{yy}\|$
- Choose *B* such that

$$B \geq B_0 := \alpha G_u^T G_u + \beta N_{yu}^T N_{yu} + N_{uu}.$$

then *s* yields descent on L^a.

$\Longrightarrow$ Convergence of One-Shot approach.

[Hamdi, Griewank, 2008], …

[Gauger, Griewank, Hamdi, Kratzenstein, Özkaya, Slawig, 2012]

In practice: BFGS-updates for the Hessian

$$\nabla_u^2 L^a = \underbrace{\alpha G_u^T G_u + \beta N_{yu}^T N_{yu} + N_{uu}}_{B} + \alpha \underbrace{(G-y)}_{\to_* 0}^T G_{uu} + \beta \underbrace{(N_y^T - \bar{y})}_{\to_* 0}^T N_{yuu}.$$

The gradient $\nabla_u L^a = \alpha (G-y)^T G_u + \beta (N_y - \bar{y})^T N_{yu} + N_u$

is evaluated by Algorithmic Differentiation (AD).

[Özkaya, Gauger, 2008]

Efficient One-Shot Approach

ONERA M6, M=0.83, α=3.01°

Drag reduction by constant lift

DLR TAU Code (Euler)

Initial geometry:

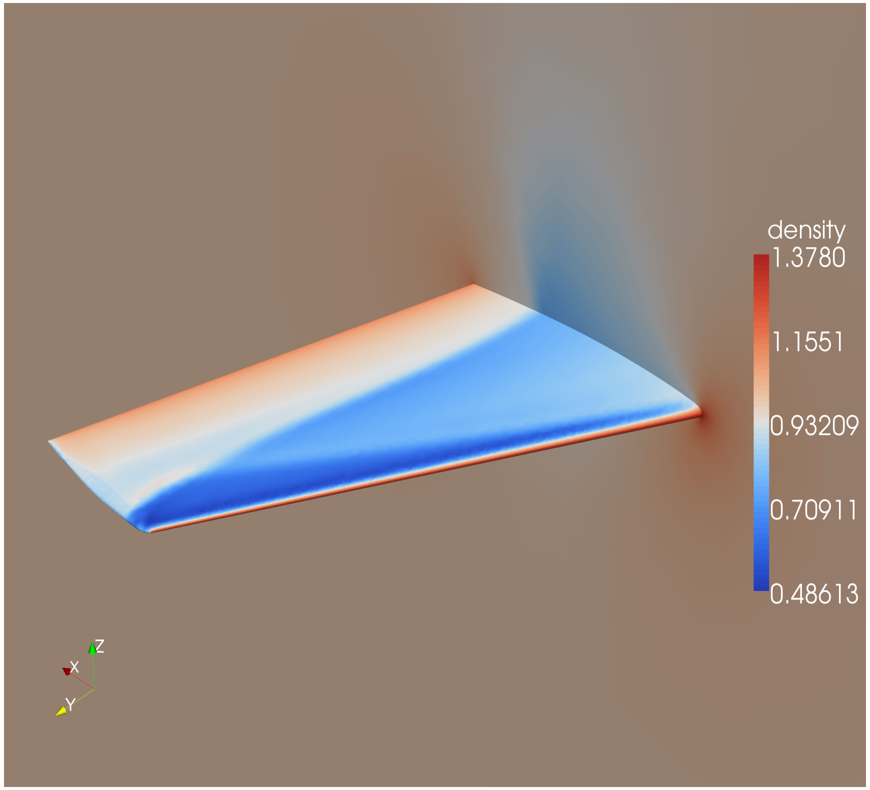
- C_D^{init} = 106 drag counts
- C_L^{init} = 27.6 lift counts

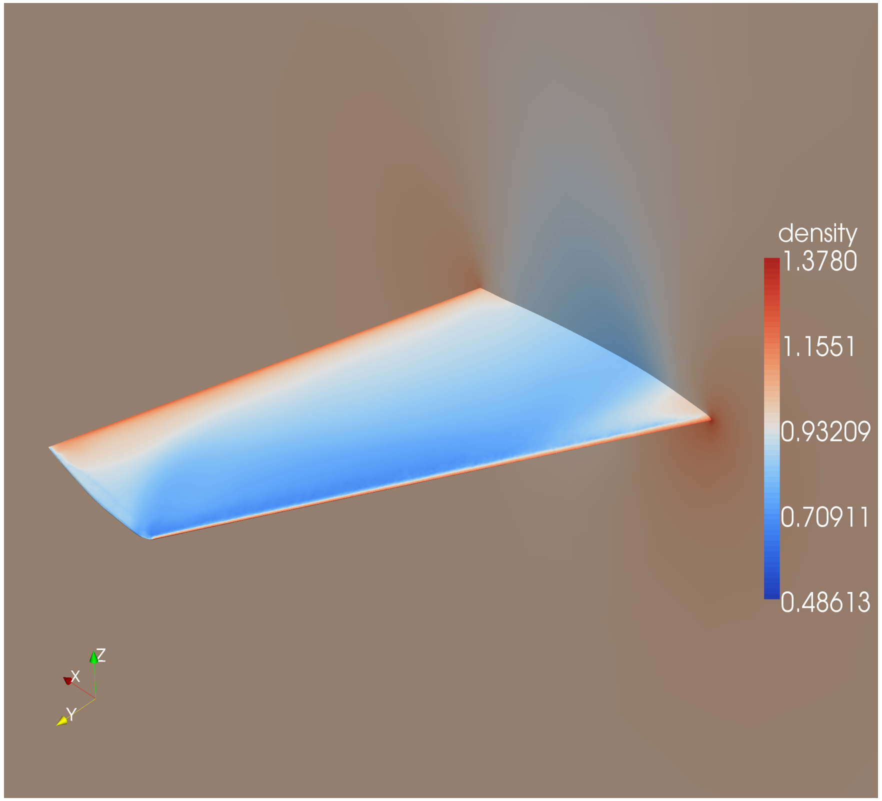
Optimized geometry:

- C_D^{opt} = 72 drag counts
- C_L^{opt} = 26.5 lift counts

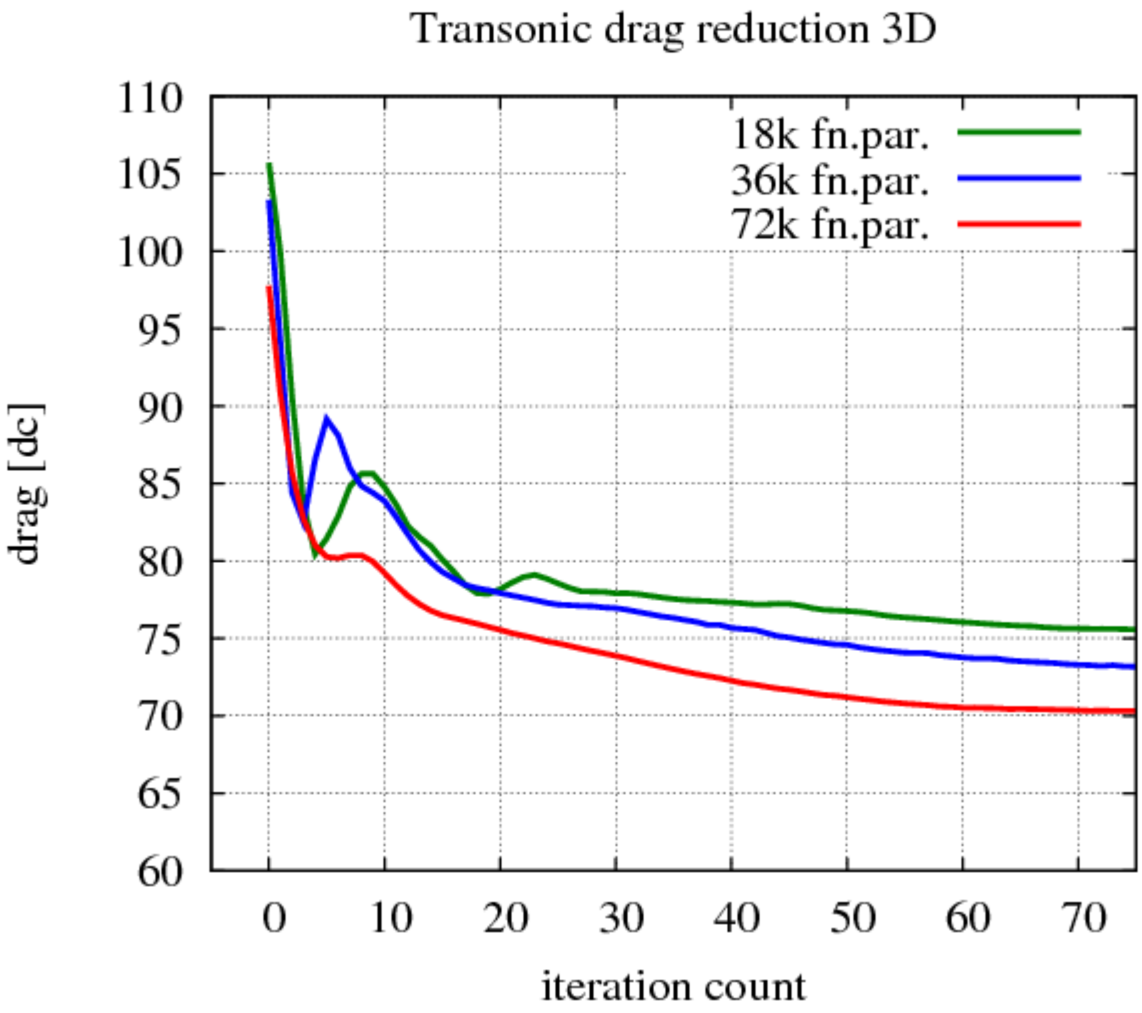
O(opt) / O(sim) = 5 (wall clock time)
= 2 (# iterations)

32% drag reduction





Scaling with Number of Parameters

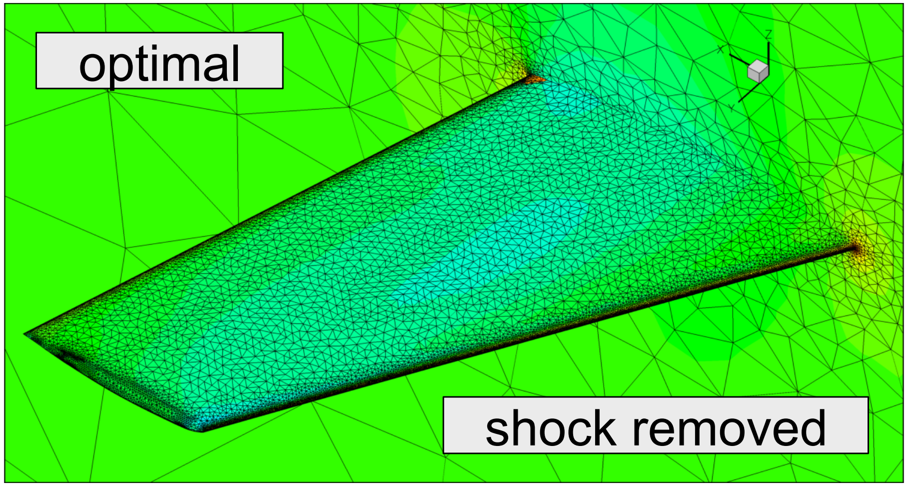


optimization to
simulation time **~5**

Approach:

- **Shape Derivatives**
- **Gradient Smoothing**
- **Preconditioning**
- **One Shot**

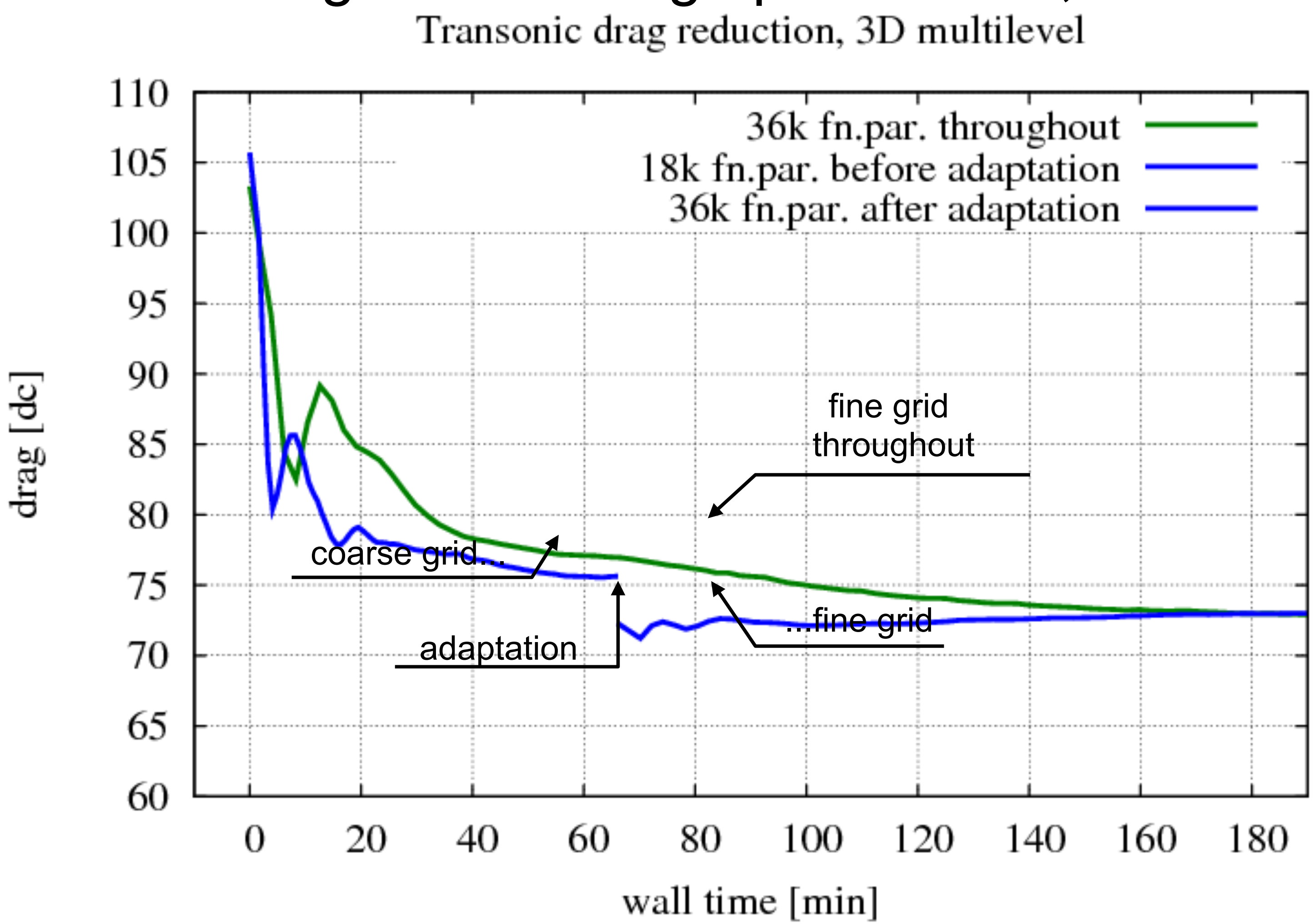
optimal drag =
induced drag (planform)
+ spurious drag (numerics)



[Ilic, Gauger, Schmidt, Schulz, 2009]

Multilevel Descent

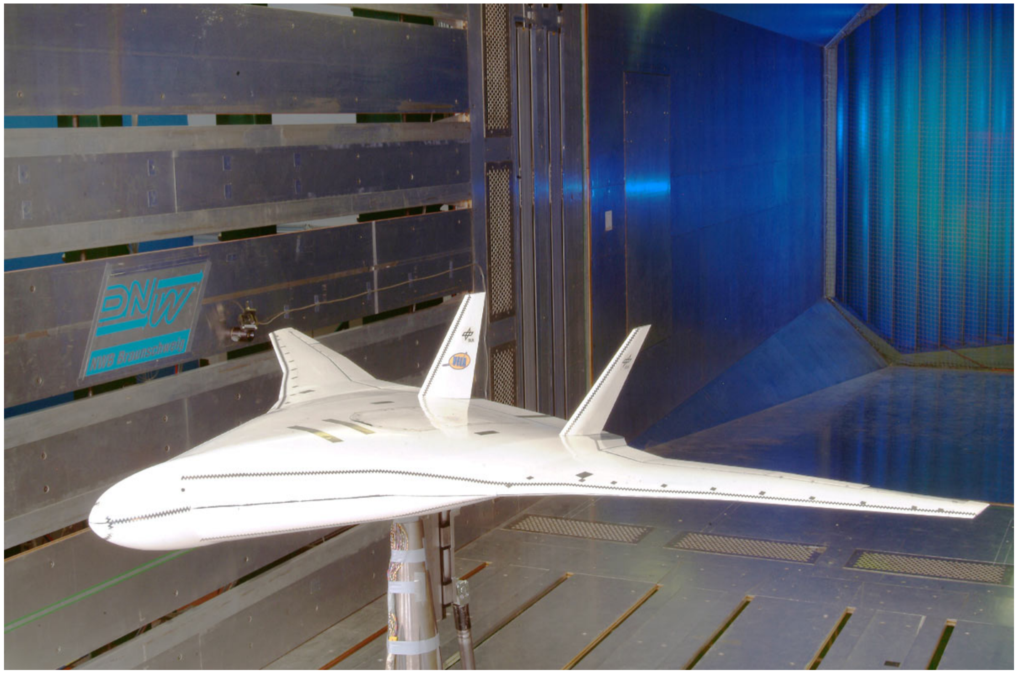
➢ 2-level: coarse grid **18k** design parameters, fine **36k**

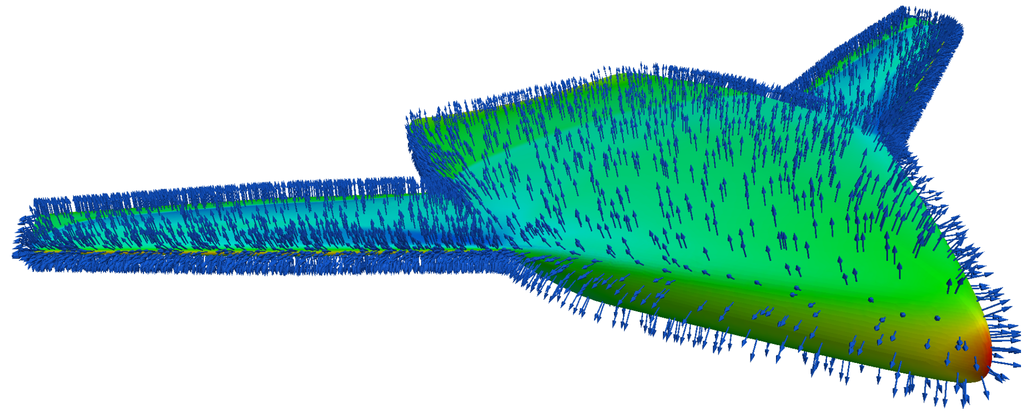


➢ 2-level iteration brings factor **~2** in optimization time

[Ilic, Gauger, Schmidt, Schulz, 2009]

One-Shot Optimization of VELA (DLR)

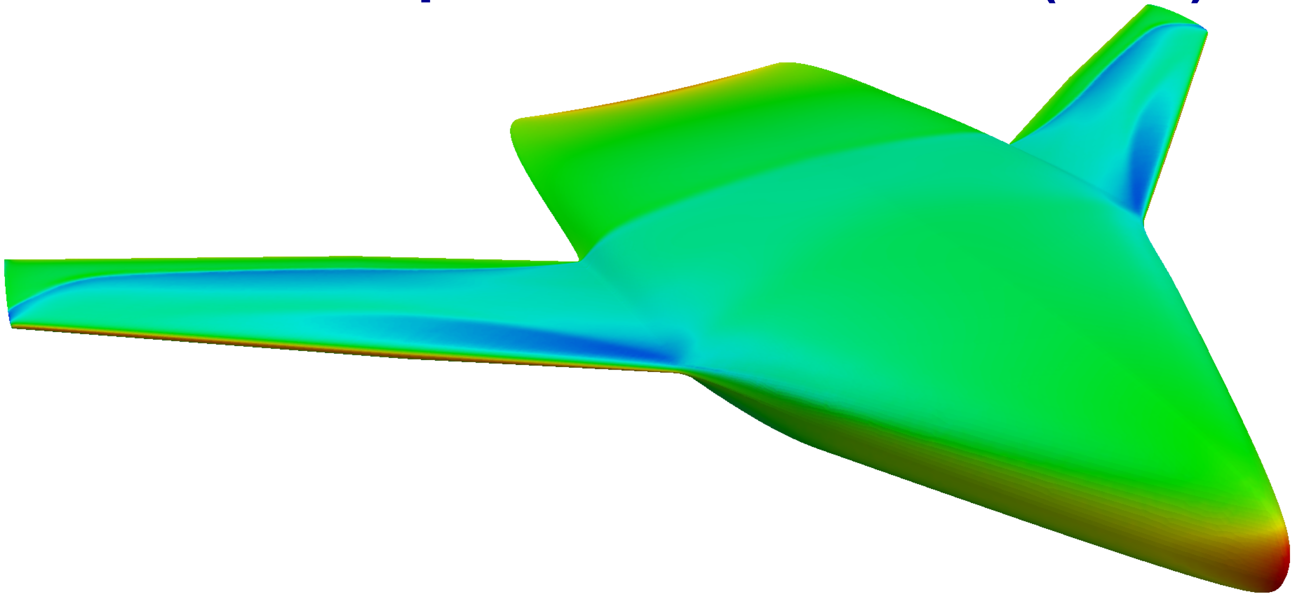




Design study for blended wing-body configurations

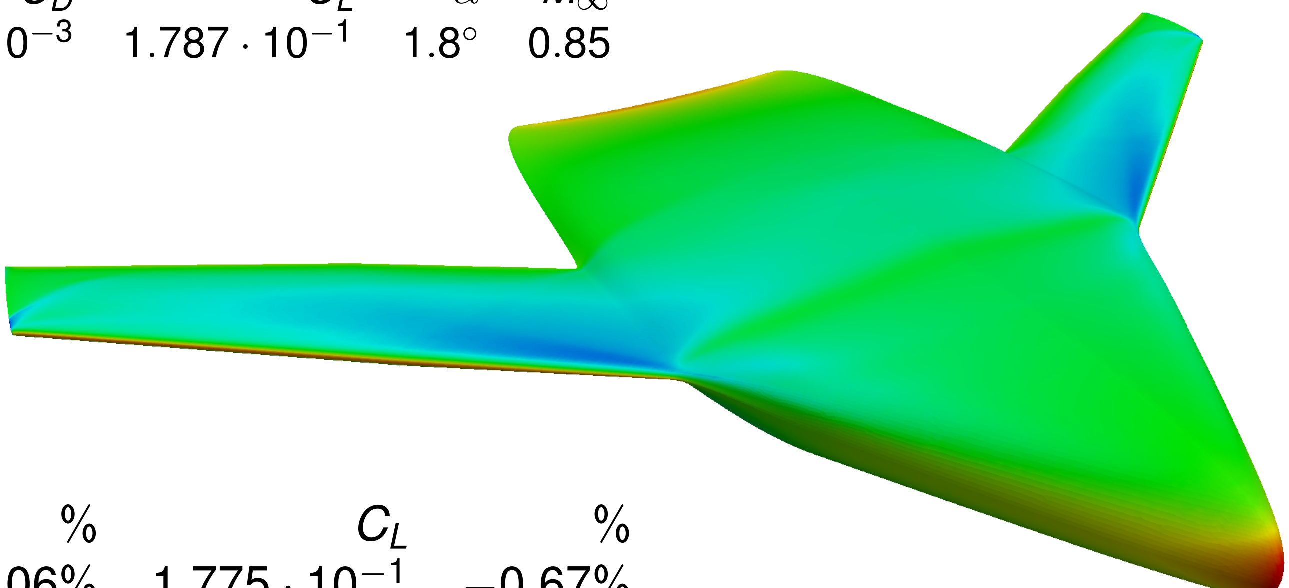
- $115,673$ surface node positions to be optimised
- Planform constant

One-Shot Optimization of VELA (DLR)



[Schmidt, Schulz, Ilic, Gauger, 2011]

Shape	State	C_D	C_L	α	M_∞
115, 673	29, 297, 175	$4.770 \cdot 10^{-3}$	$1.787 \cdot 10^{-1}$	1.8°	0.85



Shape	C_D	%	C_L	%
115, 673	$3.342 \cdot 10^{-3}$	-30.06%	$1.775 \cdot 10^{-1}$	-0.67%

Outline

- Consistent and Robust Discrete Adjoints
- Application to Separation Control on the 3D High-Lift Configuration HIREX
- One-Shot Approach for Optimization with Steady PDEs
- Adjustments for Optimization with Unsteady PDEs
- One-Shot Optimization with Unsteady RANS
- Improving the Efficiency
- Numerical Results for Van der Pol Oscillator and Advection-Diffusion Equation

Problem Setup (Unsteady PDE)

Goal: $$\min_{y,u} \frac{1}{T}\int_0^T f(y(t),u)\,\mathrm{d}t \quad \text{subject to}$$

$$\frac{\partial y(t)}{\partial t} + c(y(t),u) = 0 \quad \forall\, t \in [0,T]$$
$$y(0) = y_*^0$$

- Use existing, well-established simulation tools.
- Implicit time marching scheme:

$$\frac{y(t_i) - y(t_{i-1})}{t_i - t_{i-1}} + c(y(t_i),u) = 0$$

for each discrete time step $0 = t_0 < \cdots < t_N = T$.

Solving the Unsteady PDE

Given: Fixed point iterator G to solve the residuum equations at each time step:

for $t_1 < \ldots < t_N$:

iterate $y_{k+1}(t_i) = G(y_k(t_i), y_*(t_{i-1}), u) \stackrel{k\to\infty}{\longrightarrow} y_*(t_i)$

- Contractivity: $\left\|\frac{\partial G(y(t_i), y(t_{i-1}), u)}{\partial y(t_i)}\right\| \leq \rho < 1$

How to incorporate design updates for One-Shot?

Prepare for One-Shot

Modification of time marching scheme:

$$\text{iterate } k = 0, 1, \ldots :$$
$$\text{for } t_1 < \ldots < t_N : \; y_{k+1}(t_i) = G(y_k(t_i), y_{k+1}(t_{i-1}), u) \quad (*)$$

- Update a complete trajectory within one iteration.
- Consider $y = (y(t_1), \ldots, y(t_N)) = \left(y^1, \ldots, y^N\right) \in (\mathbb{R}^m)^N$ then

$$\text{iterate } k = 0, 1, \ldots :$$
$$y_{k+1} = H(y_k, u)$$

where H performs $(*)$.

Contractivity of H

- The Jacobian $\frac{\partial H(y,u)}{\partial y}$ is block-triangular:

$$\frac{\partial H(y,u)}{\partial y} = \begin{pmatrix} \partial_{y^1} G(y^1, y^0, u) & 0 & 0 \\ * & \ddots & 0 \\ * & * & \partial_{y^N} G(y^N, y^{N-1}, u) \end{pmatrix}$$

- Its spectral radius is bounded:

$$\text{spr}\left(\frac{\partial H(y,u)}{\partial y}\right) = \max_{i \in \{1,\dots,N\}} \text{spr}\left(\partial_{y^i} G(y^i, y^{i-1}, u)\right) \leq \rho < 1$$

$\Longrightarrow$ H is **contractive**.

[Günther, Gauger, 2013]

Outline

- **Consistent and Robust Discrete Adjoints**
- **Application to Separation Control on the 3D High-Lift Configuration HIREX**
- **One-Shot Approach for Optimization with Steady PDEs**
- **Adjustments for Optimization with Unsteady PDEs**
- **One-Shot Optimization with Unsteady RANS**
- **Improving the Efficiency**
- **Numerical Results for Van der Pol Oscillator and Advection-Diffusion Equation**

Discrete Unsteady Optimization Problem

Goal:

$$\min_{y,u} \frac{1}{N} \sum_{i=1}^{N} f(y^i, u) \quad \text{s.t.} \quad y = H(y, u)$$

- $y \in (\mathbb{R}^m)^N$ state variable, $u \in \mathbb{R}^n$ design variable
- Fixed point iterator H to solve the unsteady PDE

$$y_{k+1} = H(y_k, u) \overset{k\to\infty}{\longrightarrow} y_* = H(y_*, u)$$

- Contractivity: $\left\| \frac{\partial H}{\partial y} \right\| \leq \rho < 1$

$\Longrightarrow$ Same structure as in steady case.
$\Longrightarrow$ One-Shot method can be applied.

[Günther, Gauger, 2013]

Unsteady One-Shot Iterations

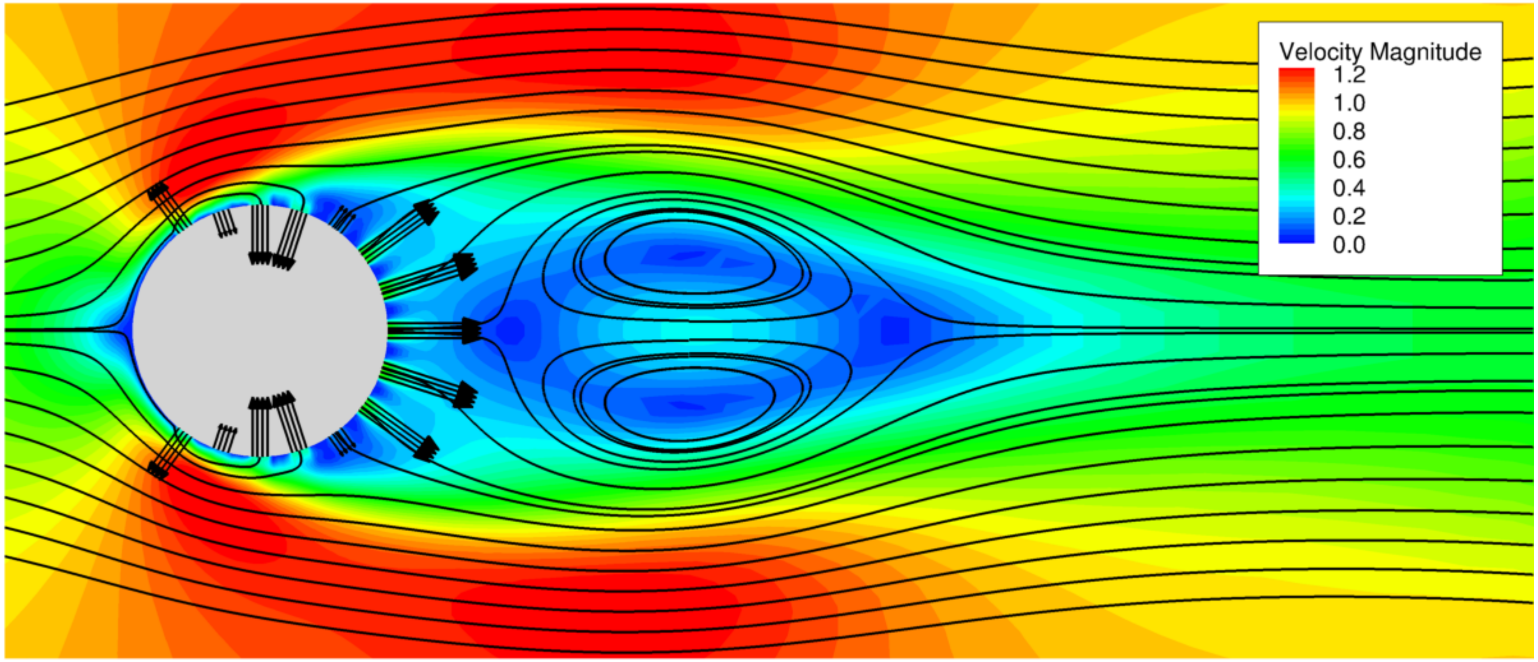
- Lagrangian function $L(y, \bar{y}, u) = J(y, u) + (H(y, u) - y)^T \bar{y}$
- Iterate simultaneously

$$\begin{bmatrix} y_{k+1} \\ \bar{y}_{k+1} \\ u_{k+1} \end{bmatrix} = \begin{bmatrix} H(y_k, u_k) \\ \nabla_y J(y_k, u_k) + H_y(y_k, u_k)^T \bar{y}_k \\ u_k - B_k^{-1} \left(\nabla_u J(y_k, u_k) + H_u(y_k, u_k)^T \bar{y}_k \right) \end{bmatrix}$$

- y_{k+1} contains loop over all time steps forward in time
- $\bar{y}_{k+1}$ contains loop over all time steps backwards in time

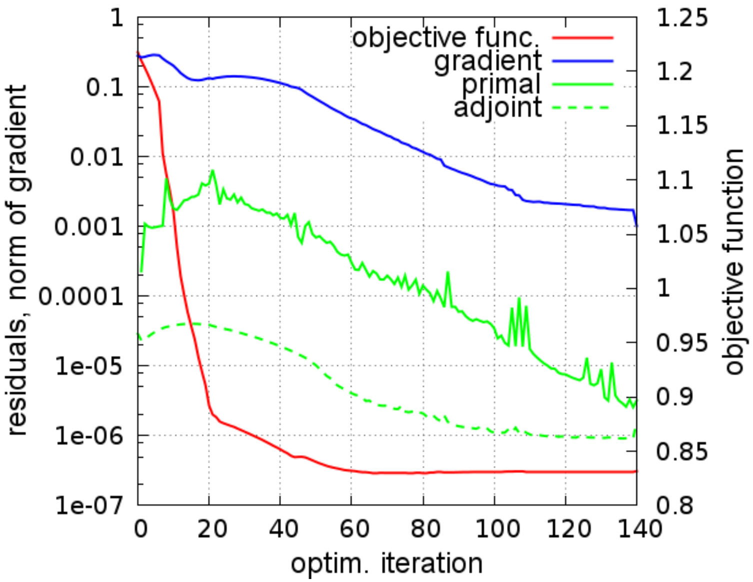
Optimal Active Flow Control

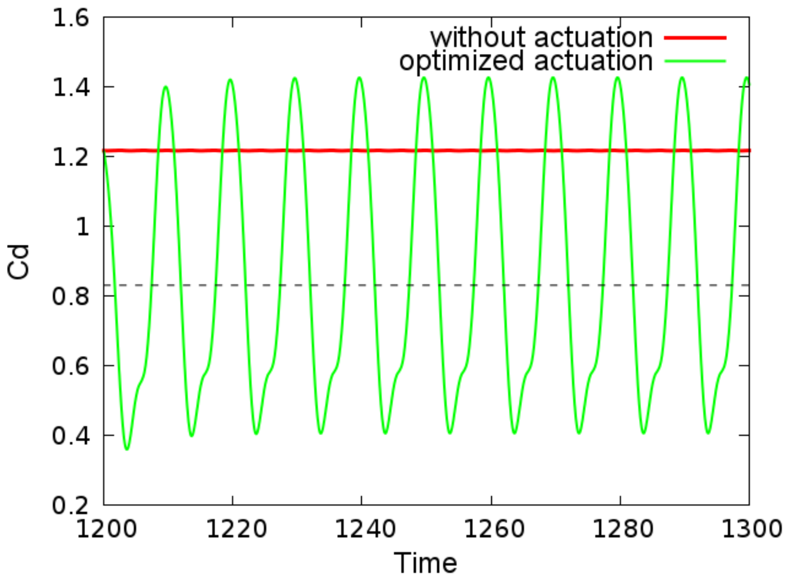
- Flow around cylinder at Re=100 governed by incompressible unsteady Reynolds-averaged Navier-Stokes equations (URANS)
- 15 actuation slots for pulsed blowing/suction
- Design parameters: Amplitude at each slot
- Objective function: Average drag coefficient



Unsteady One-Shot Optimization for Cylinder Flow

- Flow Solver ELAN (developed at ISTA, TU Berlin)
- Implicit 2nd order in space and time
- Pressure correction loops in each time step (SIMPLE algorithm)
- Modification for One-Shot framework
- Automatic Differentiation for generation of adjoint solver

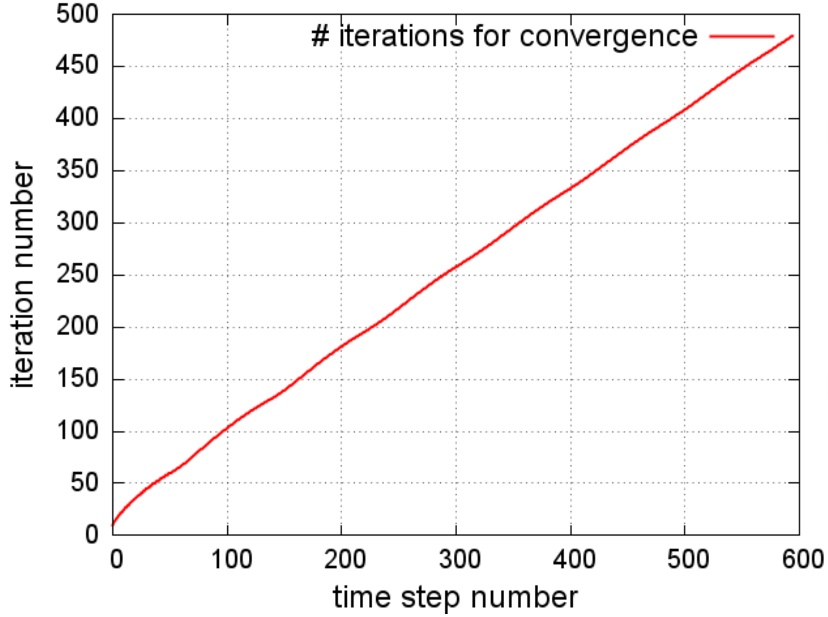




[Günther, Gauger, Wang, 2015]

Efficiency of Unsteady One-Shot Approach?

- Bounded retardation for One-Shot applications
- Performance of modified time-marching scheme:



- Number of outer iteration cycles for primal convergence depends linearly on N.
- Each iteration cycle contains a loop over entire time domain.

Outline

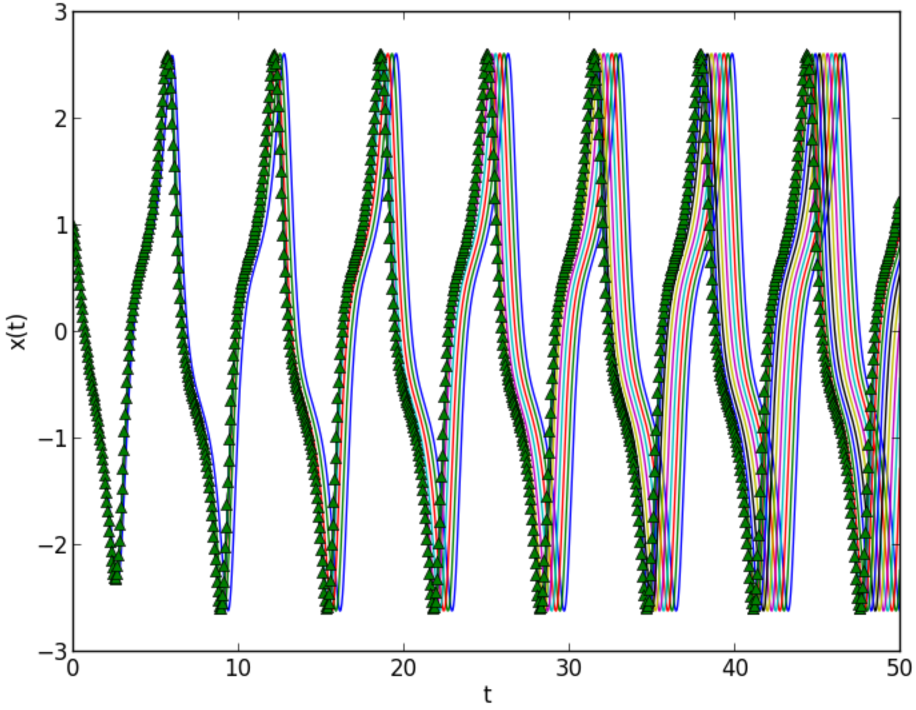
- **Consistent and Robust Discrete Adjoints**
- **Application to Separation Control on the 3D High-Lift Configuration HIREX**
- **One-Shot Approach for Optimization with Steady PDEs**
- **Adjustments for Optimization with Unsteady PDEs**
- **One-Shot Optimization with Unsteady RANS**
- **Improving the Efficiency**
- **Numerical Results for Van der Pol Oscillator and Advection-Diffusion Equation**

Model Problem: Van der Pol Oscillator

$$\begin{pmatrix} \dot{x}(t) \\ \dot{v}(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ -x(t) + u\left(1 - x(t)^2\right) v(t) \end{pmatrix} \quad \forall t \in [0, T]$$

$$\begin{pmatrix} x(0) \\ v(0) \end{pmatrix} = \begin{pmatrix} x_0 \\ v_0 \end{pmatrix}$$

- Autonomous ODE system
- Backward Euler discretization
- Quasi-Newton fixed point solver
- Modification for One-Shot framework



- **Numerical time dilation is the main error source.**

Adaptive Time Scaling

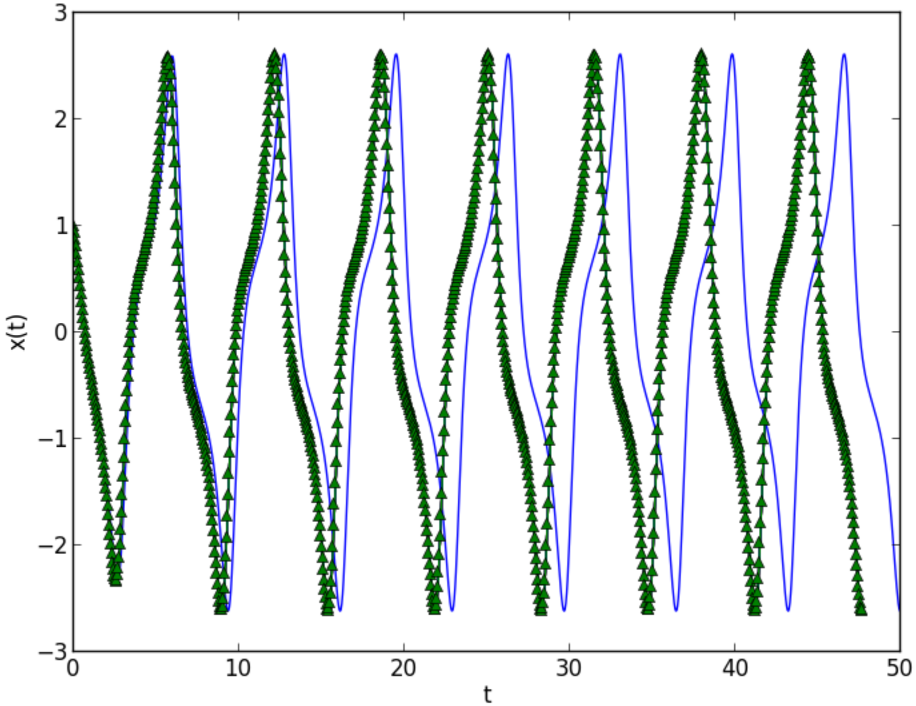
- Assign $\tilde{y}(t_i) := y(\tilde{t}_i)$ where $\tilde{t}_i$ minimizes the residuum equation

$$\min_{\tilde{t}_i} \left\| \frac{y(t_i) - y(t_{i-1})}{\tilde{t}_i - t_{i-1}} + c(y(t_i), u) \right\|_2^2$$

- For autonomous ODEs:

$$\tilde{t}_i = t_{i-1} - \frac{\langle y(t_i) - y(t_{i-1}), c(y(t_i), u) \rangle}{\| c(y(t_i), u) \|_2^2}$$

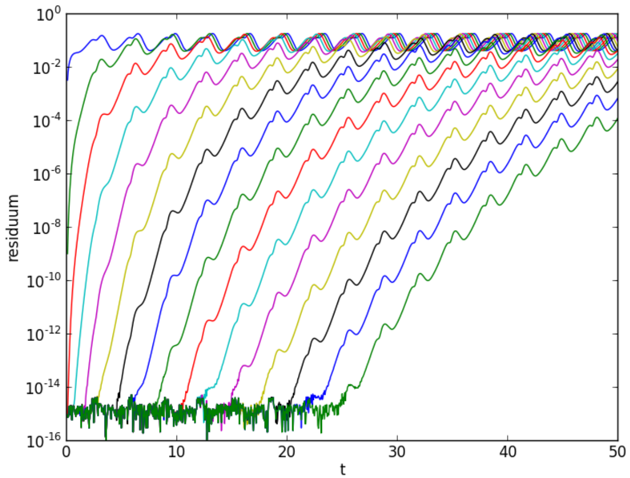
- **New trajectories are in phase with the unsteady solution.**



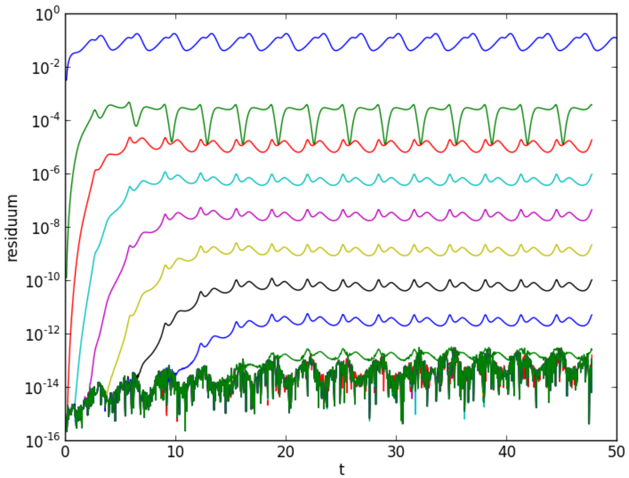
[Günther, Gauger, Wang, 2015]

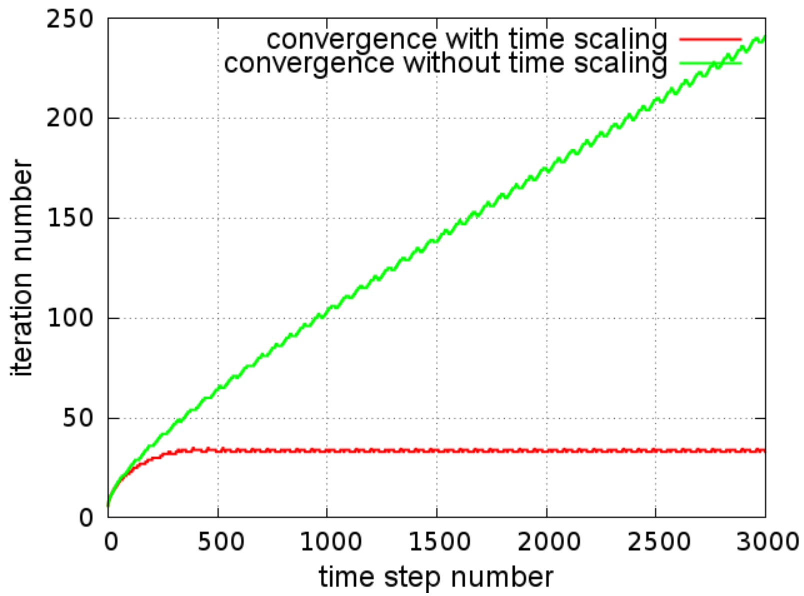
Adaptive Time Scaling

Residuals without time scaling



Residuals with time scaling





- Independence of number of time steps

[Günther, Gauger, Wang, 2015]

Outline

- Consistent and Robust Discrete Adjoints
- Application to Separation Control on the 3D High-Lift Configuration HIREX
- One-Shot Approach for Optimization with Steady PDEs
- Adjustments for Optimization with Unsteady PDEs
- One-Shot Optimization with Unsteady RANS
- Improving the Efficiency
- Numerical Results for Van der Pol Oscillator and Advection-Diffusion Equation

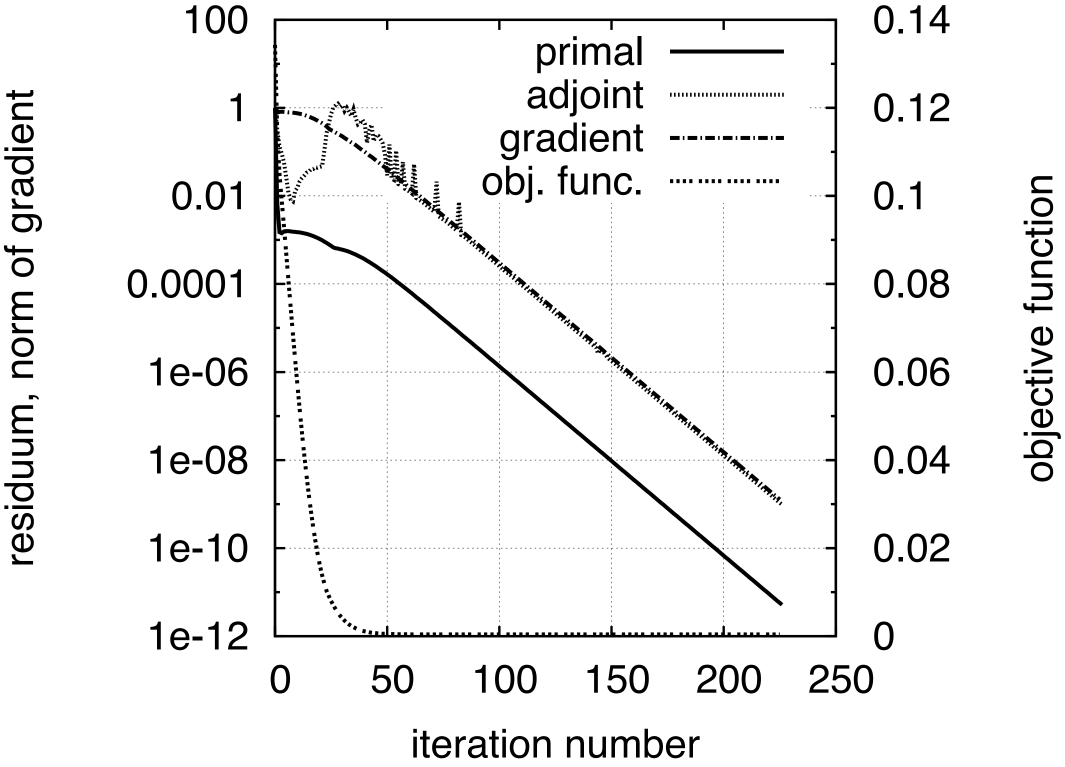
One-Shot Optimization

$$\min_{\mu,y} \int_0^T \|y - y_{ref}\|^2 + \gamma\|\mu\|^2 dt \quad \text{s.t.}$$

$$\begin{pmatrix} \dot{x}(t) \\ \dot{v}(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ -x(t) + \mu\left(1 - x(t)^2\right) v(t) \end{pmatrix} \quad \forall t \in [0, T]$$

$$\begin{pmatrix} x(0) \\ v(0) \end{pmatrix} = \begin{pmatrix} x_0 \\ v_0 \end{pmatrix}$$

- BFGS-updates for the preconditioner
- Retardation factor = 7



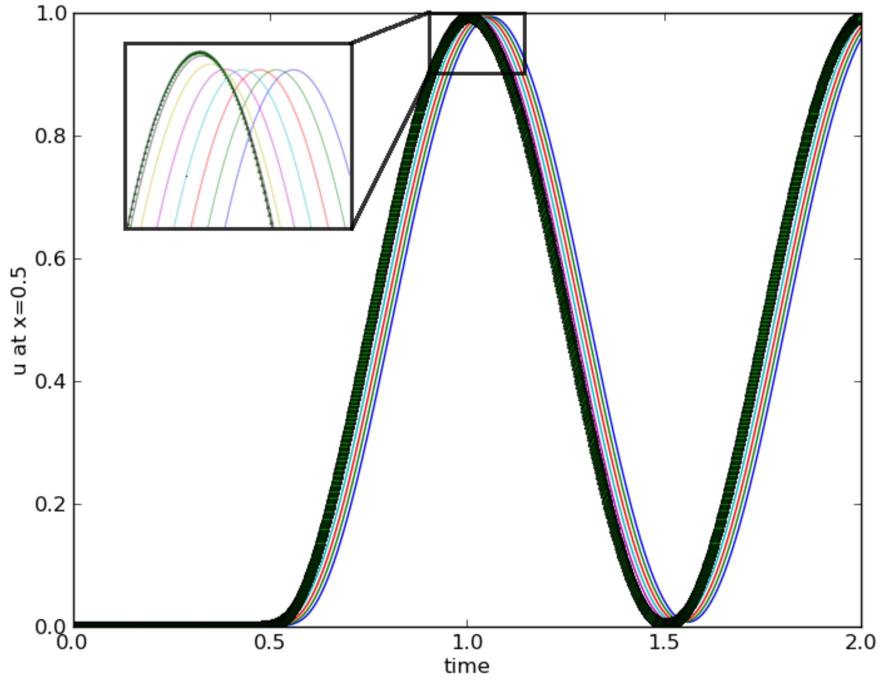
[Günther, Gauger, Wang, 2015]

Advection-Diffusion Equation

$$
\begin{aligned}
\dot{y}(t,x) + a y_x(t,x) - \mu y_{xx}(t,x) &= 0 && \forall x \in (0,1), t > 0 \\
y(t,0) &= u \cdot (\sin(2\pi t) + 1) && \forall t > 0 \\
y(0,x) &= 0 && \forall x \in [0,1]
\end{aligned}
$$

with $a = 1.0$ and $\mu = 10^{-5}$.

- Non-autonomous ODE
- Backward Euler discretization
- Quasi-Newton fixed point solver
- Modification for One-Shot framework



- Numerical time dilation plus error in amplitude

Transformation into Autonomous System

- Transform non-autonomous PDE into a system of autonomous PDEs
- State $w(t) = (y(t), s(t))$ satisfying

$$\begin{aligned} \dot{y}(t,x) + a y_x(t,x) - \mu y_{xx}(t,x) &= 0 \quad \forall x \in (0,1), t > 0 \\ y(t,0) - u \cdot (\sin(2\pi s(t)) + 1) &= 0 \quad \forall t > 0 \\ y(0,x) &= 0 \quad \forall x \in [0,1] \\ \dot{s}(t) &= 1 \quad \forall t > 0 \\ s(0) &= 0 \end{aligned}$$

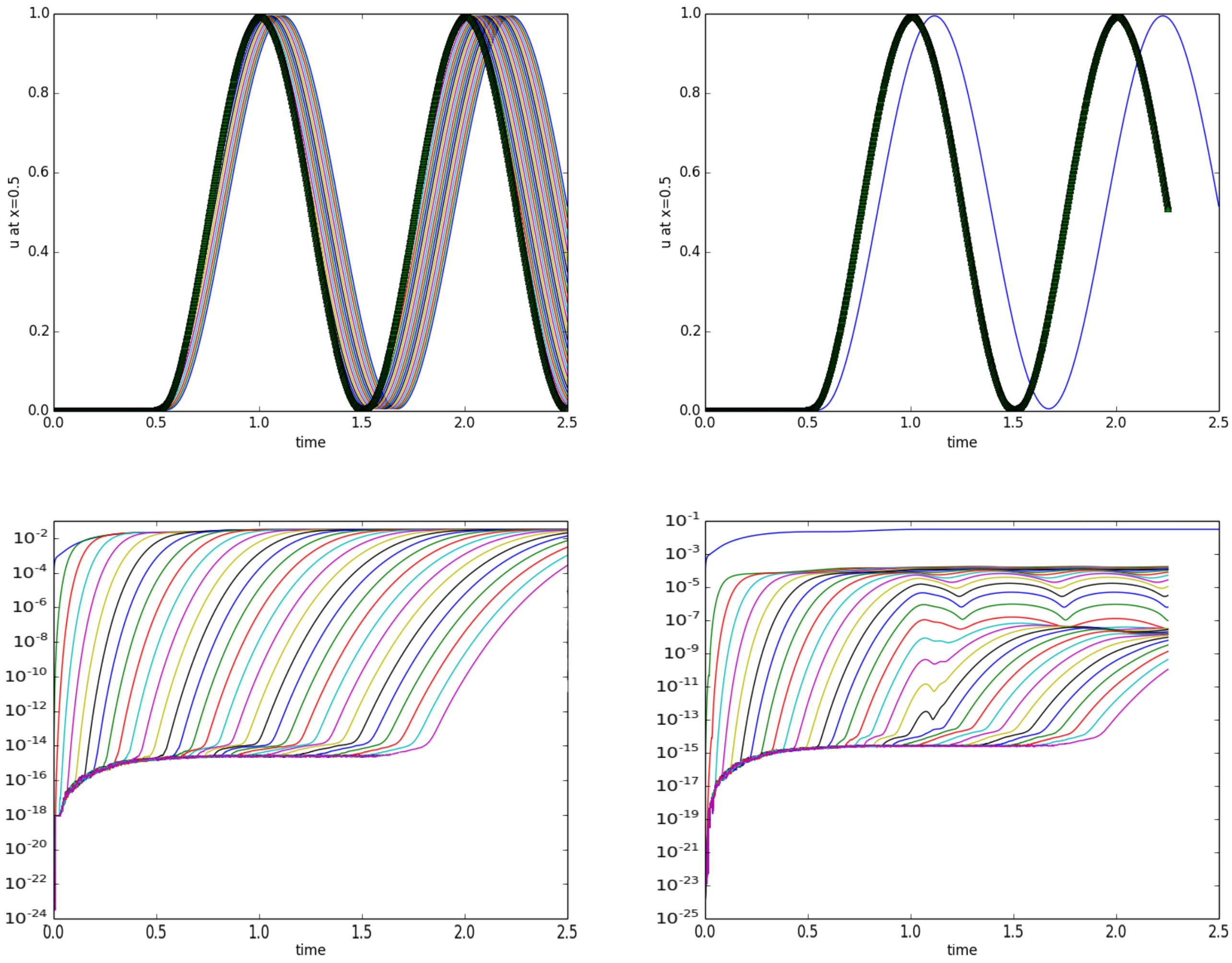
- Adaptive time scaling

$$\tilde{t}_i = t_{i-1} - \frac{\langle w(t_i) - w(t_{i-1}), c(w(t_i), u)\rangle}{\|c(w(t_i), u)\|_2^2}$$

[Günther, Gauger, Wang, in prep.]

Adaptive Time Scaling

without time scaling | with time scaling



- Numerical time dilation eliminated
- Residual improvement by 5 orders of magnitude

[Günther, Gauger, Wang, in prep.]

Content Presented

- Consistent and Robust Discrete Adjoints
- Application to Separation Control on the 3D High-Lift Configuration HIREX
- One-Shot Approach for Optimization with Steady PDEs
- Adjustments for Optimization with Unsteady PDEs
- One-Shot Optimization with Unsteady RANS
- Improving the Efficiency
- Numerical Results for Van der Pol Oscillator and Advection-Diffusion Equation

Thanks for your attention!