Fast Radial Basis Functions for Engineering Applications

Prof. Marco Evangelos Biancolini – University of Rome “Tor Vergata”

Outline

- RBF background
- Fast RBF on HPC
- Engineering Applications
- Mesh morphing
- RBF Morph CAE workflow
- Conclusions

$$s(\boldsymbol{x}) = \sum_{i=1}^{N} \gamma_i \cdot \varphi\left(\left\|\boldsymbol{x} - \boldsymbol{x}_{k_i}\right\|\right) + h(\boldsymbol{x})$$

Radial Basis Functions background

- **Radial Basis Functions** (RBF) were introduced as interpolators of scattered data in sixties. Usually the interpolation is comprised of:
  - A sum of weighted radial interactions
  - A polynomial correction
- RBF are commonly used to interpolate a **scalar function** defined in a multi-dimensional space ($\mathbb{R}^n \rightarrow \mathbb{R}$)

$$s(\boldsymbol{x}) = \sum_{i=1}^{N} \gamma_i \cdot \varphi\left(\left\|\boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{k}_i}\right\|\right) + h(\boldsymbol{x})$$

Source points

Evaluation point

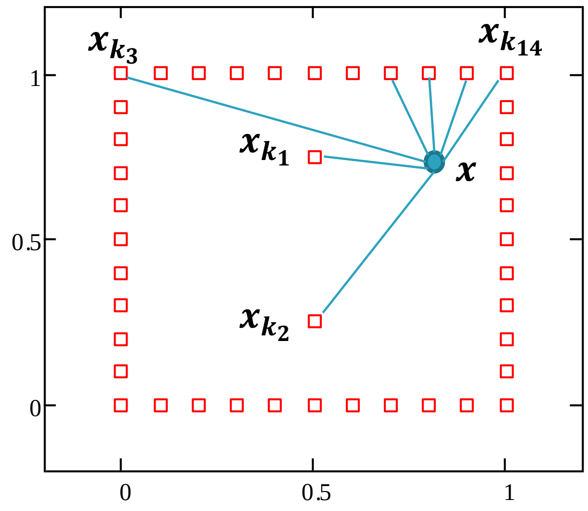
Polynomial

Radial interaction function

Weights of sources

Scalar function at evaluation point

Radial Basis Functions background

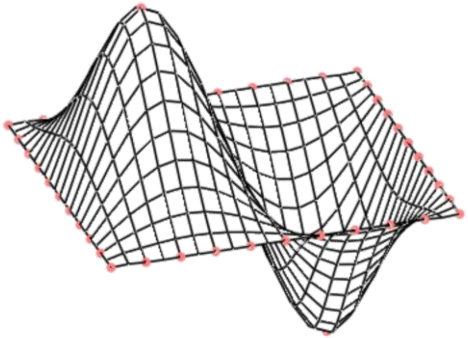


$$s(\boldsymbol{x}) = \sum_{i=1}^{N} \gamma_i \cdot \varphi\left(\left\|\boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{k}_i}\right\|\right) + h(\boldsymbol{x})$$

- Source points
- Evaluation point
- Polynomial
- Radial interaction function
- Weights of sources
- Scalar function at evaluation point

Radial Basis Functions background

$$s(\boldsymbol{x}) = \sum_{i=1}^{N} \gamma_i \cdot \varphi\left(\left\|\boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{k}_i}\right\|\right) + h(\boldsymbol{x})$$



Source points

Evaluation point

Polynomial

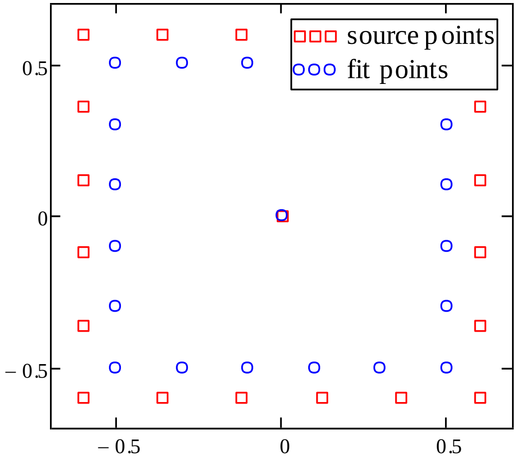
Radial interaction function

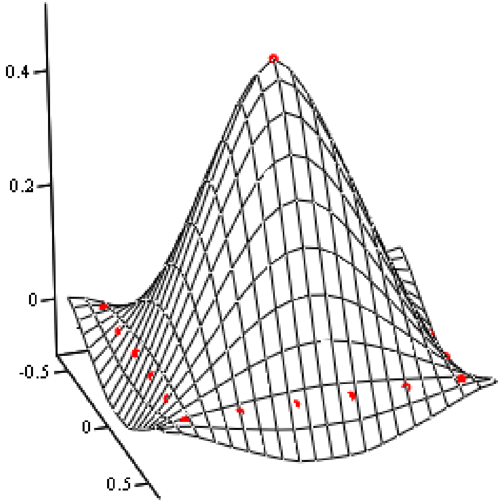
Weights of sources

Scalar function at evaluation point

Radial Basis Functions background

- The **weights of the RBF** are computed using regression/interpolation:
  - From scalar values at source points
  - From scalar values at fit points
- RBF fit (known as RBF training):
  - Solving a linear system (interpolation)
  - Using Least Squares



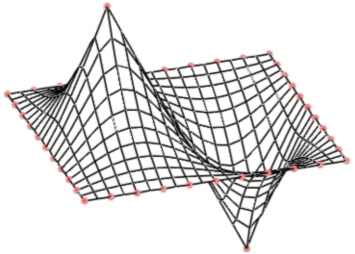


Radial Basis Functions background

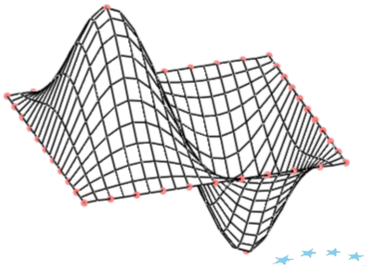
- RBF with **global support**
  - Far field interactions
  - Dense system of equations to be solved
- RBF with **compact support**
  - Local interactions
  - Sparse systems of equations to be solved

RBF with global support	$\varphi(r)$
Spline type (R_n)	$r^n, n\ odd$
Thin plate spline (TPS_n)	$r^n \log(r), n\ even$
Multiquadric (MQ)	$\sqrt{1+r^2}$
Inverse multiquadric (IMQ)	$\frac{1}{\sqrt{1+r^2}}$
Inverse quadratic (IQ)	$\frac{1}{1+r^2}$
Gaussian (GS)	e^{-r^2}
RBF with compact support	$\varphi(r) = f(\xi), \xi \le 1, \xi = \frac{r}{R_{sup}}$
Wendland C^0 (C0)	$(1-\xi)^2$
Wendland C^2 (C2)	$(1-\xi)^4(4\xi+1)$
Wendland C^4 (C4)	$(1-\xi)^6\left(\frac{35}{3}\xi^2+6\xi+1\right)$

$\varphi(r) = r$



$\varphi(r) = r^3$



Radial Basis Functions background

- Scalar
  Function values g_{s_i} known at sources $\boldsymbol{x}_{s_i}$
- Orthogonality condition
- Linear polynomial

$$s(\boldsymbol{x}_{s_i}) = g_{s_i}, 1 \leq i \leq N$$

$$\sum_{i=1}^{N} \gamma_i p(\boldsymbol{x}_{s_i}) = 0$$

$$h(\boldsymbol{x}) = \beta_1 + \beta_2 x + \beta_3 y + \beta_4 z$$

$$\sum_{i=1}^{N} \gamma_i = \sum_{i=1}^{N} \gamma_i x_{k_i} = \sum_{i=1}^{N} \gamma_i y_{k_i} = \sum_{i=1}^{N} \gamma_i z_{k_i} = 0$$

Radial Basis Functions background

- Linear system to be solved for the computation of unknown coefficients

$$\begin{pmatrix} \boldsymbol{M} & \boldsymbol{P_s} \\ \boldsymbol{P_s^T} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \boldsymbol{g_s} \\ 0 \end{pmatrix}$$

- System matrix
  - Constraint matrix $\boldsymbol{P_s}$
  - Interpolation matrix $\boldsymbol{M}$

$$\boldsymbol{P_s} = \begin{pmatrix} 1 & x_{s_1} & y_{s_1} & z_{s_1} \\ 1 & x_{s_2} & y_{s_2} & z_{s_2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{s_N} & y_{s_N} & z_{s_N} \end{pmatrix}$$

$$M_{ij} = \varphi\left(\left\|\boldsymbol{x}_{s_i} - \boldsymbol{x}_{s_j}\right\|\right), 1 \le i \le N, 1 \le j \le N$$

Fast Radial Basis Functions on HPC

Requirements and limits

- **Engineering applications** often requires large RBF clouds to be fitted (up to **millions**)
- Evaluation cloud could be even larger (up to **billions**)
- Maximum flexibility is given by direct method that **scales as N^3**
- Direct method is limited to about **10.000 points** (memory usage, time to fit)

Challenges

- The industry requires a **two order of magnitude** increment in the size of the cloud
- A further order of magnitude (billion size cloud fitting) as a reasonable target for future roadmaps
- **FastRBF** are required to fill the gap and make RBF attractive for industrial applications

Fast Radial Basis Functions on HPC

- **Options to boost performances**
  - Reducing the size of the cloud with error control (efficient algorithms for **data decimation**)
  - Decomposing the large problem in several overlapping small problems (**Partition Of Unity** POU)
  - Usage of iterative solvers for the linear system (**FGP, GMRES**)
  - Usage of compact supported RBF (**sparse** solver)
  - Approximate the RBF (**Fast Multipole Method** FMM)
  - Distributing the calculation on **multiple cores** (CPU and GPU)

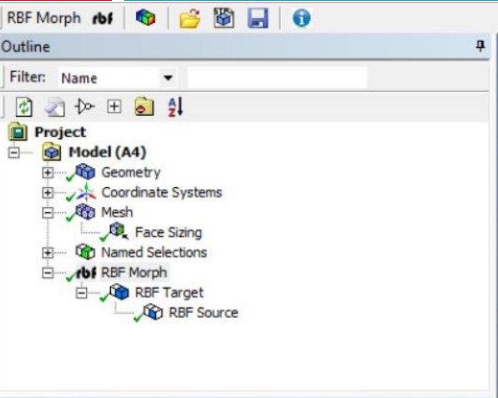
Fast Radial Basis Functions on HPC

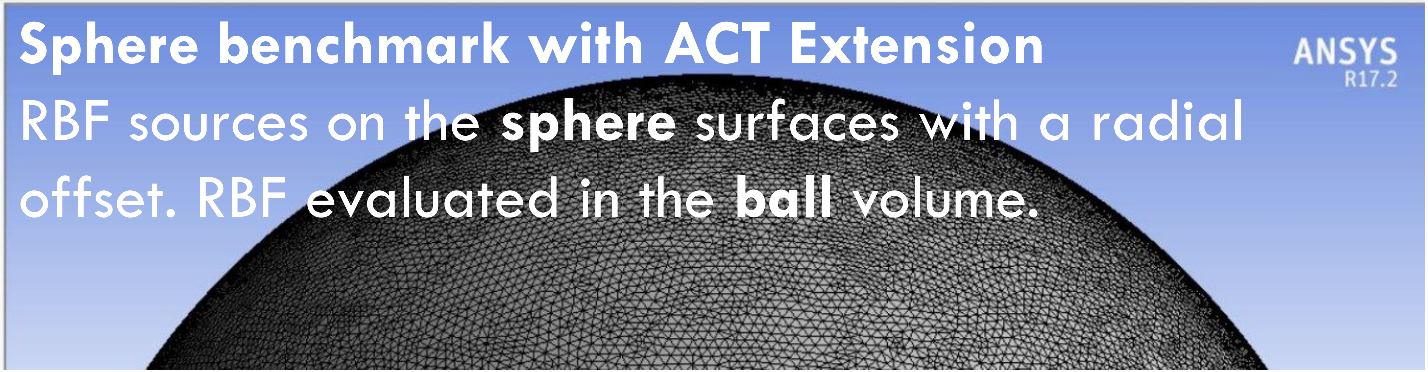
- **Fast RBF strategies**
  - The approach depends on the **RBF functions**
  - The approach depends on the **RBF** space **dimension** (2-d, 3-d, n-d)
  - It's problem dependent (are far field interactions required?)
  - **Parallelism** can be easily exploited at evaluation stage
  - Parallelism during **RBF training** is not obvious
  - POU methods can be quickly parallelised (distributing local problems)
  - Optimisation are **OS specific** and **hardware specific**

Fast Radial Basis Functions on HPC

- **RBF Morph software solver**
  - **Fluent Add On** and **Stand Alone** software feature a fast iterative solver with FMM (available for bi-harmonic kernel in 3d) + a custom POU (proprietary Local Correction Method LCM technology) with shared memory (OpenMP) parallelism (FORTRAN + C)
  - **LCM technology** can be enabled with generic kernel in 3d (reduced performances increased flexibility) (C)
  - **ACT Extension** features a fast iterative solver with advanced parallelism on CPU (OpenMP + SSE) and on GPU (CUDA) (C++)
  - RBF evaluation can be easily distributed (shared memory and distributed memory CPU, GPU)

Fast Radial Basis Functions on HPC





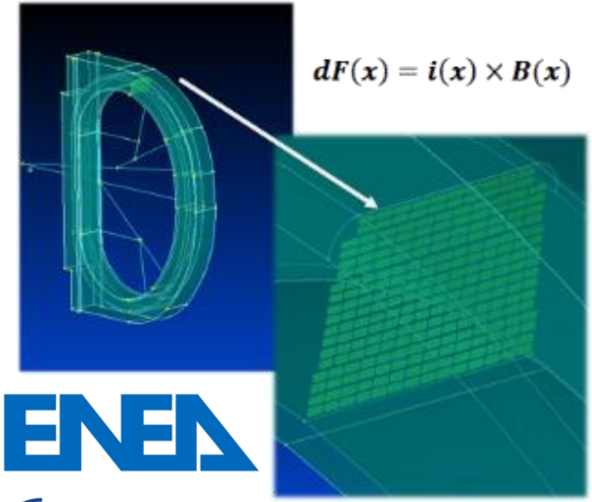
Sphere benchmark with ACT Extension

RBF sources on the **sphere** surfaces with a radial offset. RBF evaluated in the **ball** volume.

			2017 RBF solver		2015 RBF solver		
Spacing	Sources	Targets	min	sec	min	sec	speed-up
0,025	85434	205866	0	38	4	23	6,92
0,02	132598	323464	1	16	10	35	8,36
0,015	234522	574960	4	30	37	50	8,41

Engineering Applications

- **RBF mapping (DEMO+DTT)**
  - Electromagnetic loads transferred to the structural model as magnetic field
  - Can be transferred as Force density



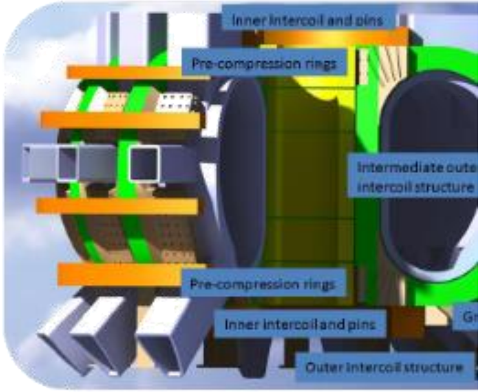
$$dF(x) = i(x) \times B(x)$$

RBF Mapper EM/FEA

$$B_x(x) = \sum_{i=1}^{N} \gamma_i^x \varphi(\|x - x_{k_i}\|) + \beta_1^x + \beta_2^x x + \beta_3^x y$$

$$B_y(x) = \sum_{i=1}^{N} \gamma_i^y \varphi(\|x - x_{k_i}\|) + \beta_1^y + \beta_2^y x + \beta_3^y y$$

$$B_z(x) = \sum_{i=1}^{N} \gamma_i^z \varphi(\|x - x_{k_i}\|) + \beta_1^z + \beta_2^z x + \beta_3^z y$$

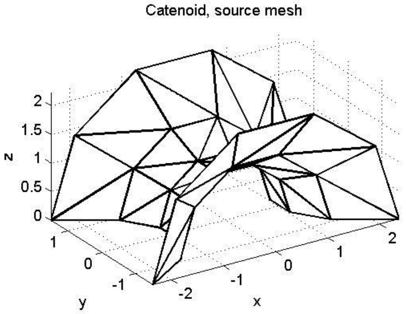


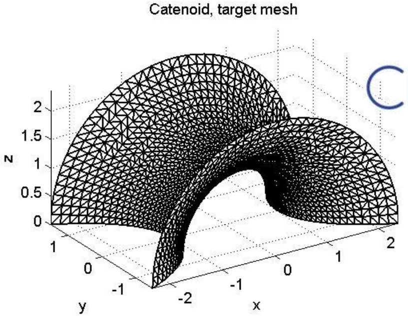
ENEA

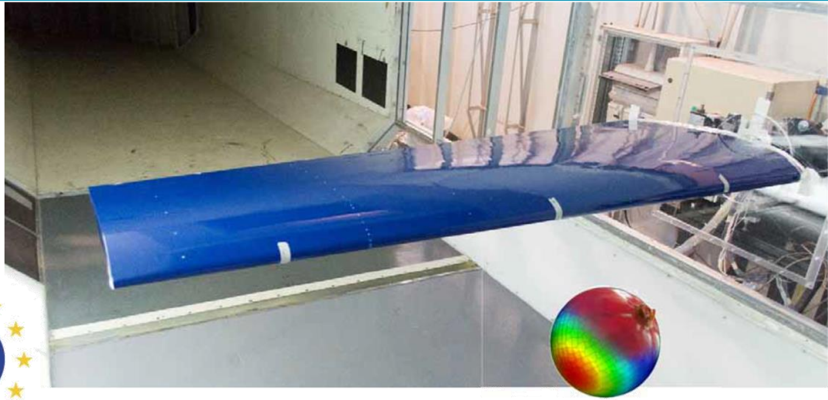
EUROfusion

Engineering Applications

- **RBF mapping (RIBES)**
  - Pressure field computed on surface (CFD) onto structure (FEA)
  - Temperature field mapped i the volume

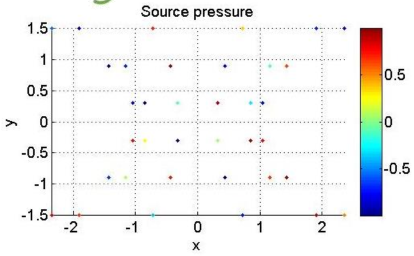


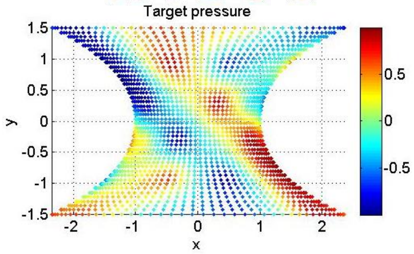




Clean Sky

R I B E S



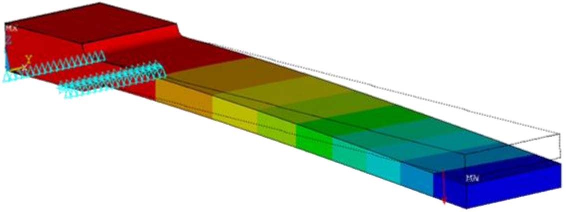


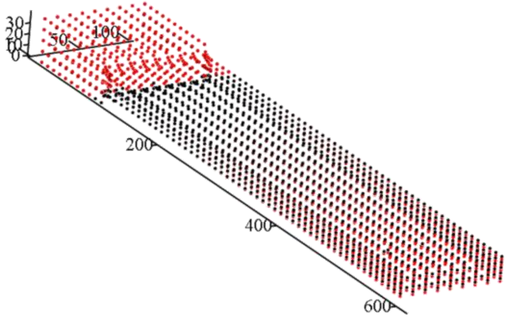
(rbf-morph)™

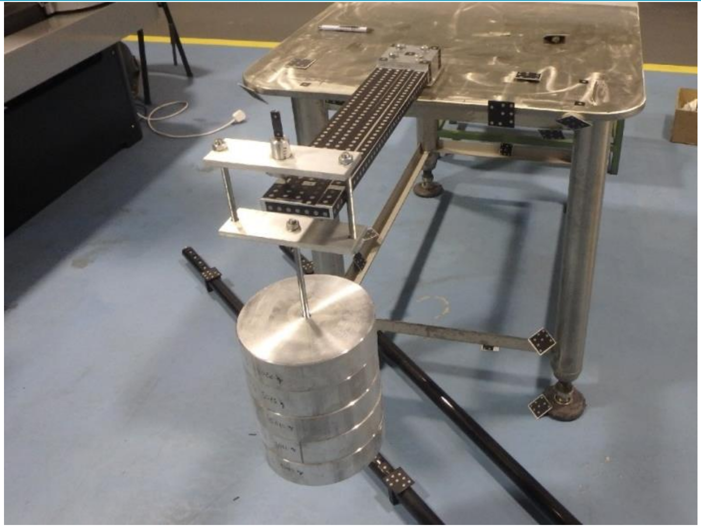
PRACE

Engineering Applications

- **Compensation of metrological data (RBF4METRO)**
  - Environment modelled using FEA
  - Acquired points compensated using RBF



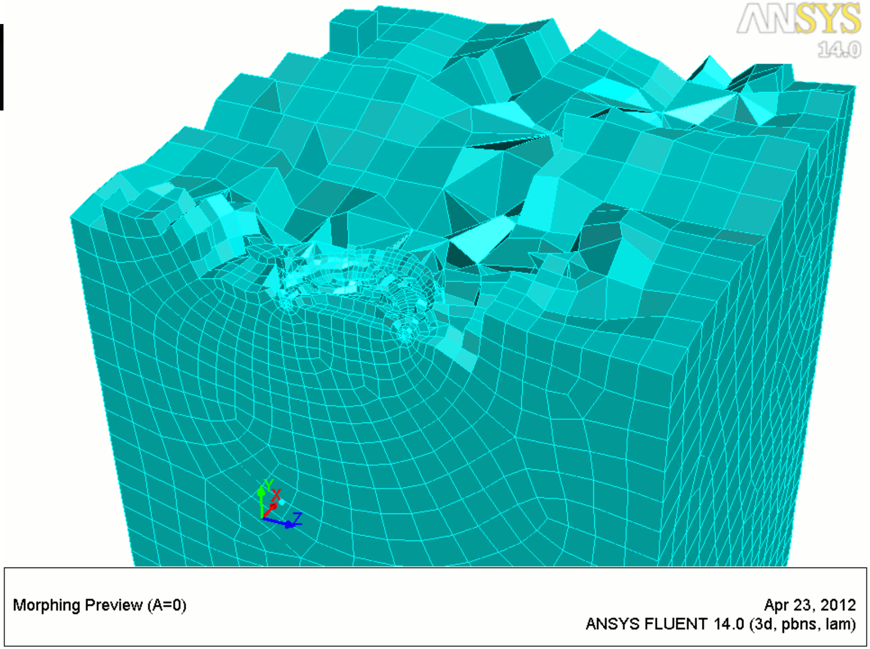




FUSION FOR ENERGY

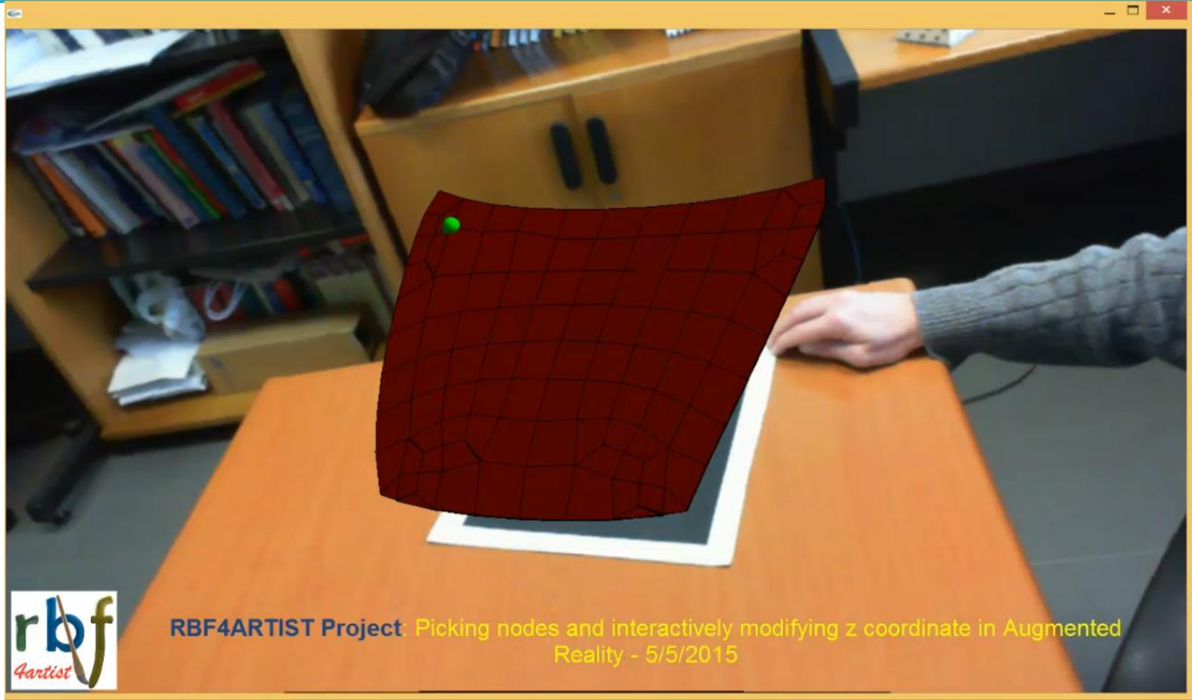
Engineering Applications

- **Crack propagation (RBF4CRACKS)**
  - Local driving force computed using FEA
  - RBF to interpolate driving force and morph the FEA mesh



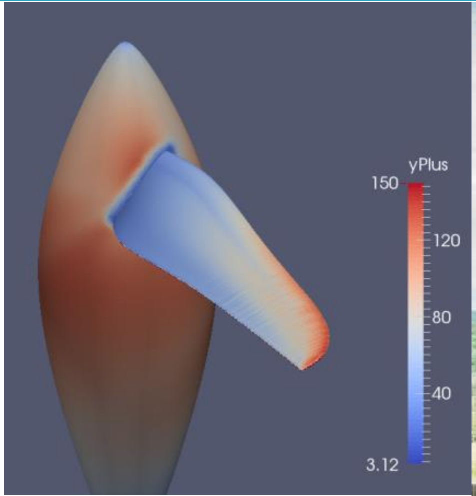
Engineering Applications

- **Interactive sculpting (RBF4ARTIST)**
  - Augmented reality
  - Force feedback system
  - Real time reactivity requires high performances!
  - youtu.be/74yjd7ZWcNk



Engineering Applications

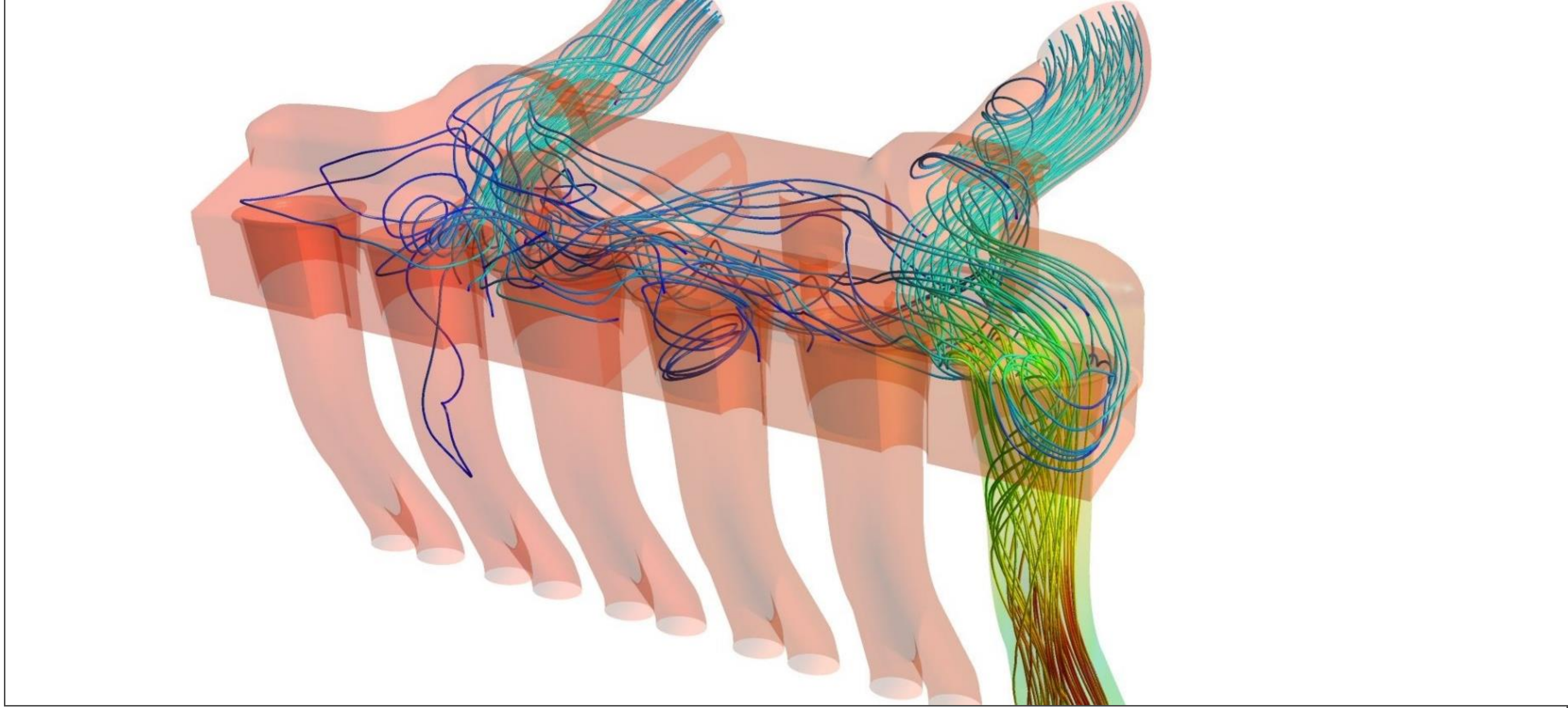
- **FSI optimisation (RBF4AERO now on FF2)**
  - Mesh morphing for shape parametrization of numerical grids
  - FSI based on mapping and modal superposition
  - Optimisation run on the flexible model
  - www.rbf4aero.eu/
  - youtu.be/eThibFzEPNI
  - youtu.be/A0WPDyhIr8Q



RBF4AERO

(rbf-morph)™

PRACE

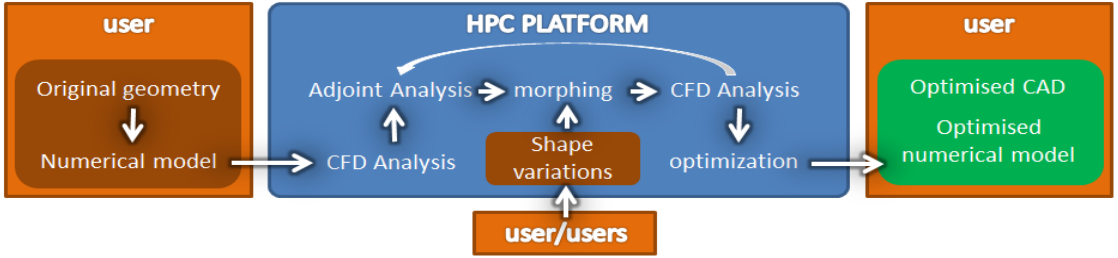


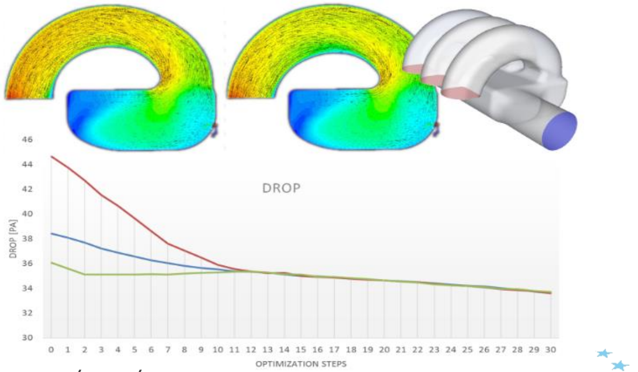
RBF Morph CAE workflow

Fortissimo experiment 515 - **Cloud-based Additive Manufacturing**

Fortissimo EU Project

- **F**actories **O**f the Future **R**esources, **T**echnology, **I**nfrastructure and **S**ervices for **SI**mulation and **MO**delling
- Our experiment: **"Virtual Automatic Rapid Prototyping Based on Fast Morphing on HPC Platforms"**
- **HSL srl, Trento; University of Rome "Tor Vergata"; CINECA**





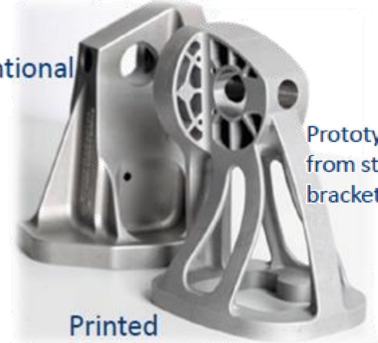
Motivation

- Fortissimo Call submission on January 2014
- Fortissimo **WP515** “Cloud based modelling for the 3-d printing of complex shapes” started on October 2014
- August 2015 **Lamborghini** on board as a first user of the method
- New service on the **Fortissimo Marketplace**

Concept

- 3d printing already in use for **one-off** projects
- Moving to **small production** lots looks reasonable (especially for top cars)
- Full exploitation of 3d printing potential requires **new CAE concepts**
- Shape optimisation based on **mesh morphing** (with parameters or without parameters, **adjoint**) could be a meaningful answers

conventional



Prototype of an optimized Airbus A380 bracket made from stainless steel powder, with conventional bracket behind

GE Unveils Additive Manufacturing Factory Plan

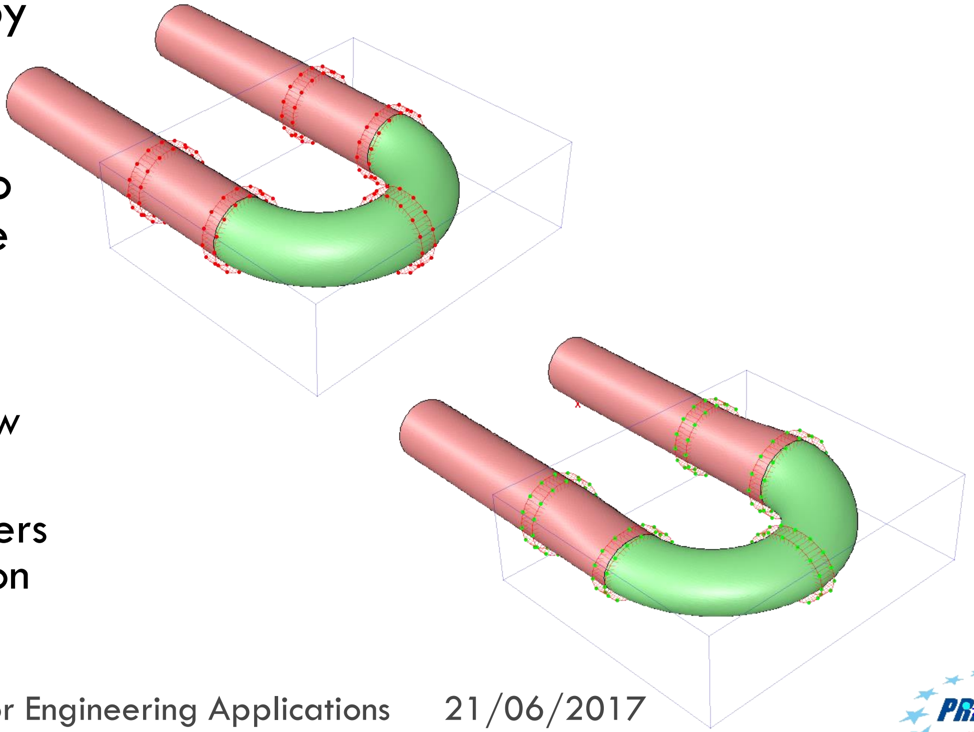
Guy Norris | *AWIN First* — *Jul 15, 2014*

FARNBOROUGH -- General Electric has revealed plans to develop the aerospace world's first large, dedicated additive manufacturing facility for jet engine parts in Auburn, Alabama.

40k nozzles/year by 2020 for
- Airbus A320neo
- Boeing 737 MAX
- Comac's C919

Shape parameterization strategy

- Geometric parameterization by **Mesh morphing**
- The principle is to take the control on a set of point and to transfer the deformation to the whole mesh
- Adjoint sensitivities
  - filtered and used to have "flow sculpted" shapes
  - derivatives of shape parameters for gradient based optimisation



Radial Basis Functions for mesh morphing

- **Radial Basis Functions** (RBF) can be used to drive mesh morphing (smoothing) from a list of source points and their displacements.
  - Surface shape changes (exact nodes control)
  - Volume mesh smoothing.
- RBF are recognized to be one of the **best mathematical tool** for mesh morphing.

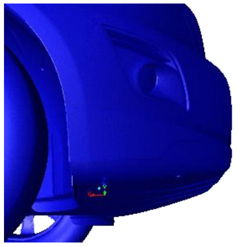
$$\begin{cases} s_x(\boldsymbol{x}) = \sum_{i=1}^{N} \gamma_i^x \varphi(\|\boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{s}_i}\|) + \beta_1^x + \beta_2^x x + \beta_3^x y + \beta_4^x z \\ s_y(\boldsymbol{x}) = \sum_{i=1}^{N} \gamma_i^y \varphi(\|\boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{s}_i}\|) + \beta_1^y + \beta_2^y x + \beta_3^y y + \beta_4^y z \\ s_z(\boldsymbol{x}) = \sum_{i=1}^{N} \gamma_i^z \varphi(\|\boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{s}_i}\|) + \beta_1^z + \beta_2^z x + \beta_3^z y + \beta_4^z z \end{cases}$$

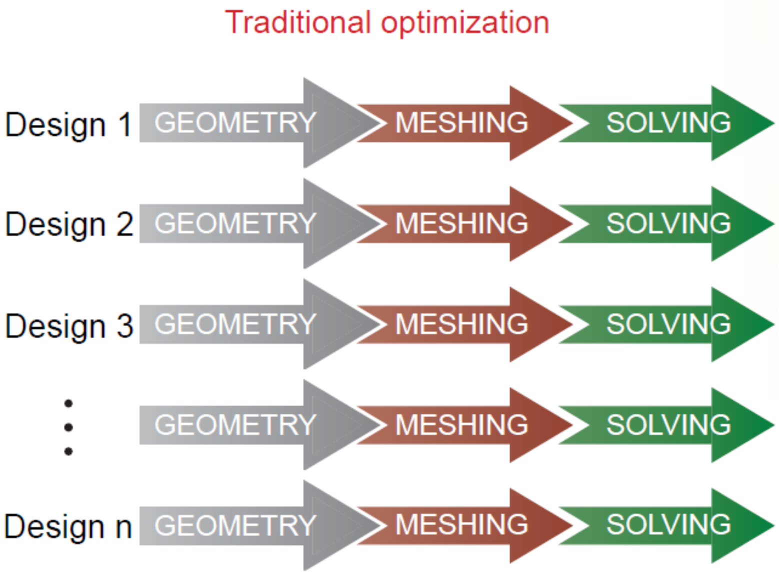
Radial Basis Functions for mesh morphing

$$\begin{cases} s_x(\boldsymbol{x}) = \sum_{i=1}^{N} \gamma_i^x \varphi(\|\boldsymbol{x} - \boldsymbol{x}_{s_i}\|) + \beta_1^x + \beta_2^x x + \beta_3^x y + \beta_4^x z \\ s_y(\boldsymbol{x}) = \sum_{i=1}^{N} \gamma_i^y \varphi(\|\boldsymbol{x} - \boldsymbol{x}_{s_i}\|) + \beta_1^y + \beta_2^y x + \beta_3^y y + \beta_4^y z \\ s_z(\boldsymbol{x}) = \sum_{i=1}^{N} \gamma_i^z \varphi(\|\boldsymbol{x} - \boldsymbol{x}_{s_i}\|) + \beta_1^z + \beta_2^z x + \beta_3^z y + \beta_4^z z \end{cases}$$

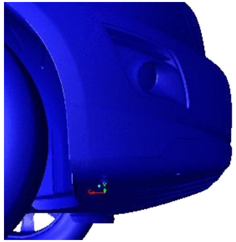
- Main advantages
  - No re-meshing
  - Can handle any kind of mesh
  - Can be integrated in the CFD solver
  - Highly parallelizable
  - Robust process
- Main disadvantages
  - Computationally expensive (HPC for large grids)
  - Back to CAD procedure required

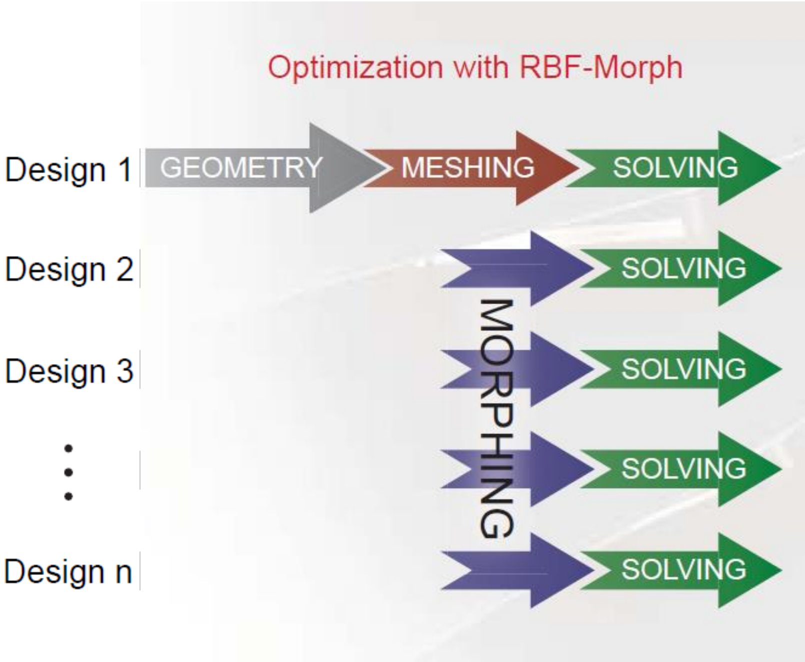
Example – 50:50:50 procedure





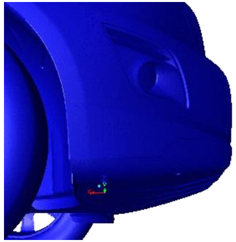
Example – 50:50:50 procedure

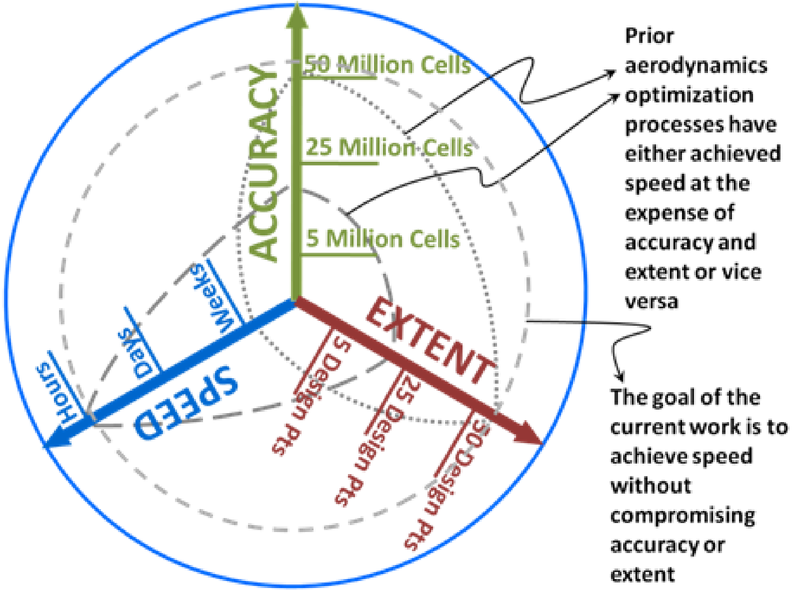




(rbf-morph)™

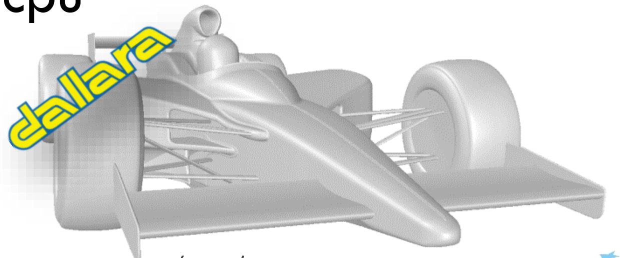
Example – 50:50:50 procedure



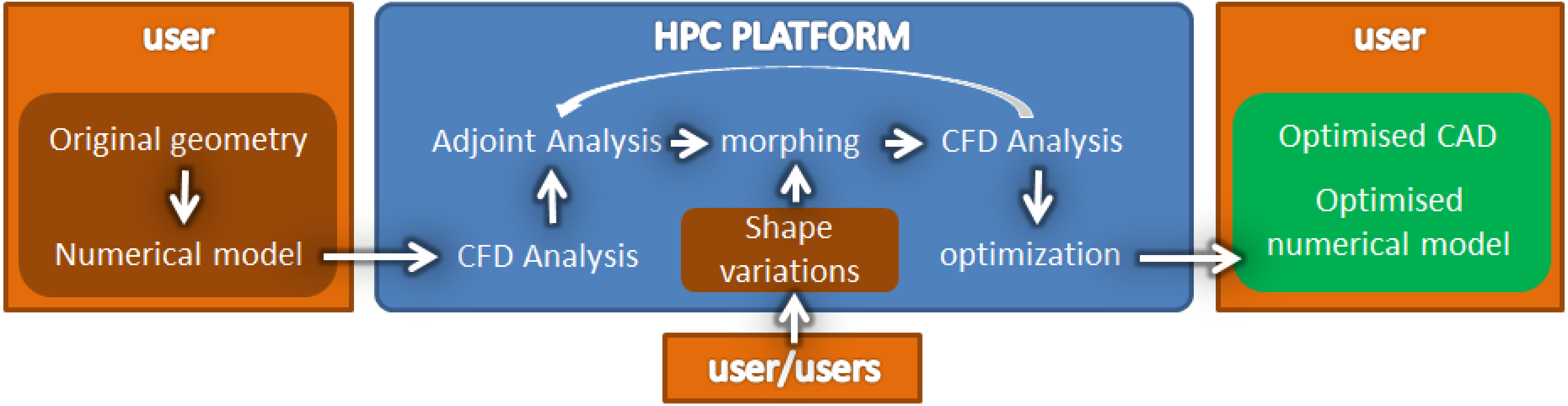


HPC performances

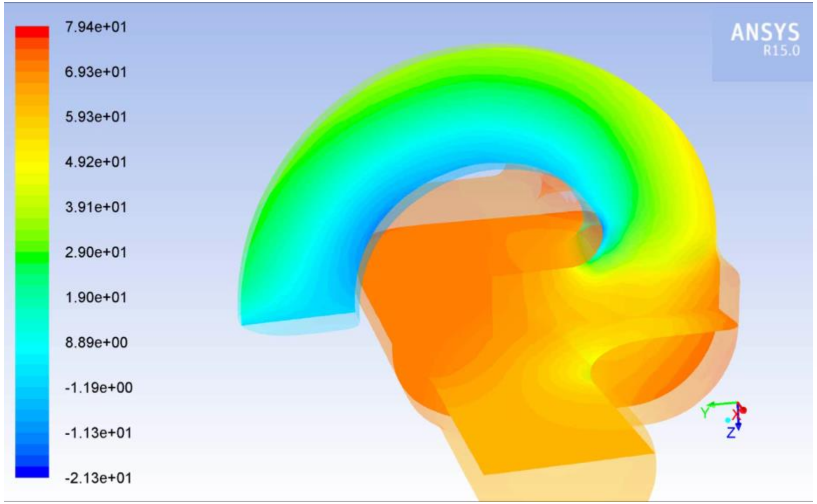
- 14 mill. cells, 60.000 points, PC 4 cpu 2.67 GHz
  - RBF training: 53 sec. (serial)
  - morphing: 3.5 min.
- 50 mill. cells, 30.000 points, HPC 140 cpu
  - RBF training : 25 sec. (serial)
  - morphing : 1.5 min.
- 100 mill. cells, 200.000 points, HPC 256 cpu
  - RBF training : 25 min.
  - morphing : 5 min.



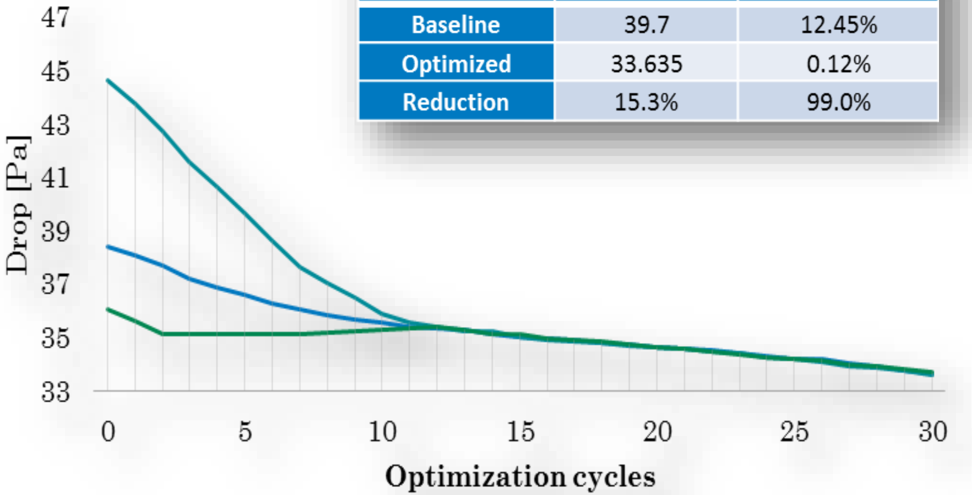
Fortissimo WP515



Fortissimo Benchmark

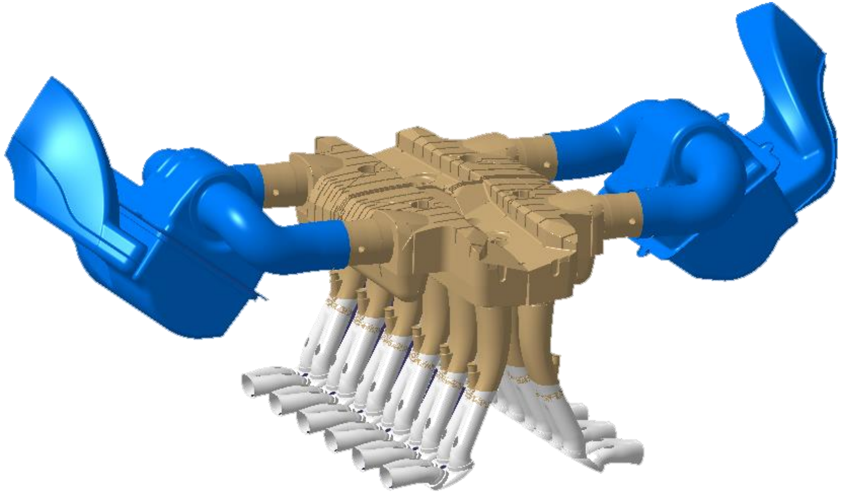


	Mean pressure Drop [Pa]	Unbalance
Baseline	39.7	12.45%
Optimized	33.635	0.12%
Reduction	15.3%	99.0%



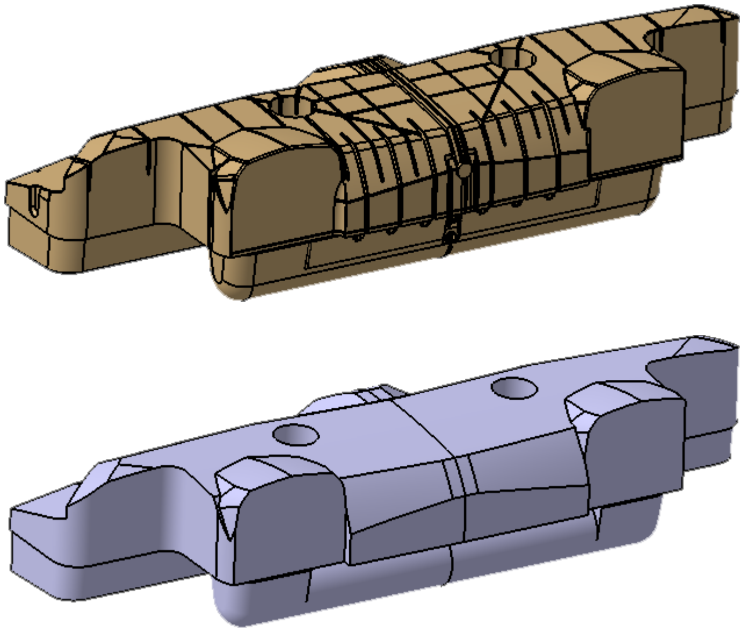
Fortissimo Case study

- Airbox of the **Lamborghini Aventador**
- Detailed CFD analyses of intake runners pressure drops (compressible!)
- Define a new shape for charging efficiency maximization



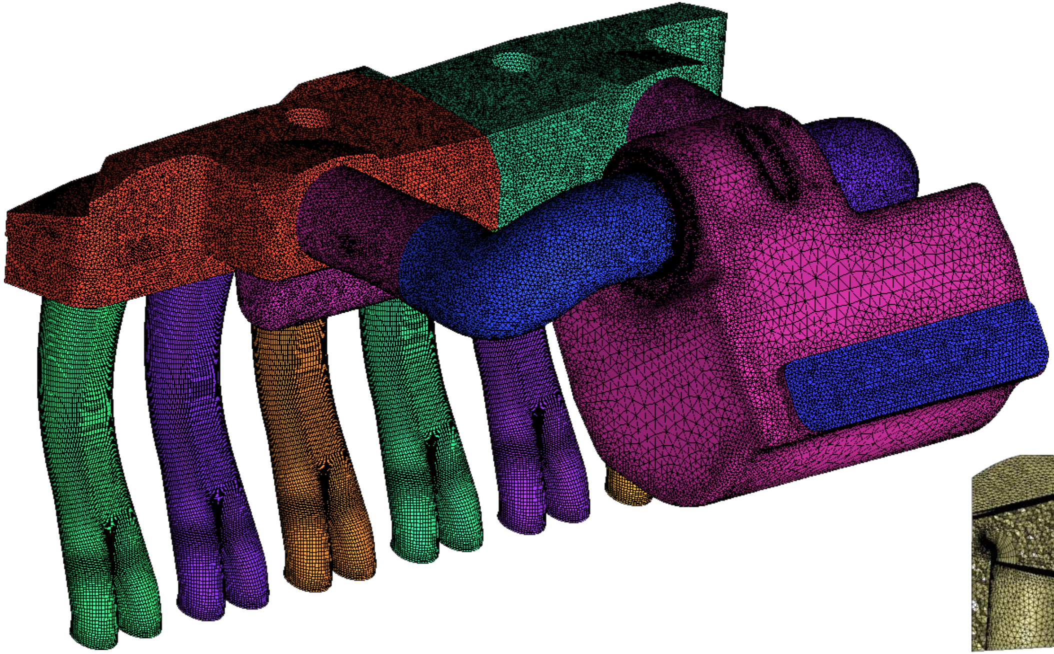
CAD model preparation

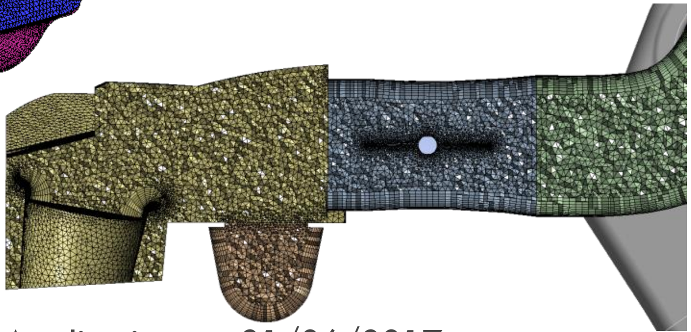
- CAD model rebuilt to:
  - simplify the geometry eliminating reinforcements (reduced mesh dimension)
  - clean the surfaces (steps, gaps, holes) to be suitable for CFD



Mesh assembly

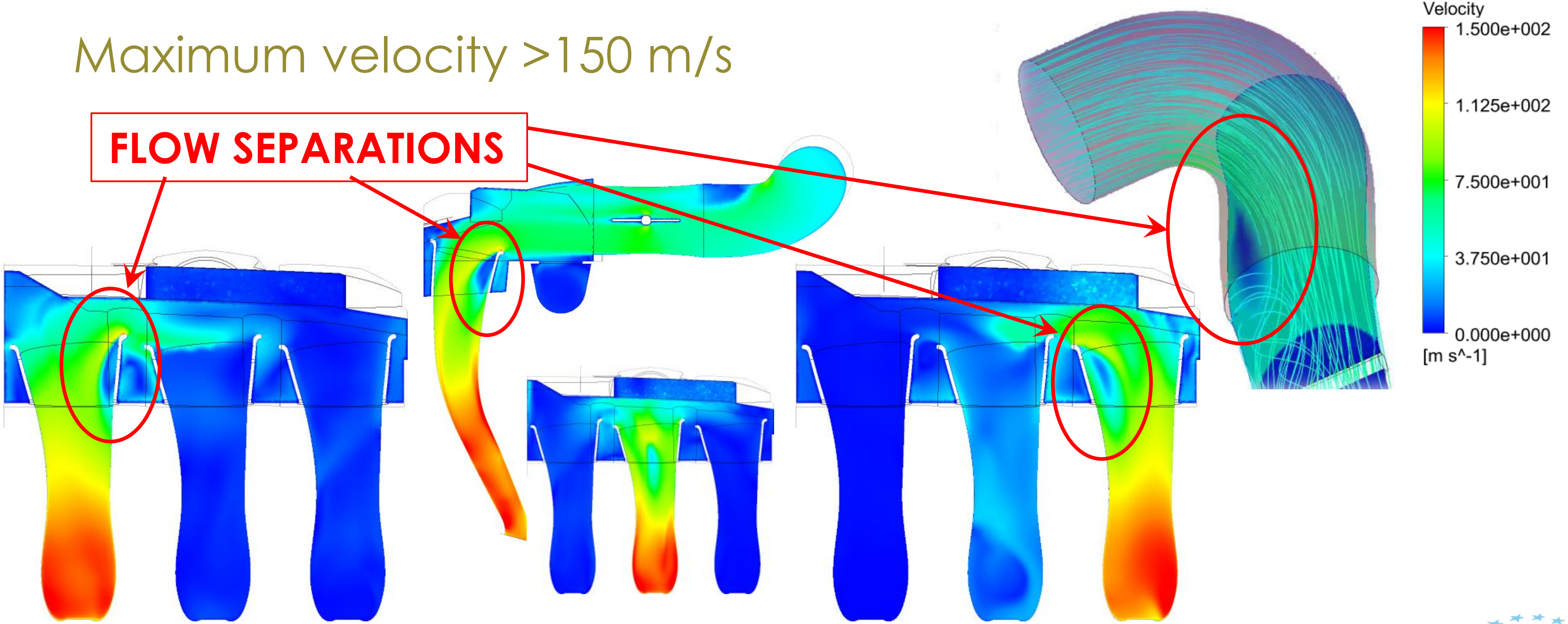
Maximum dimension
9.5 millions





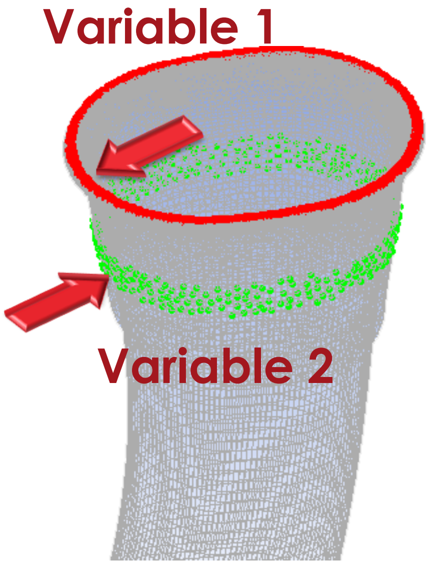
Critical regions

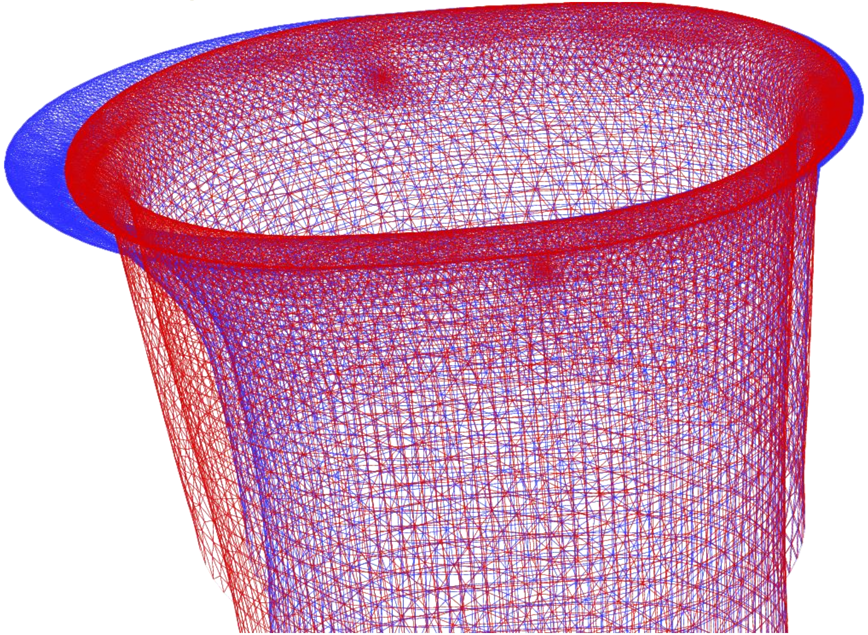
Maximum velocity >150 m/s



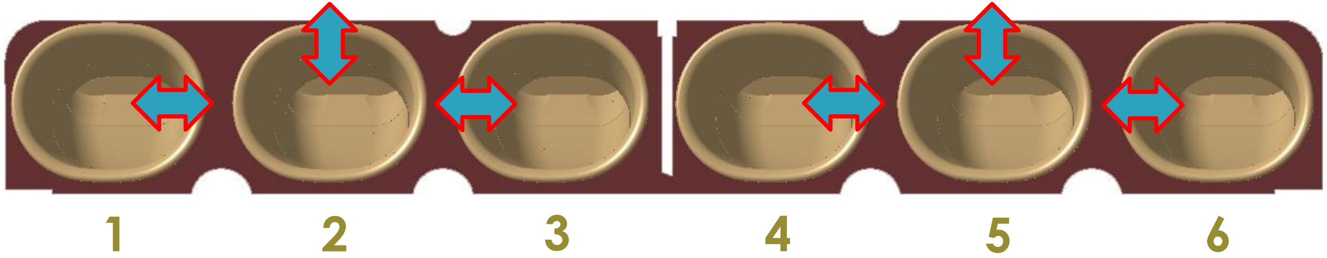
RBF setup

Two shape modifiers for each runner acting in the region of separation



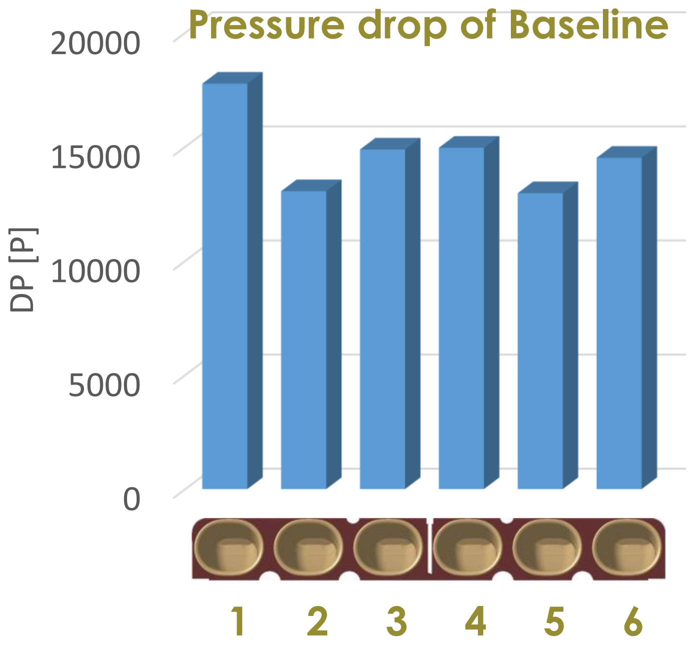


RBF setup for all runners

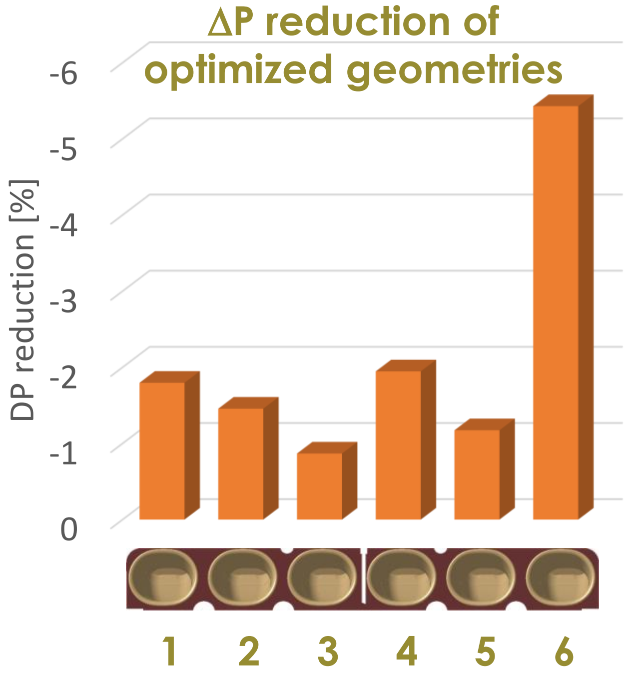


Locations of morphing actions

Results



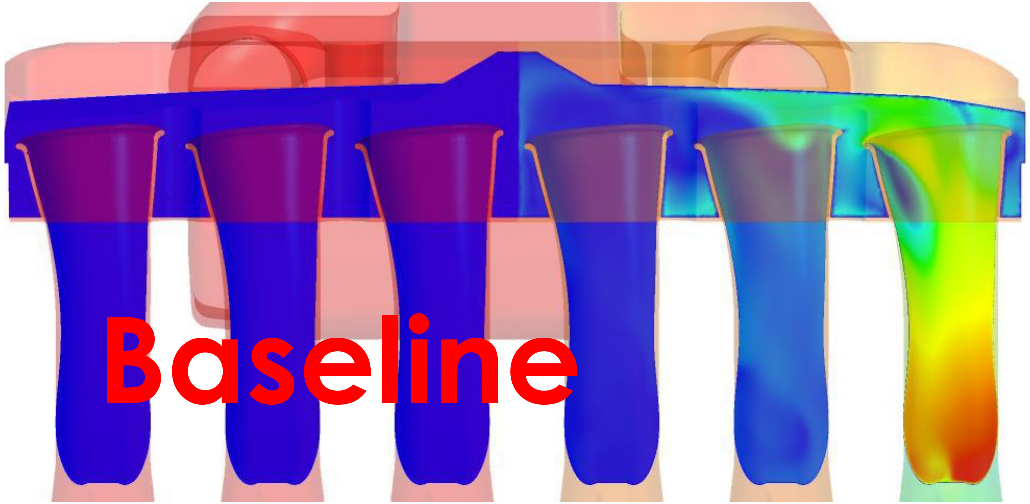
Pressure drop of Baseline
20000
15000
10000
5000
0
DP [P]
1
2
3
4
5
6



ΔP reduction of
optimized geometries
-6
-5
-4
-3
-2
-1
0
DP reduction [%]
1
2
3
4
5
6

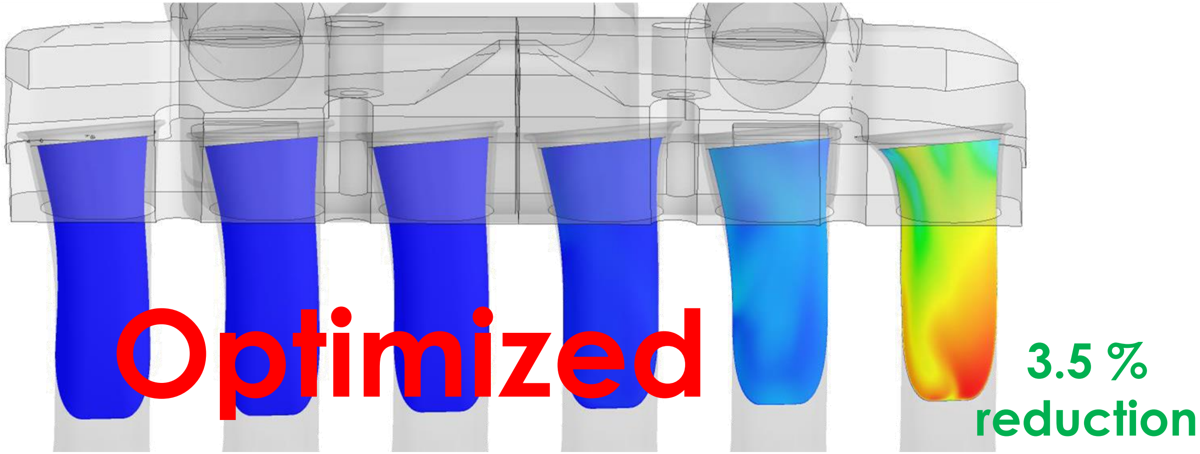
Results

DP = 15044.8 P



Results

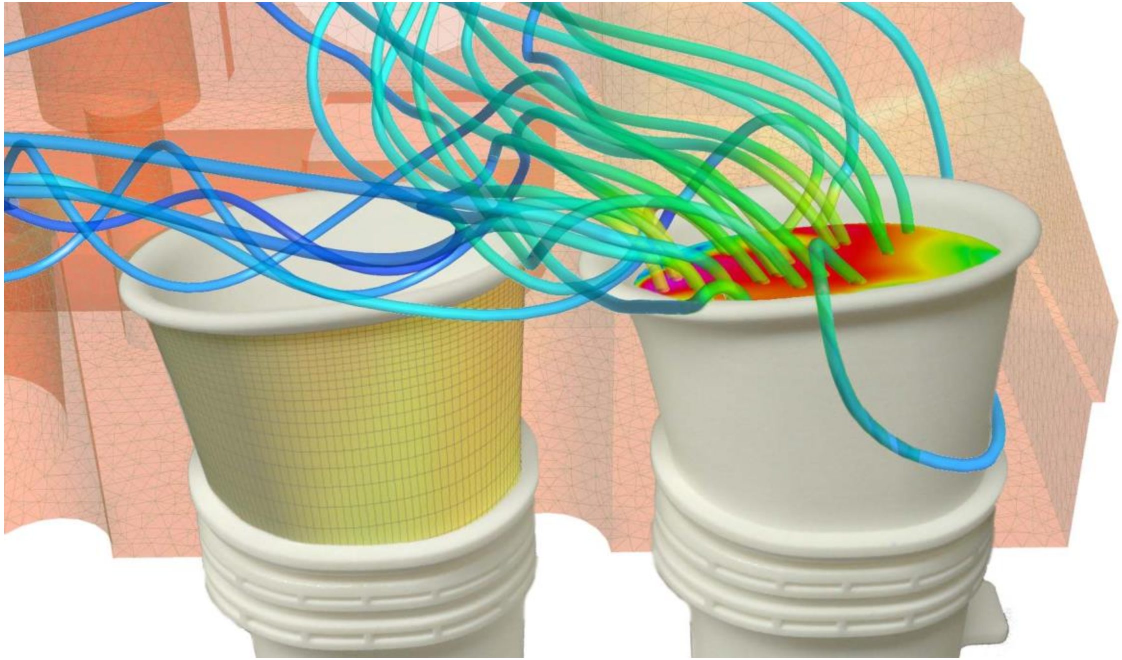
DP = 14584.8 P



New 3D-printed part


New 3D-printed part

New 3D-printed part



New 3D-printed part

Conclusions

- **Radial Basis Functions** have a great potential to be more and more adopted in **Engineering Applications**
- **Fast RBF** are a paramount to tackle industrial cases. Users are hungry of performances.
- **RBF Morph** software (first industrial mesh morphing tool based on RBF) is representative of the **industrial needs**
- Various **engineering applications** demonstrated (ranging in research/industrial active projects)
- A detailed example of a cloud HPC workflow of **Fortissimo** fully demonstrated

(rbf-morph)™

goo.gl/1svYd

twitter.com/RBFMorph

linkedin.com/company/rbf-morph

youtube.com/user/RbfMorph

rbf-morph.com

Many thanks for your kind attention!

biancolini@ing.uniroma2.it