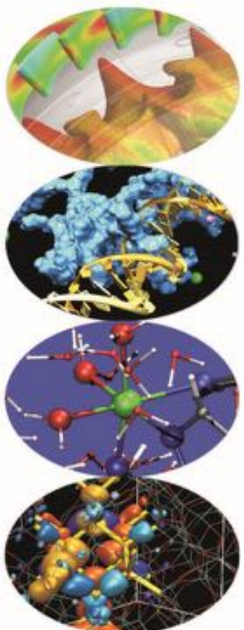


FFD, mesh morphing & reduced order models: Enablers for efficient aerodynamic shape optimization

F Salmoiraghi, G Rozza *SISSA mathlab*

A Scardigli, **H Telib** *OPTIMAD engineering srl*

Outline

1. **Practical problems in shape optimization**
2. **Enabling of large scale aerodynamic shape optimization**
   - **Shape and mesh morphing**
   - **Efficient sampling strategies**
   - **ROMs based on POD**
3. **Questions & hopefully answers**

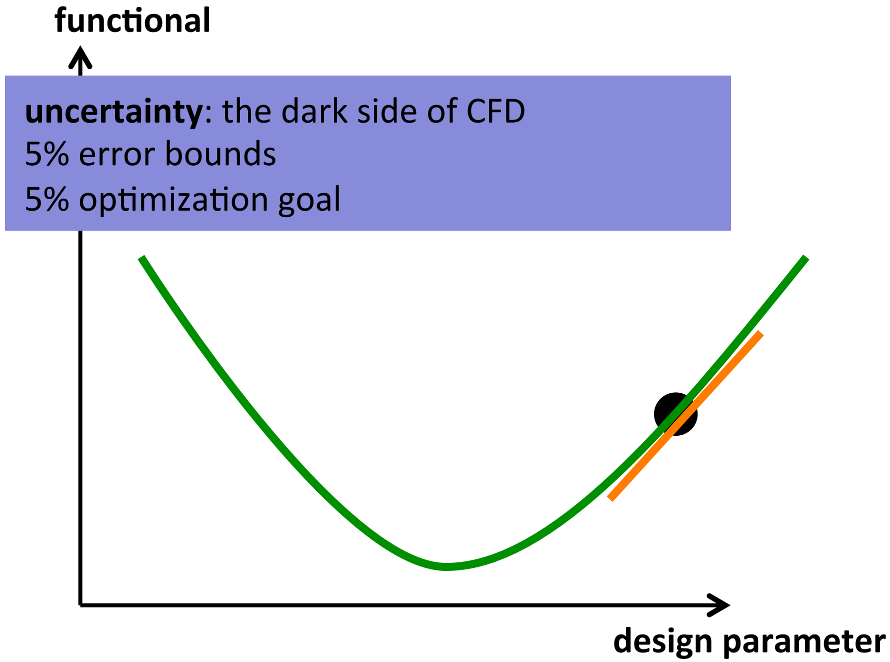
Motivation

1. **Difficult to set-up (Integration)**
   - identification of parameters, parameterization itself etc
   - totally automatized (geometry creation, pre-processing)
   - especially critical if at advanced design

2. **Expensive (Availability)**
   - computing resources sized for analysis
   - licenses CAD, CFD

Information provided by solver

- **Evaluate functional (Comparative)**
  - –Industry standard
- **Evaluate gradient (convergence, direction)**
  - –Advanced capabilities
- **Evaluate Hessian (step, local topology)**
  - –Cutting edge



Time & Costs

• cost of real life RANS approx.	**C_{HFM} =2000 cpuh**
• # of design variables	**O(10)**
• cost of computing	**0.1€/cpuh**
• cost of licenses	**0**

2 Level multi-fidelity approach using response surface (neglectable cost) + HFM

• global optimization run O(100) - O(1000)	2.e5 – 2.e6 cpuh
• computing resources O(10) – O(1000)	2. e2 – 2.e4 h

1week – 2years **:stop we you have to**

20K€ - 200K€ **:convince your management**

3 Level multi-fidelity approach using response surface (neglectable cost) + ROM + HFM

• cost of Reduced Order Model	$C_{ROM} = \sigma \, C_{HFM}$
• global optimization run O(1000) ROM + O(10) HFM	2.e6 σ + 2.e4 cpuh
• necessary saving factor σ O(1month), O(10K€)	**1 – 1/100**

About optimization

- uncertain
  1. hope in “systematic errors” or “conservation of trends”
  2. what if your new prototype(!!) performs worse than original??
  3. mastered by empirical knowledge
  4. limited basin of validity

- it takes specialized technical staff
  1. to set all optimization parameters (parameters?, strategy)
  2. to build an automatic workflow (geometry??, mesh??, 1month)
  3. and to do some preliminary investigations (sensitivity, uncertainty) (1.5 month)
  4. which help you to set up the optimization run (0.5 month)

- very costly wrt to analysis
  1. computing: 10x – 100x
  2. staff: 10x – 100x

Automatic shape optimization is used only if strategic

HPC view

- 2 level parallelization
  - concurrent jobs
  - parallel execution of single job
- serial part of workflow 0.01-0.1*(parallel part)
  - geometry & mesh processing

Free-Form Deformation & Mesh-Morphing using Level-Sets

Requirements to geometrical engine

Geometry represented as surface triangulation (CAD neutral)

1. **parameterization of complex geometries**
   - Free-Form Deformation developed by Desideri et al @INRIA
   - Mesh-Morphing using RBFS
2. **constraints handling**
   - C^0, C^1, C^2 conditions on arbitrary boundaries
   - no-penetration condition
3. **features & curvature based surface mesh adaptation**
   - if deformed geometry needs finer surface mesh than original geometry

Different approaches

Direct morphing of surface & mesh

- surface constraints are handled by choosing wisely CPs (e.g. inner points of lattice, rbf nodes with given distance to boundaries)
- mesh constraints handled via limitations on bounds of CP
- very cheap, since only evaluations of Deformation Lattice or RBF required

http://mathlab.sissa.it/pygem

used in the final examples!!

1. **surface deformation**
2. **mesh morphing**

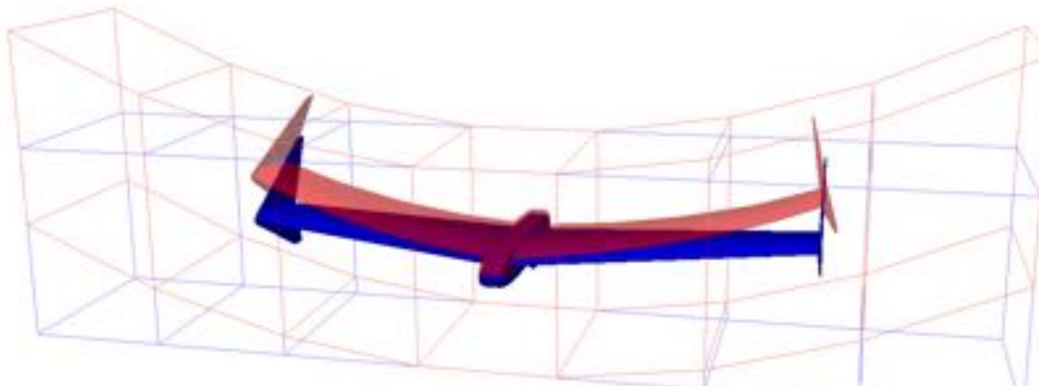
- surface constraints via topological information (geodesic distances) on surface
- volume constraints via computation of Euclidean distances
- expensive, since constraints are computed explicitly at each deformation
- very expensive, since mesh morphing is formulated as an interpolation (minimization) pb on surface deformation

https://github.com/optimad/MIMMO

http://www.optimad.it/products/camilo

explicit surface constraints

Free-Form Deformation applies a displacement vector $N_i = S_i + D(S_i)$

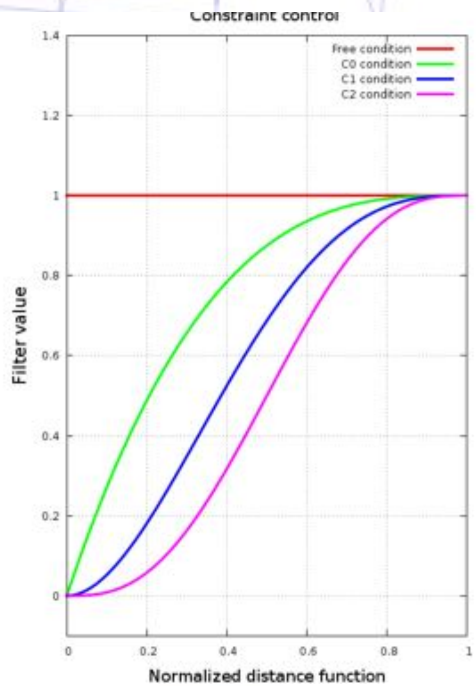


- difficult to impose regularity
- conditions on an arbitrary
- shaped boundary **Γ**

- our approach introduces a weight function
- $N_i = S_i + w[\ \phi(S_i|\Gamma)\ ]\ D(S_i)$

- with $w(0) = 0$ for C^0 condition
- with $w(0) = 0\ ,\ w'(0) = 0$ for C^1 condition
- with $w(0) = 0\ ,\ w'(0) = 0\ ,\ w''(0)=0$ for C^2 condition

- **$\phi(S_i|\Gamma)$ must provide topological information**
- **but it is requires that $\phi(S_i|\Gamma)$ is C^0, C^1 and C^2 respectively**

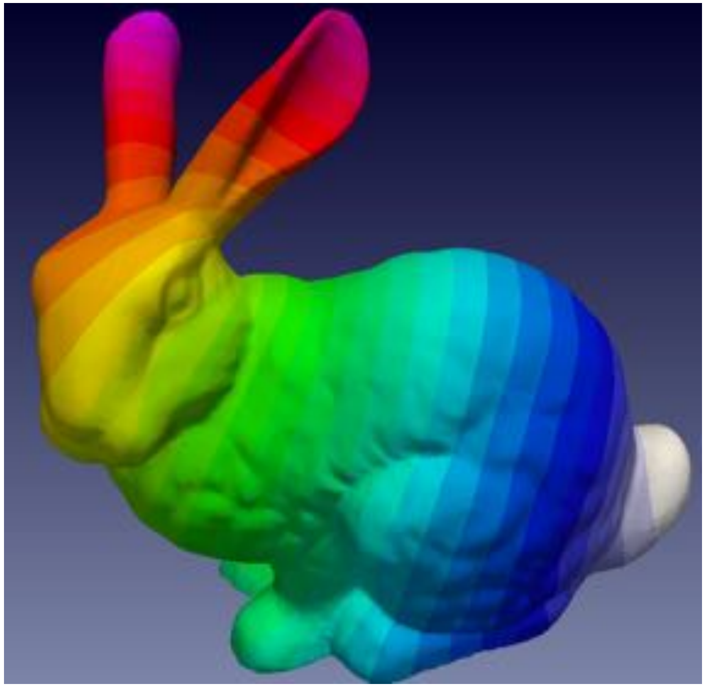


topological information 1: exact geodesic distances

source: Crane et al.

resulting function is only C^0, cannot impose higher regularity

topological information 2: smoothed geodesic distances



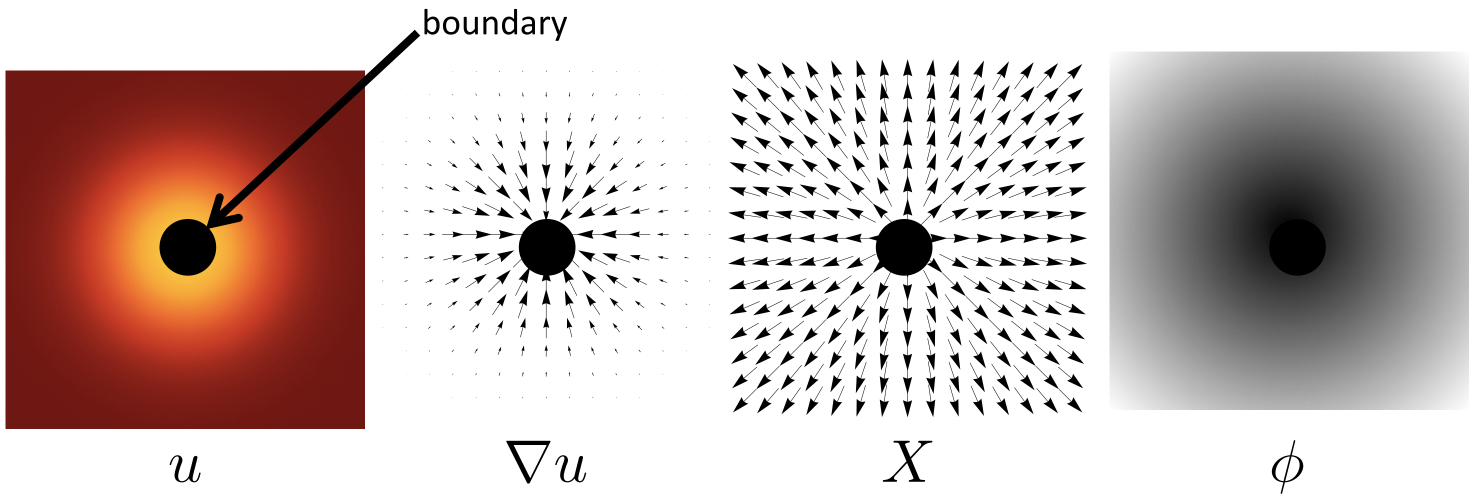
we impose the following optimization problem:

- as close as possible to geodesic LS to keep topology
- constraint on C^2 continuity
- infinite solutions -> smoothing parameter

this leads to the solution of 1 parabolic and 1 elliptic PDE

- sparse direct solver with reordering is used

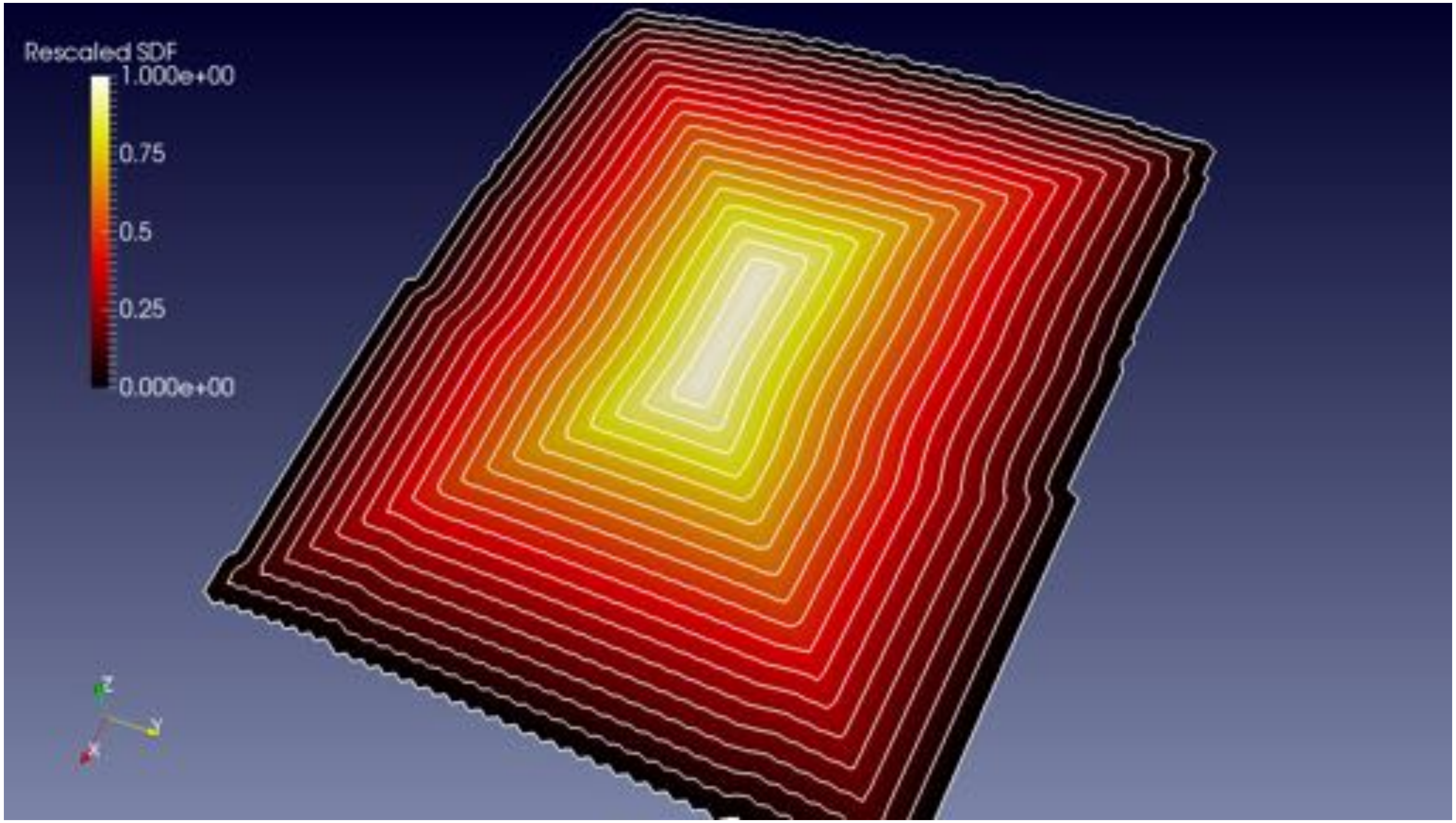
Geodesics based on heat kernel



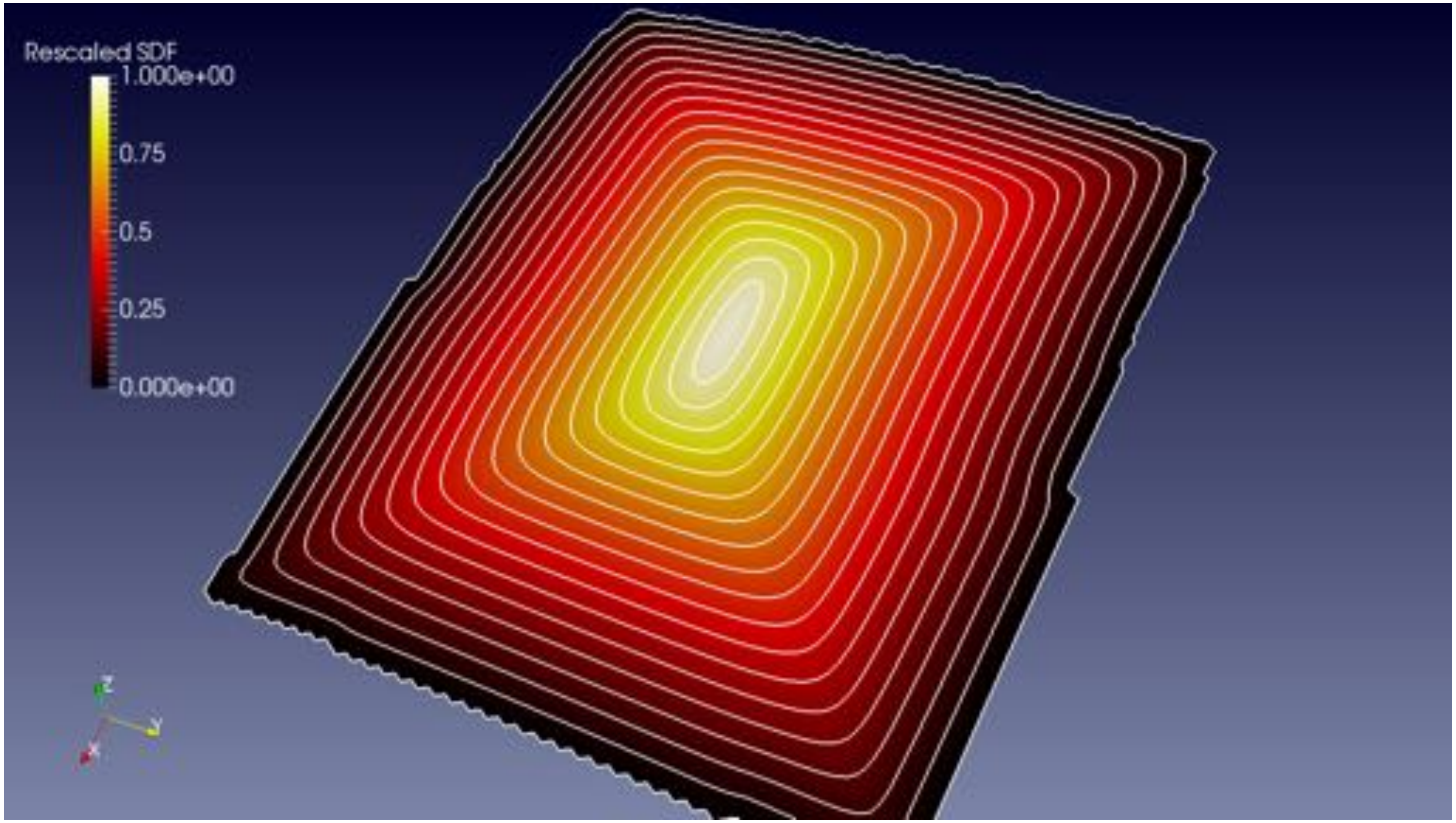
1. resovle heat equation $\mathbf{u_{,t} = -u_{,xx}}$ for a given time (parameter for smoothing)
2. calculate **X = -grad u / |grad u|**
3. solve **lap Φ = div grad X**

As similar as possible to geodesic distance, but imposes smoothness

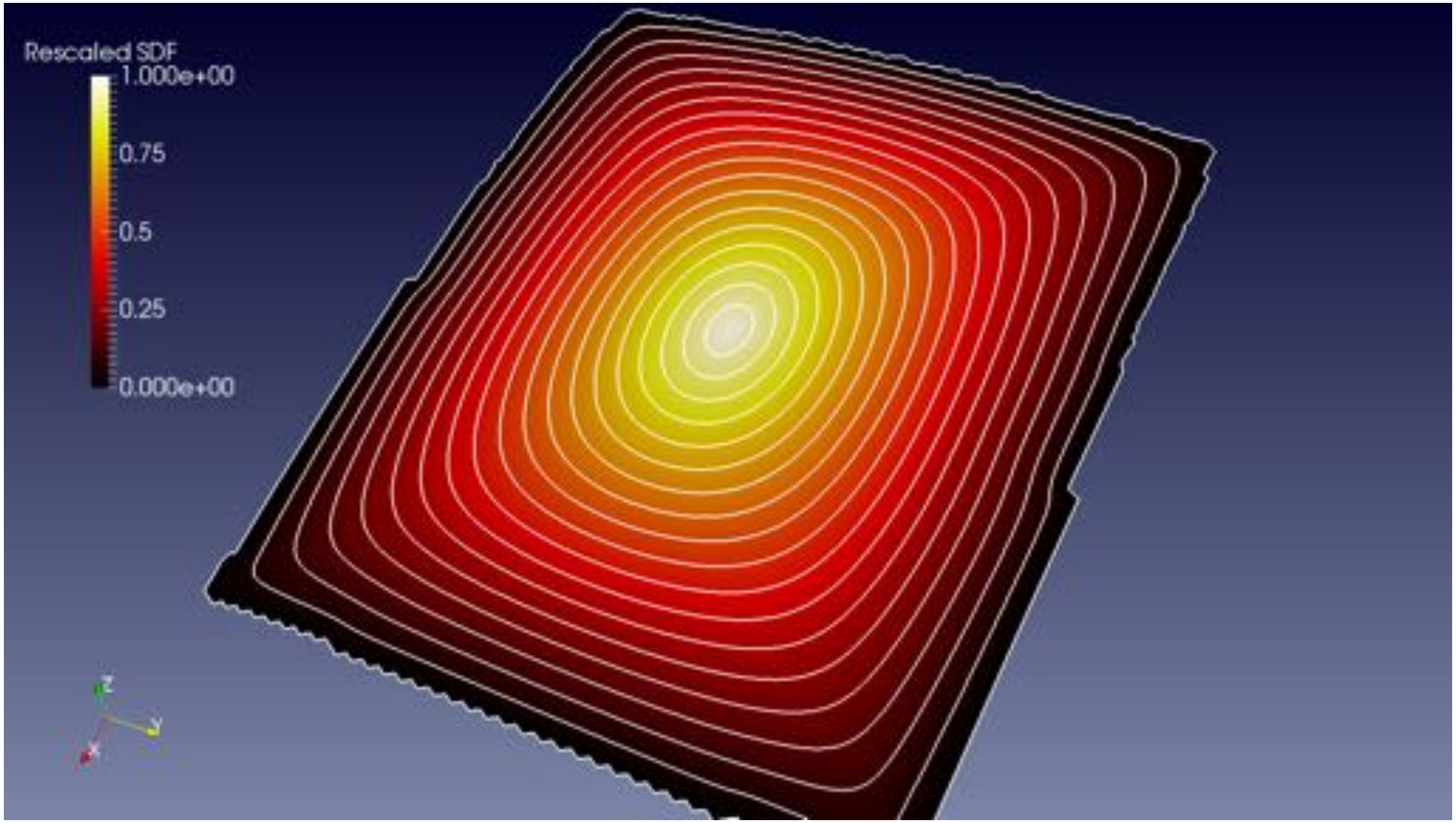
Heat kernel solution: t=0



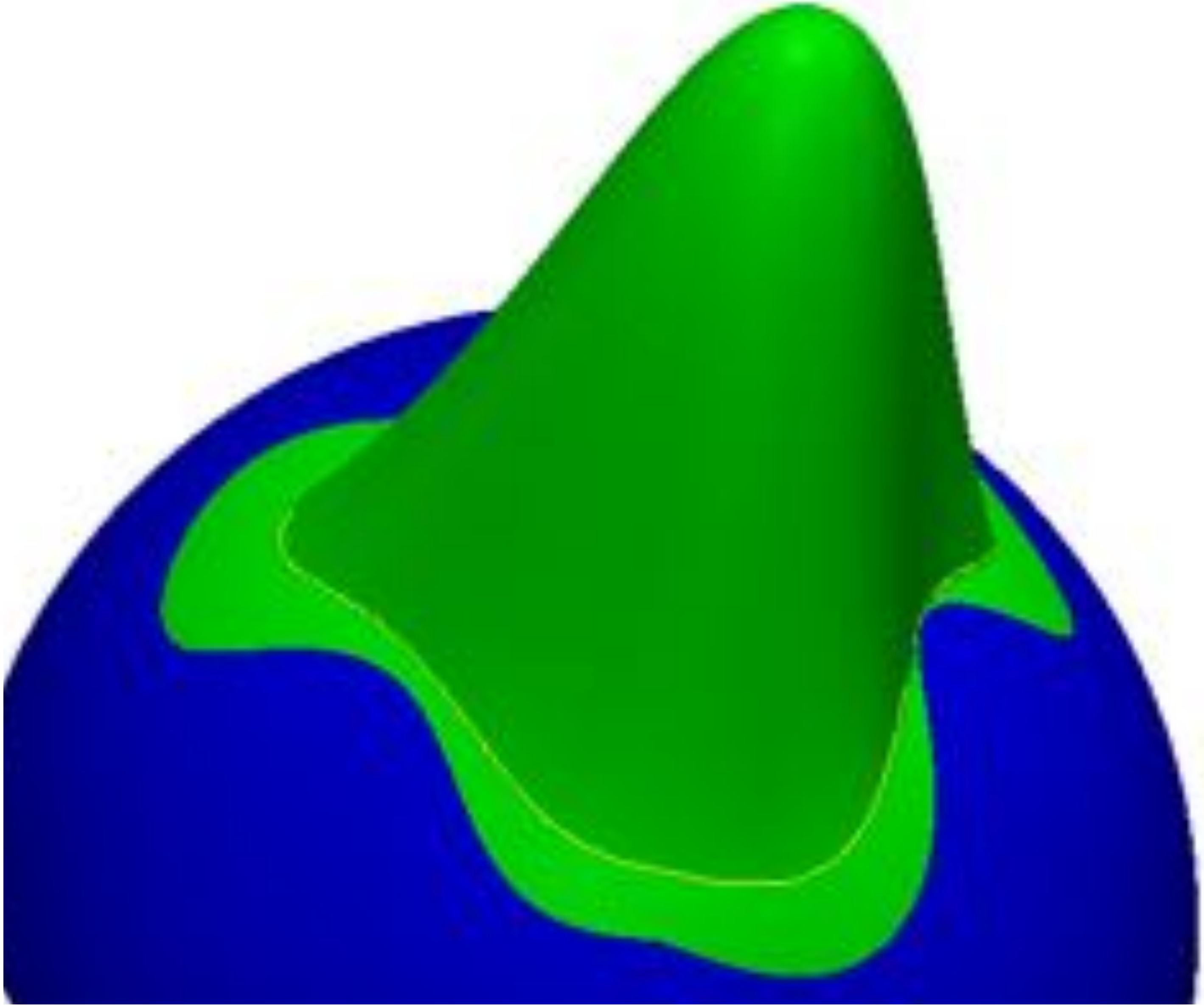
Heat kernel solution: t=0.1



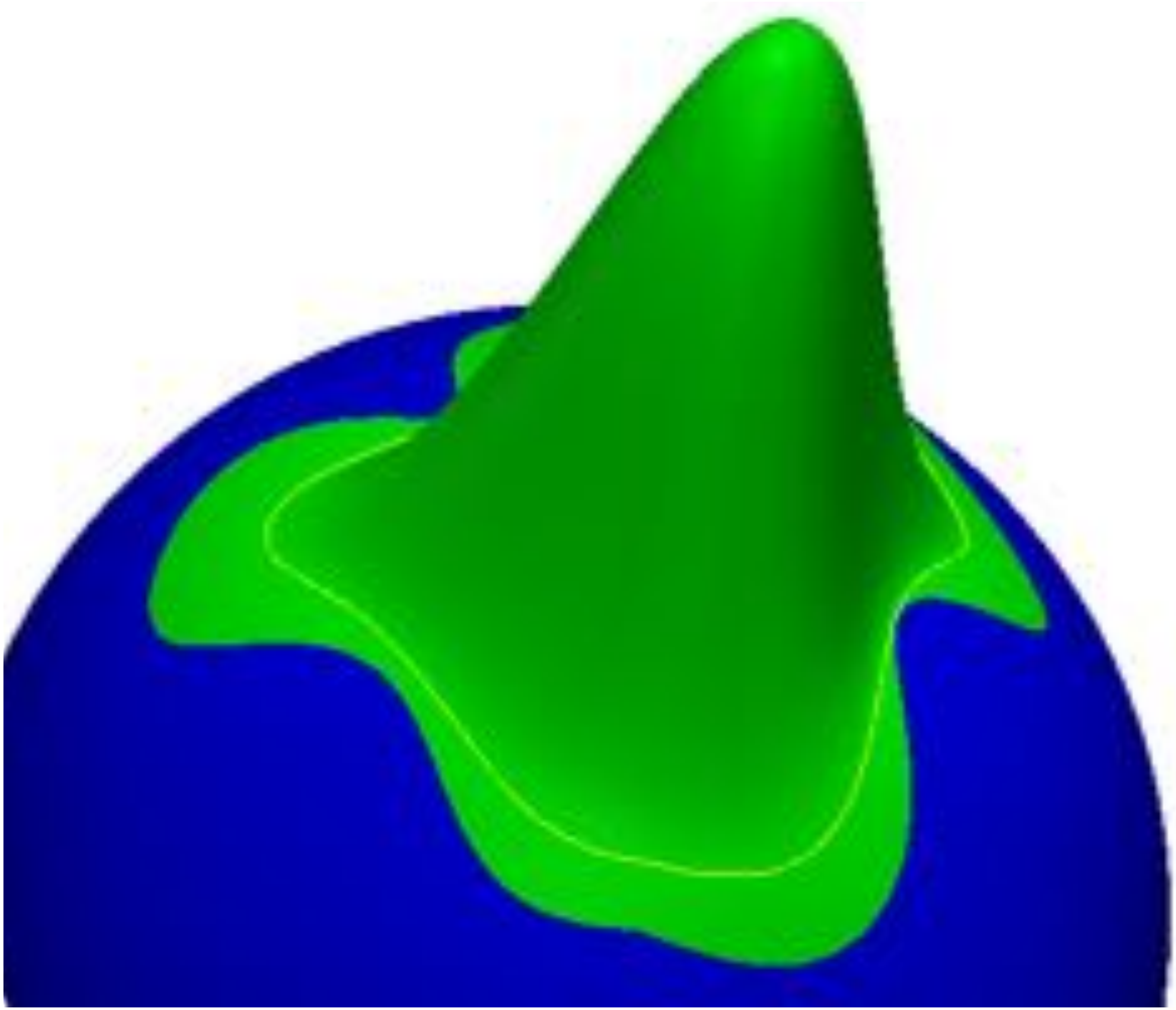
Heat kernel solution: t=1.0



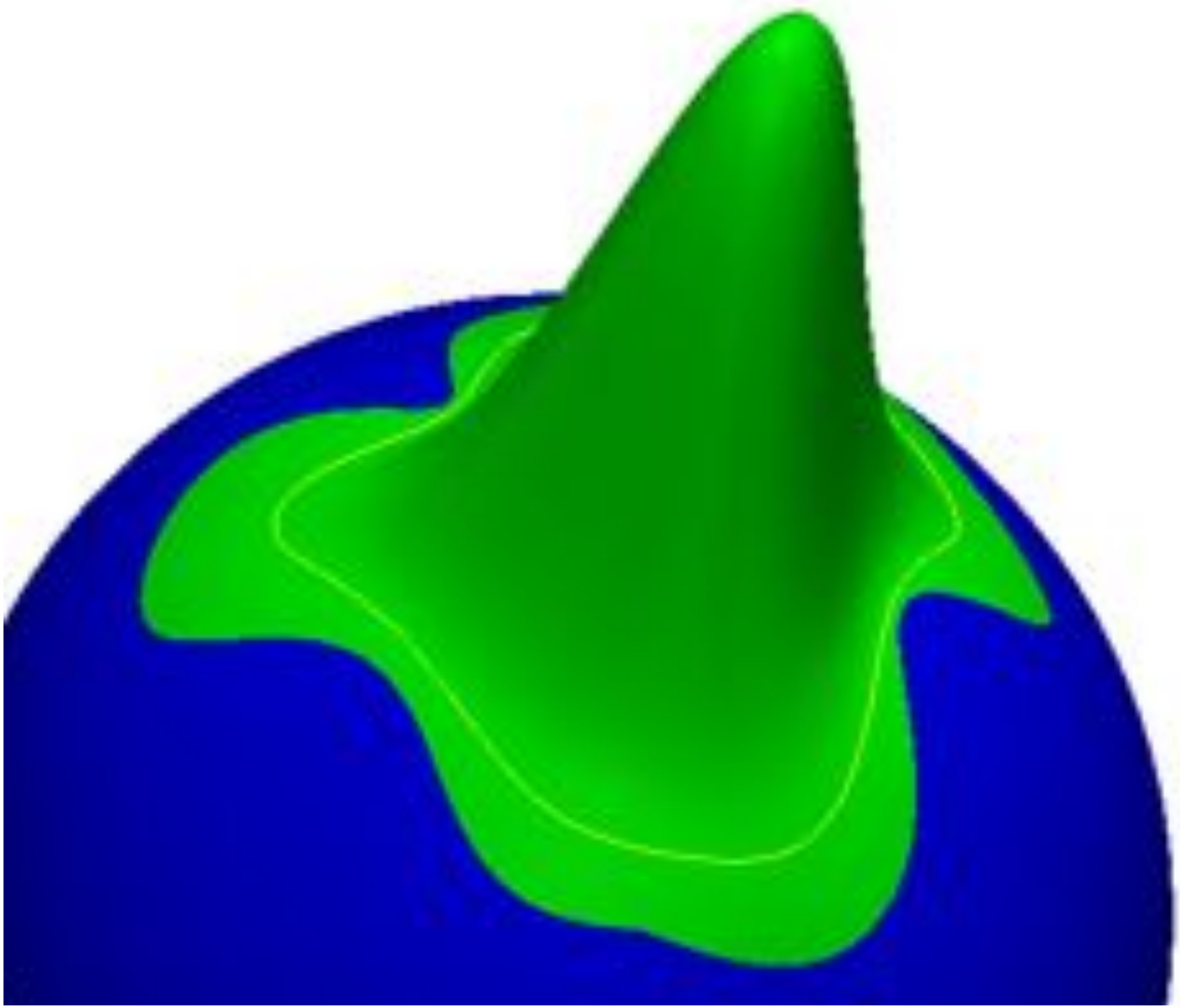
Deformation using C^0 constraint



Deformation using C^1 constraint



Deformation using C^2 constraint



volume constraints

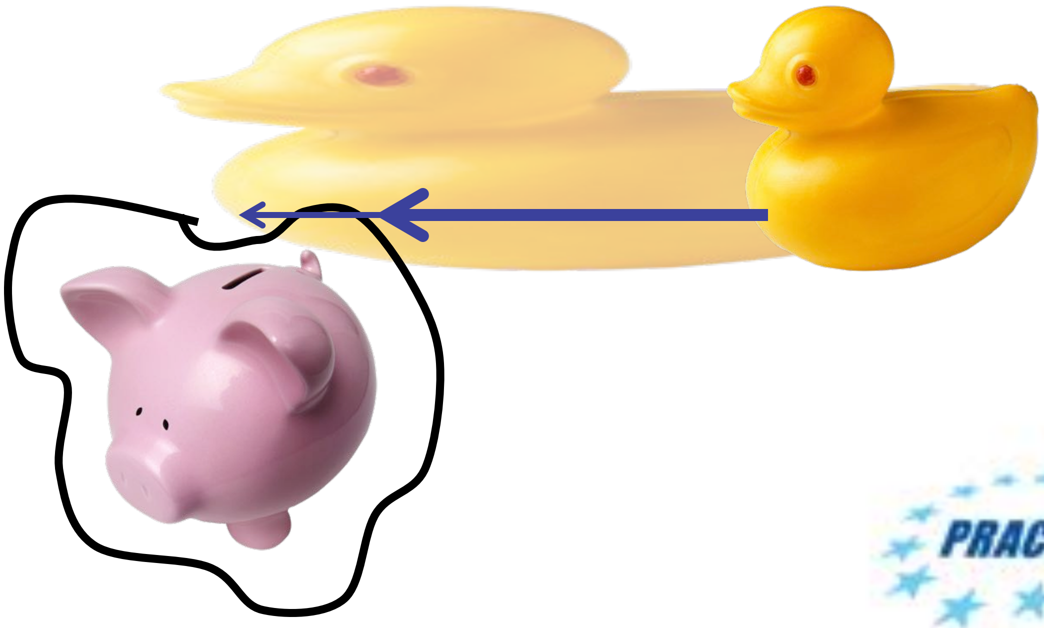
Common constraint: Distance to a given surface should be maintained

- 1. User should indicate only intuitive information
  - surface (open or closed)
  - distance to be maintained
  - bounds on CP non-intuitive and often not efficient
- 2. Combined control algorithm
  - Ray-tracing
  - Level Set
  - Line search
- 3. Two different types of rescaling algorithms available

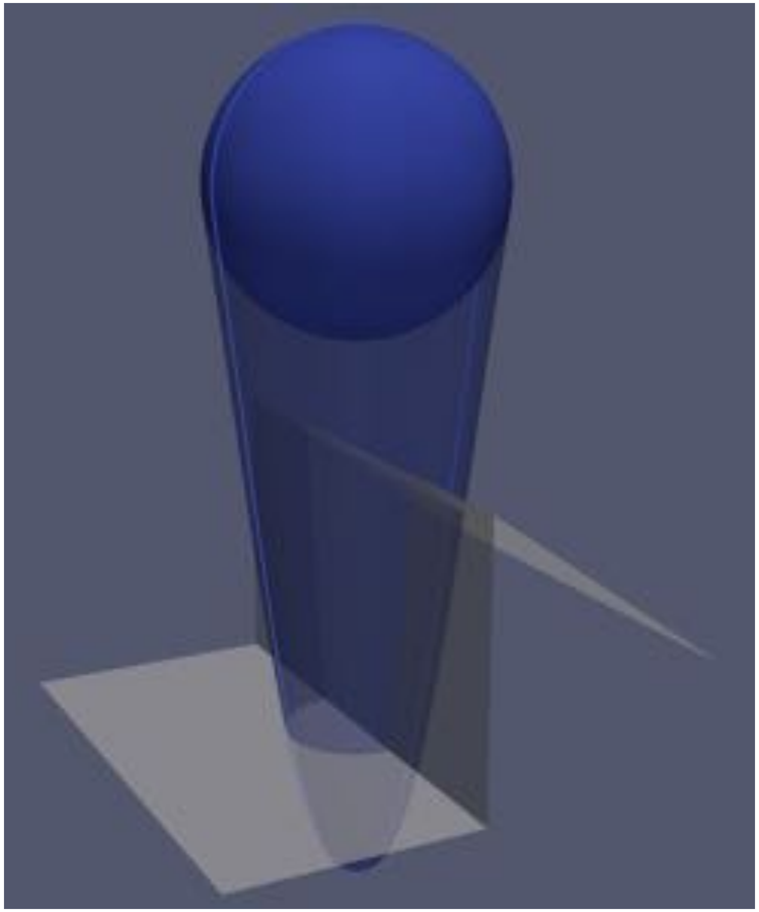
basic algorithm

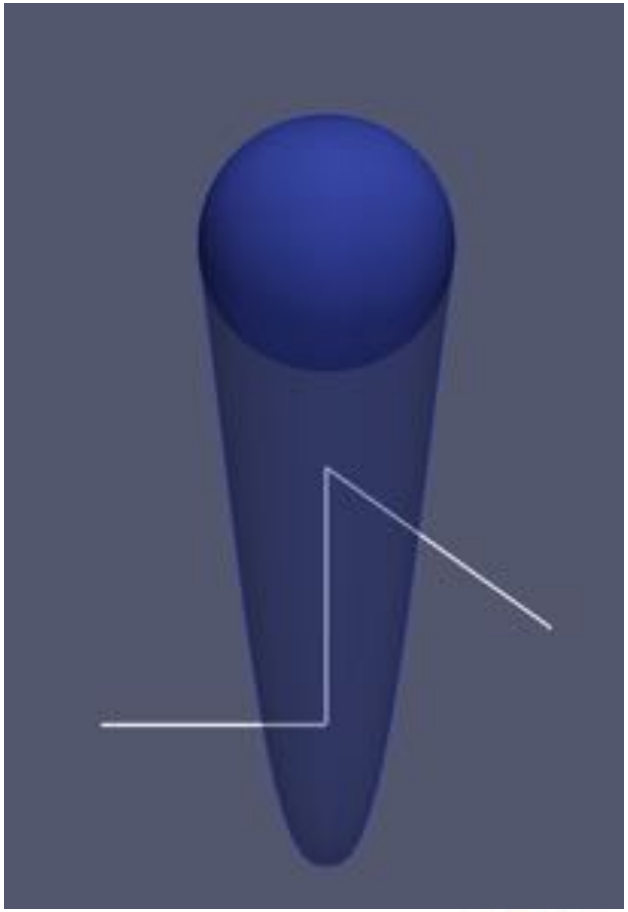
- Given an unconstraint deformation field and a control surface
- Calculate the Level-Set function (signed distance) of constraint
- Perform a ray-tracing step using deformation field as rays
- Compute for each surface point allowable fraction

- All steps are extremely scalable

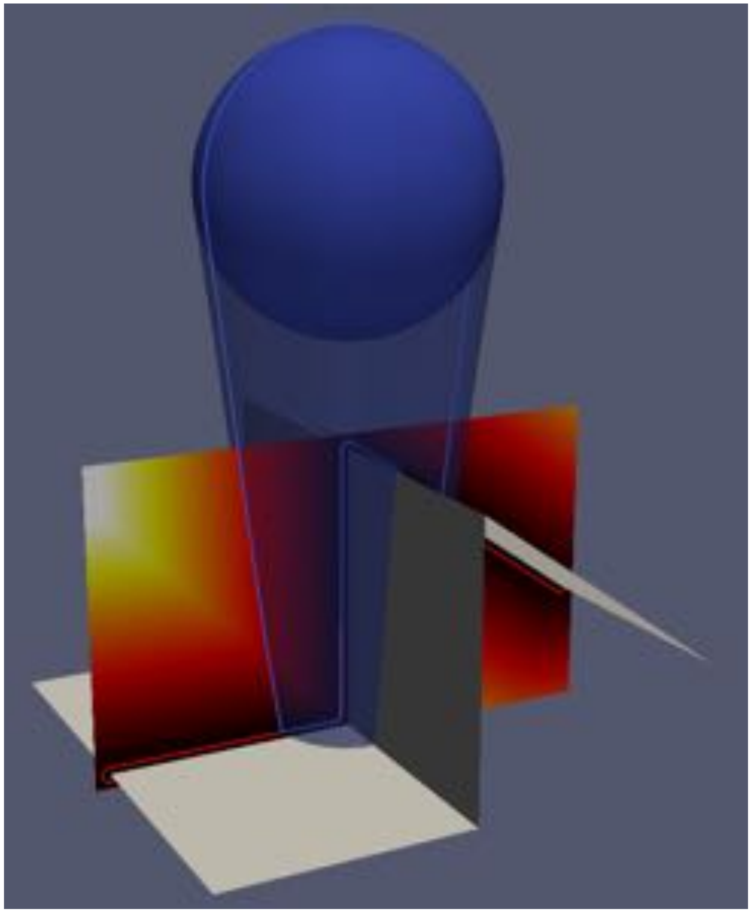


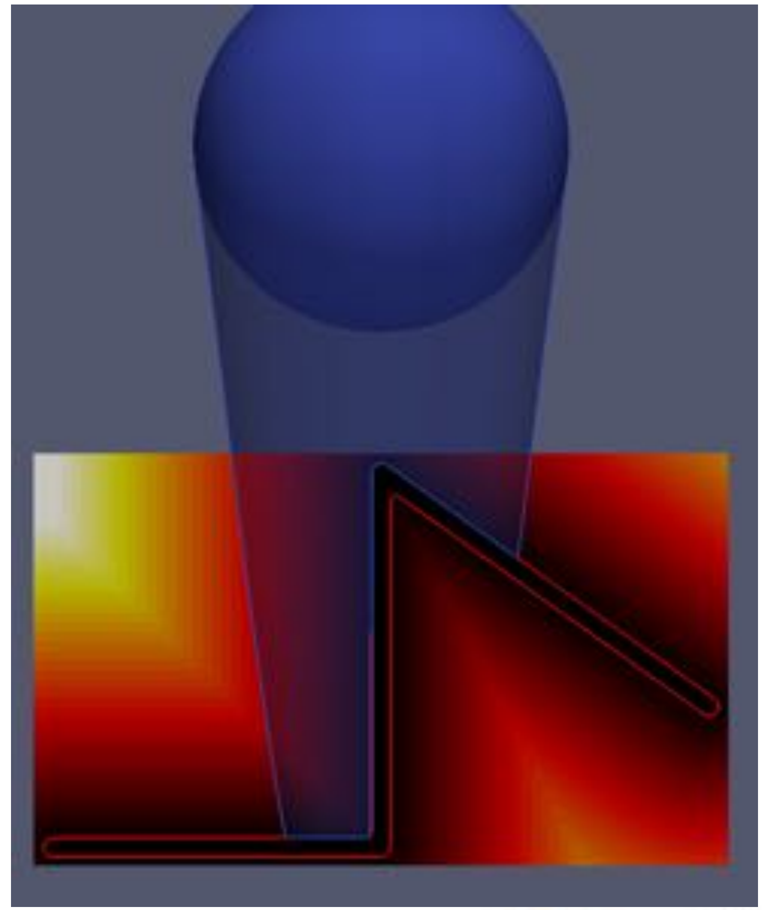
Controll off





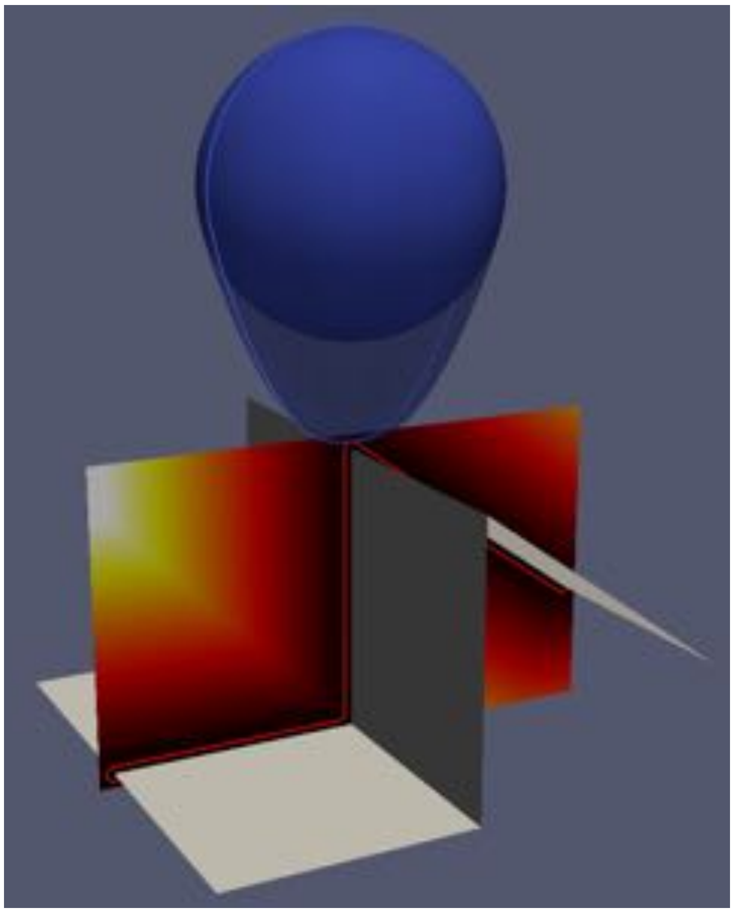
Local rescaling

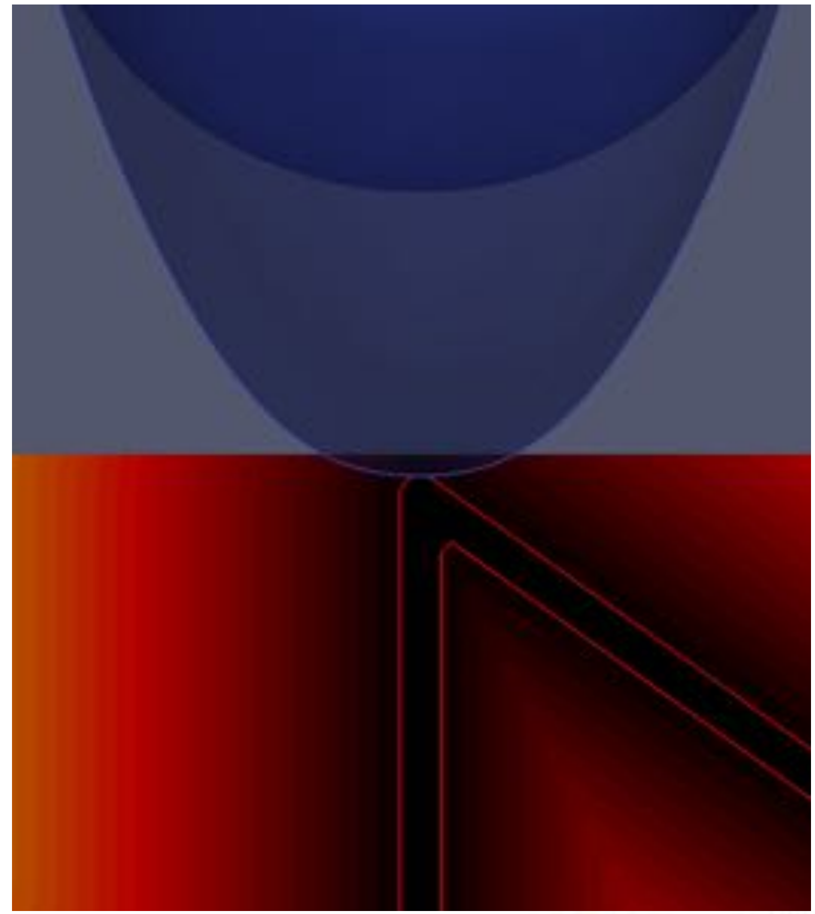




distance wrt tangential projection

Global rescaling





identification of critical point

Mesh-Morphing

- propagate surface deformation to mesh
- avoid usage of sHM but creation of one initial high quality grid
- brute force: RBF with one node on each surface vertex

- very costly (N_V volume grid nodes, N_S surface grid nodes):
  - solve phase: dense linear system N_S DoF
  - evaluation phase: N_V * N_S operations (n^5)

- greedy algorithm:

  initial RBF with one node @ largest surface displacement;

  calculate initial error;

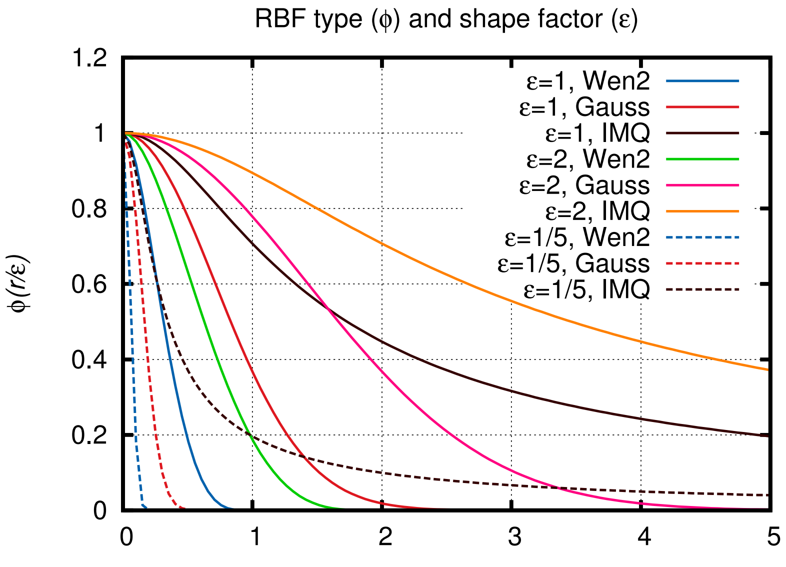
  while (maxError > tolerance) do

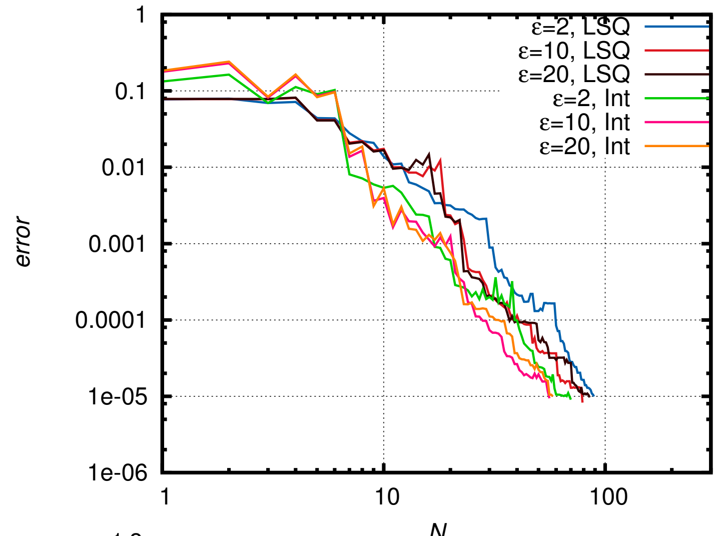
  - evaluate RBF at each surface node
  - calculate error = ||realDispl-reconDispl||
  - add new node @maxError

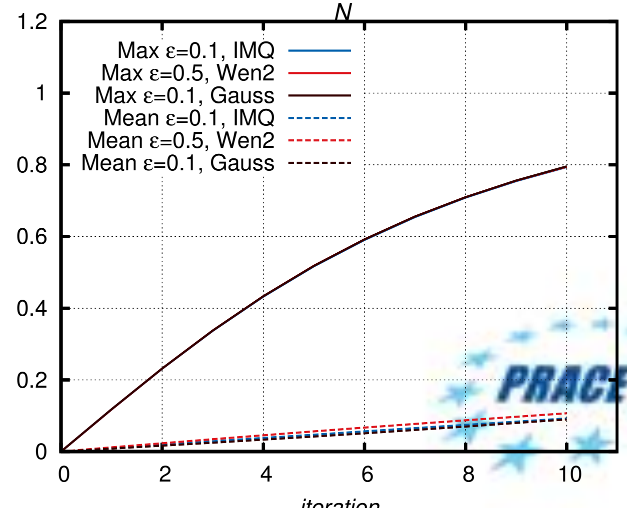
  enddo

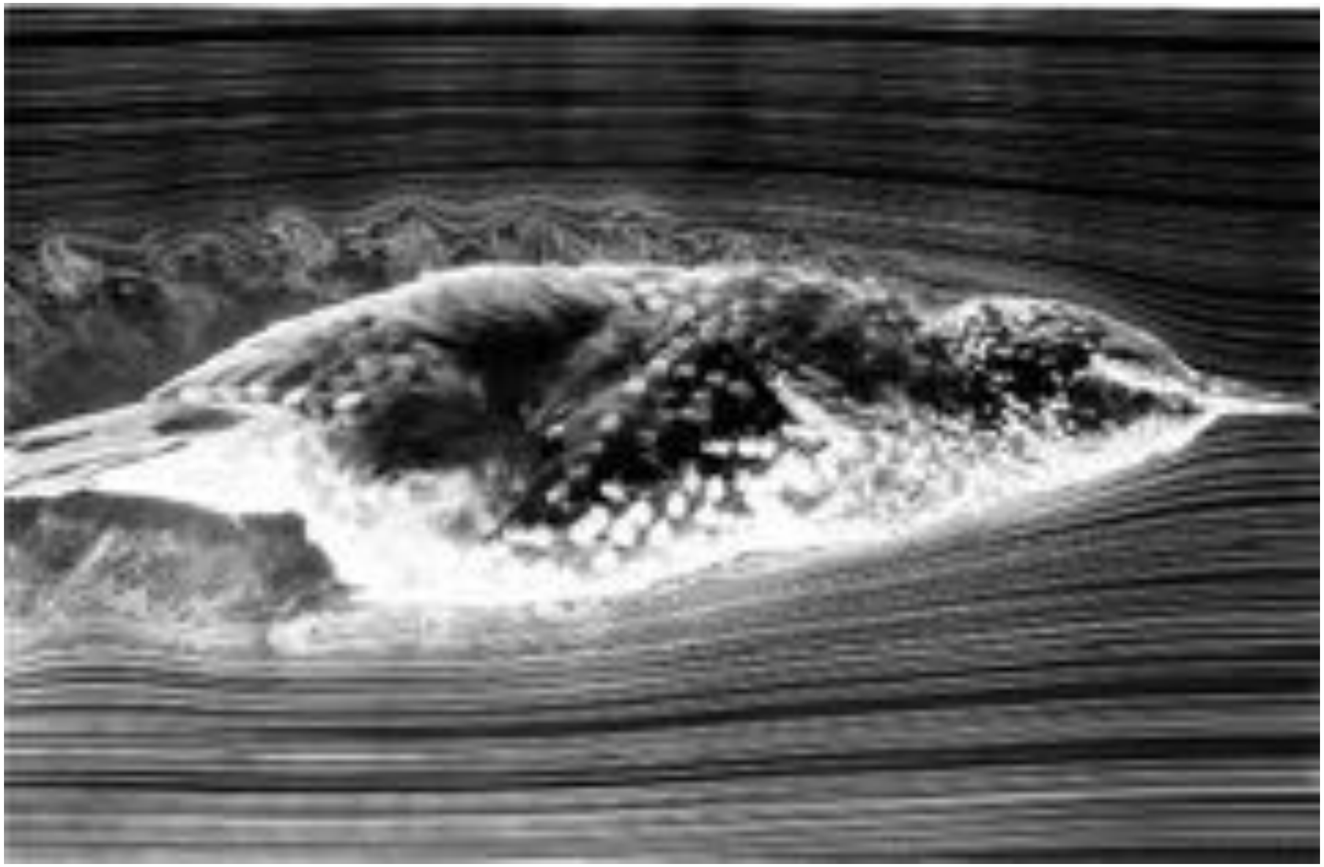
Mesh-Morphing

- RBF types, convergence and quality, $N_S = O(10^4)$









podFOAM & ezRB: Reduced Order Models based on POD

HPC based HF + ROMs

- Models tend to saturate HPC resources
  - bigger & more complex (e.g. DES, multi-physics)
  - more reliable & accurate (??)
- Computing time does not decrease as computing power increases
  - big challenge for optimization
- Scenario
  - use high-fidelity simulations to build a knowledge database (few, but which?)
  - recycle your data through semi-empirical Reduced Order Models

Scenario 1

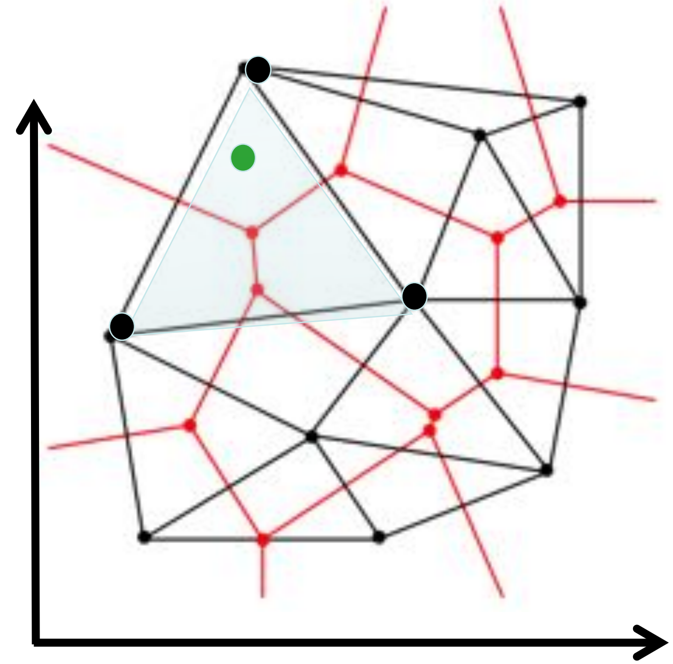
you knew from the very beginning of your project that you would do shape optimization

- parametric geometrical model ($a_0 \ldots a_{N-1}$)
- create a database of solutions (DoE)
- associate set of parameters a_i to each solution of DB

- make a Voronoi tesselation of the parameter space
- build a linearized model for each simplex

ezRB

- tessellation of parameter space
  - can be done efficiently in N dimensions
  - small problem size
- requested solution
  - locate right simplex
  - interpolate solution at simplex vertices
- can be performed efficiently via POD



Scenario 2

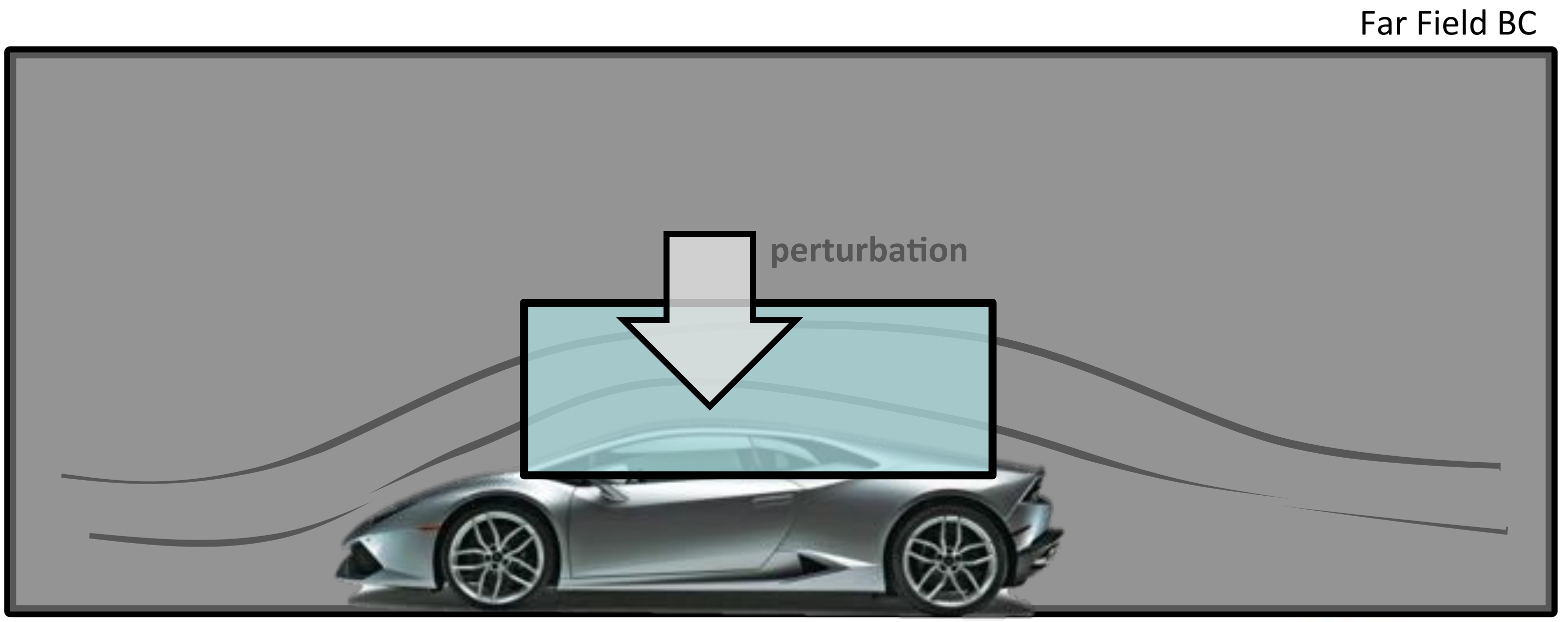
you didn't know that you would need some shape optimization

- database of loose solutions

- parametric geometrical model ($a_0 \ldots a_{N-1}$)
- feed your CFD with information from DB to reduce cost

podFOAM

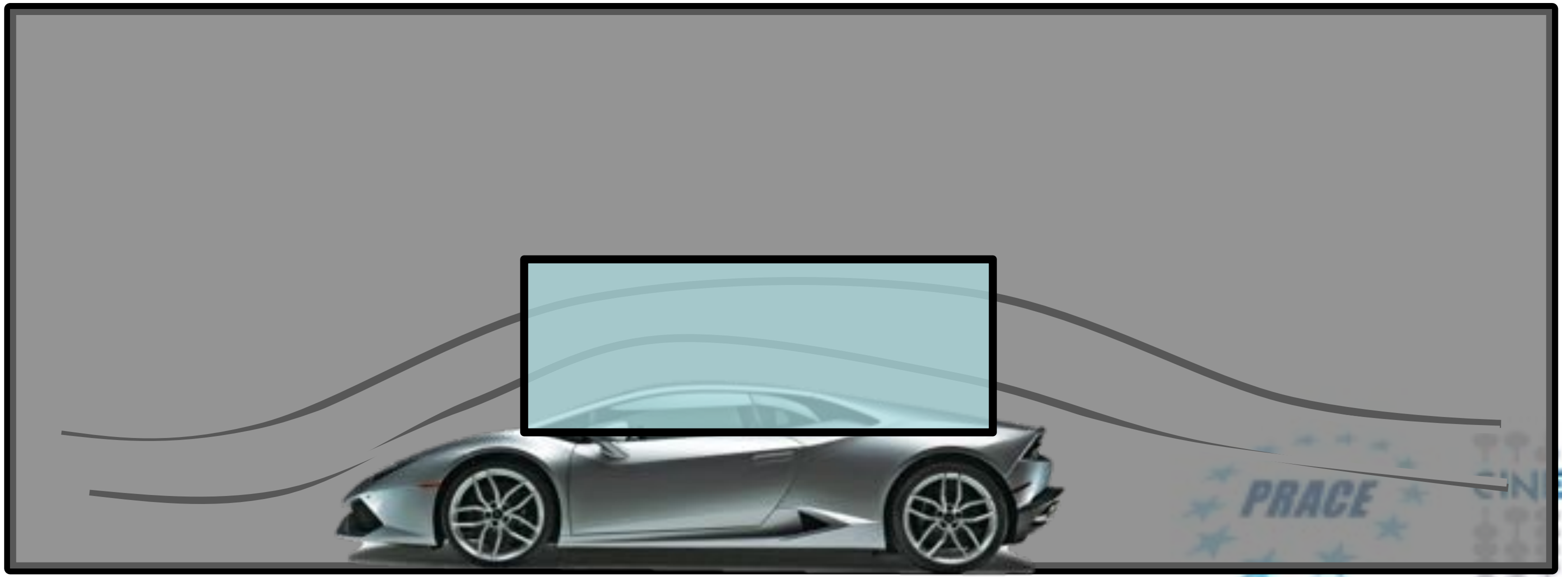
inner zone: use non-linear CFD
outer zone: use simplified model to impose BC to inner zone



podFOAM

- Our assumptions are that
  - in the outer zone, the perturbation becomes linear
  - the **information needed** for describing the flow field of outer zone, is **already available** in the data stored on your HD

- **Represent the green zone by Proper Orthogonal Decomposition**
- **Couple to CFD in blue zone** through a Least-Squares Problem on the data at the interface



POD

- **Proper Orthogonal Decomposition**

  - representation of a solution as $\mathbf{u^j_i(x) = \Sigma\, a^j_i\, \Phi_i(x)}$ for i= 0…N-1
  - $\Phi_i(x)$ I sorthogonal POD basis, which can be found by solving the eigen-problem of the **snapshot correlation matrix**
  - **no series converges faste**r than POD; identification of **coherent structures**; **very few modes** to capture 99% of the energy

Sampling strategy

- both ROMs zero error at solution of DB
- the ROM should be reliable in entire parameter space
- if *a priori* error available, additional snapshots in critical zones

- Leave-One-Out strategy to determine pseudo error

foreach solution of DB

– remove solution from DB
– recalculate ROM
– evaluate ROM at solution point
– calculate error = $||U_{HF} – U_{ROM}||$

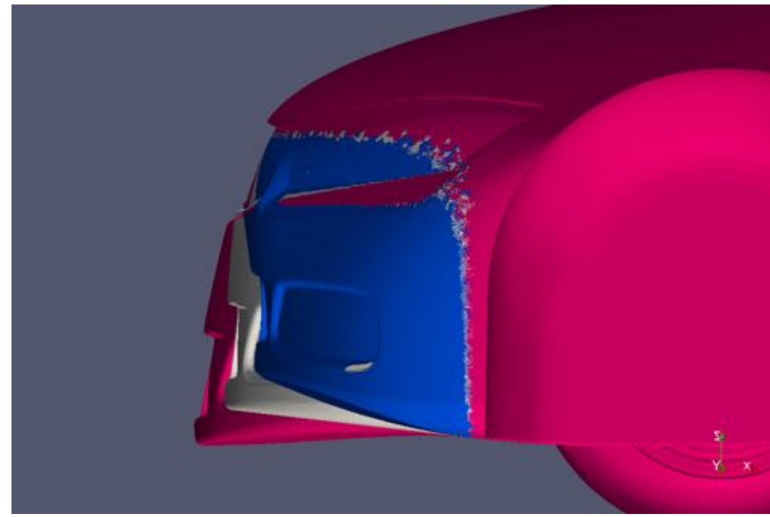
end foreach

add new snapshot where indicator is high & far from points

DrivAer model

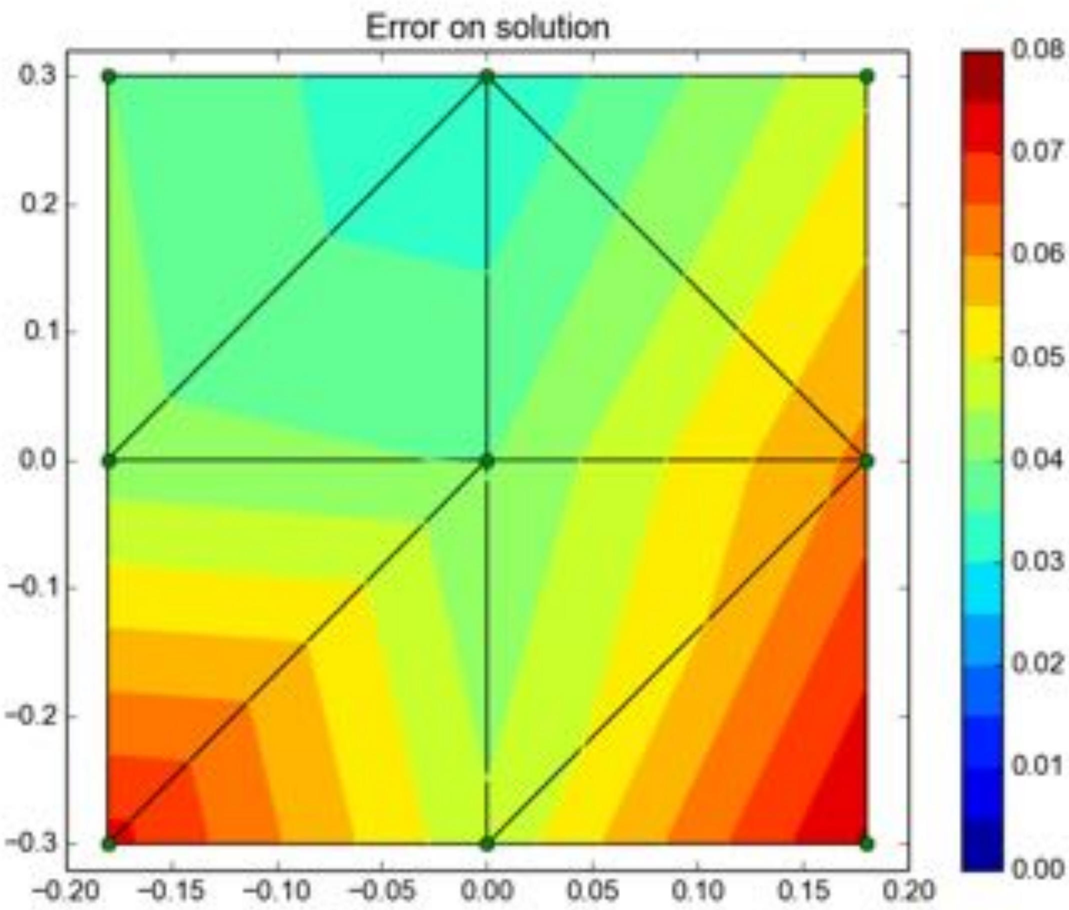
Free model by TU Munich in collaboration with Audi & BMW

- Clean symmetric model: 14M cells
- 2 control parameters
- $S_{forces} = 0.1$



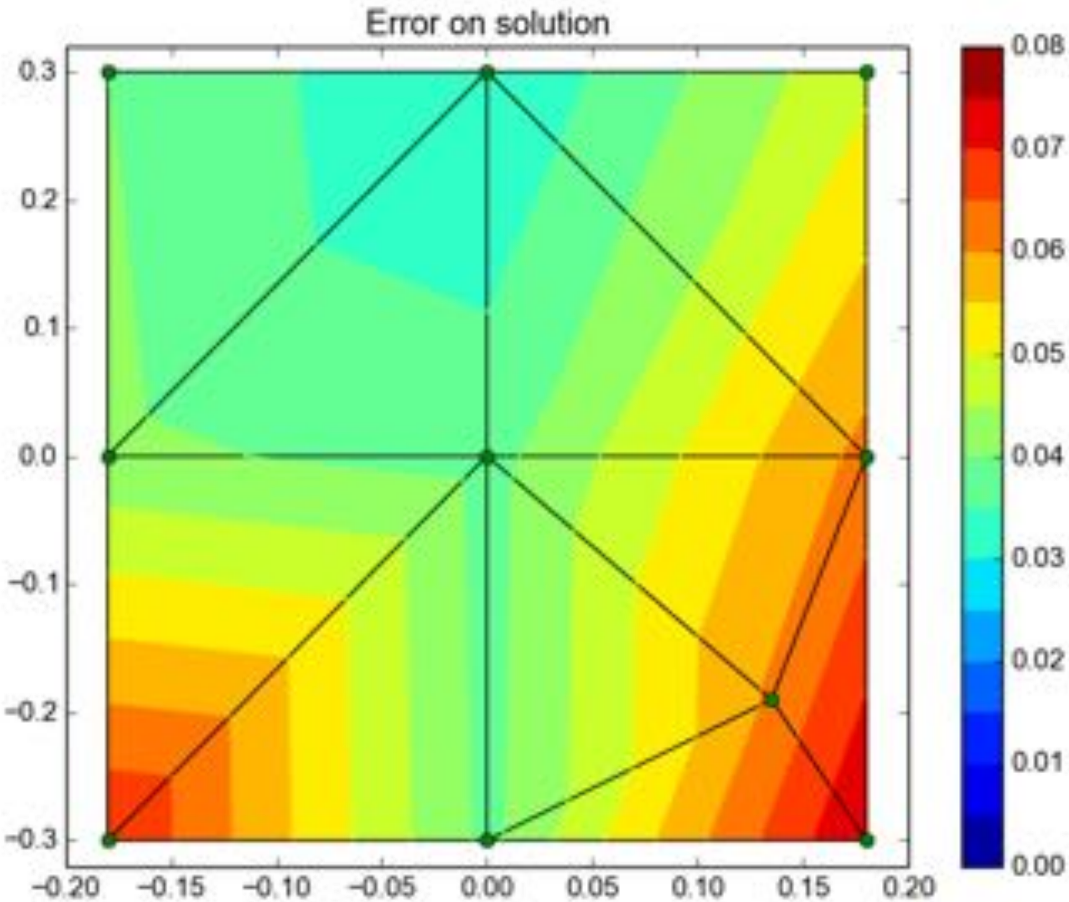
Sampling strategy

Iteration 0



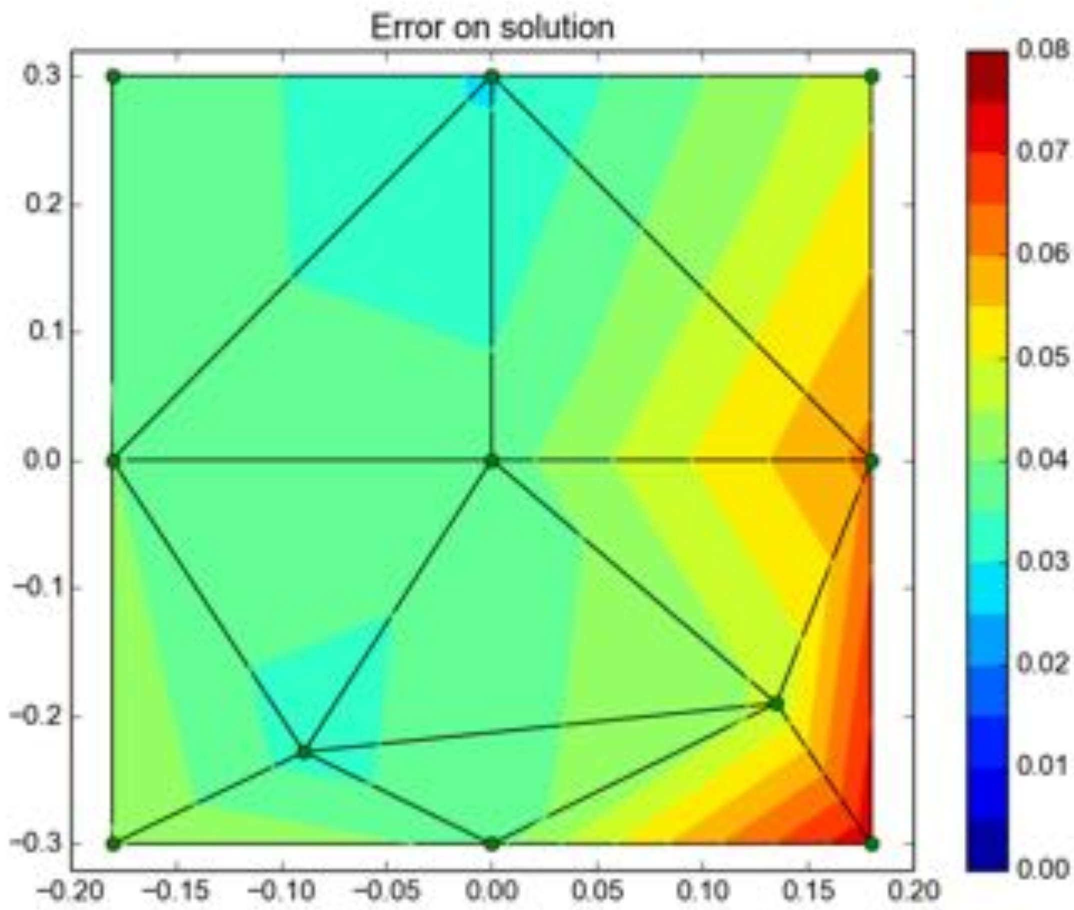
Sampling strategy

Iteration 1



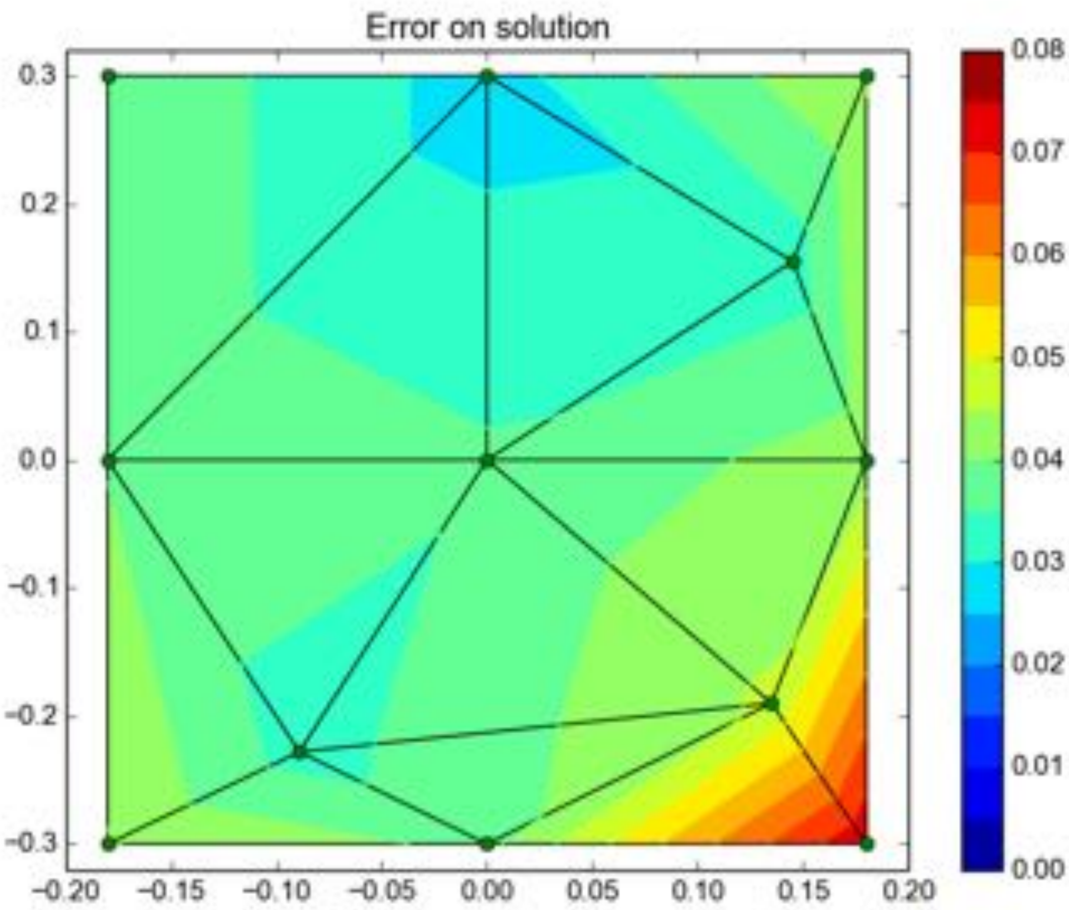
Sampling strategy

Iteration 2



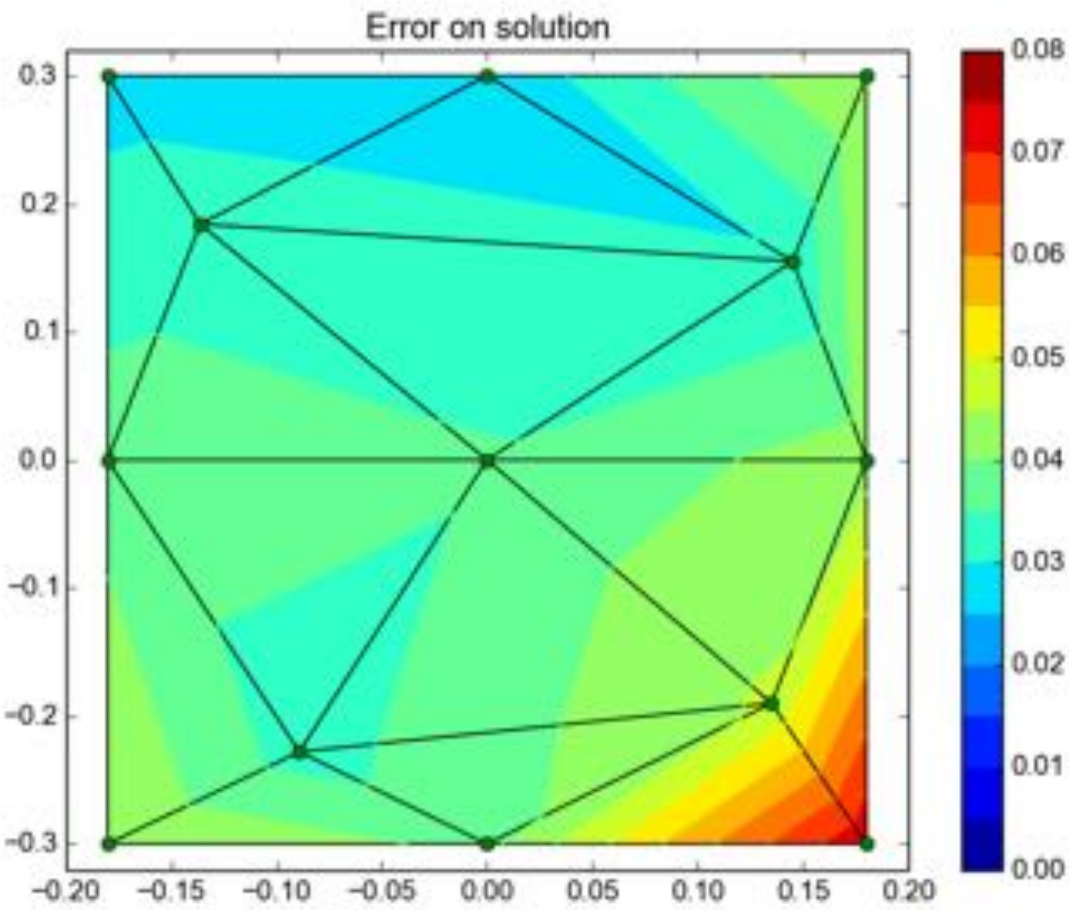
Sampling strategy

Iteration 3



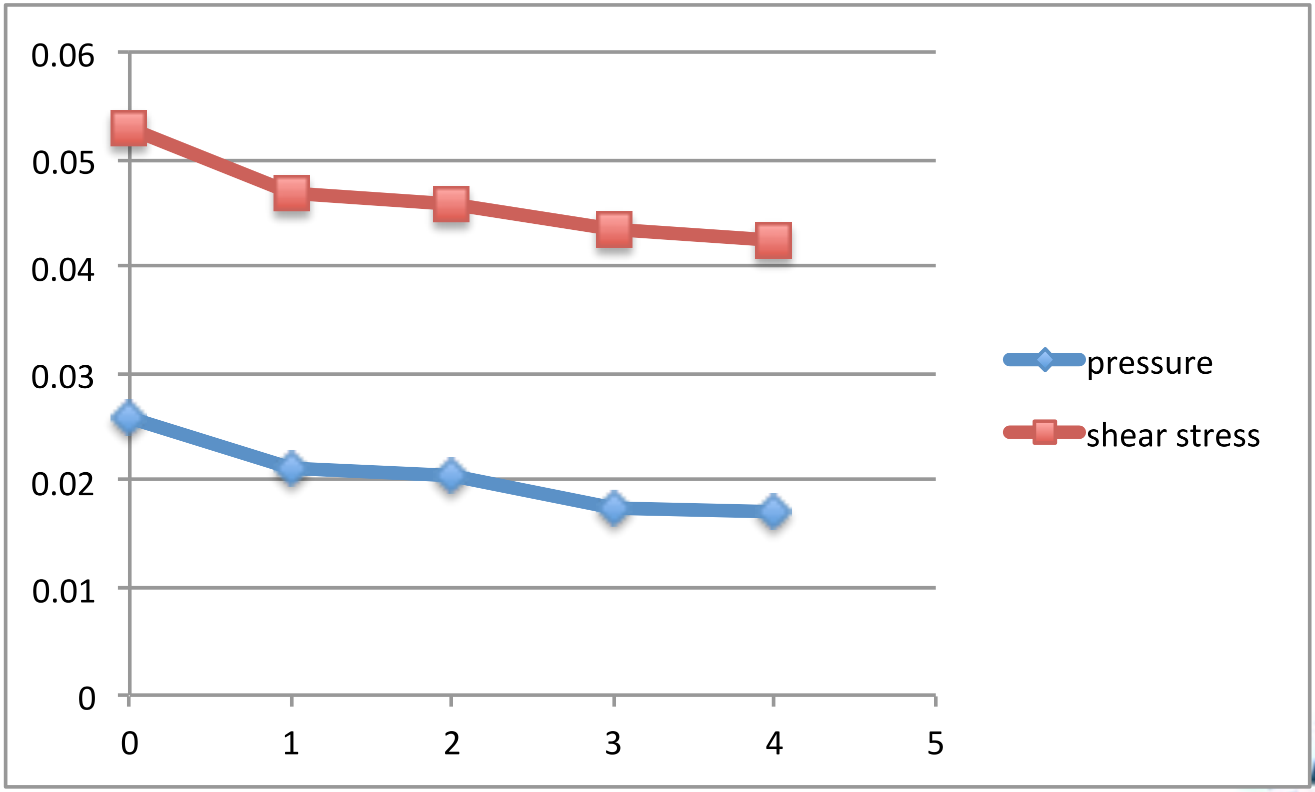
Sampling strategy

Iteration 4

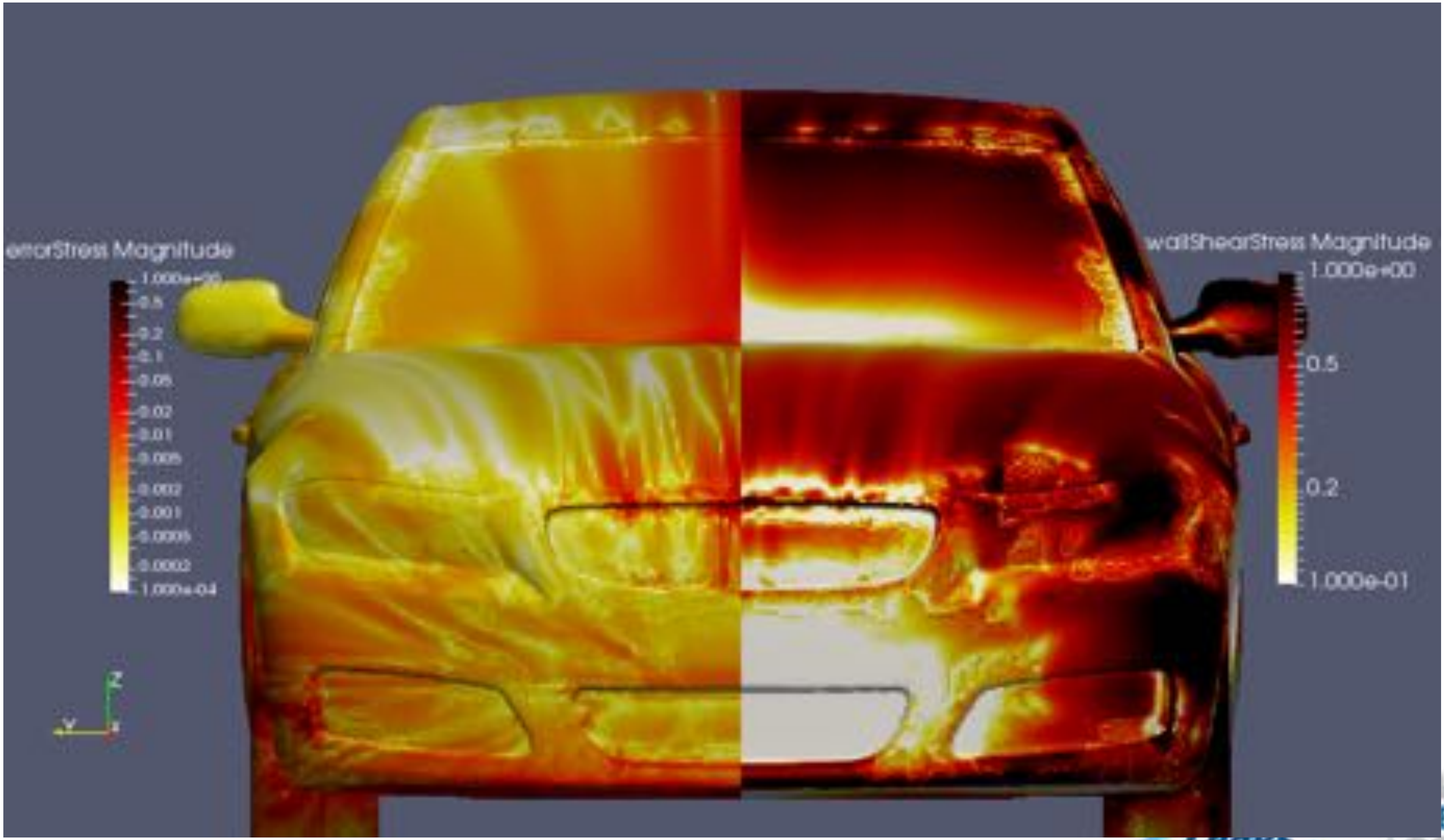


ezRB

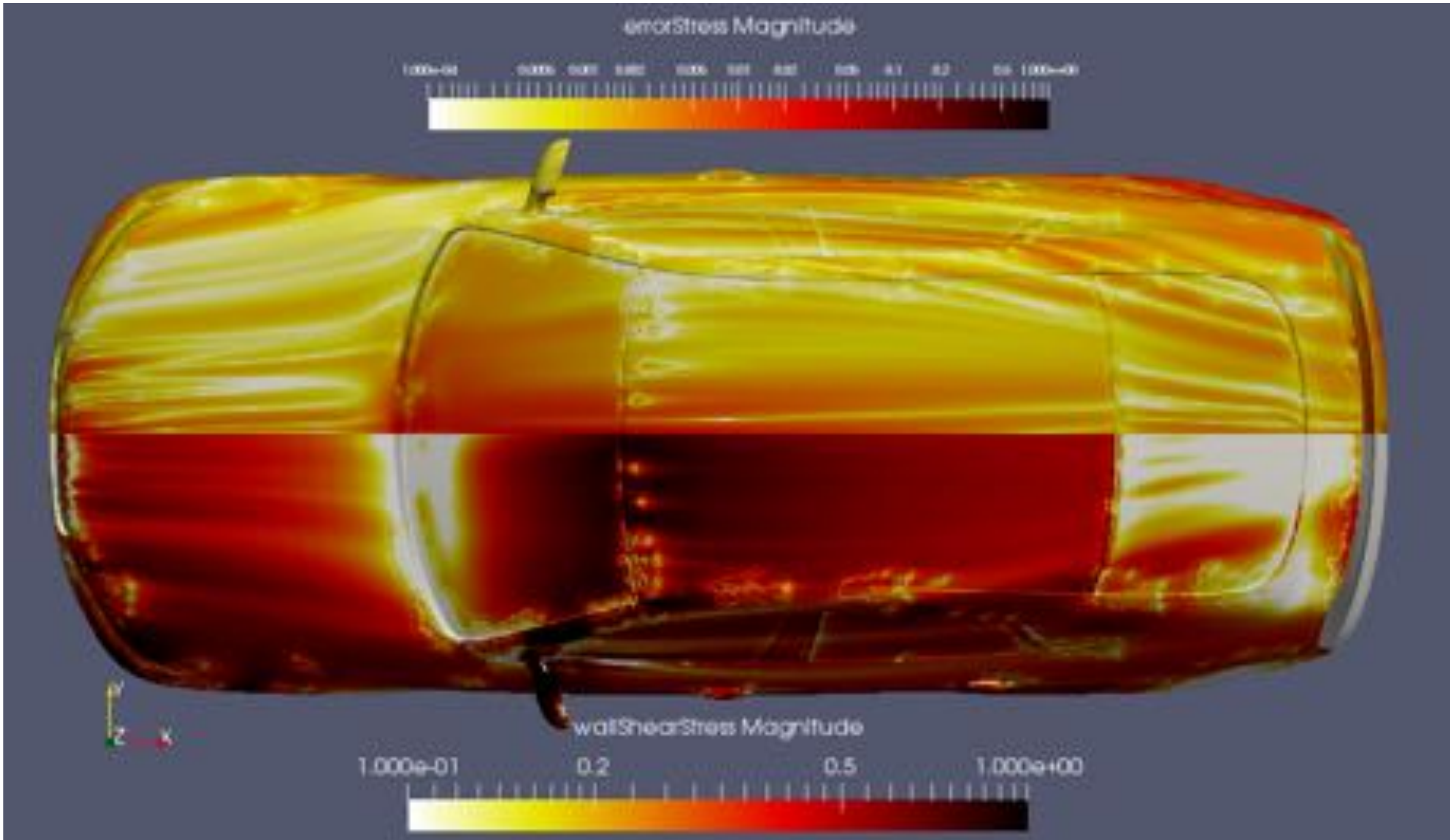
- reconstruction of surface pressure and shear stress
- mean error over 4 random configurations out-of-DB
- Cost O(s)



ezRB


errorStress Magnitude
0.5
0.2
0.1
0.05
0.02
0.01
0.005
0.002
0.001
0.0005
0.0002
1.000e-04
wallShearStress Magnitude
1.000e+00
0.5
0.2
1.000e-01

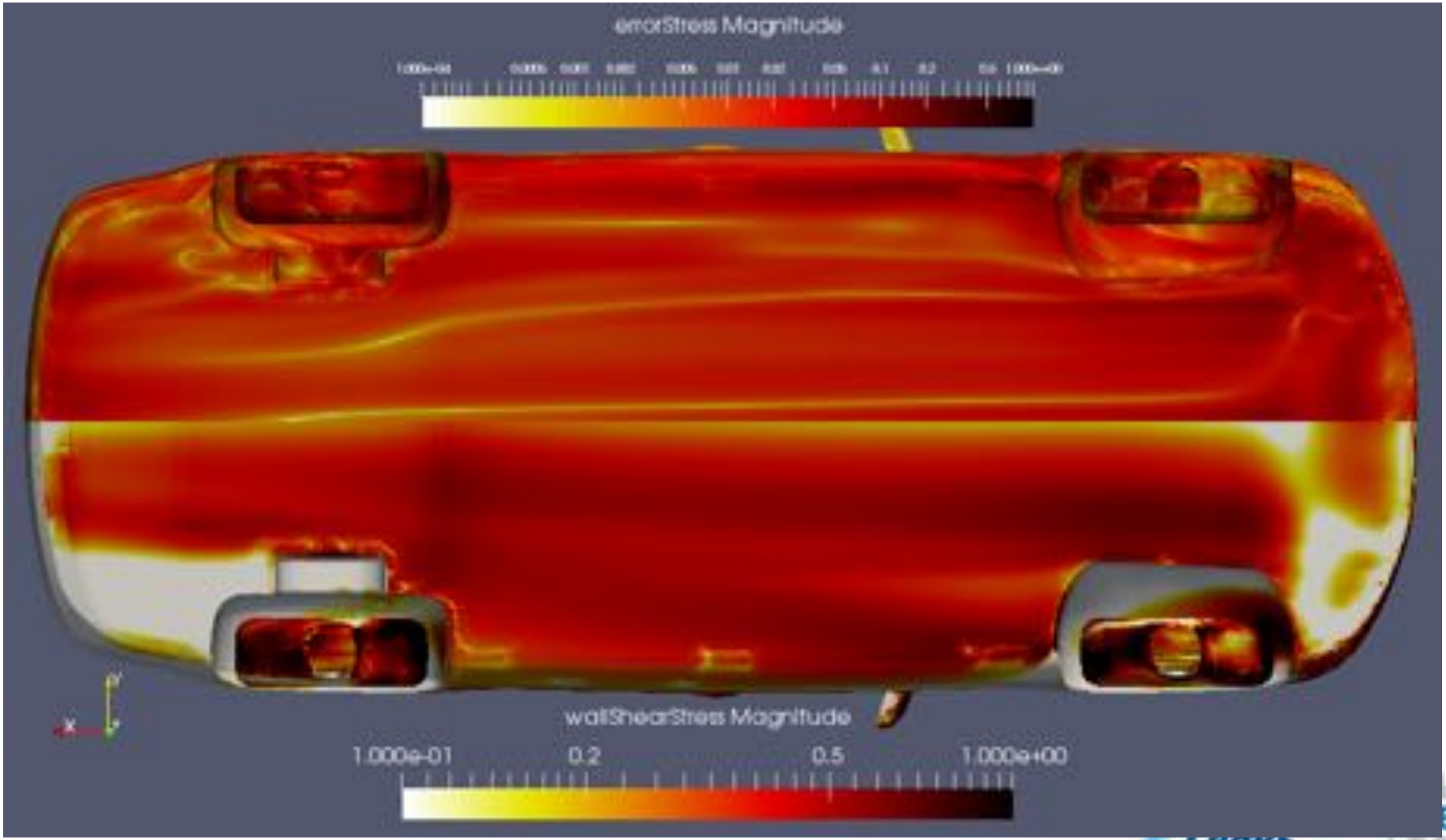
ezRB


errorStress Magnitude
wallShearStress Magnitude
1.000e-01
0.2
0.5
1.000e+00

CINECA SCAI SuperComputing Applications and Innovation

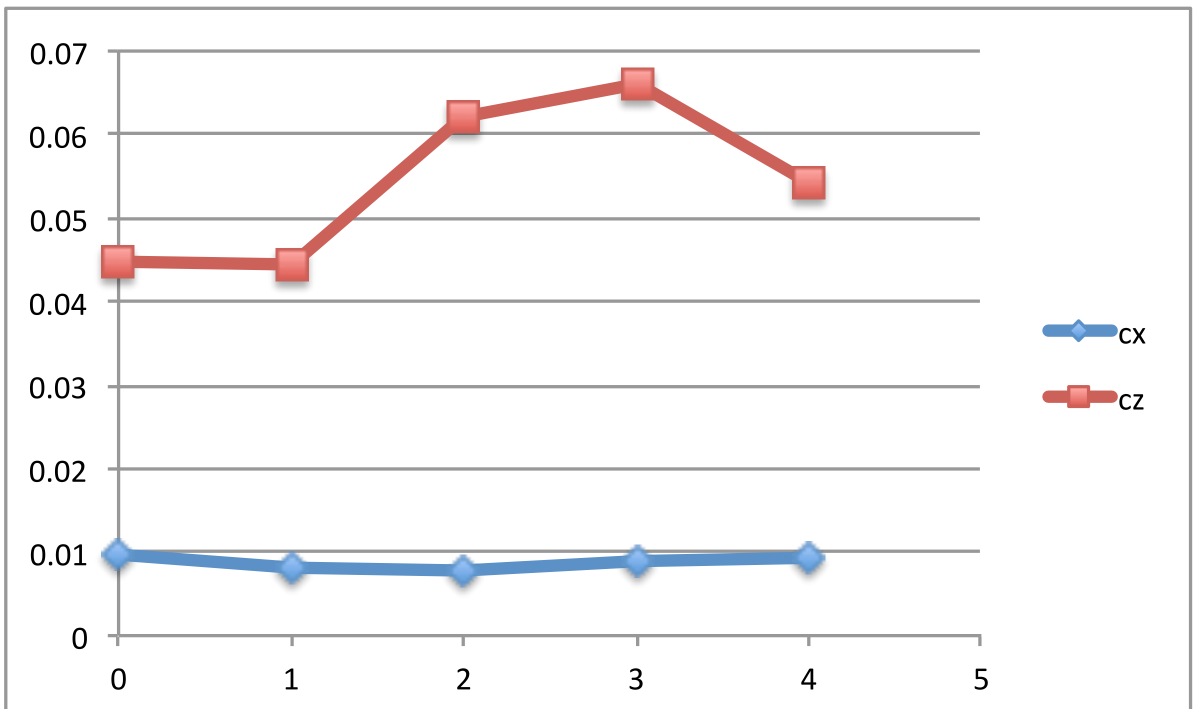
Cineca TRAINING High Performance Computing 2016

ezRB

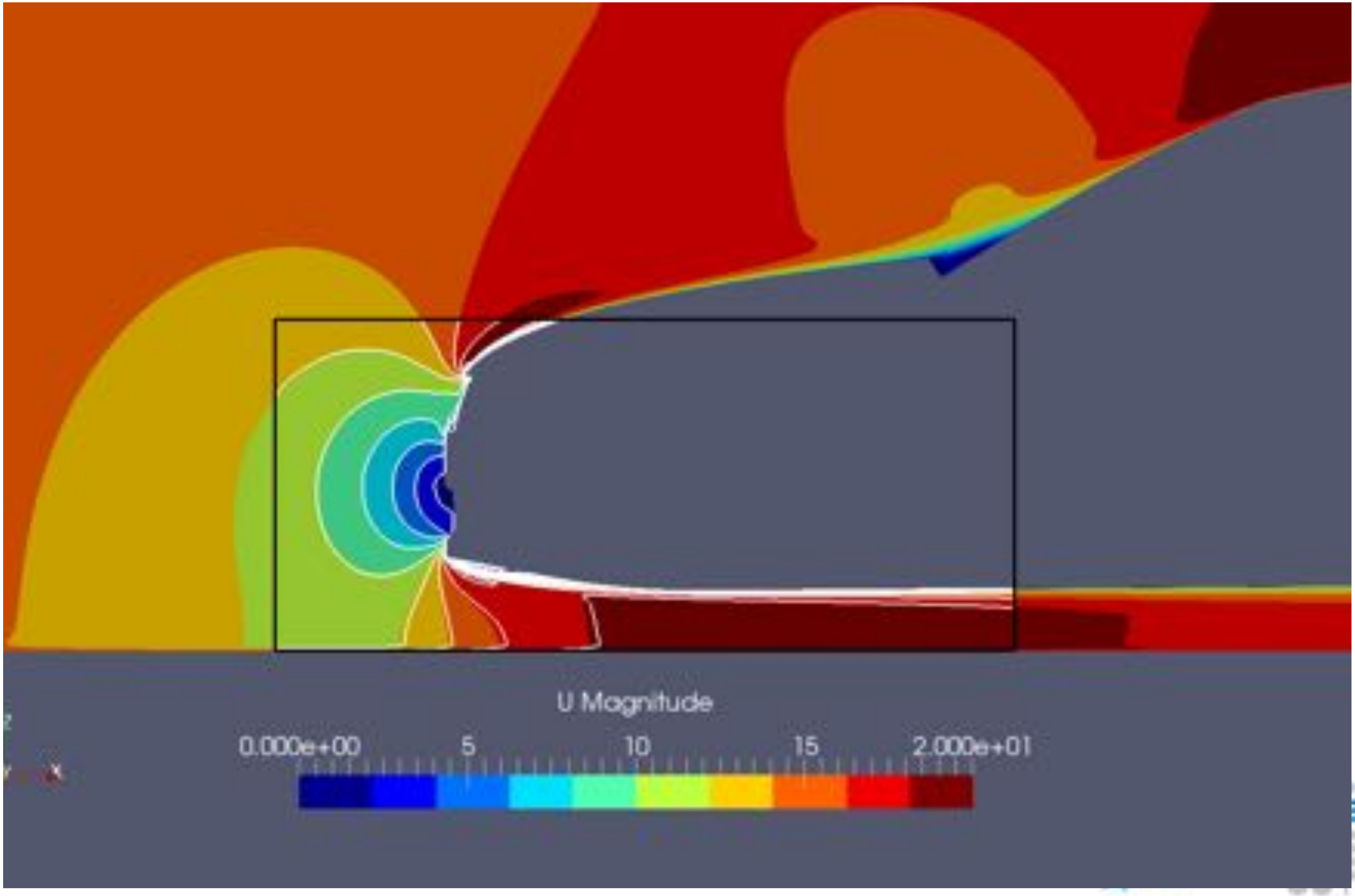


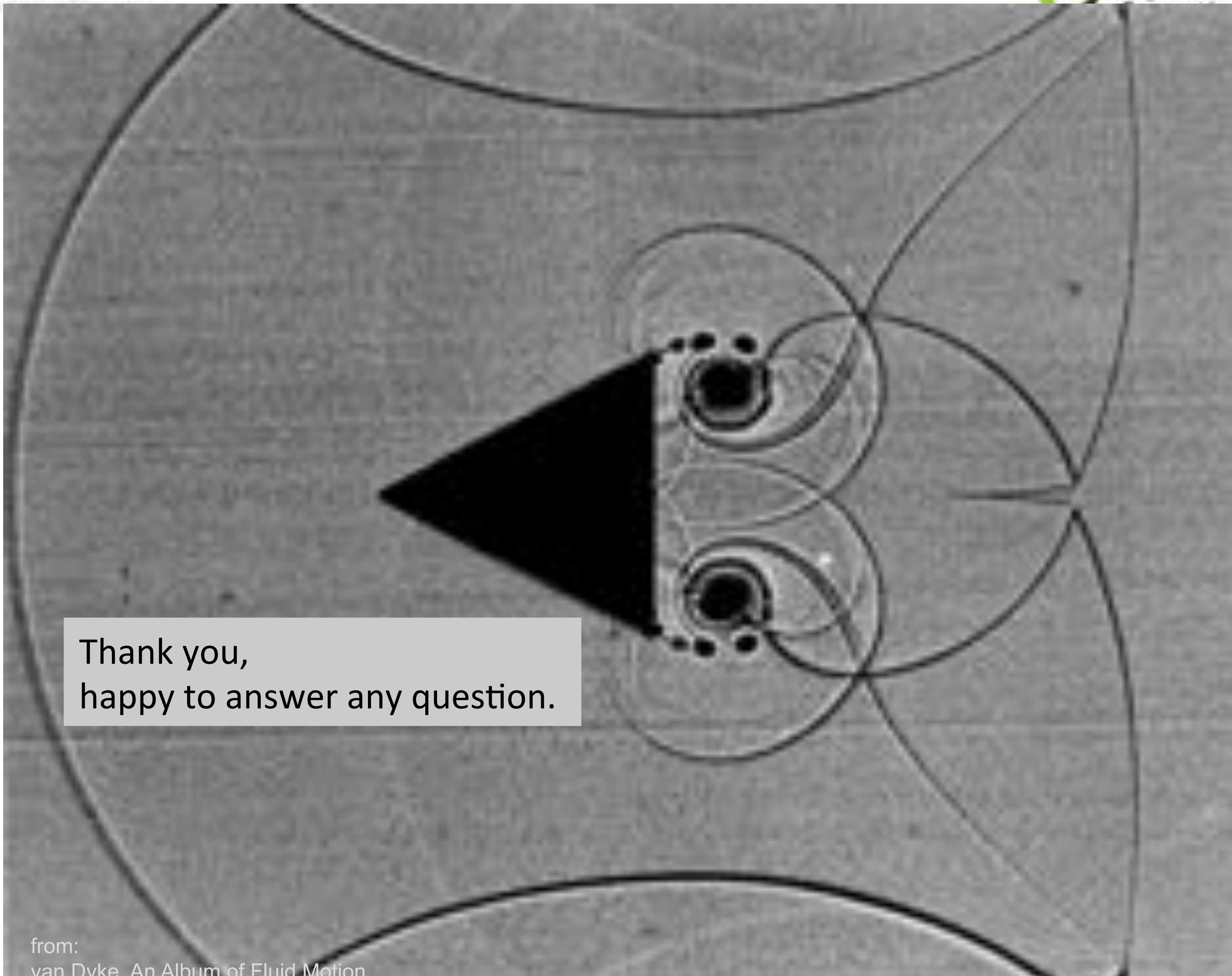
podFOAM

- recalculation of inner & outer flow field
- mean error over 4 random configurations out-of-DB
- speed-up O(50)



podFOAM





Thank you,
happy to answer any question.

from:
van Dyke, An Album of Fluid Motion