European Conference on Computational Optimization (EUCCO)
Chemnitz, July 17-19, 2013

# Adjoint Gradient-Based Approach for Aerodynamic Optimization of Transport Aircraft

Caslav Ilic


DLR

# Outline

- Optimization problem
- Optimization method
- Application examples
- Issues and outlook

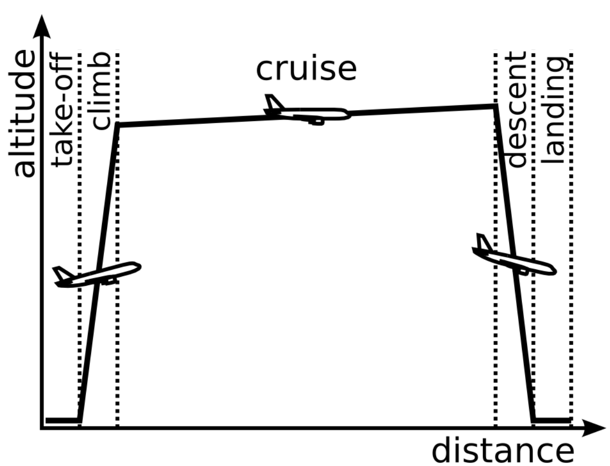
# Optimization Problem

# Transport Aircraft Performance

- Transport aircraft: passengers (airliners) or cargo (freighters).


http://www.airliners.net/photo/Lufthansa/Airbus-A330-343X/2054700

- Mission profile (simplified): distance vs. altitude




- Possible design goals for cruise:
  - maximize distance (range)

$$R = \frac{a}{g}\frac{1}{c_f} M \frac{L}{D} \ln\left(1 + \frac{m_f}{m_e}\right)$$

  - minimize fuel expenditure

$$m_f = m_e\left(\exp\left(\frac{R}{\frac{a}{g}\frac{1}{c_f} M \frac{L}{D}}\right) - 1\right)$$

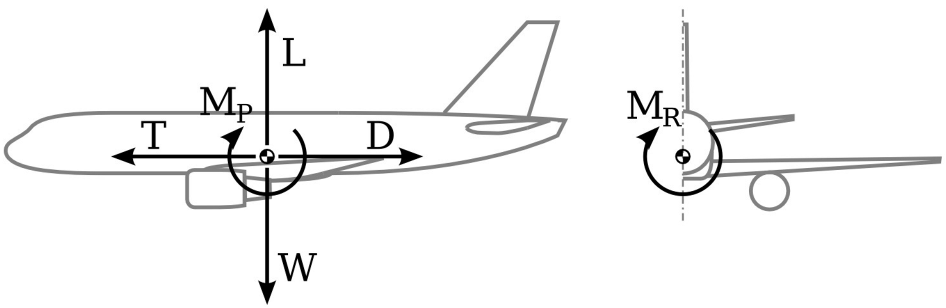
aerodynamics

structure

propulsion

# Aerodynamic Cruise Performance

- Force balance in horizontal flight:




| steady flight equations | non-dim. quantities |
|---|---|
| $L - W = 0$ | $L \rightarrow C_L$ |
| $T - D = 0$ | $D \rightarrow C_D$ |
| $M_p = 0$ | $M_P \rightarrow C_M$ |
| | $M_R \rightarrow C_R$ |

- Maximize Mach-scaled lift-to-drag ratio, at several near-design flight Mach numbers (multi-point optimization):

$$\left( \sum_k M_k \frac{L_k}{D_k} = \right) \sum_k M_k \frac{C_{L_k}}{C_{D_k}} \rightarrow \max , \quad k = 1..\mathrm{p}$$

- Under the constraints (with signs of moments as pictured):

$$C_{L_k} = C^T_{L_k}; \quad C_{M_k} = 0 \ \left(\text{or} \geqslant C^T_{M_k}\right); \quad C_{R_k} \geqslant C^T_{R_k}; \quad G_l = G^T_l; \quad i = 1..\mathrm{p}; \quad l = 1..\mathrm{q}$$

- By modifying the aircraft outer shape through design parameters:

$$D_i , \quad i = 1..\mathrm{n}$$

# Optimization Method

# Character of The Optimization Problem

- Small number of cost functions (goal and constraints) ~ O(10).
- Large (compared to #CF) number of design parameters ~ O(100).
- *Very high* computational cost of cost function evaluation:
  - CFD simulation based on RANS equations.
  - Simulation run-time in hours, using O(100) CPU cores.
- The requirements on the optimization algorithm:
  - The algorithm must converge using small number of cost function evaluations → *gradient-based*.
  - Algorithm internal computation and storage cost (e.g. linear system) is insignificant compared to cost function evaluation.
  - Constraints must be handled explicitly (not e.g. as penalties).
- Therefore, we use:
  - SQP (sequential quadratic programming) as the optimizer.
  - Evaluation of the cost function gradients by the *adjoint* method.

# Adjoint Gradient Computation Theory

$D$ design parameters
$W$ flow state
$X$ CFD mesh points

$J(W, X)$ cost function $(C_L, C_D, ...)$ flow
$R(W, X)=0$ flow state equations (RANS, SA turb.)
$T(X, D)=0$ mesh state equations (linear elasticity)

construct $\tilde{J}=J+R\Lambda_f+T\Lambda_m \quad (\equiv J)$
with $\Lambda_f, \Lambda_m$ arbitrary fields on $X$

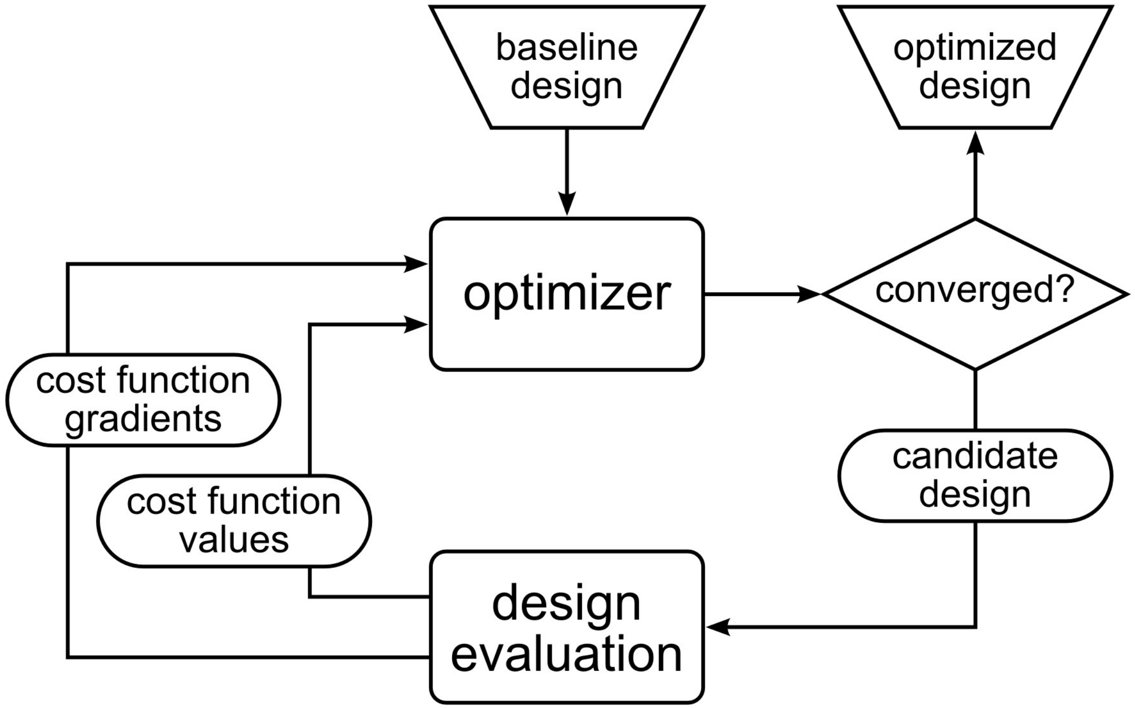
$$\frac{d\tilde{J}}{dD}=\frac{\partial J}{\partial W}\frac{dW}{dD}+\frac{\partial J}{\partial X}\frac{dX}{dD}+\frac{\partial R}{\partial W}\frac{dW}{dD}\Lambda_f+\frac{\partial R}{\partial X}\frac{dX}{dD}\Lambda_f+\frac{\partial T}{\partial X}\frac{dX}{dD}\Lambda_m+\frac{\partial T}{\partial D}\Lambda_m$$

$$=\left(\frac{\partial J}{\partial W}+\frac{\partial R}{\partial W}\Lambda_f\right)\frac{dW}{dD}+\left(\frac{\partial J}{\partial X}+\frac{\partial R}{\partial X}\Lambda_f+\frac{\partial T}{\partial X}\Lambda_m\right)\frac{dX}{dD}+\frac{\partial T}{\partial D}\Lambda_m$$

compute $\Lambda_f, \Lambda_m$ s.t. $\frac{\partial R}{\partial W}\Lambda_f=-\frac{\partial J}{\partial W}$; $\quad \frac{\partial T}{\partial X}\Lambda_m=-\frac{\partial J}{\partial X}-\frac{\partial R}{\partial X}\Lambda_f$ f-adj m-adj j-defo

finally the gradient becomes $\frac{d\tilde{J}}{dD}\equiv\frac{dJ}{dD}=\frac{\partial T}{\partial D}\Lambda_m$ s-pert s-dot

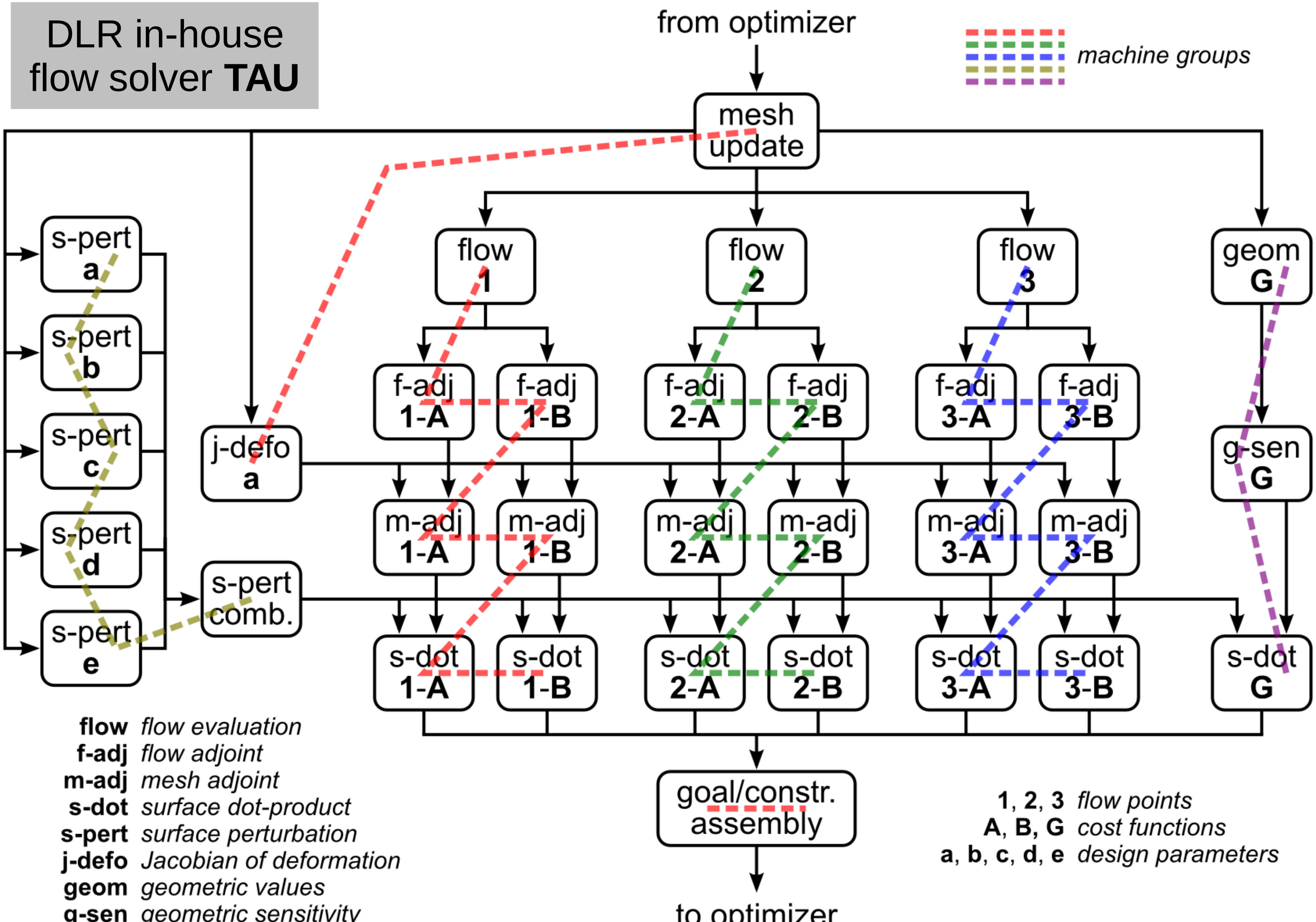
# Optimization Workflow: Optimizer Loop




DLR in-house optimization framework **Pyranha**

# Optimization Workflow: Design Evaluation

DLR in-house flow solver **TAU**

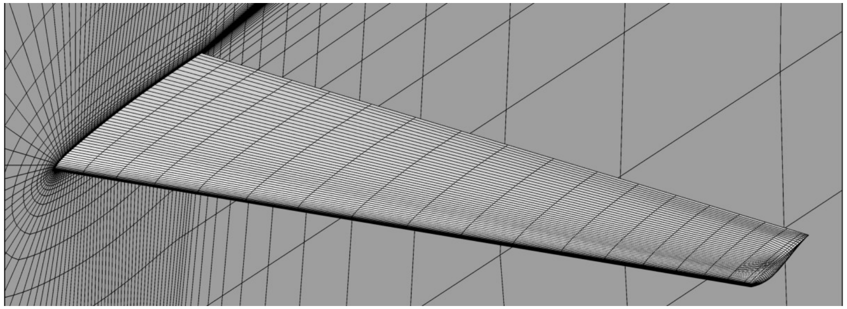


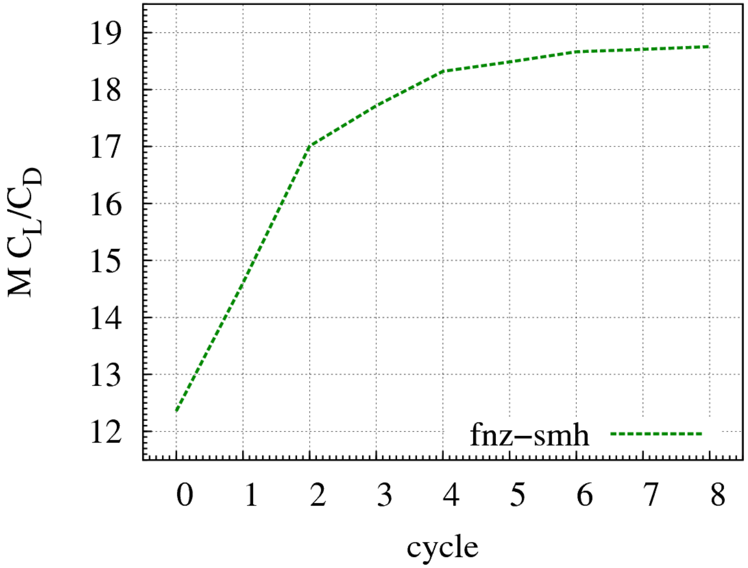
# Application Examples


DLR

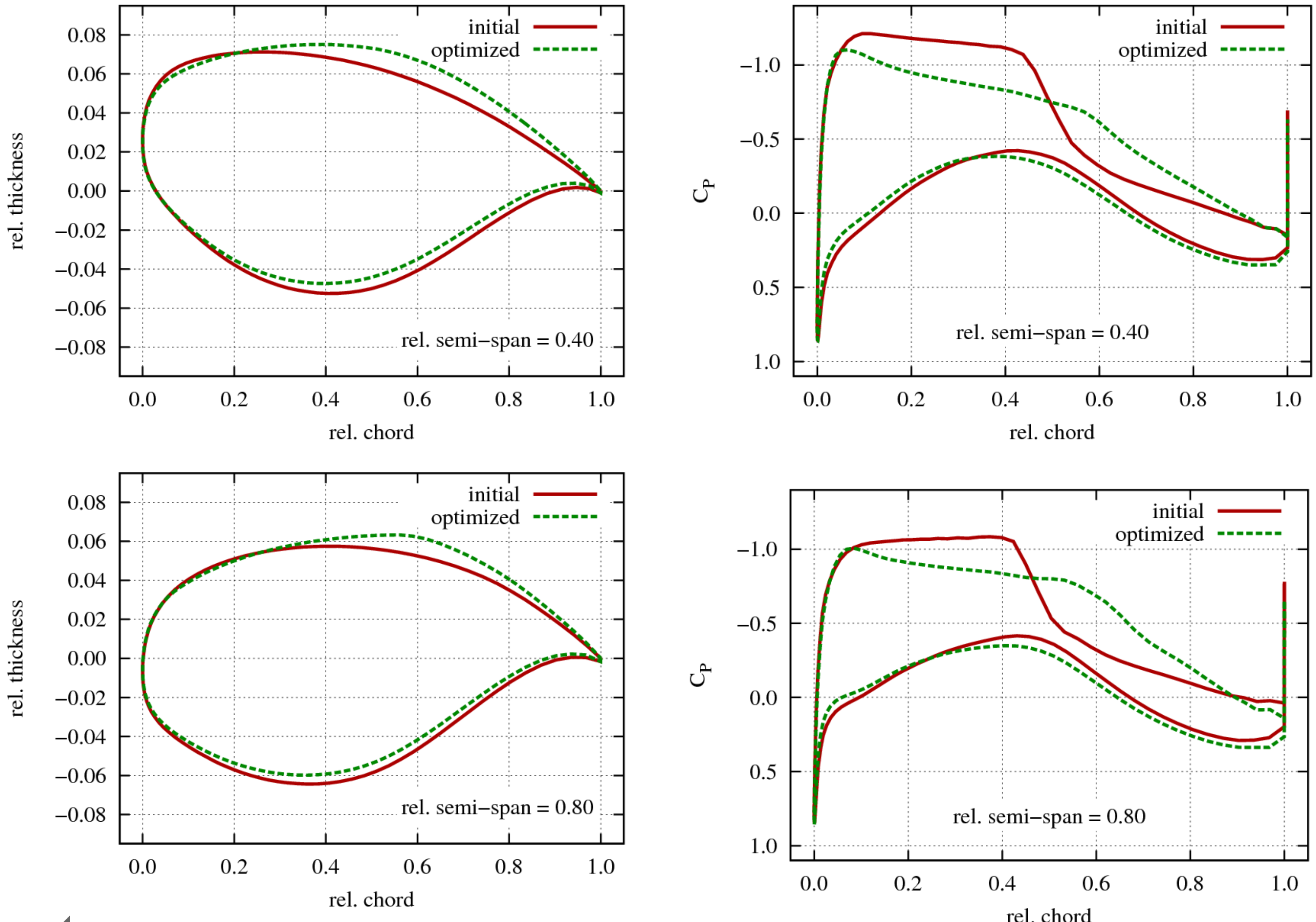
# Transonic Wing: Setup and Convergence

- Simple problem, but *every mesh surface node* a design parameter.
- LANN wing:
  - AR = 7.9, n = 0.4, t/c = 0.12, $\Lambda$ = 25°, supercritical sections.
  - M = 0.82, Re = 7.3 M, $C_L$ = 0.53.
- Optimization setup:
  - Objective: maximize M $C_L$ / $C_D$.
  - $C_L$ implicitly constrained through flow solver fix-point iteration.
  - Internal volume explicitly constrained.
  - Free-node (z-direction), 3250 design parameters.
  - Trailing edge nodes fixed.



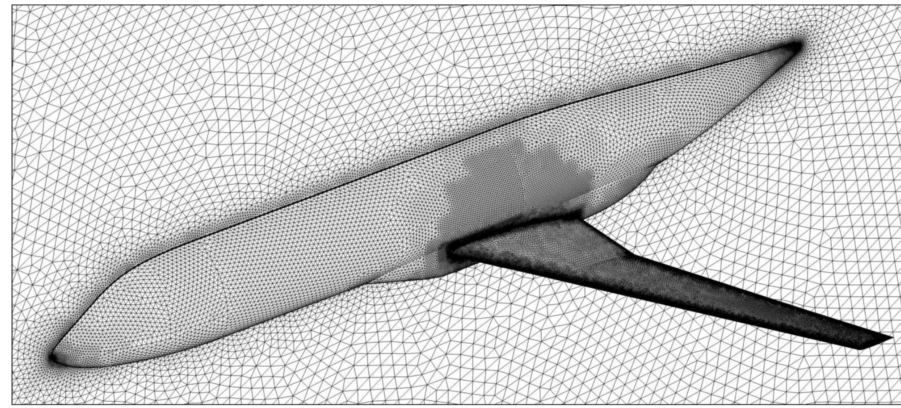


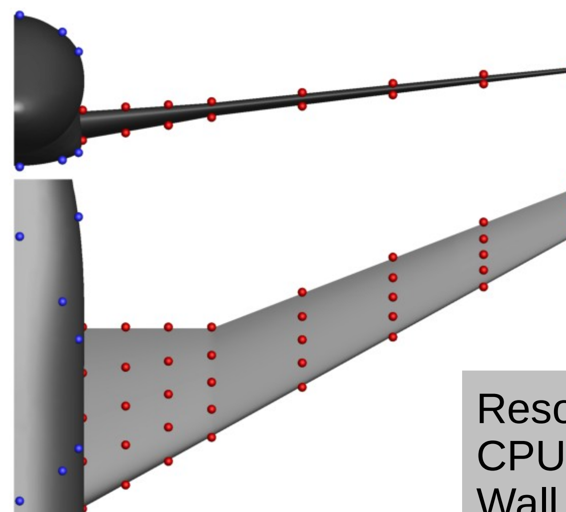
# Transonic Wing: Results



# Transonic Wing-Body: Setup

- Wing-body based on the Do-728.
- Design point:
  M = 0.80, Re = 21 M, $C_L$ = 0.55.
- CFD mesh:
  hybrid-unstructured, 3 M points.
- Parametrization: 80 FFD (free-form deformation) control pts. on the wing.
- Single- and multi-point optimization:
  **SP**: M = 0.80
  **MP 3**: M = 0.78, 0.80, 0.82
  **MP 5**: M = 0.76, 0.78, 0.79, 0.80, 0.81
- Goal: maximize sum(M $C_L$ / $C_D$).
- $C_L$ implicitly constrained through flow solver fix-point iteration.
- Wing thickness implicitly constrained by linking upper-lower control points.
- Explicit constraints: $C_M$ (every point), $C_R$ (design point).

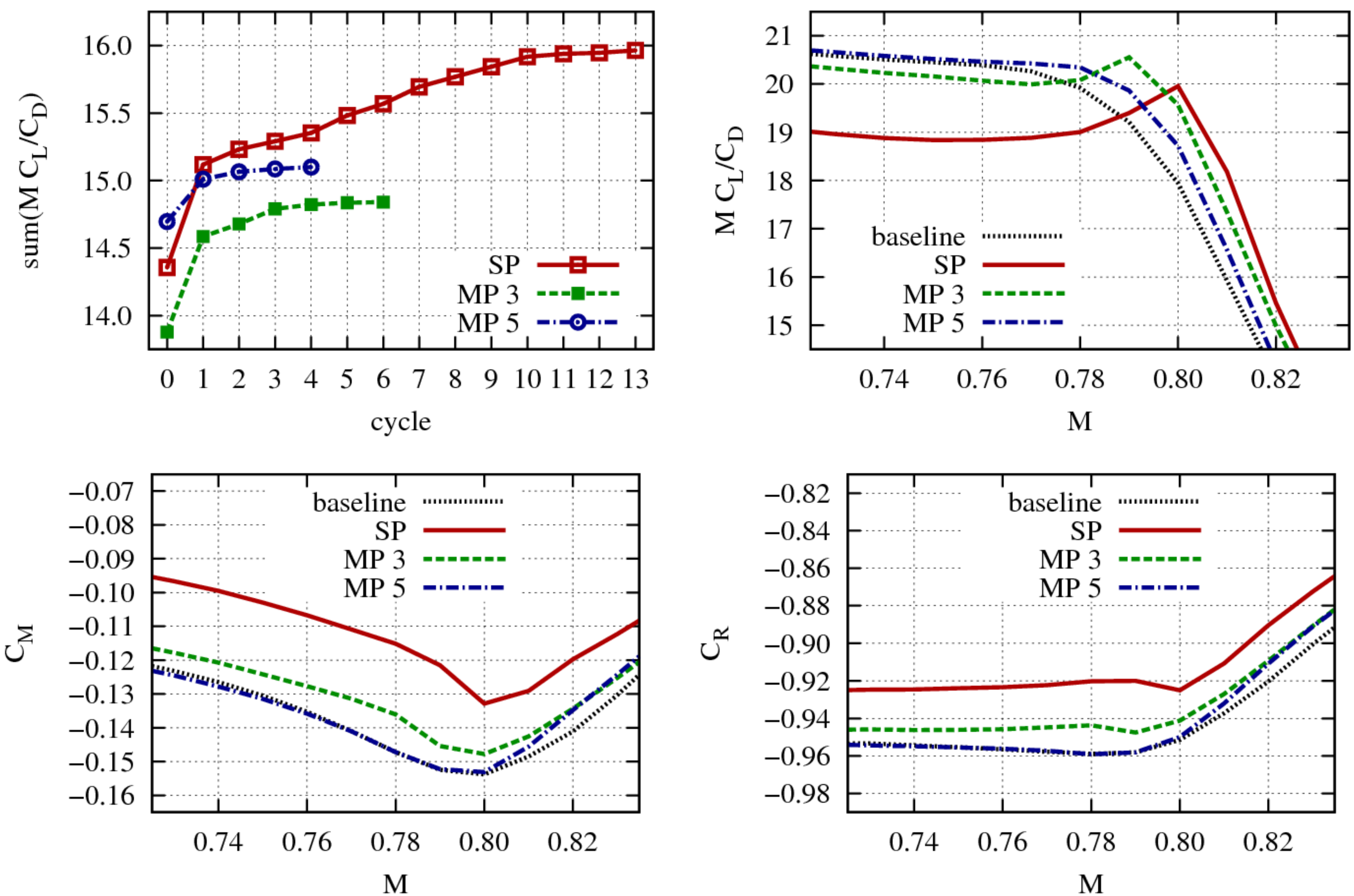




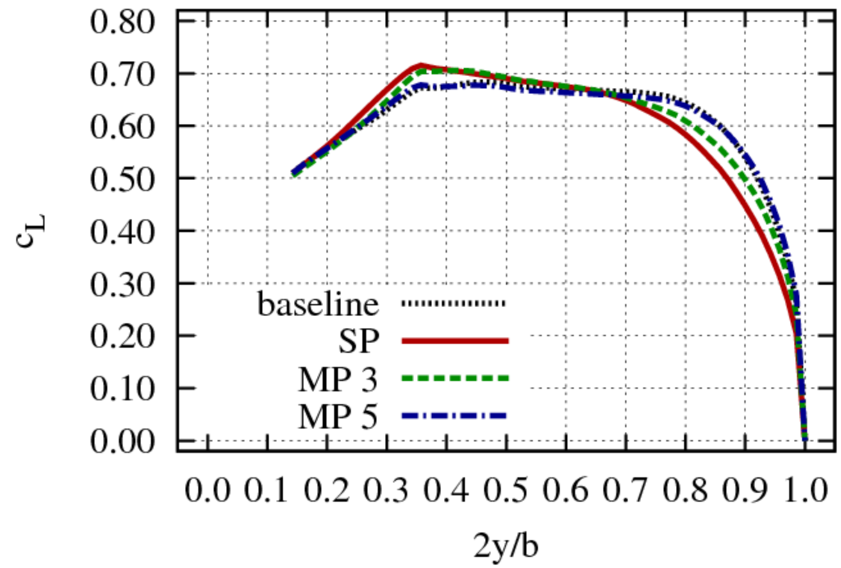
Resources for MP 5:
CPU: 480 cores
Wall time: ~36 hrs

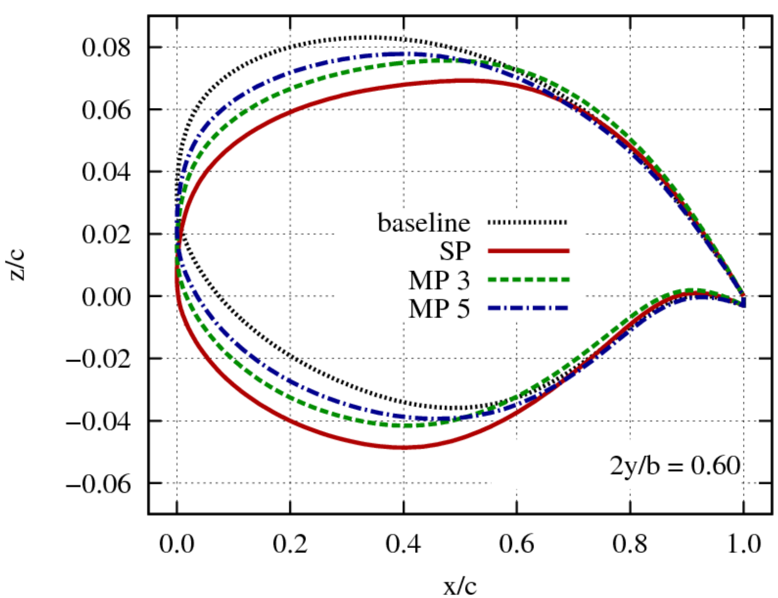
DLR

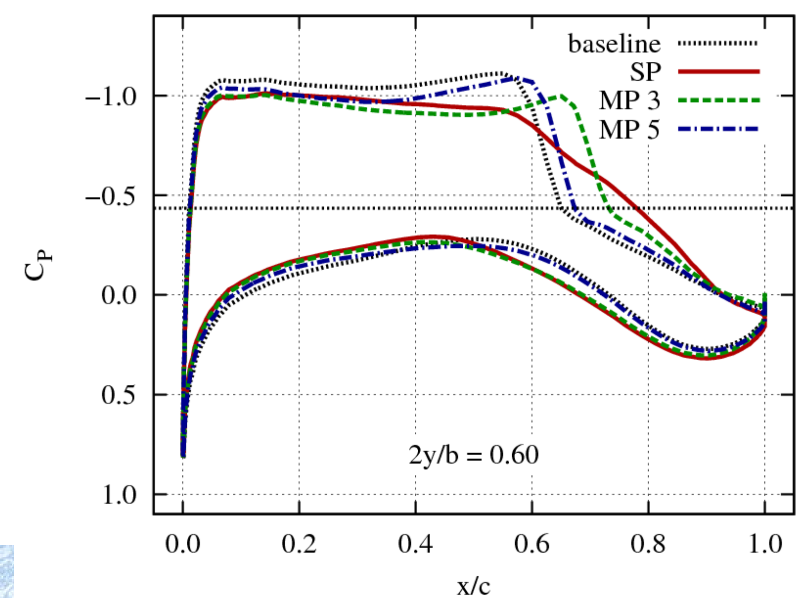
# Transonic Wing-Body: Convergence, Performance



# Transonic Wing-Body: Spanwise and Section Load



# Issues and Outlook


DLR

# Issues and Outlook

- Issues:
  - Cost function evaluation always “noisy” in practice (e.g. due to less-than-perfect convergence of flow/adjoint simulations).
  - Optimizer “cheats” as much as possible (exploits any insufficient constraining or non-considered operating conditions).
  - Not quite “user-friendly” optimization tool chain.

- Ongoing work:
  - Find gradient-based optimization algorithms that are:
    - more robust in face of noise in cost function value/gradient;
    - preserving feasibility as much as possible during cycles.
  - Find a way to pick relevant operating conditions to consider in multi-point optimization (not too many, but significant).
  - Add more “primitive” cost functions (value *and* gradient).
  - Assemble well-documented and deployable optimization tool chain (also with a GUI).

# Thank you for your attention!


DLR