One-shot methods review of existing approaches

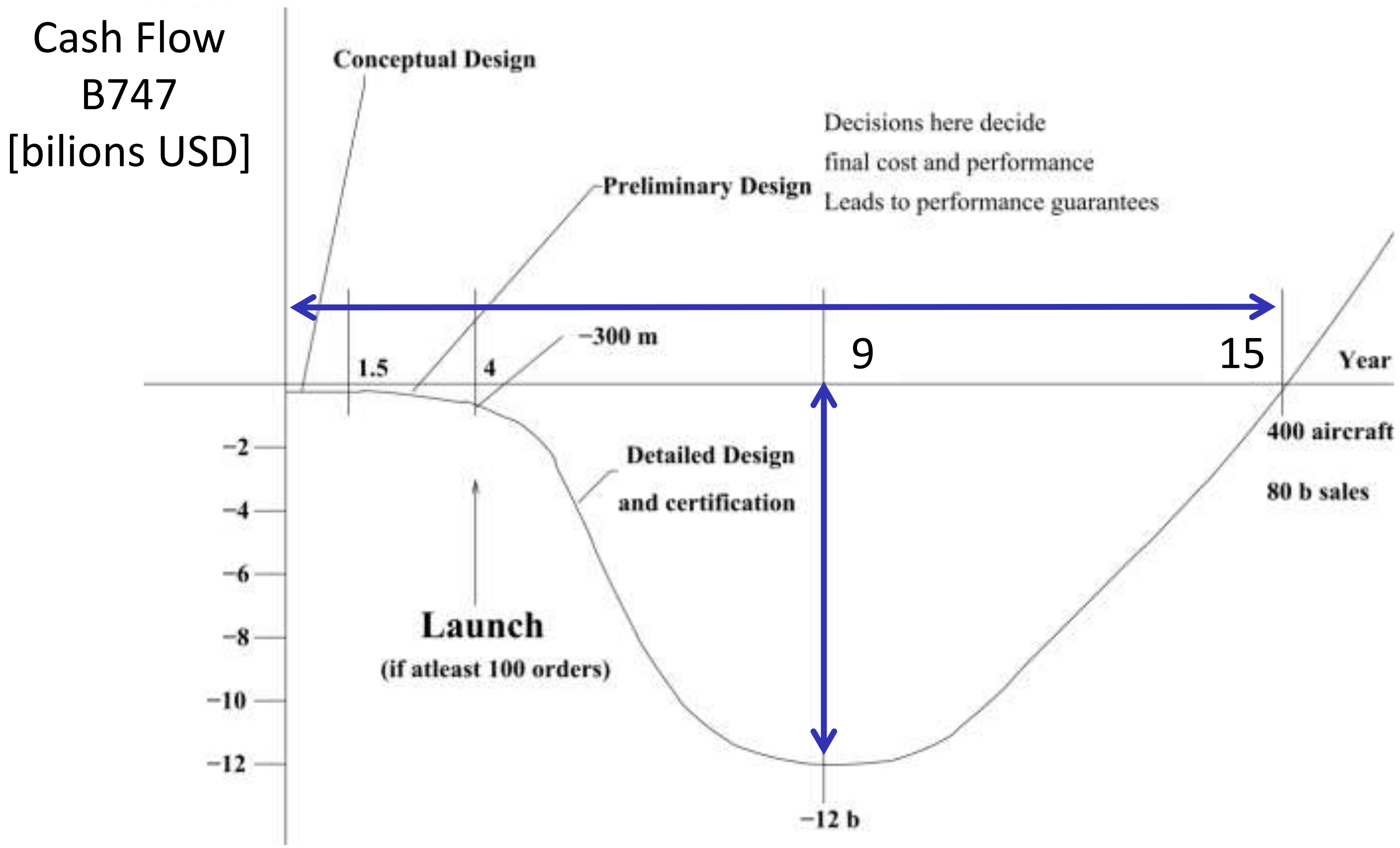
Armen Jaworski

armen@meil.pw.edu.pl

24.06.2015 r.

Motivation

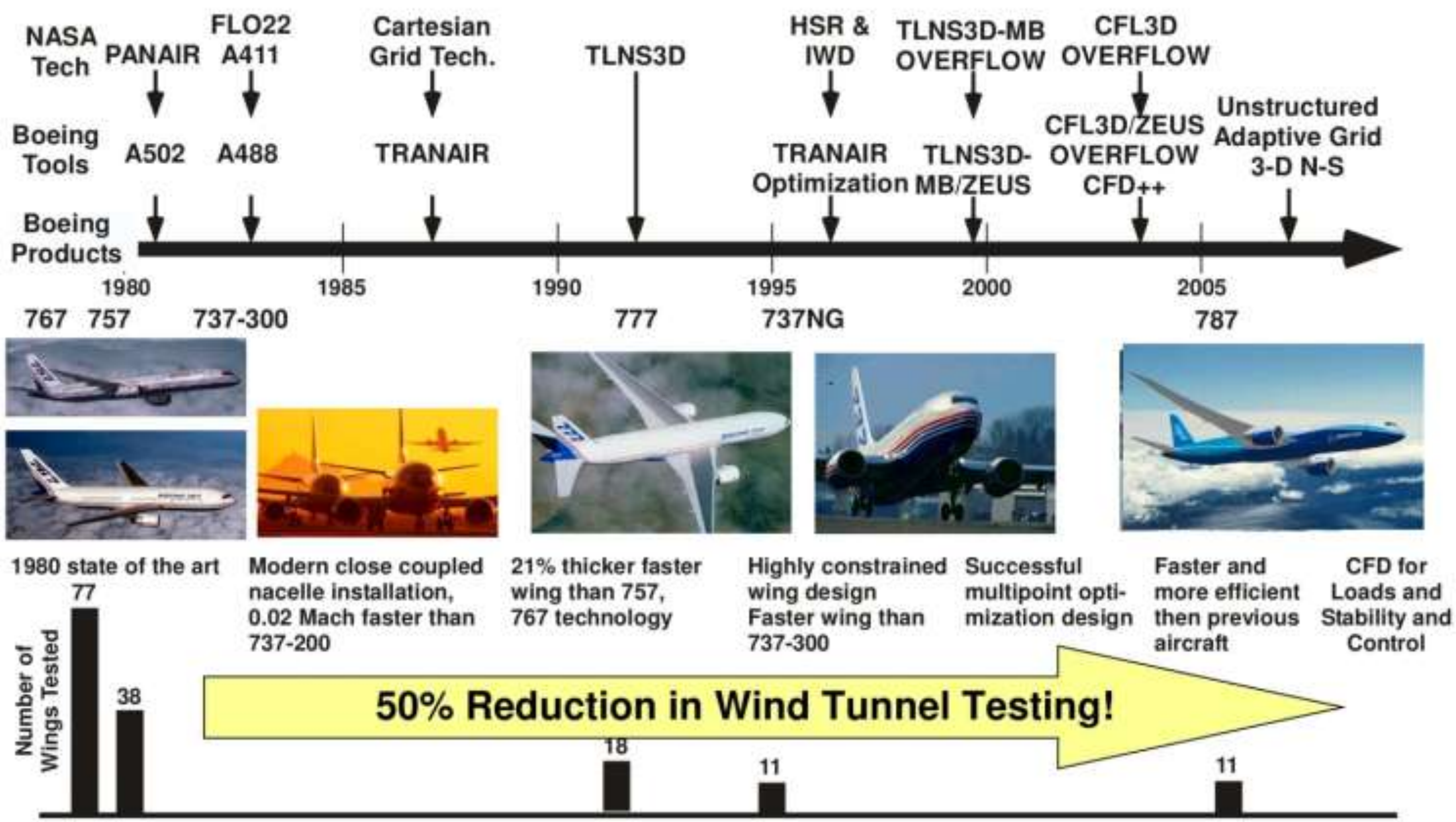
- Increasing demand on efficiency and emission targets
- Development of a new airplane or car requires high investment and time



Source: Antony Jameson, „Airplane Design with Aerodynamic Shape Optimization", Shanghai 2010

Motivation

- Numerical simulations have an increasing share in the design process



Source: A. Jameson, "Airplane Design with Aerodynamic Shape Optimization", Shanghai 2010

Motivation

CFD simulations

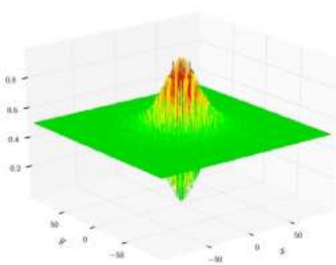
- becoming common practice in the design process
- Limited by computational power

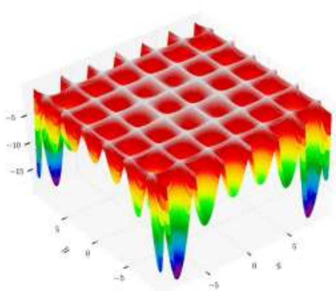
Optimisation

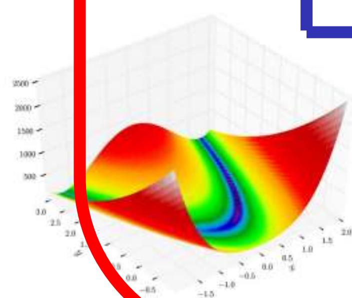
- Require many CFD calculations
- Needs many simulations - computational cost is the main limiation
- **It is necessary to develop algorithms which can speed up the process of aerodynamic optimisation to enable its wider application in practice**

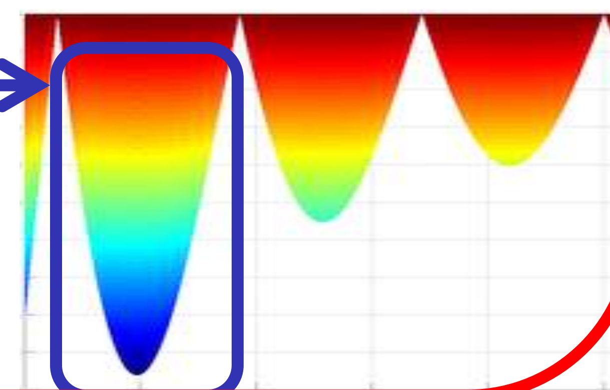
Aerodynamic Optimisation

Evolutionary algorithms	Gradient based
▶ Requires: objective function **+**	▶ Requires: objective function **–** + gradient **–**
▶ Effective for "difficult" problems **+**	▶ Ineffective for difficult problems **+**
▶ Requires around 1000 calculations of the objective function **–**	▶ Convergence with cost comparable to 1-2 f & g calculations **+**
▶ Limited number of parameters (10-20) **–**	▶ Large number of parameters (at the order of 1000)


Shaffer function

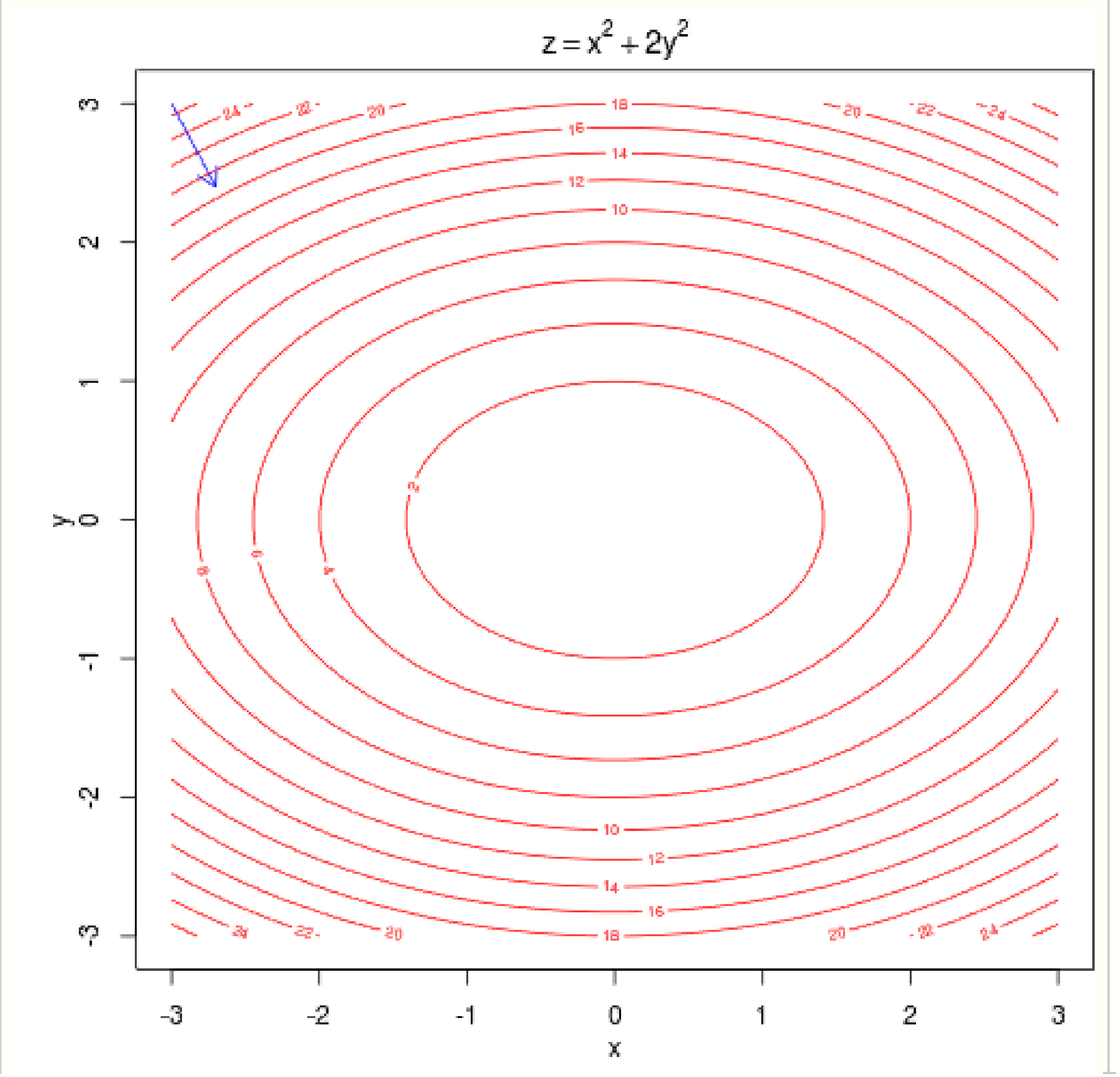

Holder table function


Rosenbrock's function



Gradient based optimisation

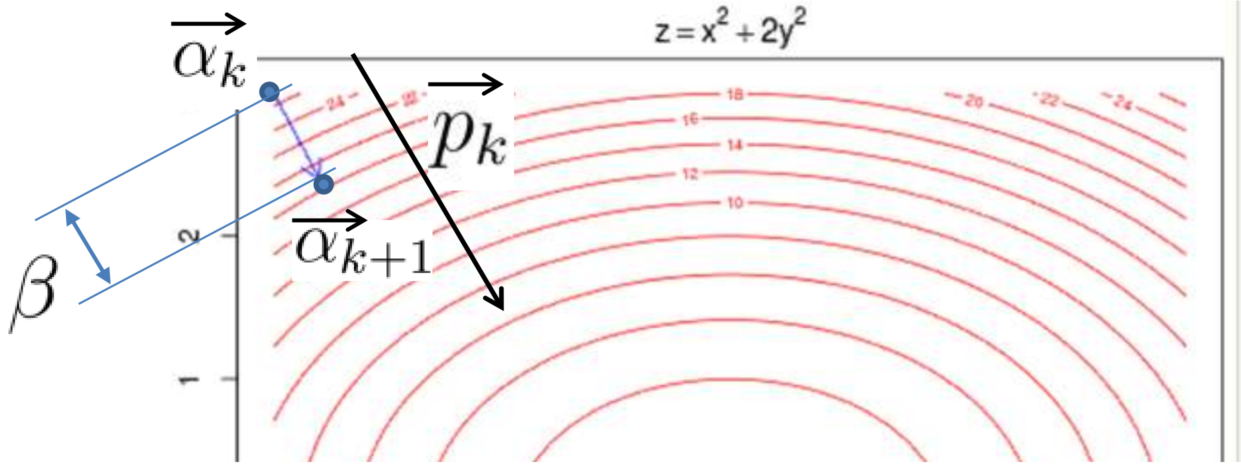
Parametr 1



Parametr 2

- Gradient is required

Gradient based optimisation



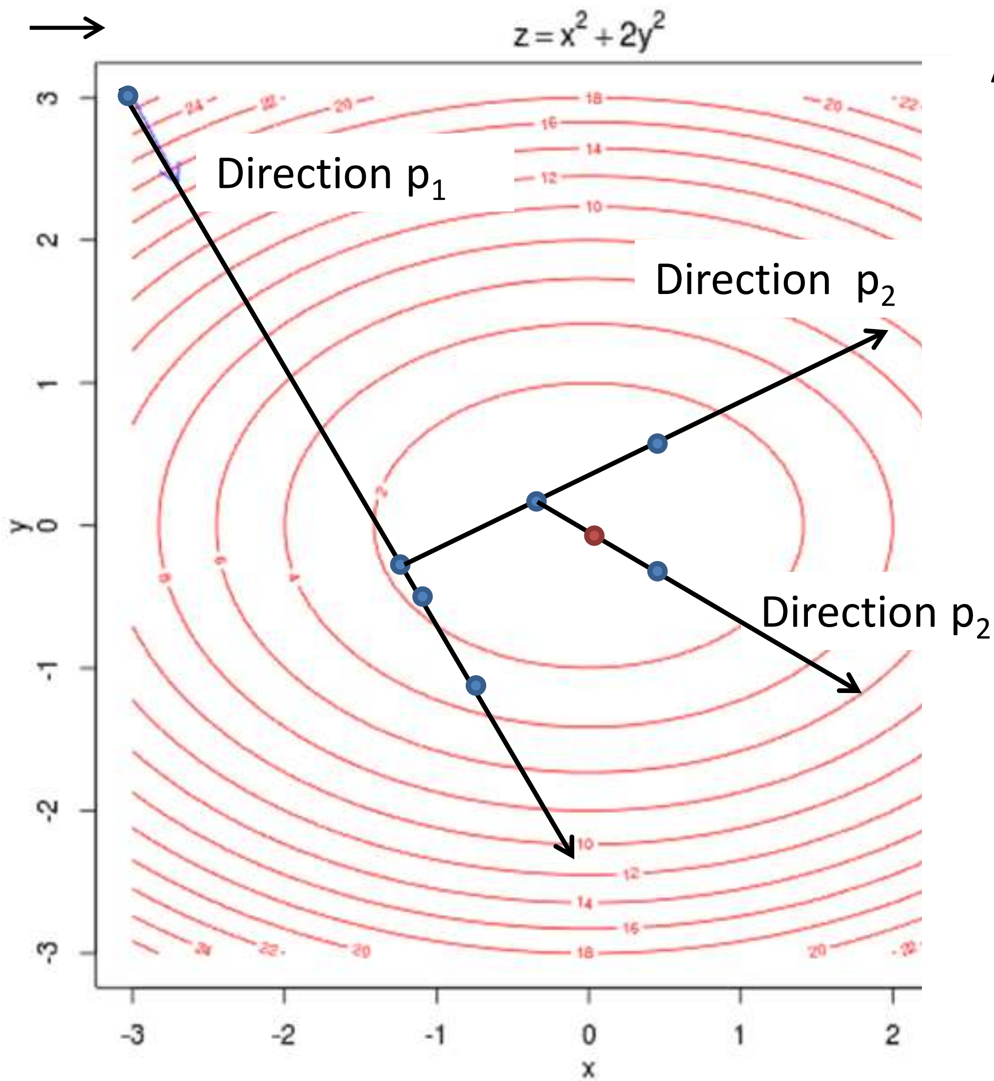
$$\overrightarrow{\alpha_{k+1}} = \overrightarrow{\alpha_k} - \beta \cdot \overrightarrow{p_k}$$

$\overrightarrow{\alpha_k}$ - Design variables

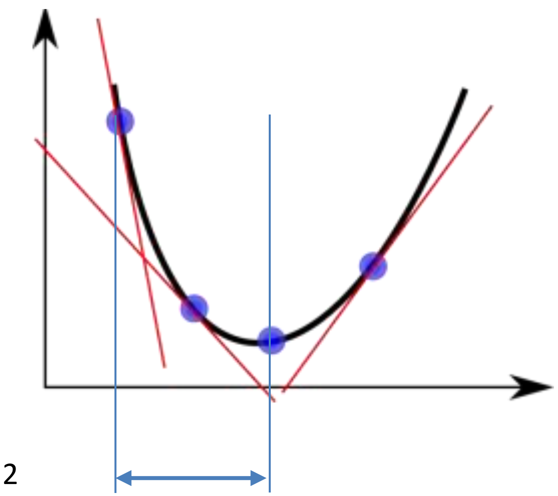
β - Step lenght

$\overrightarrow{p_k}$ - Descent direction

L-BFGS-B



At each search direction
1D problem

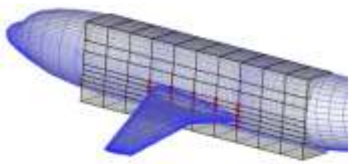


Optimal β at the direction

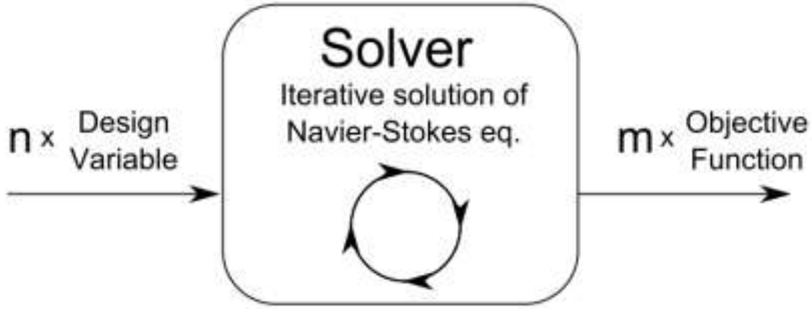
Aerodynamic Optimisation

Number of parameters

n – order of 1000

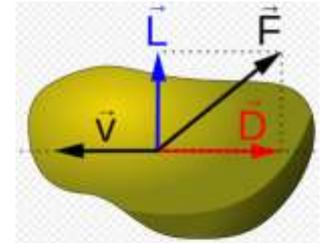


Single SIMULATION



Number of objectives

m – order of 1-3

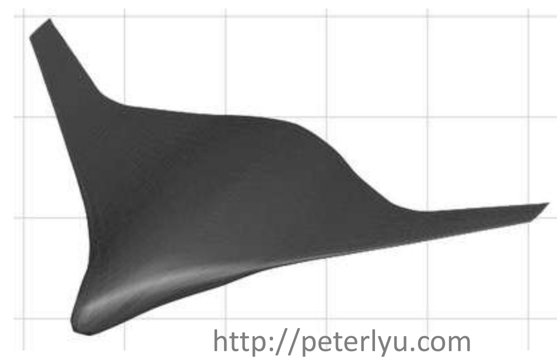


⇩

Adjoint method

- **Pironneau** (**1964**) Optimum shape design for subsonic potential flow
- **Jameson** (**1988**) Optimum shape design for transonic and supersonic flow modeled by the transonic potential flow equation and the Euler equations

⇨



Adjoint method

- Linearisation in reverse – adjoint method

$$L(Q,\alpha) \Rightarrow \frac{dL}{d\alpha} = \frac{\partial L}{\partial \alpha} + \boxed{\frac{\partial L}{\partial Q}}\frac{\partial Q}{\partial \alpha}$$

$$\frac{dL}{d\alpha} = \frac{\partial L}{\partial \alpha} + \overbrace{\boxed{\left(\frac{\partial L}{\partial R}\right)^T}\frac{\partial R}{\partial Q}\frac{\partial Q}{\partial \alpha}}$$

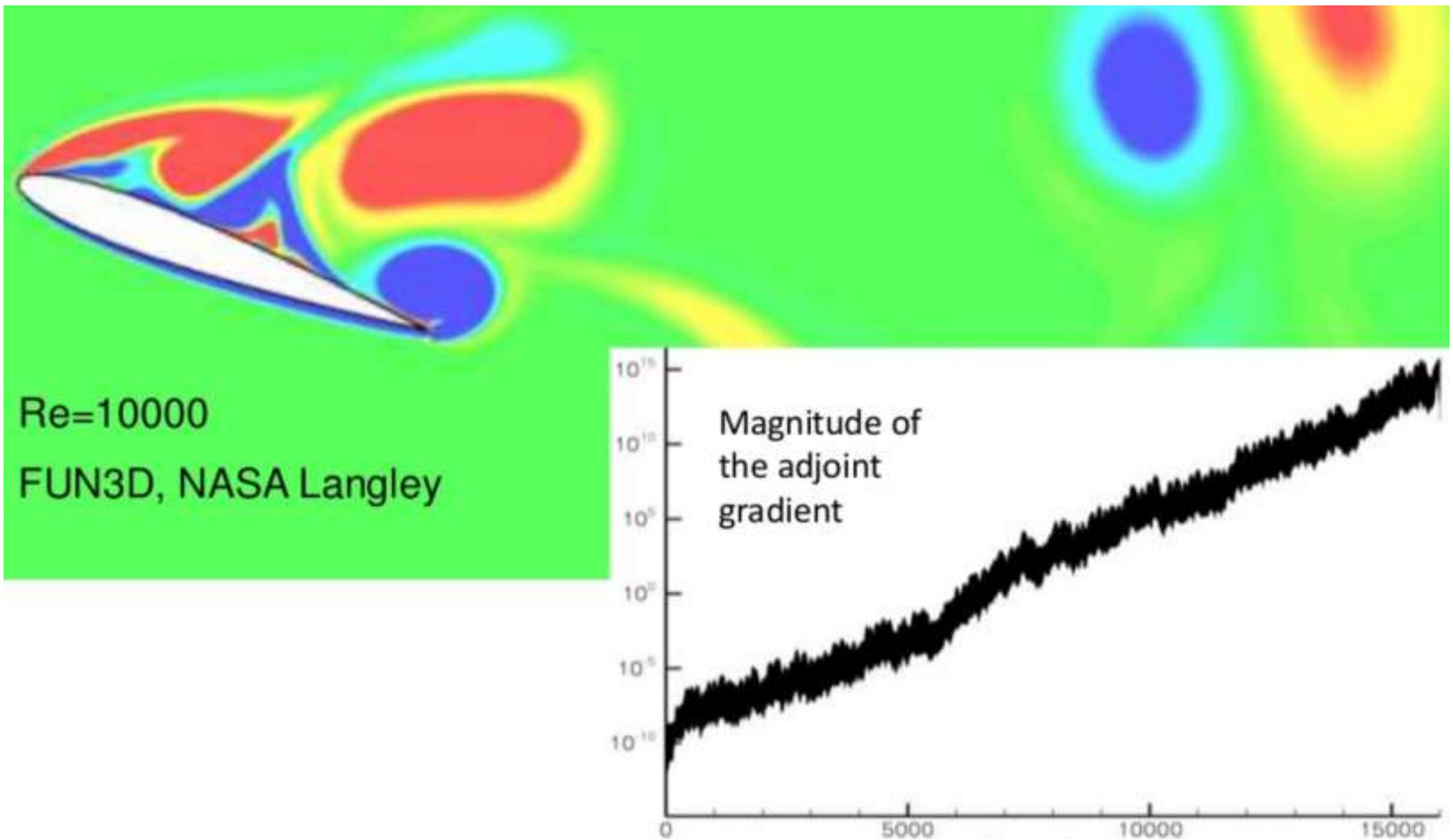
$$\left(\frac{\partial \mathrm{R}}{\partial \mathrm{Q}}\right)^{\mathrm{T}} \mathrm{v} = \left(\frac{\partial \mathrm{L}}{\partial \mathrm{Q}}\right)^{\mathrm{T}}$$

Linear system $\mathrm{A}^{\mathrm{T}}\mathrm{v} = \mathrm{g}$

- **Cost ~ m** (objective functions ~1-3) x (linear system ~ 10^8 variables)
- <u>The cost is independent of the number of design variables</u>

Adjoint method

- Problems with chaotic flows



Source: QiQi Wang, „Least squares shadowing” Dartmouth 2014

Adjoint method - limitations

Use of Computational Simulations	**Analysis** Simulation on manually picked geometry and condition	**Design** Beyond single simulation, towards optimization and parametric study
Low fidelity simulation Potential flow solver, RANS, URANS		**THE FRONTIER**
High fidelity Simulation Large Eddy Simulation (LES), Detached Eddy Simulation (DES), Unsteady Multi-physics Simulations	**THE FRONTIER**	

Source: QiQi Wang, „Least squares shadowing” Dartmouth 2014

Cost of gradient based optimisation

I_k

1.	mesh $G_k \rightarrow$	CFD solver	$\rightarrow$ the flow $Q_k,\ L_k$	$I_k T_Q$
2.	$G_k,\ Q_k \rightarrow$	Adjoint solver	$\rightarrow$ the adjoint $v_k,\ \nabla L_k$	$I_k T_v$
3.	$\alpha_k,\ L_k,\ \nabla L_k \rightarrow$	Optimisation algorithm	$\rightarrow$ the new design α_{k+1}	T_{opt}
4.	$\alpha_{k+1},\ G_k \rightarrow$	Mesh Morphing	$\rightarrow$ the new mesh G_{k+1}	T_m

N

$$\text{Total Cost} = \sum_k^N I_k \cdot (T_Q + T_v) + N \cdot (T_{opt} + T_m)$$

How to reduce the optimisation cost

C) Reduce the number of design iterations

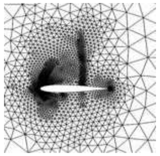
$$\text{Total Cost} = \sum_{k}^{N} I_k \cdot \underbrace{(T_Q + T_v)} + N \cdot (T_{opt} + T_m)$$

A) Reduce the number of flow and adjoint iterations

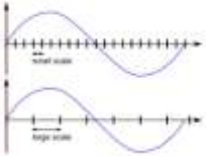
B) Limit time of single iteration of flow and adjoint

- A $\Rightarrow$ **One-shot method**

- B $\Rightarrow$ Mesh adaptation



- C $\Rightarrow$ Multigrid in optimisation



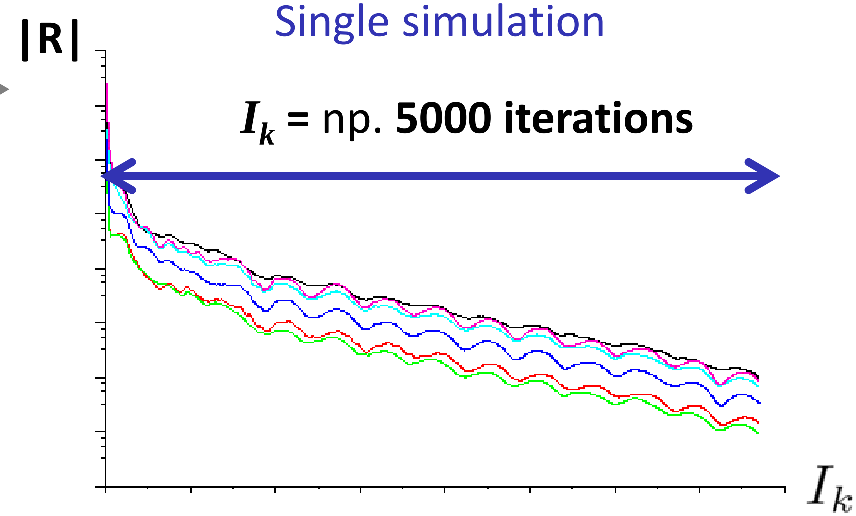
One-shot method

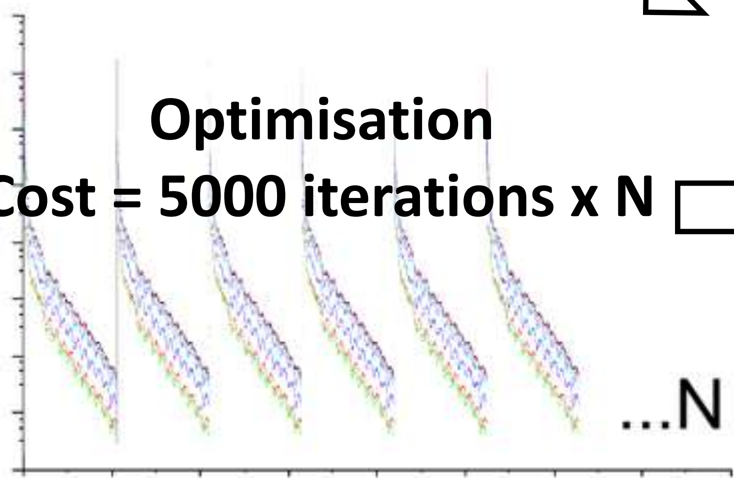
Typical optimisation

- Full convergence of simulation in each optimisation step

N

1. CFD solver
2. Adjoint solver
3. Optimisation algorithm
4. Mesh Morphing





Optimisation
Cost = 5000 iterations x N

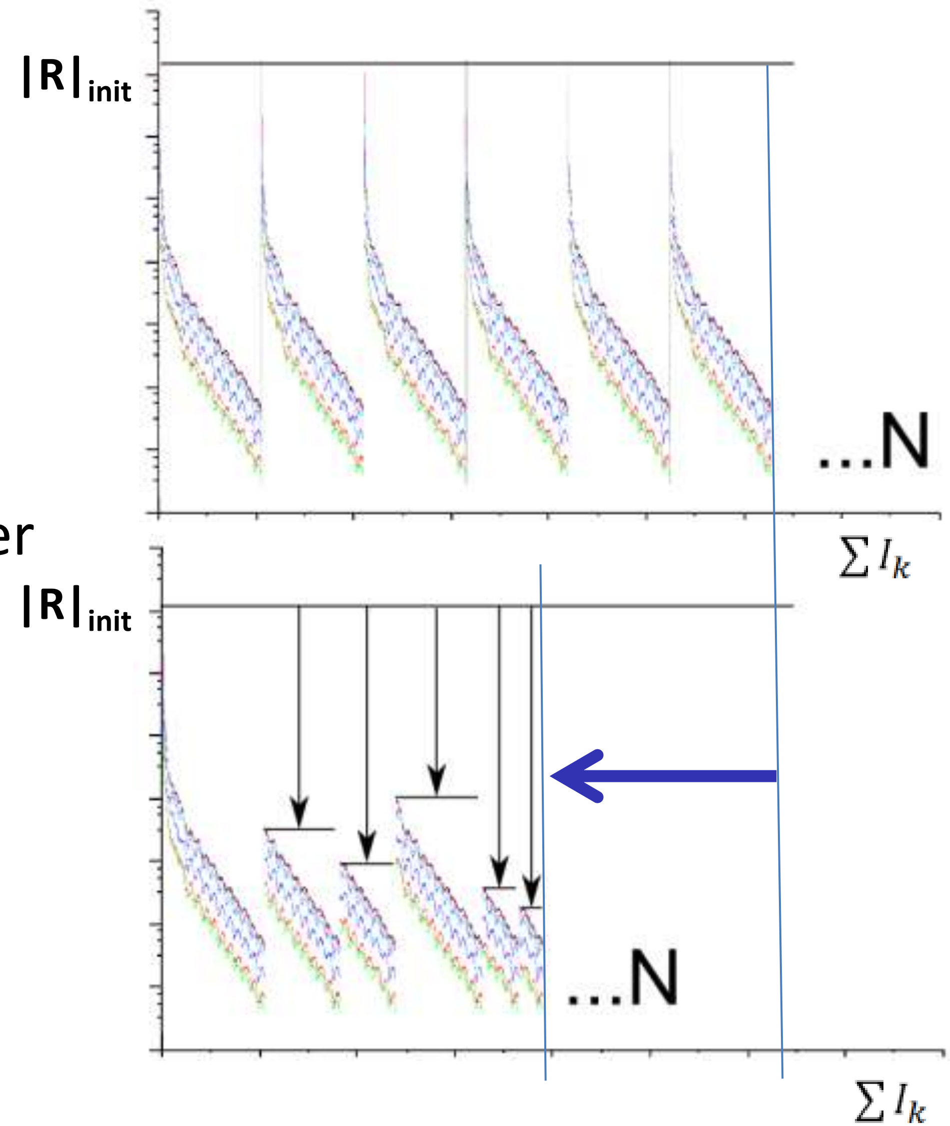
$$\text{Total Cost} = N \cdot I_k \cdot (T_Q + T_v)$$

Simple improvement

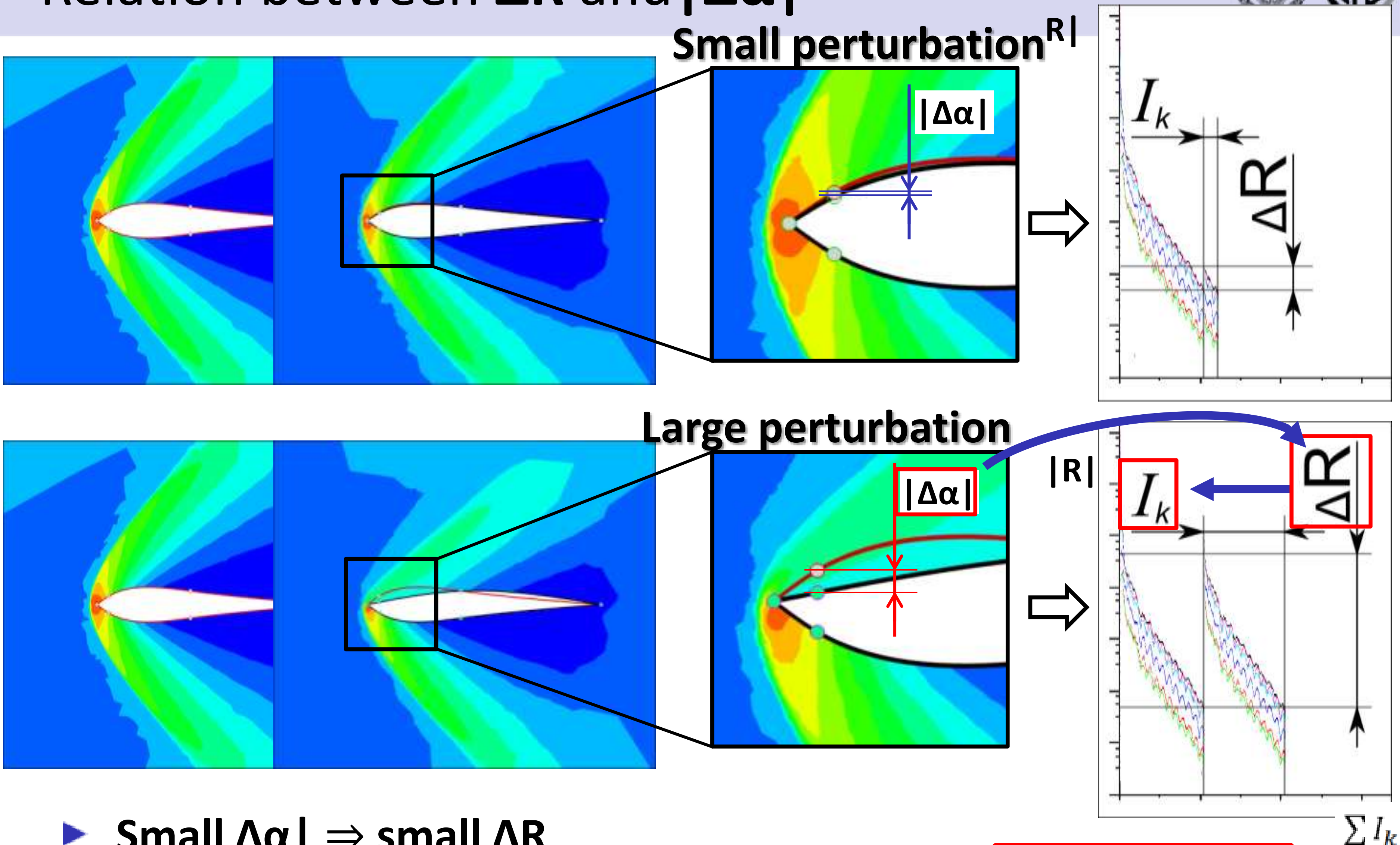
- The use of simulation result from the previous step optimization

⇩

- Further steps start from lower value of residuals R
- Lower $\boldsymbol{I_k}$
  ⇒ lower optimisation cost



Relation between ΔR and |Δα|

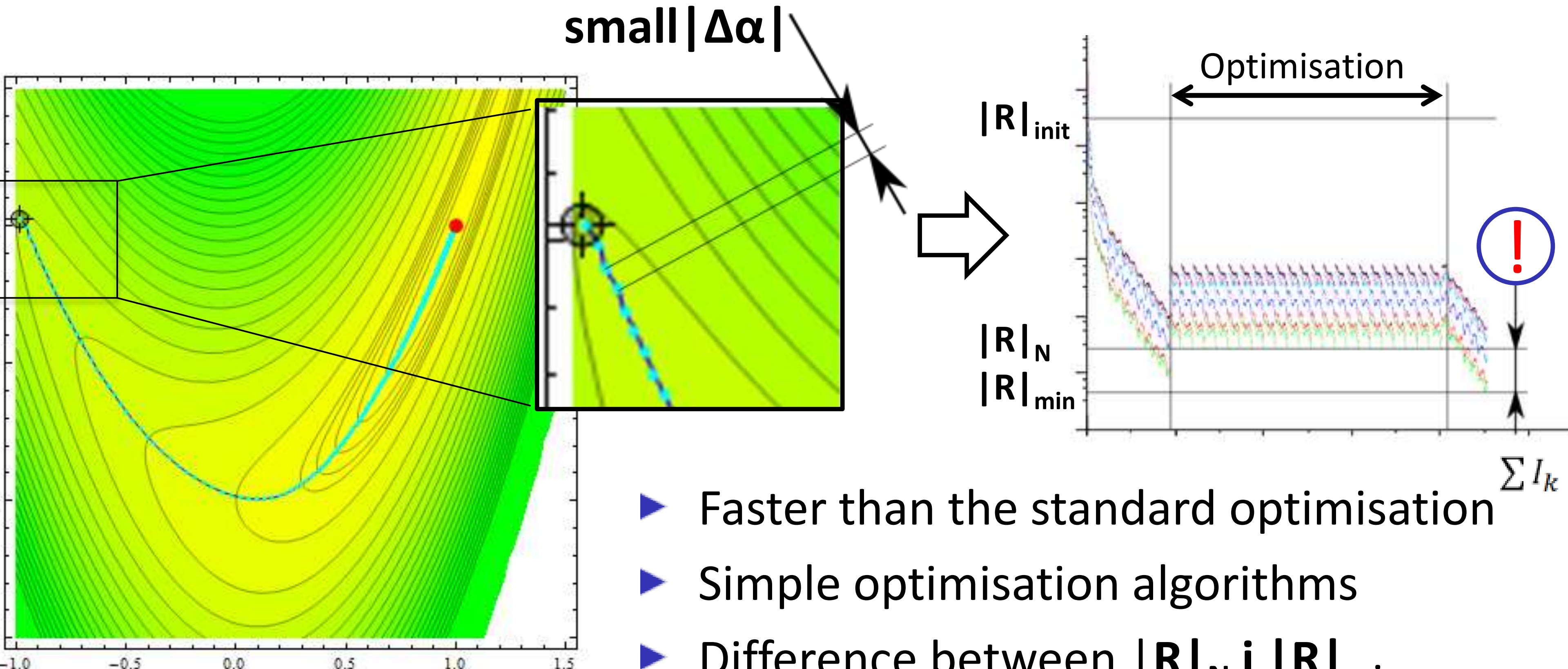


- **Small Δα| ⇒ small ΔR**
- **Large|Δα| ⇒ large ΔR**

⇨ **ΔR ⇒ I_k** ⇨ $\boxed{|\Delta\alpha| \Rightarrow I_k}$

One-shot ⇒ „ optimisation in one shot”

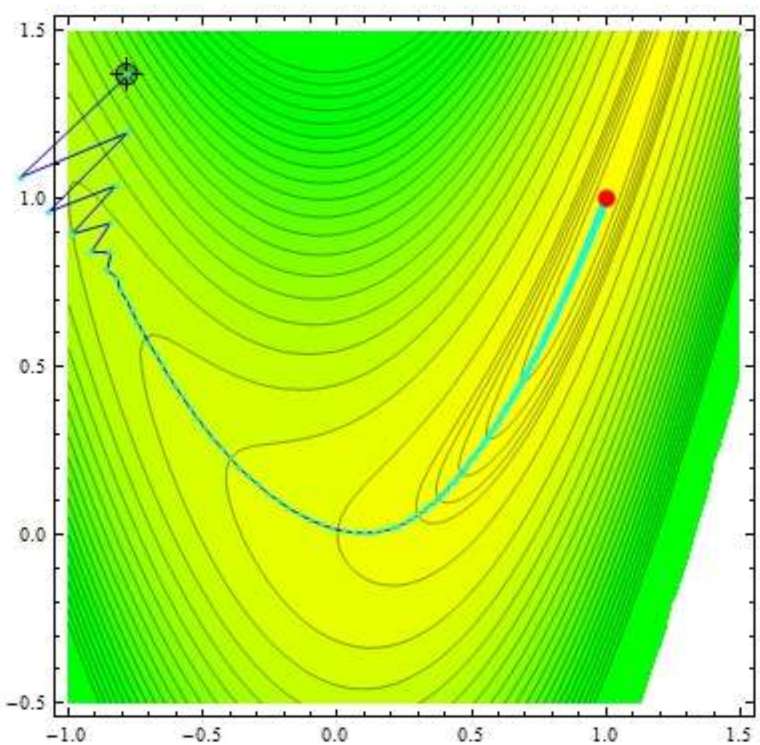
- **Solving simultaneously the flow and optimization - in one shot**
- Many opt. steps (**N**↑) with very small (**|Δα|**↓ ⇒ I_k ↓)



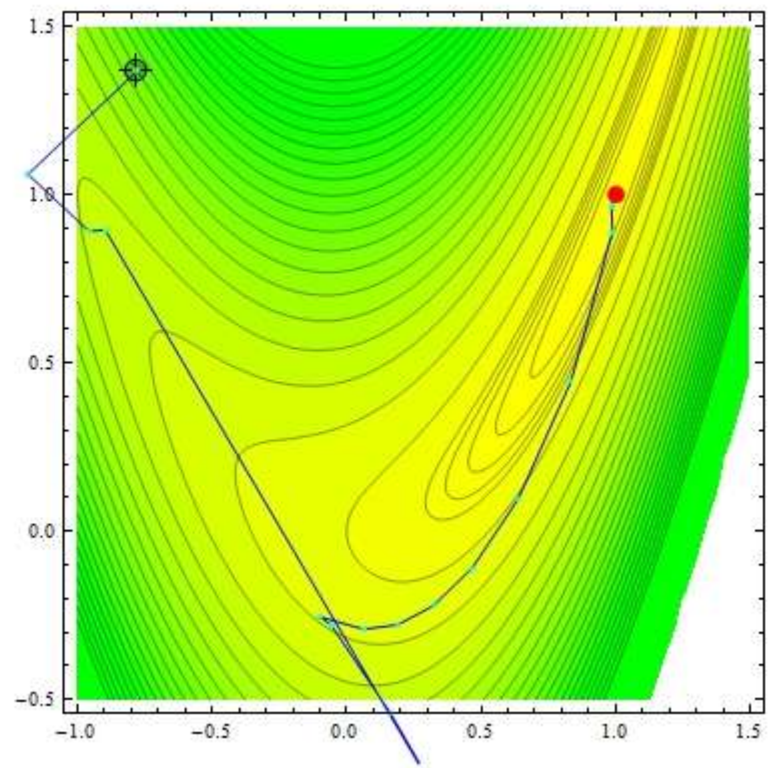
- Faster than the standard optimisation
- Simple optimisation algorithms
- Difference between **|R|**$_N$ **i** **|R|**$_{min}$

Optimisation algorithms

- „steepest descent”

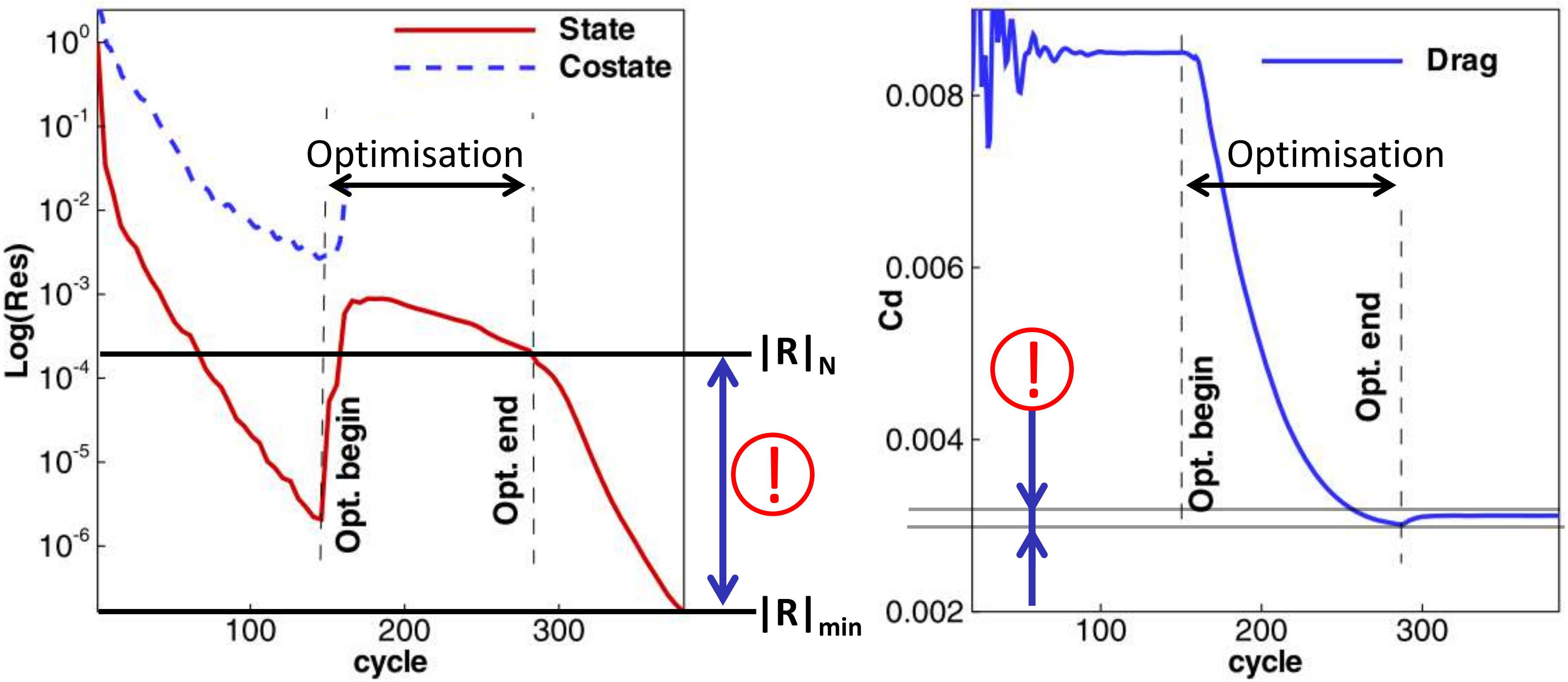


- „quasi-Newton”



An example of one-shot method

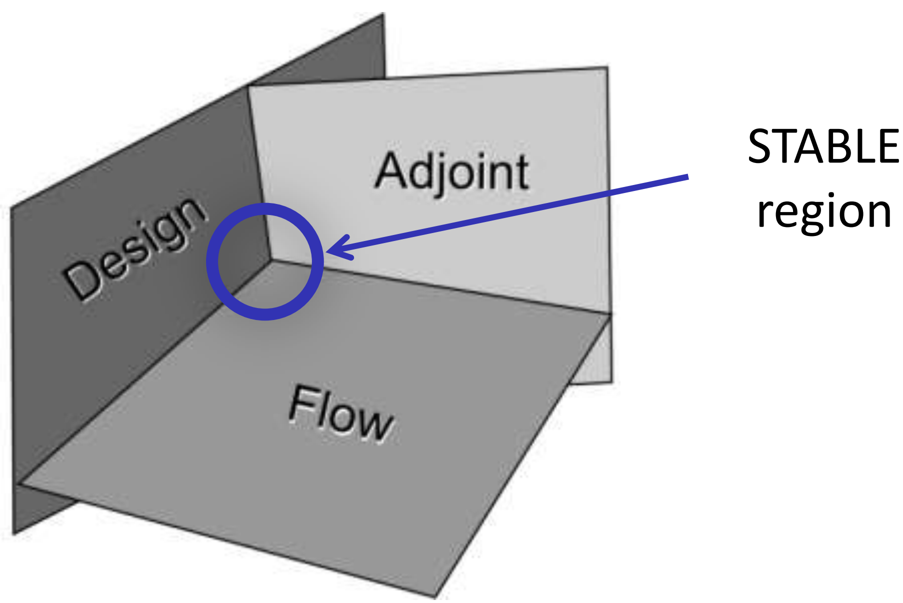
- Optimizing with a moderate accuracy level
- Exact solution only after finishing the optimisation



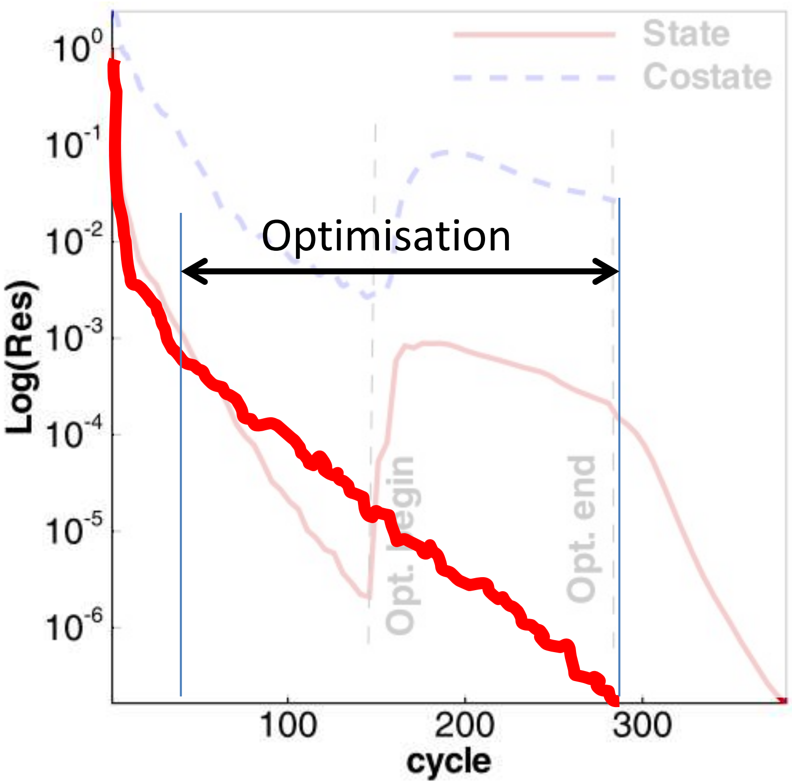
S. Hazra. Aerodynamic shape optimization using simultaneous pseudo-time-stepping. In Large-Scale PDE-Constrained Optimization in Applications, volume 49 of Lecture Notes in Applied and Computational Mechanics, pages 81–104. Springer Verlag, 2010.

One-shot ⇒ "optimization in one shot"

- **At the same time** ("in one shot") solve the flow equations optimisation problem (inaccurate simulation)
- **Stability** - requires appropriate balance between convergence solutions (flow and adjoint)? And optimization
- **One-shot method** – how to satisfy the stability condition with lowest optimisation cost



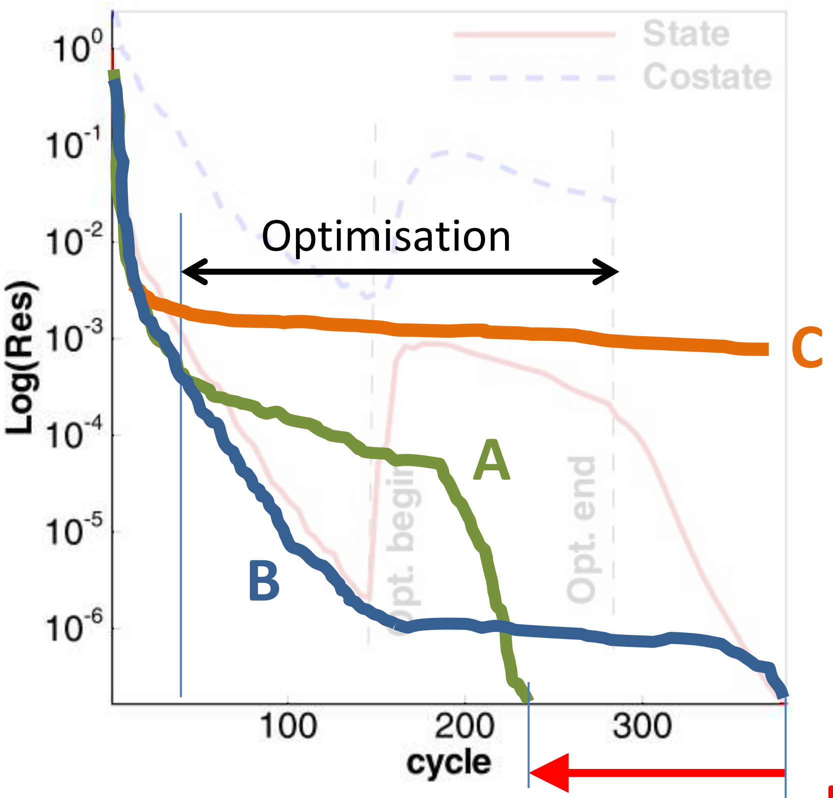
Another approach to one-shot



- $|R|_N = |R|_{min}$
- Increasing simulation accuracy during the optimization progress

Another approach to one-shot

$$\text{Total Cost} = \sum_{k}^{N} \boxed{I_k} \cdot (T_Q + T_v) + N \cdot (T_{opt} + T_m)$$



- $|R|_N = |R|_{min}$
- Increasing simulation accuracy during the optimization progress

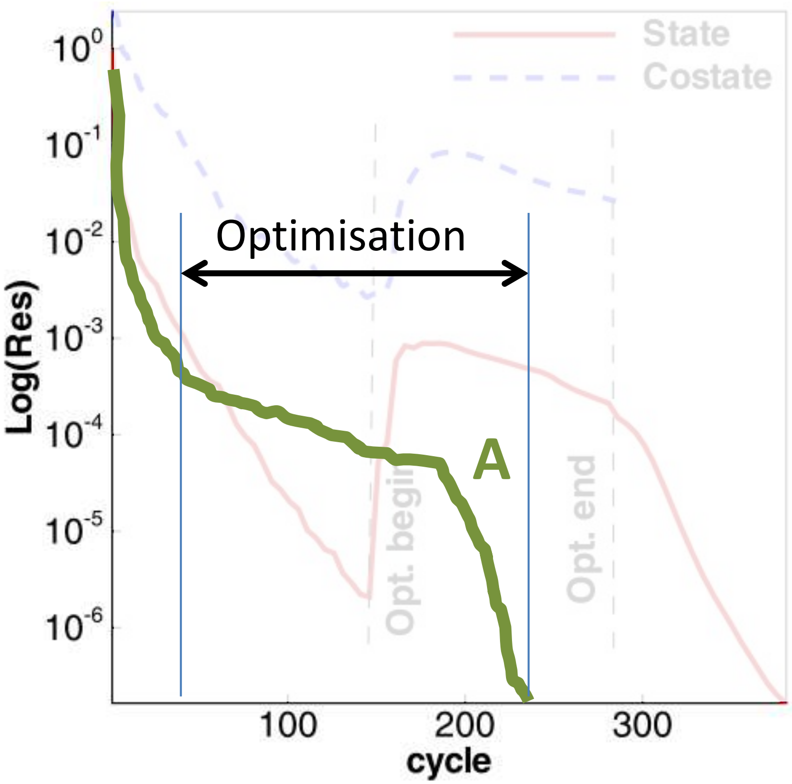
  A: Low accuracy during optimisation = **small I_k**

  B: High accuracy during optimisation = **large I_k**

  C: No convergence

$$\boxed{\text{Total Cost}_A \ll \text{Total Cost}_B}$$

Another approach to one-shot

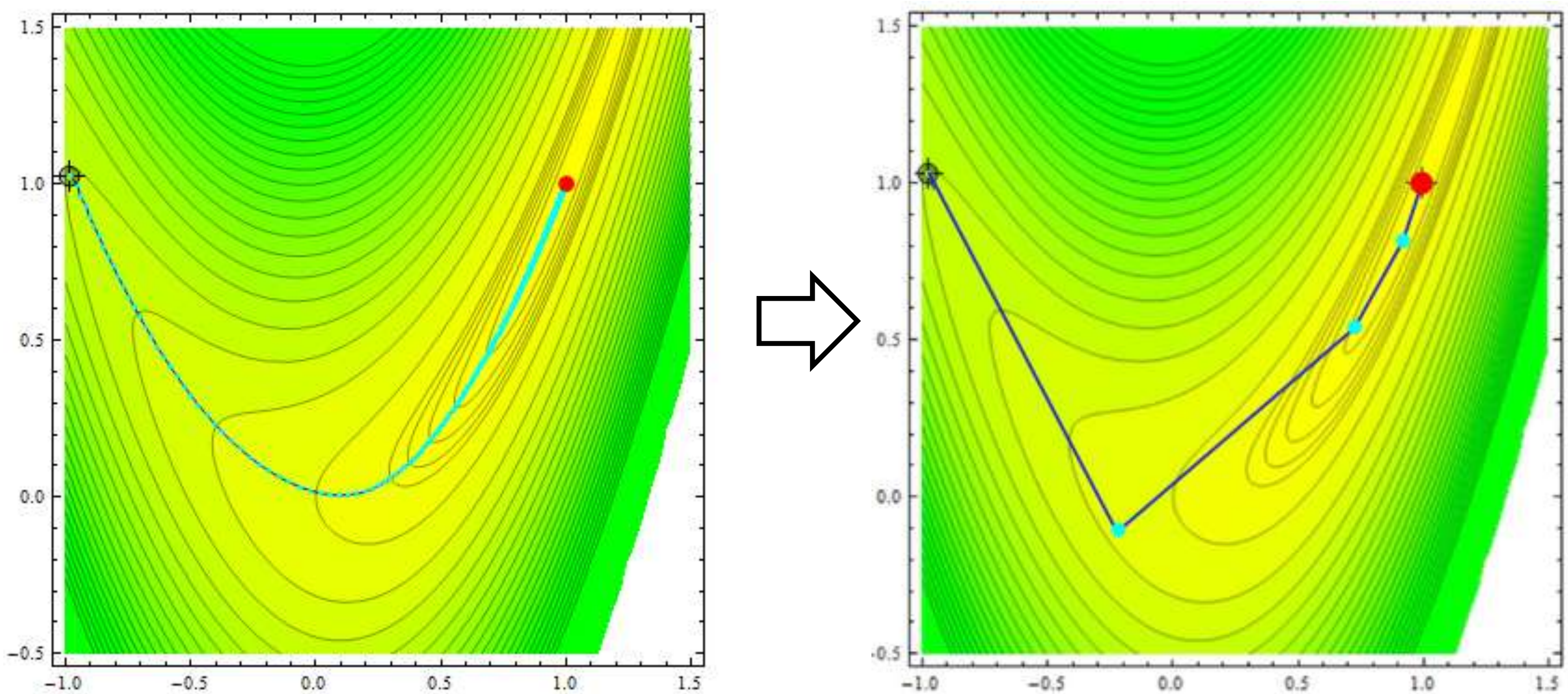


- $|R|_N = |R|_{min}$
- Increasing simulation accuracy during the optimization progress
- Optimisation with lowest possible accuracy

Another approach to one-shot

- During optimisation **|R|=** R_k^{Θ} +/- **Δ**
- **Step size |Δα| determined by the optimizer, and not by the condition STABILITY** ⇒ it is possible to use more complex and efficient optimisation algorithms

„steepest descent” „quasi-Newton”



Another approach to one-shot

- During optimisation **|R|=** R_k^{Θ} +/- **Δ**

- How to determine lowest possible accuracy which will not prevent the optimisation from convergence?

Another approach to one-shot

- The relation between gradient norm and its target value

Accuracy ~ $$\frac{\log||\nabla L_k||}{\log(g_{\min})}$$

- The relation between change in the objective function between search directions compared expected accuracy at convergence

Accuracy ~ $$\frac{\log(\Delta L_m)}{\log(\delta L_N)},$$

- The parameter of desired accuracy of the objective function

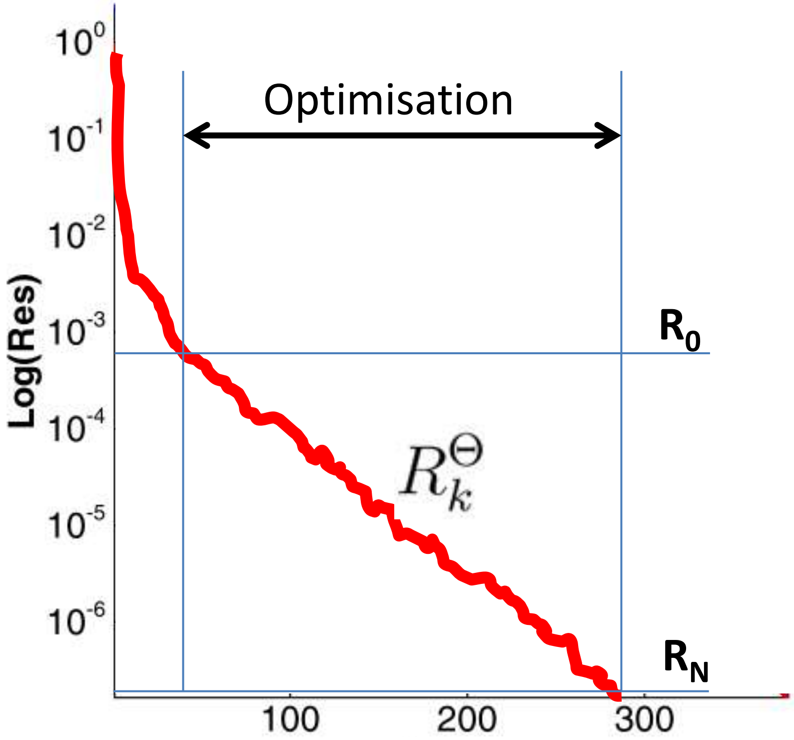
$$\phi_k = \max\left(\frac{\log||\nabla L_k|| - \log||\nabla L_0||}{\log(g_{\min}) - \log||\nabla L_0||}, \frac{\log||\Delta L_m|| - \log||\Delta L_0||}{\log\delta L_N - \log||\Delta L_0||}\right)$$

$$\phi_k \in (0,1)$$

Another approach to one-shot

- Residual during the optimisation
  |R|= R_k^{Θ} +/- **Δ,**

$$R_k^{\Theta} = R_0 \left(\frac{R_N}{R_0} \right)^{\phi}$$



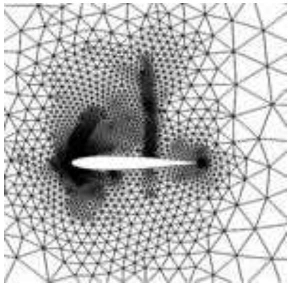
R_0 – residual threshold|R| at the begining

R_N – residual threshold|R| at opt. convegence

$$\phi_k = \max \left(\frac{\log ||\nabla L_k|| - \log ||\nabla L_0||}{\log(g_{\min}) - \log ||\nabla L_0||}, \frac{\log ||\Delta L_m|| - \log ||\Delta L_0||}{\log \delta L_N - \log ||\Delta L_0||} \right)$$

One-shot + mesh adaptation

+



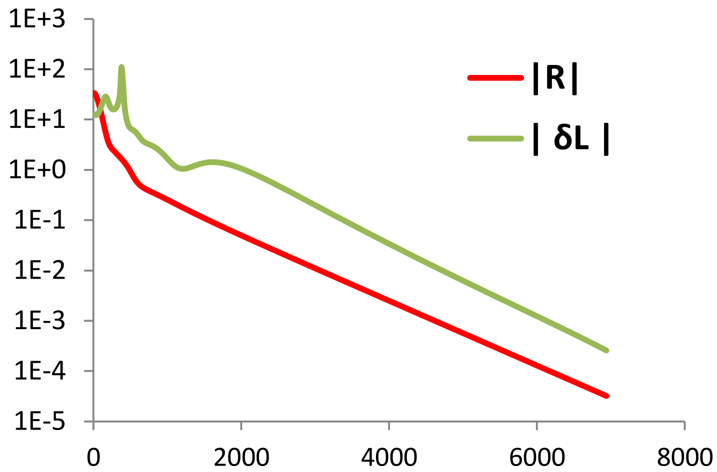
Error sources

- **Error due to incomplete convergence**

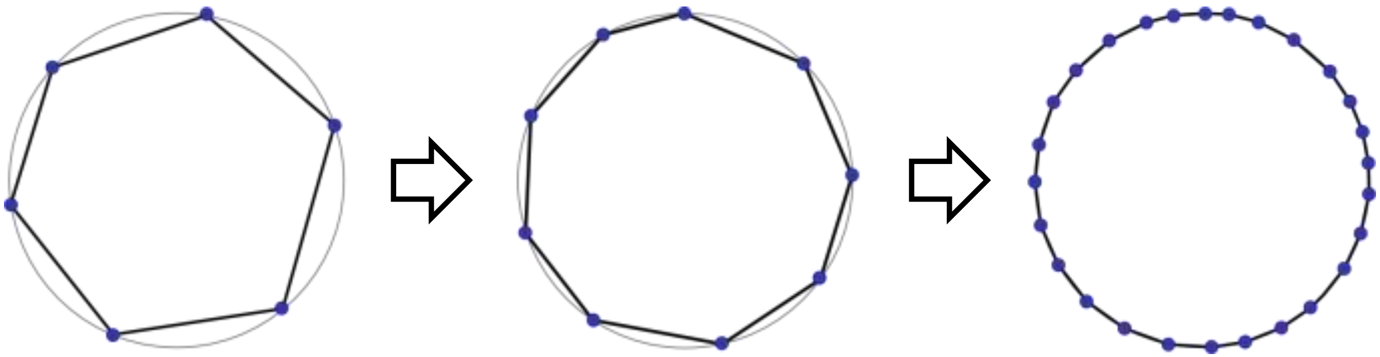
$$\delta L \approx \frac{\partial L}{\partial Q} \delta Q$$

$$\delta Q \approx A^{-1} R$$

$$||\delta Q|| \lesssim ||A^{-1}|| \; ||R||$$



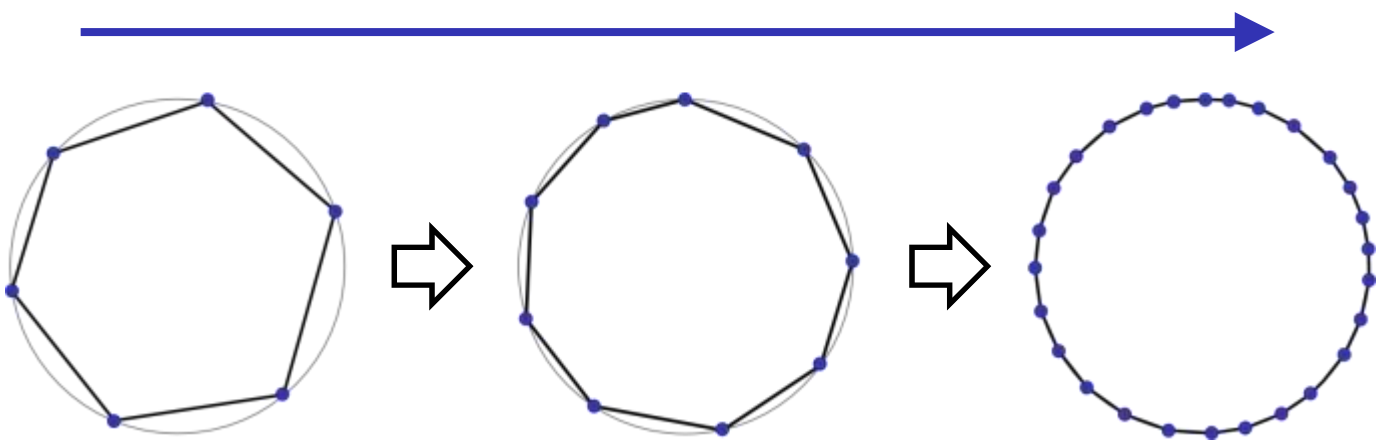
- **Discretisation error**



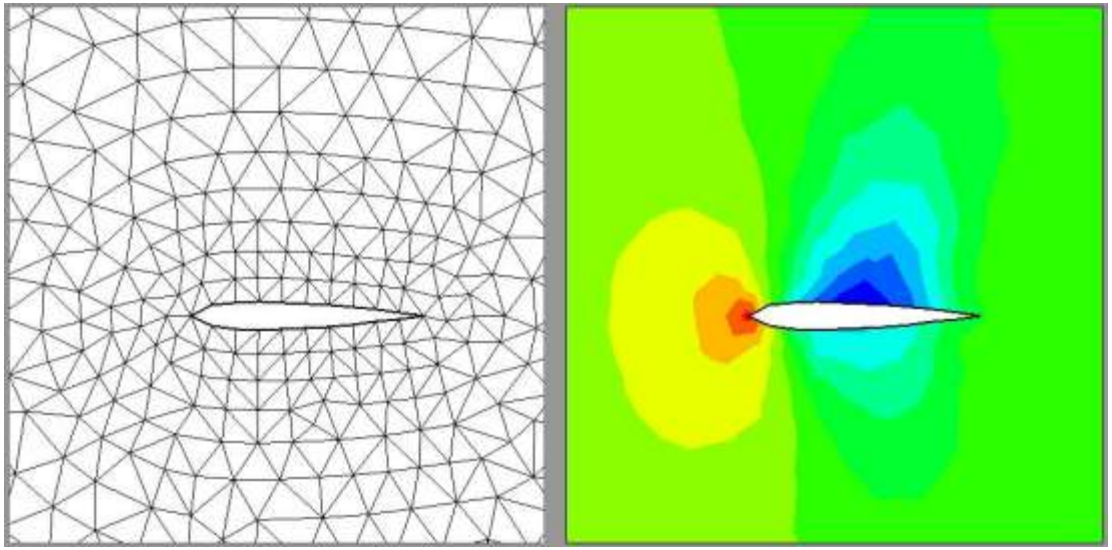
Discretisation – optimisation cost

$$\text{Total Cost} = \sum_{k}^{N} \boxed{I_k} \cdot \boxed{(T_Q + T_v)} + N \cdot (T_{opt} + T_m)$$

Increase in $T_Q + T_v$

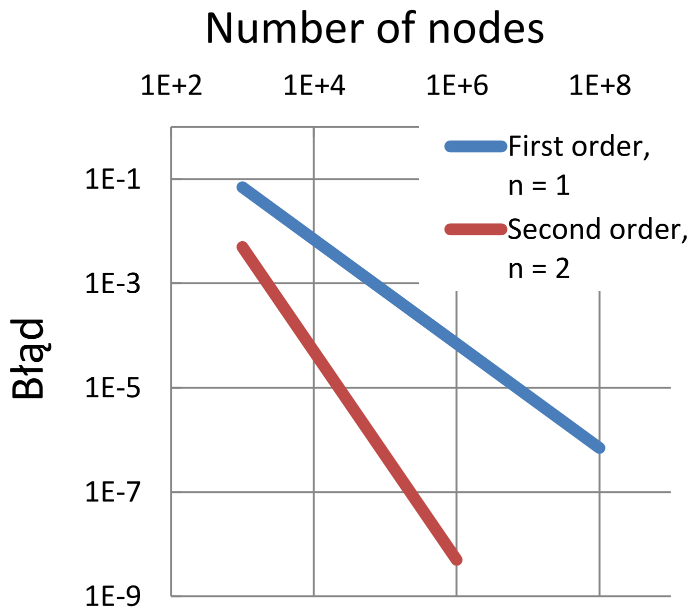


Discretisation error



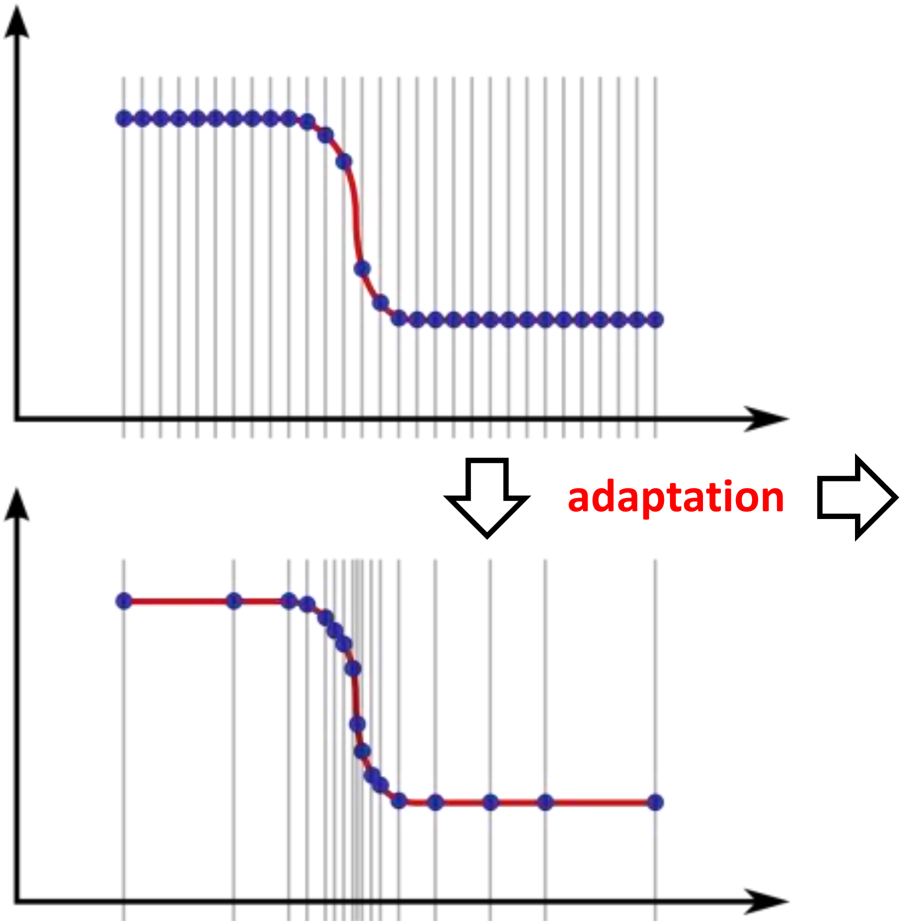
- Uniform grid
- Discretisation error:

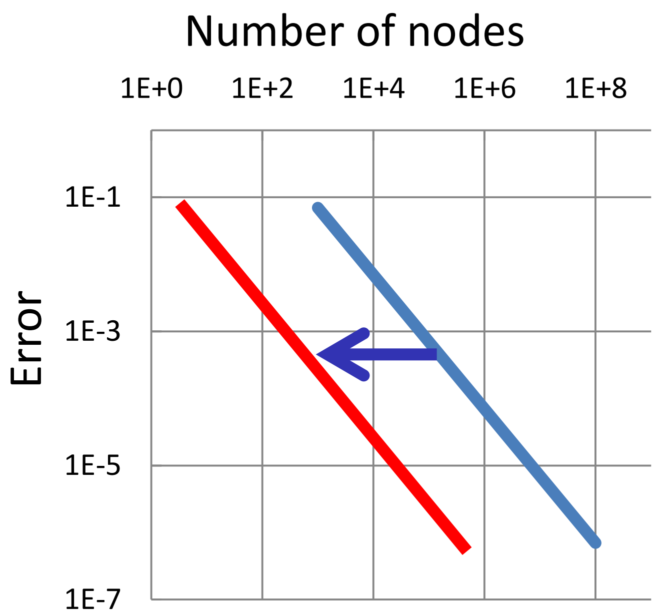
$$E(h) = Ch^n$$



Mesh adaptation

- Looking for optimal distribution of mesh nodes

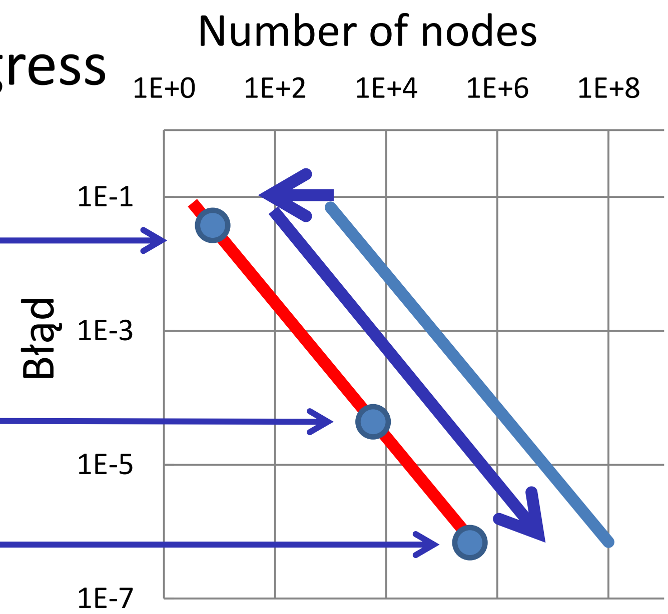




- Similar error with lower number of DOFs
- Lower comp. cost

One-shot + adaptation

- Adjusting the discretization error according to the optimization progress



δL_0^h at the begining of opt.

δL_k^Θ during the optimisation

δL_N^h at the opt. convergence

where:

$$\delta L_k^\Theta = \delta L_0^h \left(\frac{\delta L_N^h}{\delta L_0^h} \right)^{\phi}$$

ϕ set by one-shot accuracy parameter

Error estimation

- For the 2nd order method interpolation error is proportional to the second derivative of the solution (Hessian) $\mathcal{H}_{i,l}$

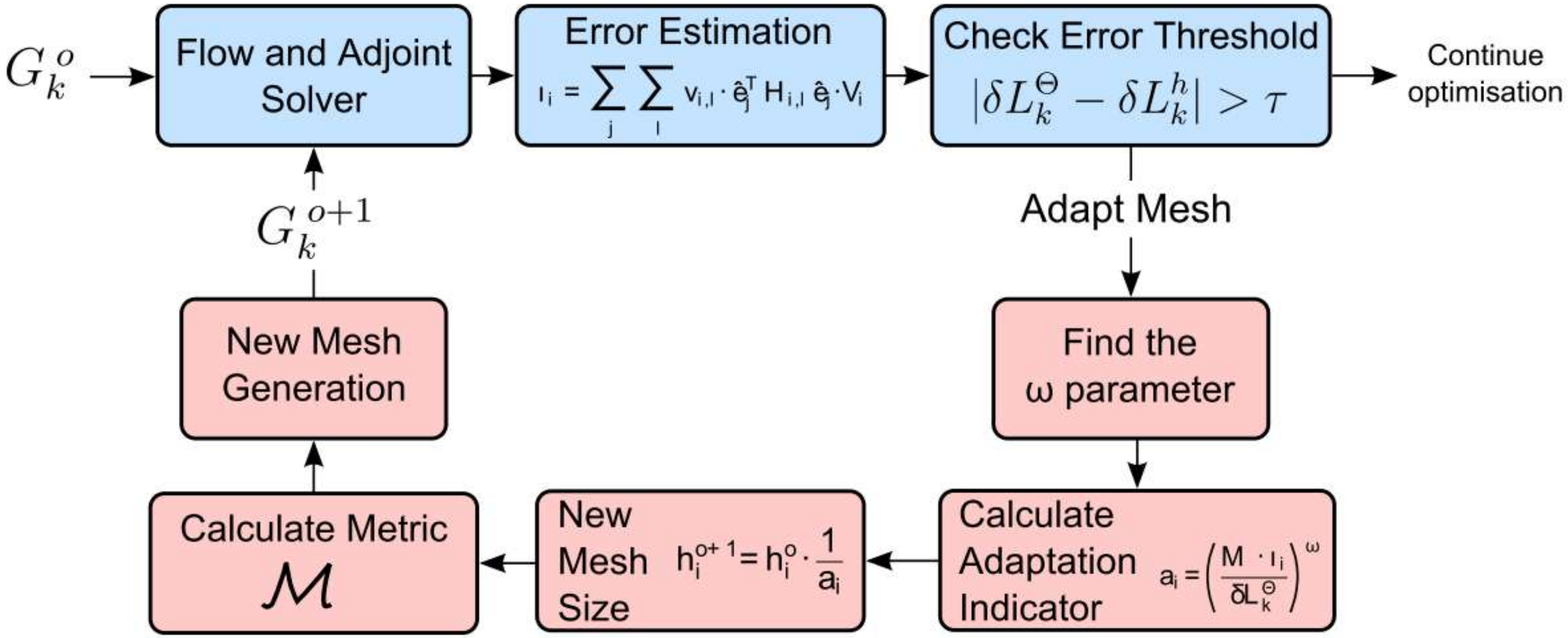
$$|f(x) - f_h(x)| \leq \frac{h^2}{8} \boxed{|f''(x)|} + \mathcal{O}(h^3)$$

- Adjoint variable may show the impact of the local error on the objective function (adaptation indicator)

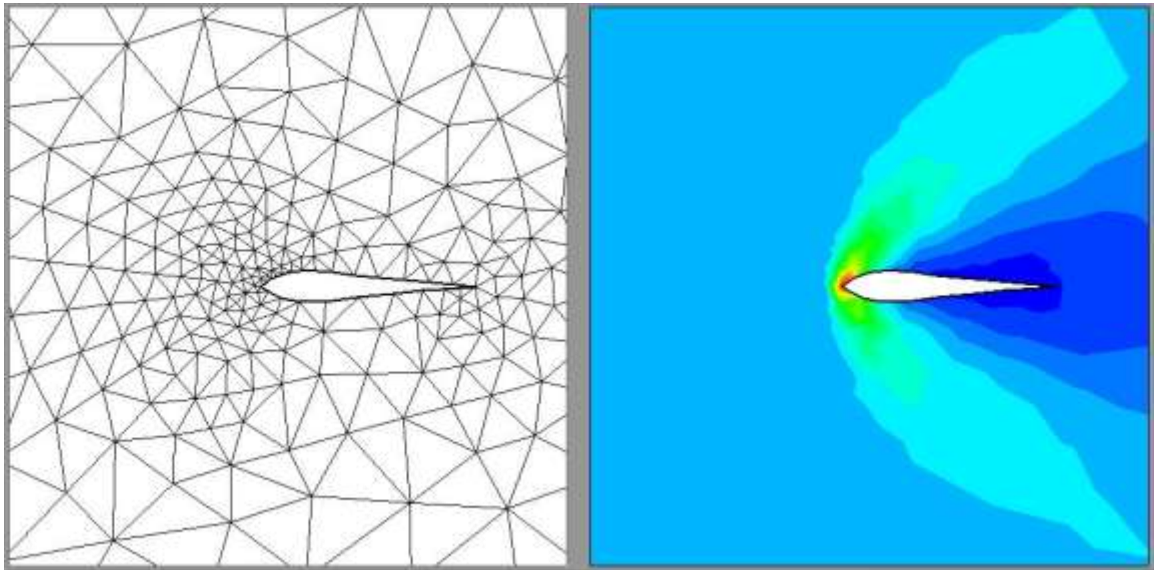
$$\delta L_k^h = \sum_i \sum_j \sum_l \boxed{v_{i,l}} \cdot \left| \hat{e}_j^T \, \mathcal{H}_{i,l} \, \hat{e}_j \right| \cdot V_i$$

Mesh adaptation process

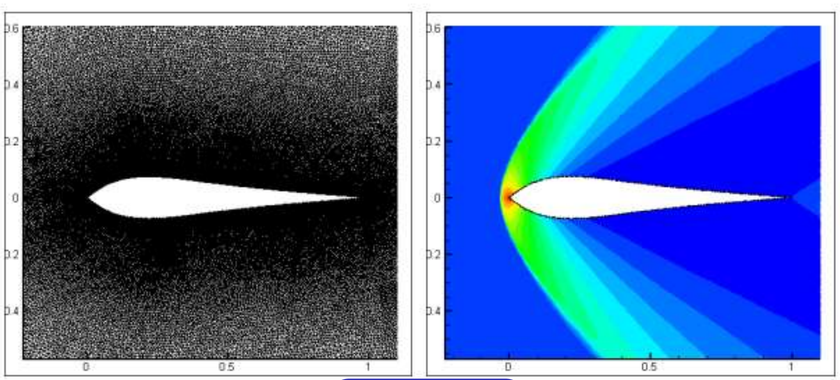
- Another loop nested within the optimisation



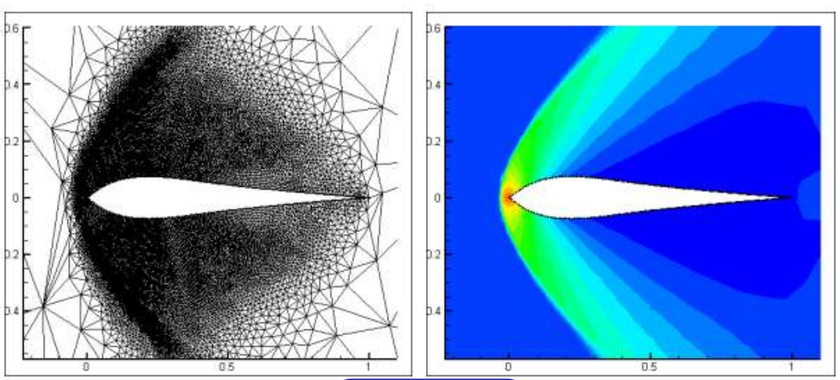
Example of adjoint-based adaptation



Uniform grid



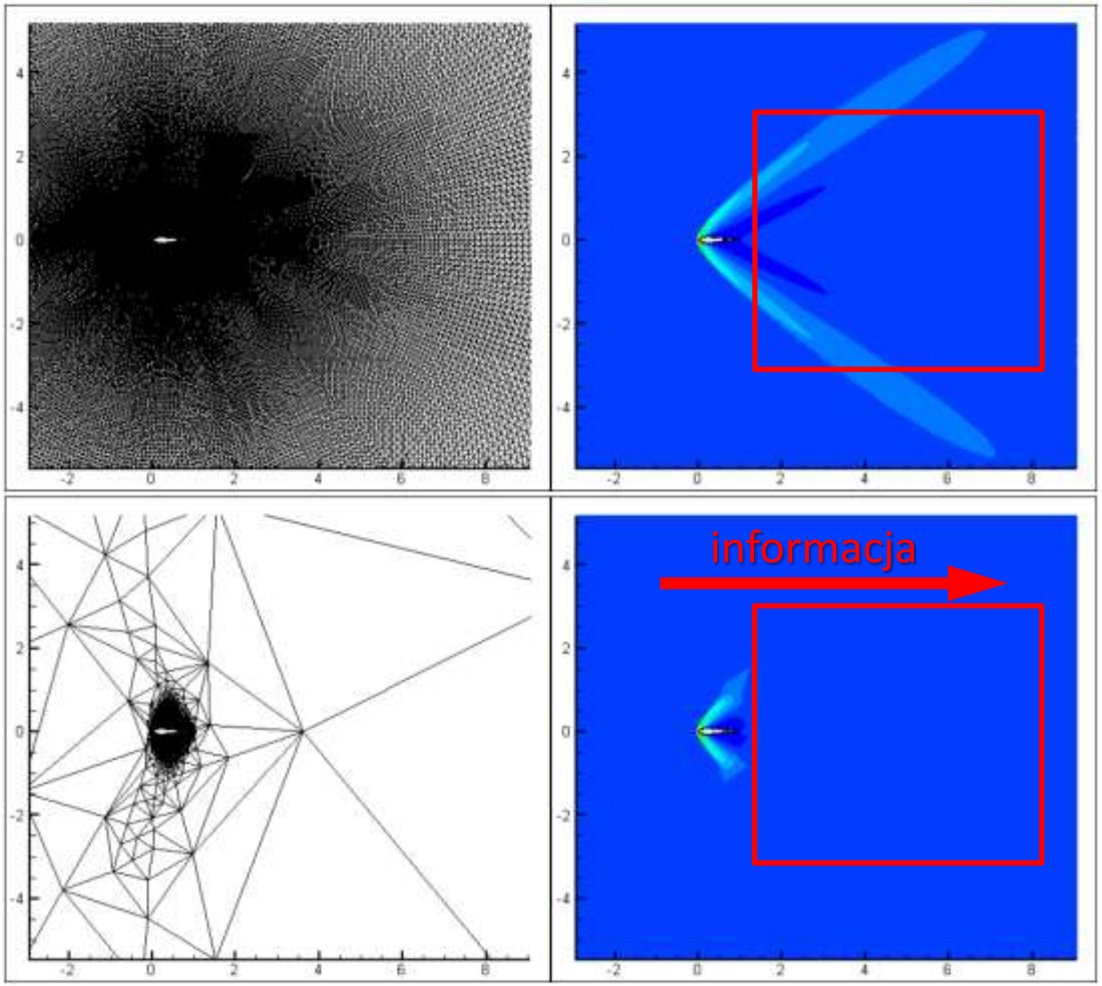
Adaptation



142 000 → 19 000

Example of adjoint-based adaptation

- Mesh is refined only in regions important for estimation of the objective function



Region behind the airfoil
⇒ no influence on the objective function based on lift and drag

One-shot + adaptation

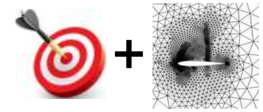
- Typical optimisation

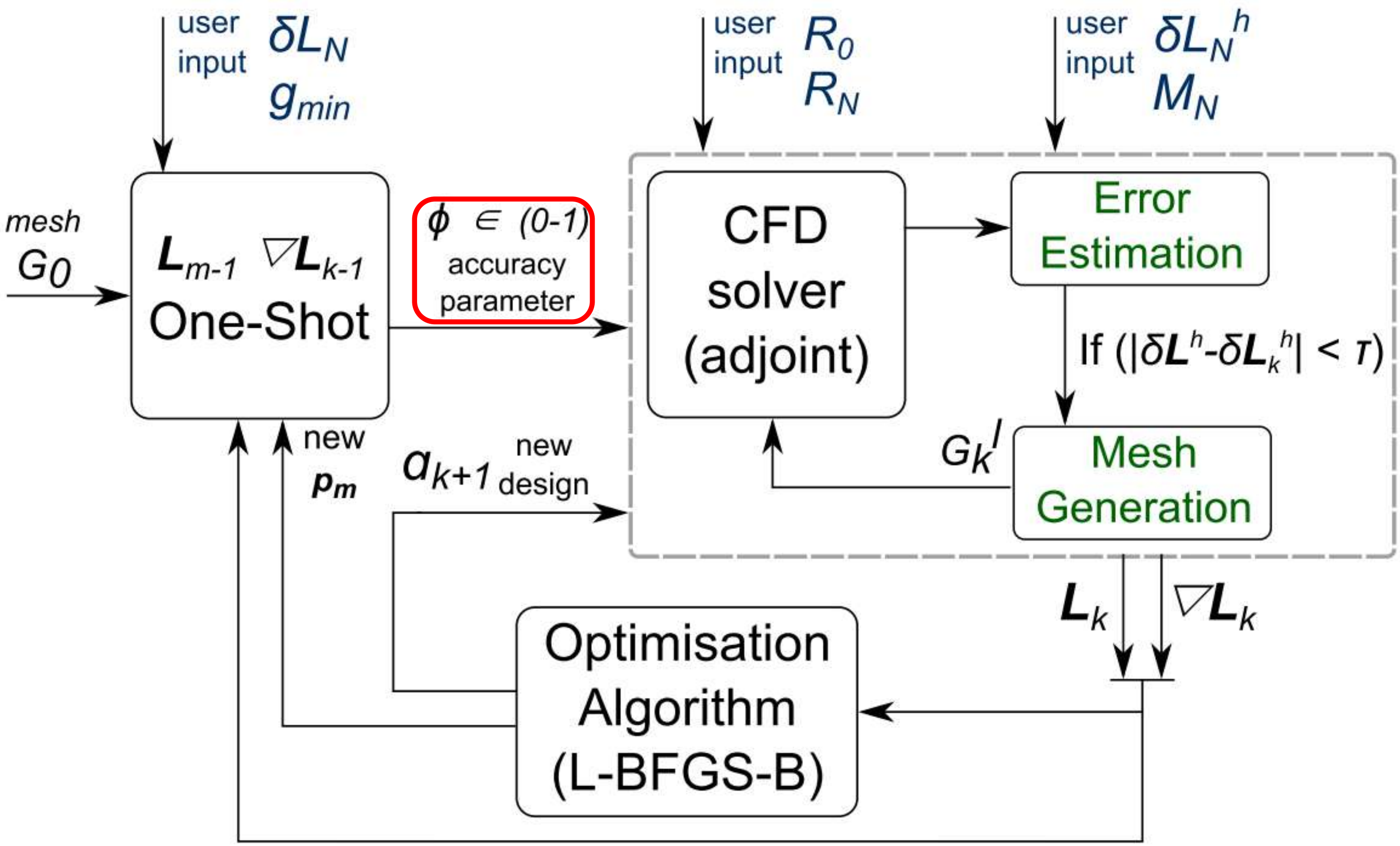
1.	mesh $G_k \rightarrow$	CFD solver	$\rightarrow$ the flow Q_k, L_k
2.	$G_k, Q_k \rightarrow$	Adjoint solver	$\rightarrow$ the adjoint v_k, ∇L_k
3.	$\alpha_k, L_k, \nabla L_k \rightarrow$	Optimisation algorithm	$\rightarrow$ the new design α_{k+1}
4.	$\alpha_{k+1}, G_k \rightarrow$	Mesh Morphing	$\rightarrow$ the new mesh G_{k+1}

- One-shot + adaptation ⇩

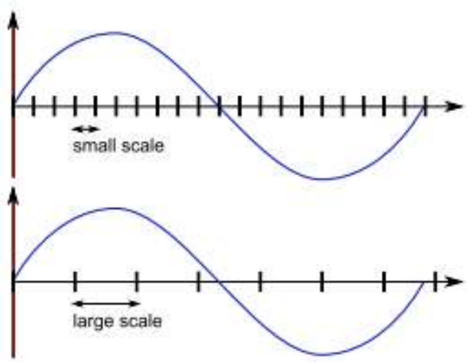
1.	mesh $G_k^o \rightarrow$	CFD solver	$\rightarrow$ flow Q_k, L_k
2.	$G_k, Q_k \rightarrow$	Adjoint solver	$\rightarrow$ adjoint v_k, ∇L_k
2a.	$Q_k, v_k \rightarrow$	If $(R_k > R_k^{\Theta})$ then	$\rightarrow$ **goto 1**
2b.	$Q_k, v_k \rightarrow$	If $(\delta L_k^h > \delta L_k^{\Theta})$ then	$\rightarrow$ **new** G_k^{o+1}, **goto 1**
3.	$\alpha_k, L_k, \nabla L_k \rightarrow$	Optimisation algorithm	$\rightarrow$ new design α_{k+1}
4.	$\alpha_{k+1}, G_k \rightarrow$	Mesh Morphing	$\rightarrow$ new mesh G_{k+1}

One-shot + adaptation

- Discretisation error defined by the one-shot approach 
- It is possible to use any type of optimisation algorithm

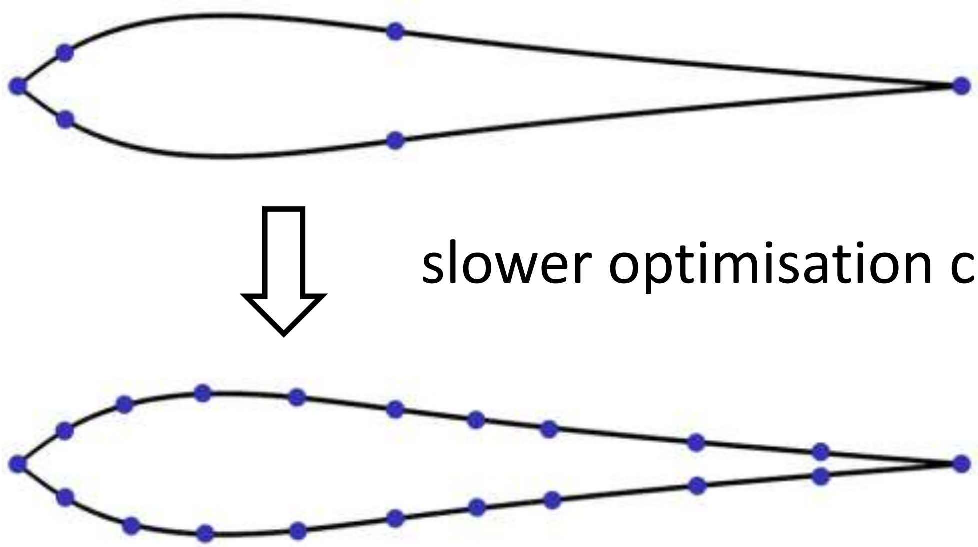


Multigrid in optimisation



Multigrid in optimisation

- Optimized for larger number of parameters requires more f & g evaluations



Dim $\vec{\boldsymbol{\alpha}}$ = 6

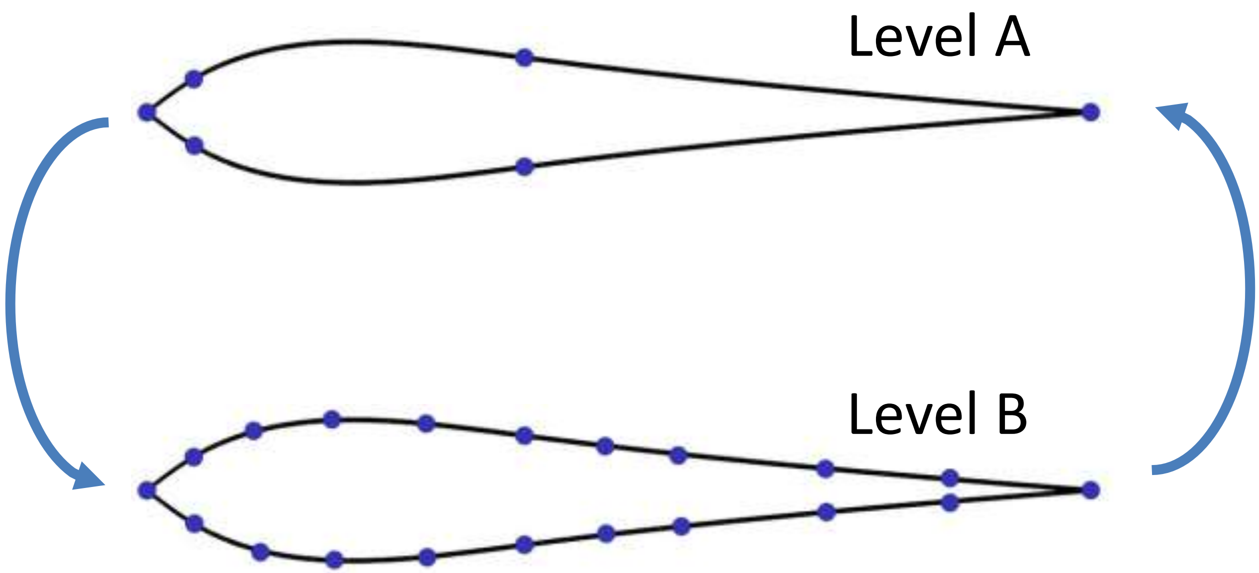
slower optimisation conv. ⇒ higher N

Dim $\vec{\boldsymbol{\alpha}}$ = 20

$$\text{Total Cost} = \sum_{k}^{\boxed{N}} I_k \cdot (T_Q + T_v) + N \cdot (T_{opt} +$$

Multigrid in optimisation

- Optimized for larger number of parameters requires more f & g evaluations

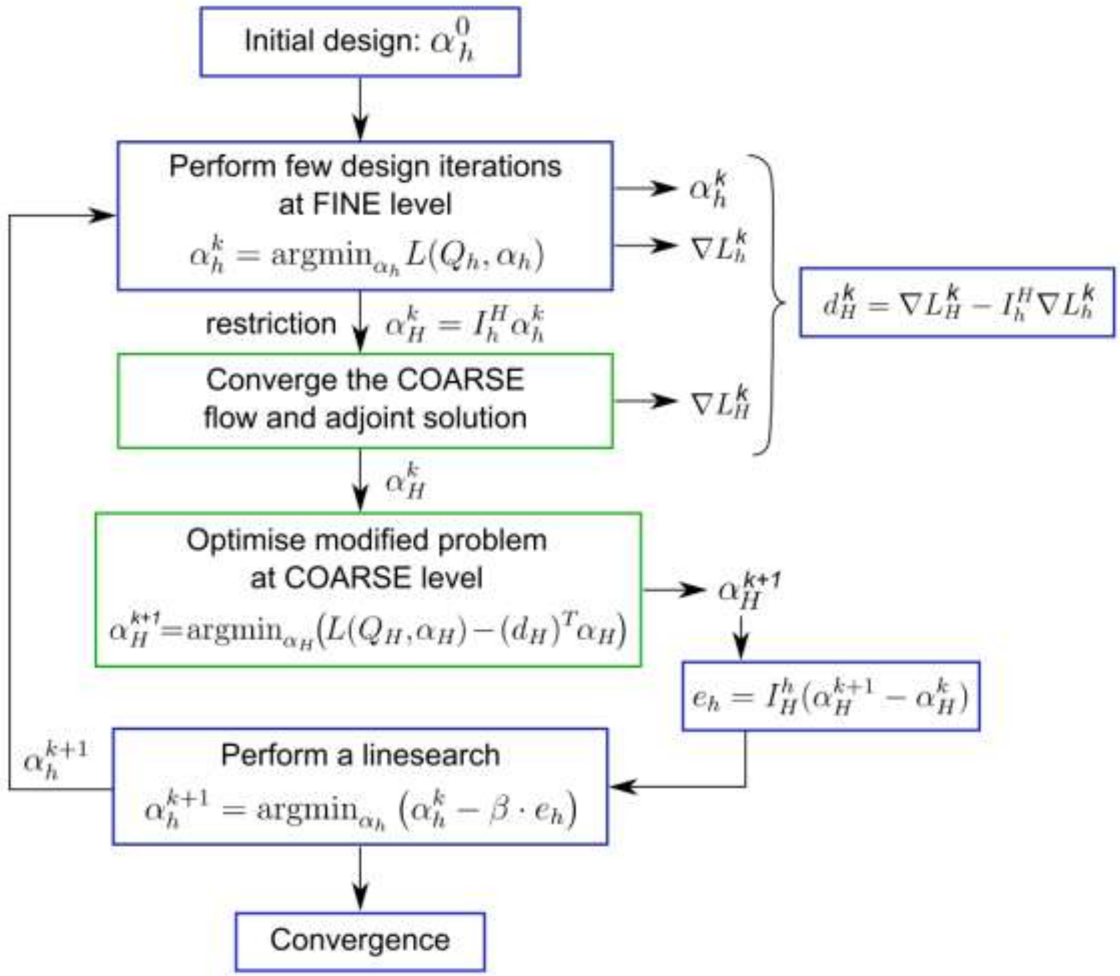


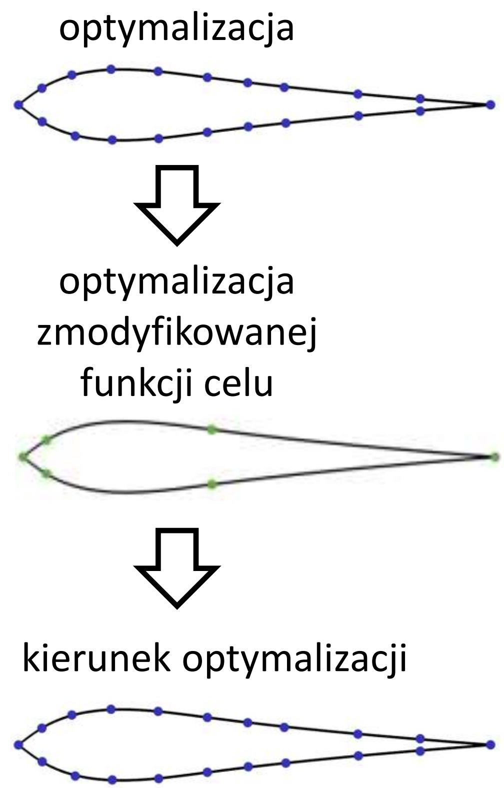
You can use information from level A to accelerate the optimisation at the level B
⇒ MULTIGRID in optimisation

- Aim: obtain opt. convergence independent on the number of parameters

Multigrid in optimisation

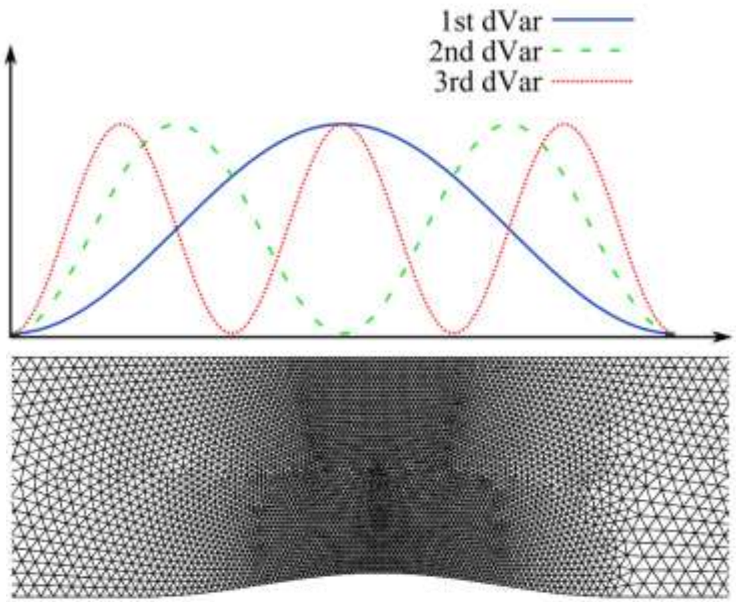
- The algorithm proposed by Lewis and Nash

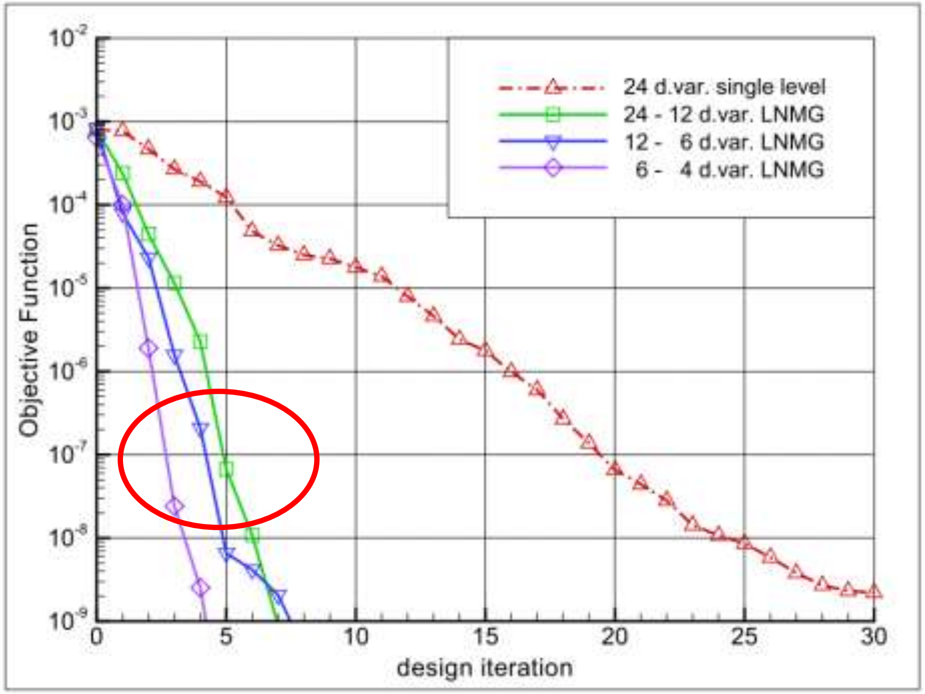




Multigrid in optimisation

- The key aspect is to choose an appropriate parameterisation

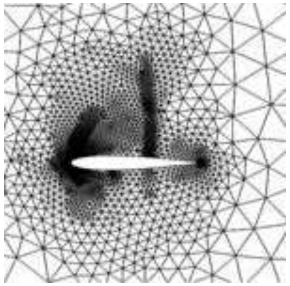




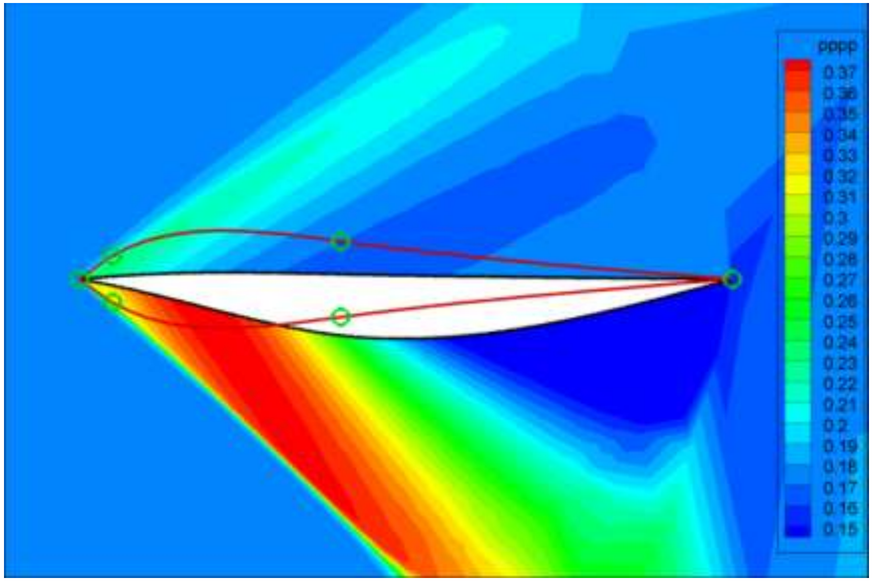
- The coincidence of the number of independent parameters
- Positive result only for Fourier parameterisation - difficult to use in realistic cases

Numerical examples

+



- Optimization target: min drag D for a given lift Z_t, the objective function: $L = D + \sigma \cdot |Z - Z_t|$
- **M = 2.0**, **4 design parameters**



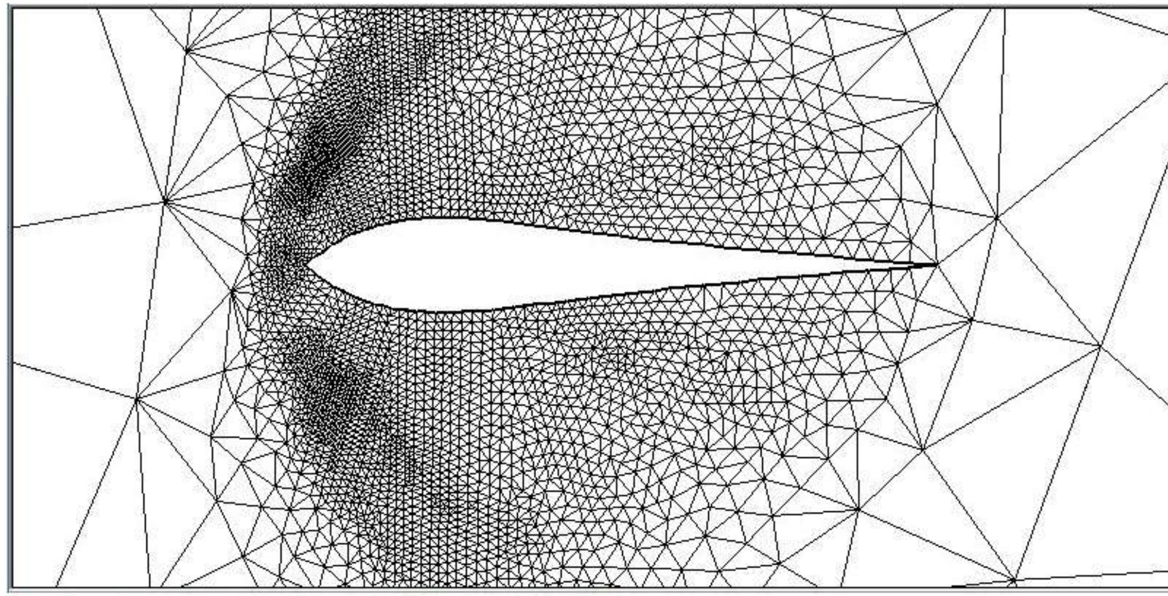
δL_N	g_{min}	R_0	R_N	δL_N^h	M_0	M_N
10^{-4}	0.05	10^{-7}	10^{-10}	$0.01(abs.)$	600	10 000

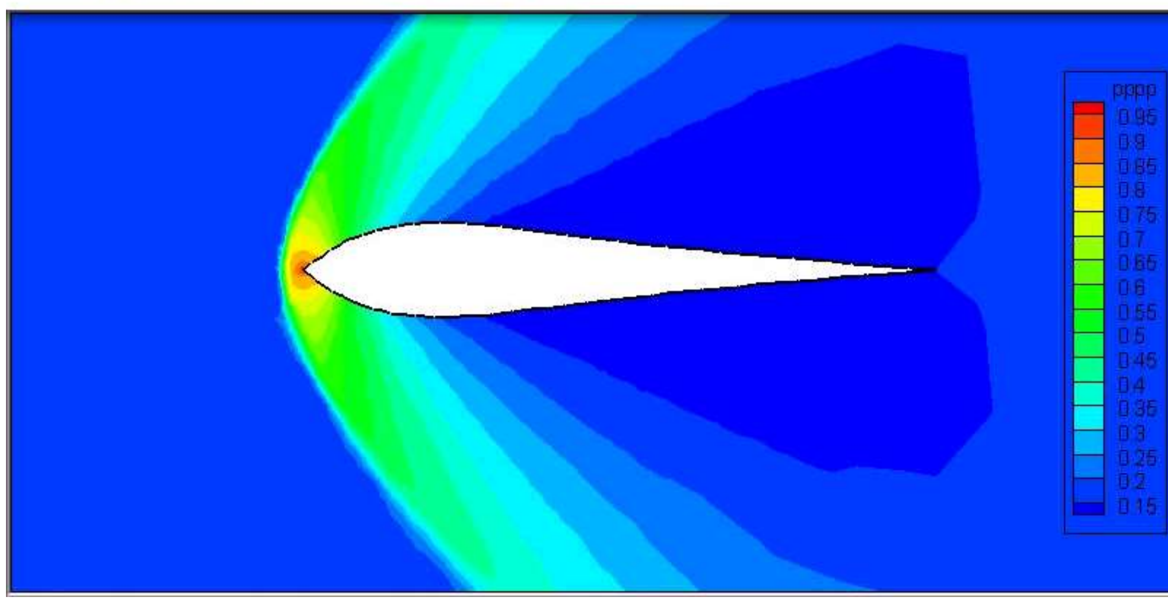
1. Wave-rider: optimisation + adaptation

- min drag D for a target lift Z_t

$$L = D + \sigma \cdot |Z - Z_t|$$

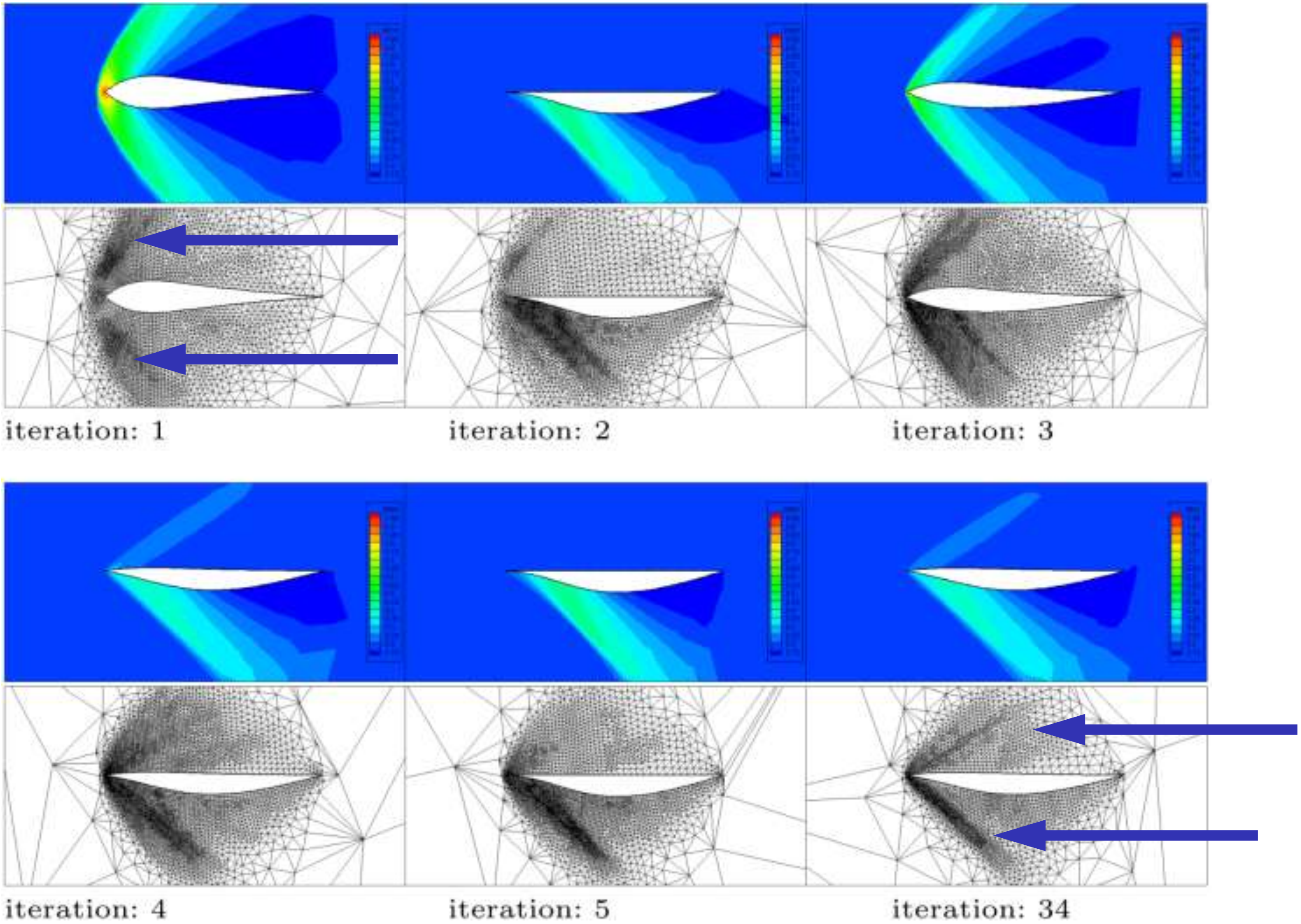
- Supersonic flow
  M = 2.0



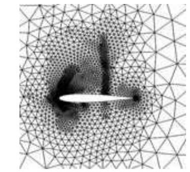


1. Wave-rider: optimisation + adaptation

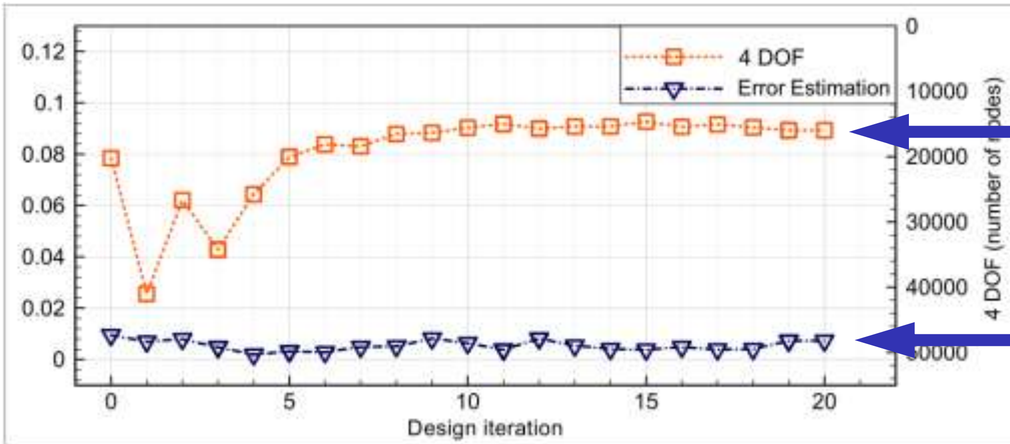
- In each optimisation step the optimum discretisation is different



1. Wave-rider

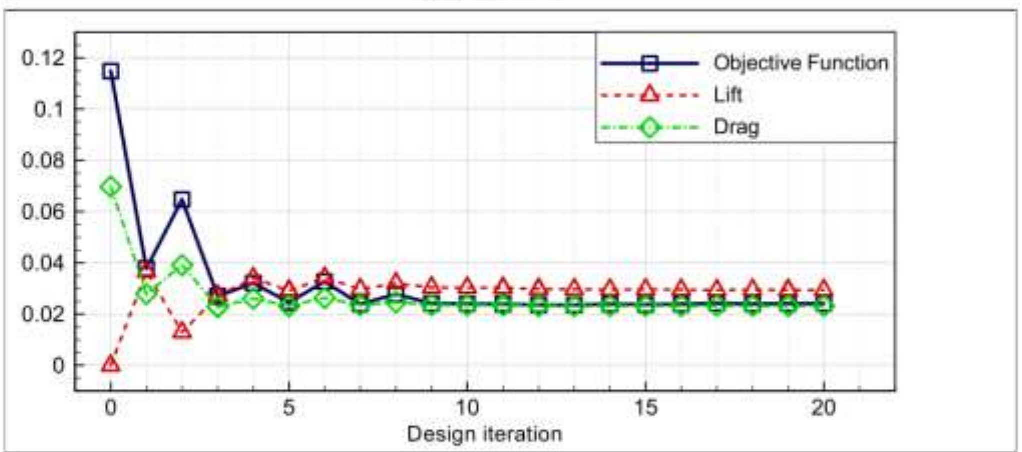


- Optimisation with adaptation (Φ = 1)
- Constant relative error during optimisation



The number of DOFs is increasing

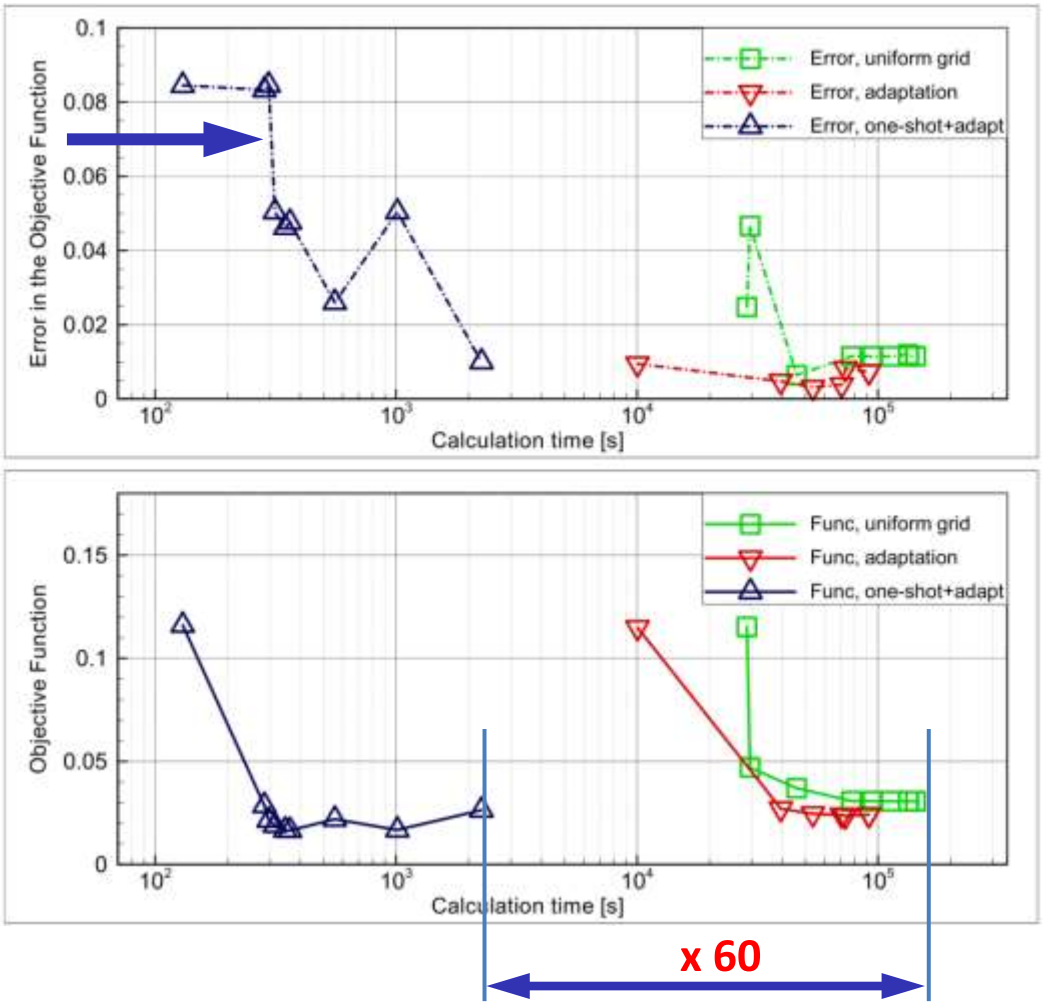
Constant error

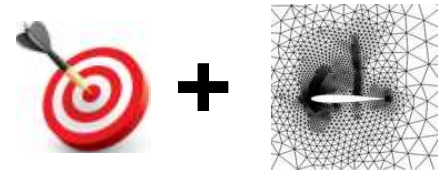


1. Wave-rider

- **speedup of** 60 is reached for **one-shot with adaptation**

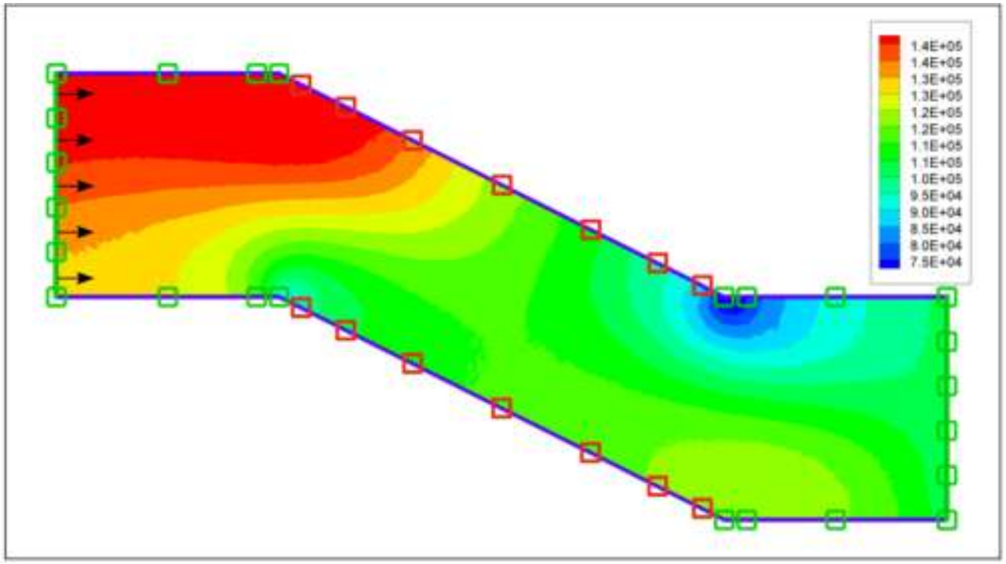
One-shot + adaptation error is decreasing during optimisation





2D sbend: one-shot + adaptation

- Laminar flow, Re = 300, ANSYS Fluent v14 adjoint solver
- Optimisation task - minimize pressure drop
- 14 design variables, sequence of meshes

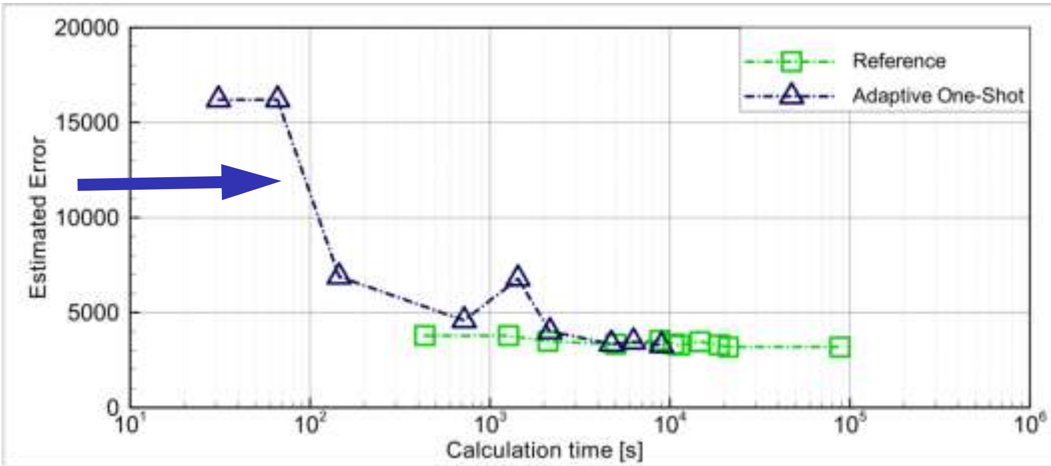


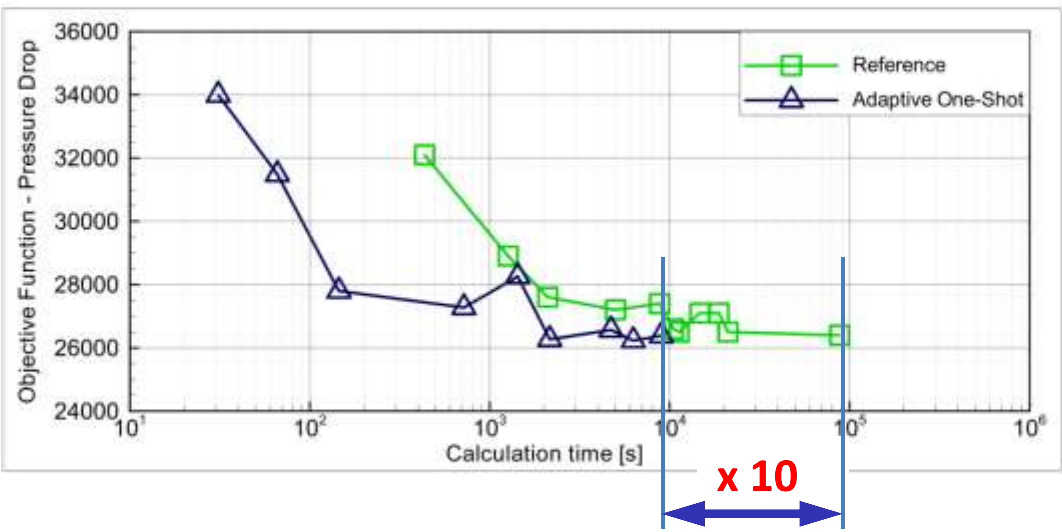
δL_N	g_{min}	R_0	R_N	δL_N^h	M_0	M_N
100	4500	10^{-10}	10^{-10}	$20\%(rel.)$	1 000	15 000

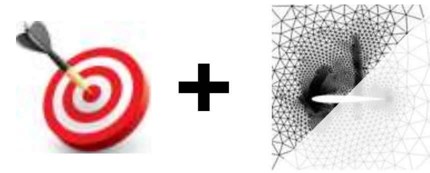
2D sbend: one-shot + adaptation

- **Speedup of 10 is reached**

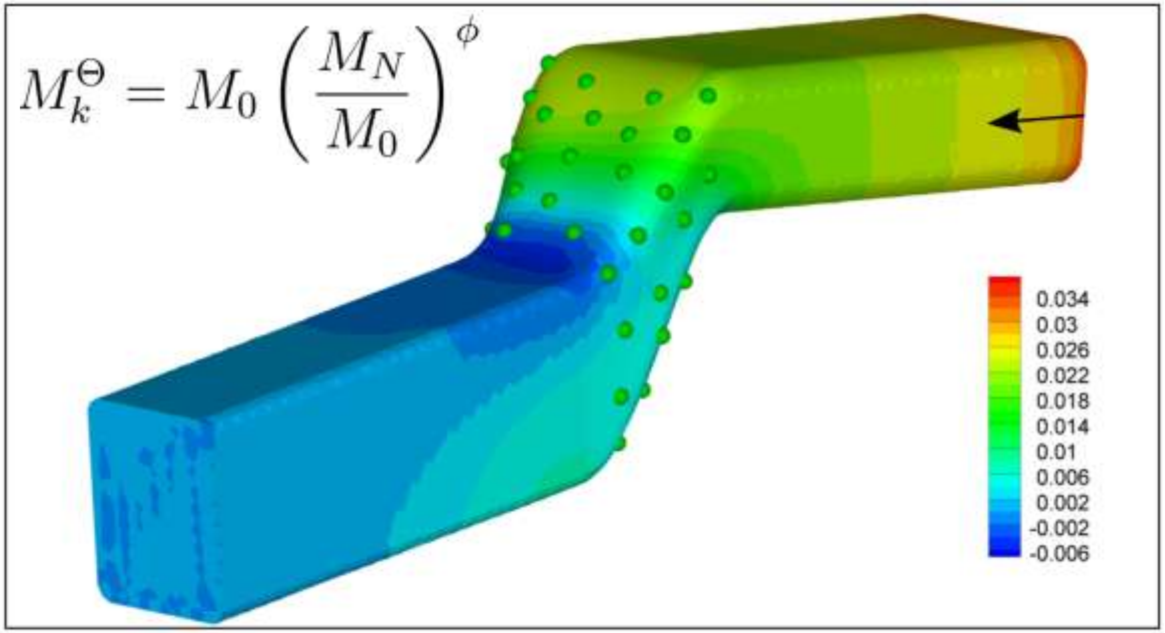
One-shot + adaptation error is decreasing during optimisation







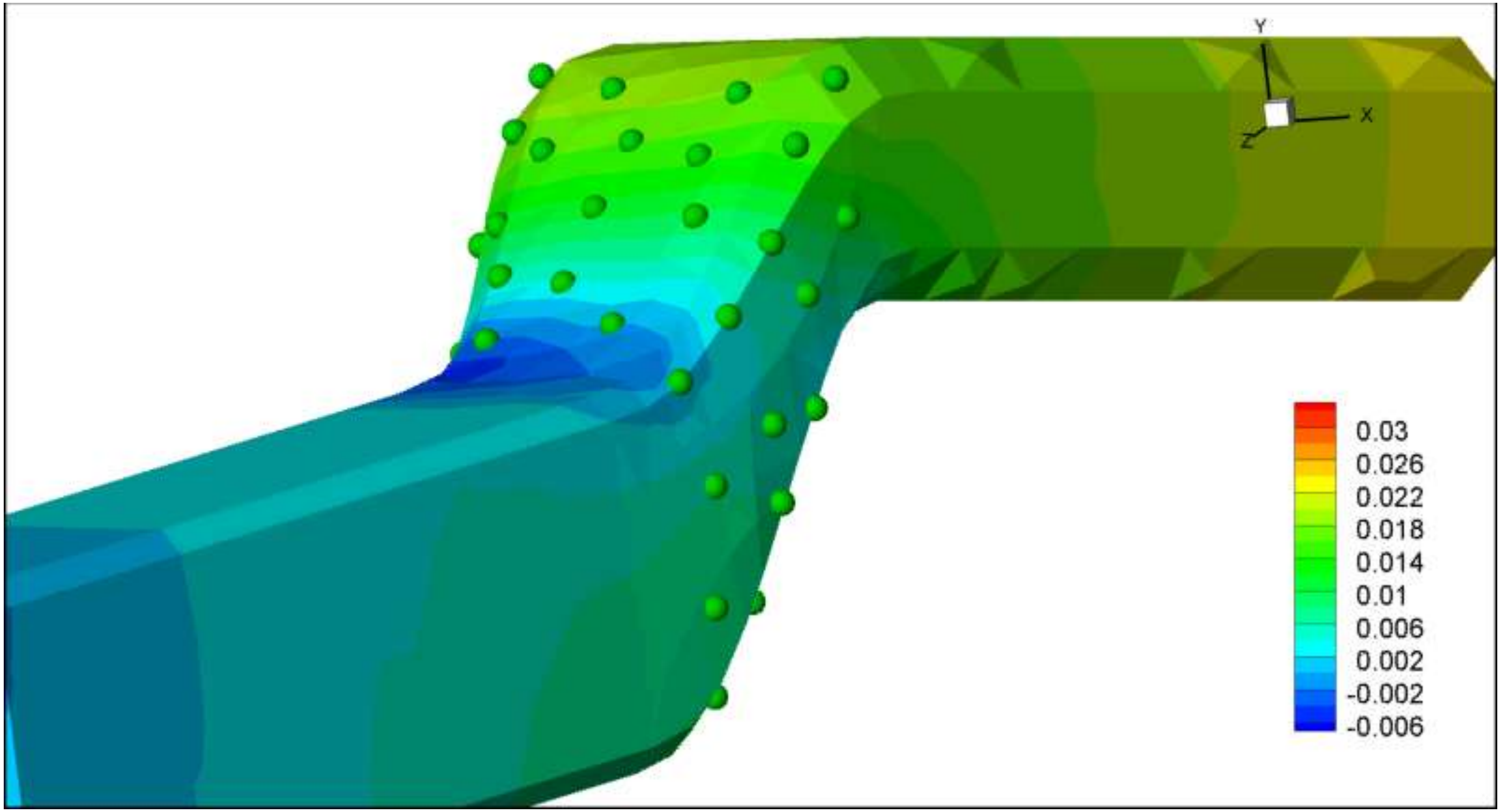
- Laminar flow, Re = 300, ANSYS Fluent v14 adjoint solver
- Optimisation task - minimize pressure drop
- 150 design variables
- Sequence of meshes 1 913 ⇒ 99 796 nodes



δL_N	g_{min}	R_0	R_N	δL_N^h	M_0	M_N
$5 \cdot 10^{-3}$	$2 \cdot 10^{-2}$	10^{-4}	10^{-10}	–	1 900	100 000

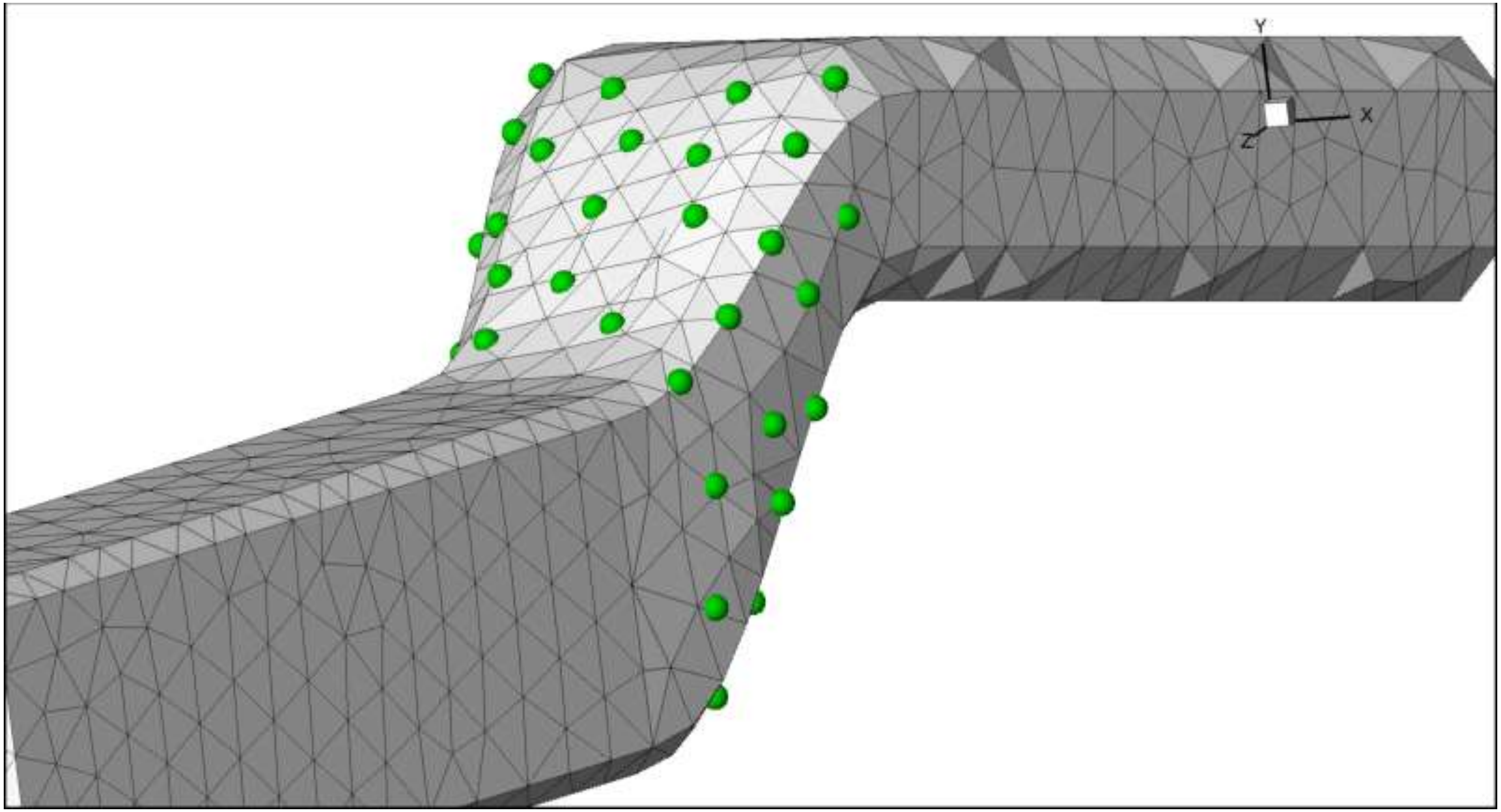
3D sbend: one-shot + adaptation

- Optimisation



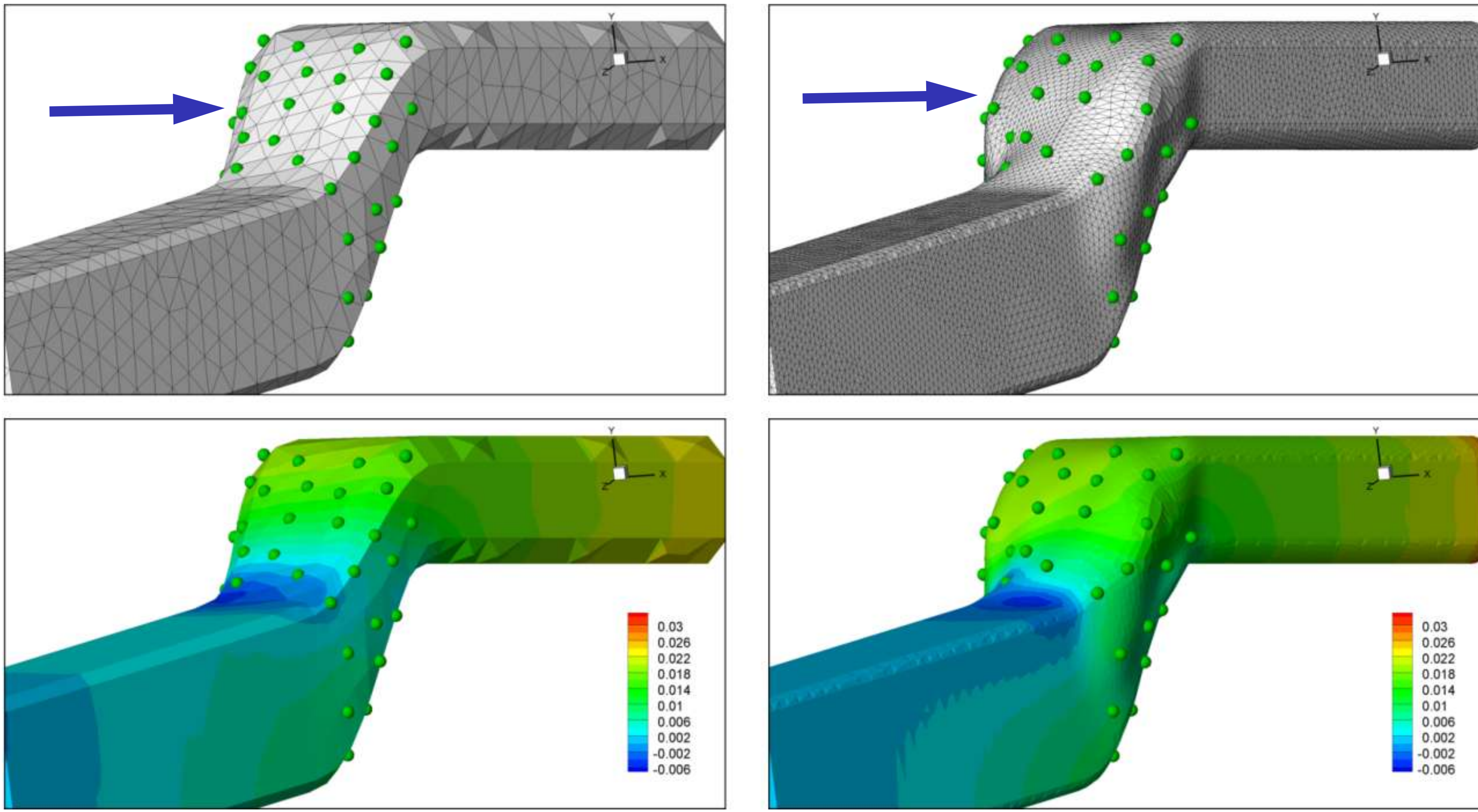
3D sbend: one-shot + adaptation

- **Change in the mesh density during optimisation**



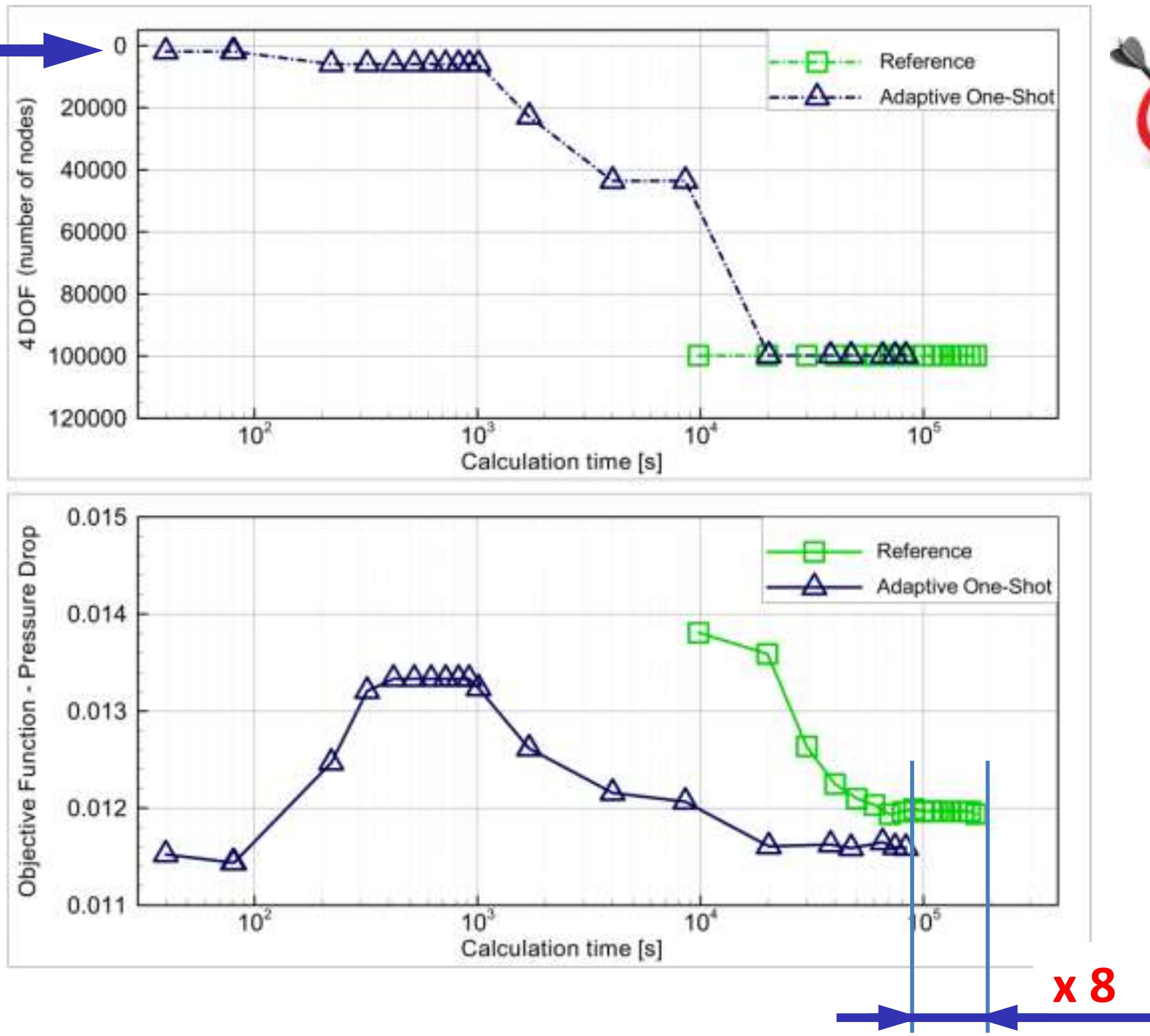
3D sbend: one-shot + adaptation

- **Change in the mesh density during optimisation**

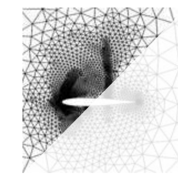


- **Speedup of 8 is reached**

One-shot + adaptation error is decreasing during optimisation

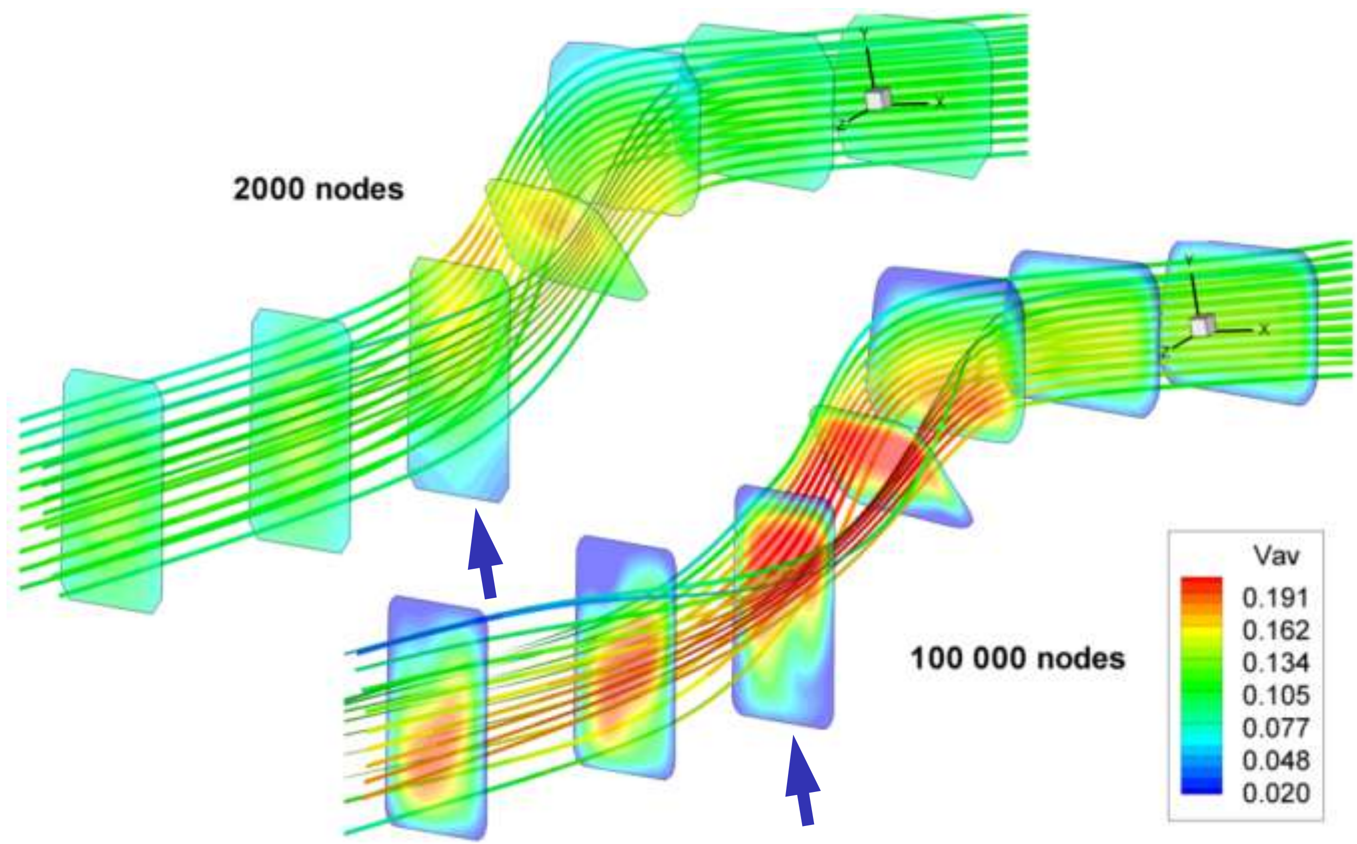


+



3D sbend: one-shot + adaptation

- The impact of simulation accuracy – initial geometry
- Coarse discretisation is not resolving important flow features



Summary

No.	Test-case	One-shot	Adaptation Approach	Speed-up
1.	2D Wave-rider	δL_k^R and δL_k^h	Adjoint based adaptation	60
2.	2D s-bend	only δL_k^h	Uniform adaptation	10
3.	3D s-bend	δL_k^R and M_k	Sequence of meshes	8

- One-shot method in combination with the adaptation achieves from 8 to 60 times faster optimization

This work was supported by EU FP7 FlowHead Project
(Fluid Optimisation Workflows for Highly Effective Automotive Development Processes, SCP7-GA-2008-218626)

Thank you!