(rbf-morph)™

Welcome to the World of Fast Morphing!

Virtual Automatic Rapid Prototyping on HPC Platforms + Fast Morphing tutorial

C. Groth, M. E. Biancolini

University of Rome Tor Vergata

corrado.groth@uniroma2.it

info@rbf-morph.com

Outline

- RBF Morph software line
- Mesh morphing with RBF
- Accelerating the solver
- How it works
- Ongoing RBF researches
- Test Case: video tutorial
- Industrial applications

Morphing & Smoothing

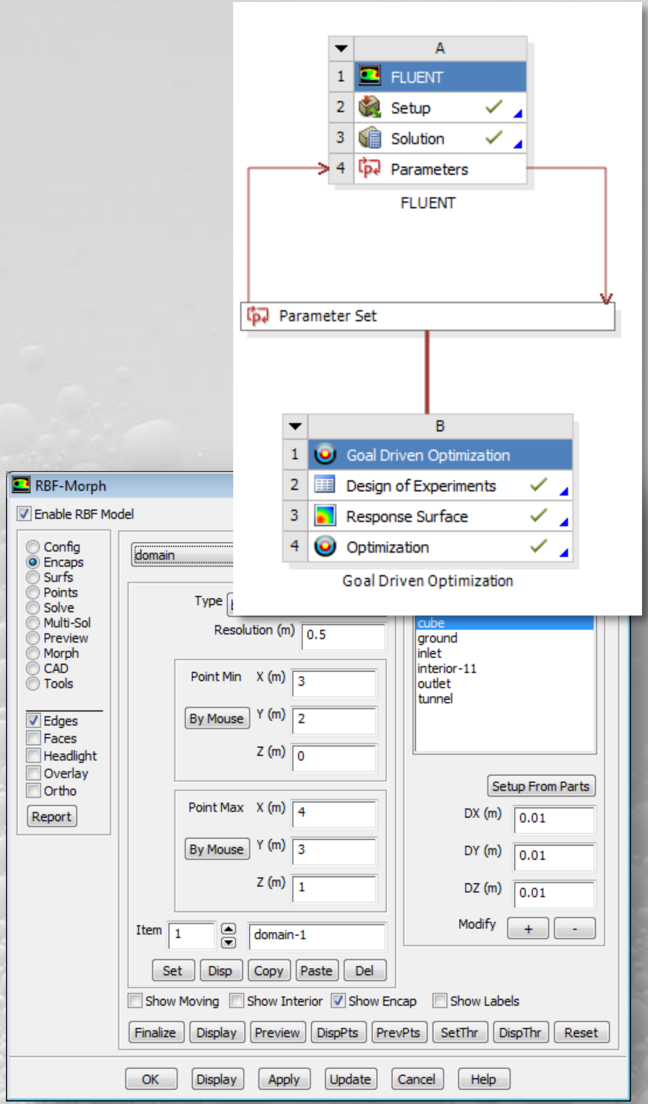
- A mesh morpher is a tool capable to perform **mesh modifications**, in order to achieve arbitrary shape changes and related volume smoothing, without changing the mesh topology.
- In general a morphing operation can introduce a reduction of the **mesh quality**
- A **good** morpher has to minimize this effect, and maximize the possible shape modifications.
- If mesh quality is well preserved, then using the same mesh structure it's a **clear benefit** (remeshing introduces **noise**!).

RBF Morph software line

- HPC RBF **general purposes** library (state of the art algorithms, parallel, GPU). This is the numerical kernel of our software. **Millions** of RBF centers can be fitted in a short time.
- Awarded mesh morphing software available as an add-on for **ANSYS Fluent** CFD solver
- **Stand alone** morphing software + smoothing commands for different mesh formats
- ANSYS Mechanical **ACT extension**

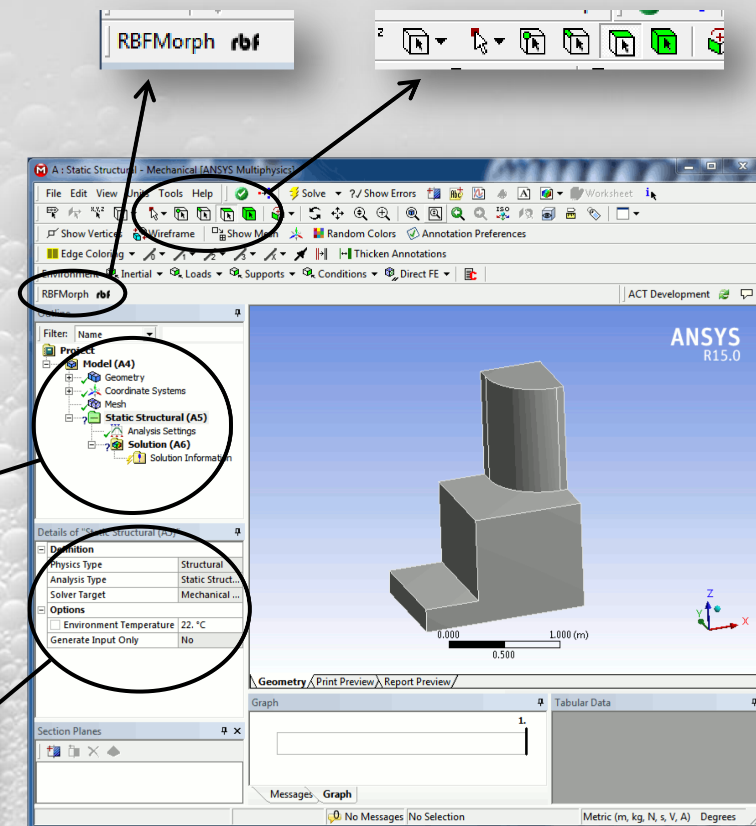
Fluent add-on

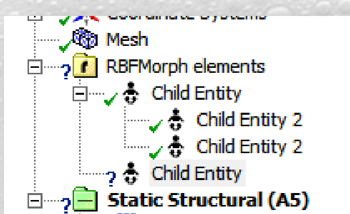
- **Add on** fully integrated within **Fluent** (GUI, TUI & solving stage), **Workbench** and **Adjoint Solver**
- **Mesh-independent** RBF fit used for surface mesh morphing and volume mesh smoothing
- **Parallel** calculation allows to morph **large size** models (many millions of cells) in a short time
- Management of **every kind of mesh** element type (tetrahedral, hexahedral, polyhedral, etc.)
- Support of the **CAD re-design** of the morphed surfaces
- **Multi fit** makes the Fluent case truly parametric (only 1 mesh is stored)
- **Precision**: exact nodal movement and exact feature preservation (**RBF** are better than **FFD**)

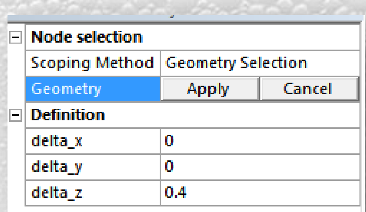


ACT extension for Mechanical

- Deeply integrated in ANSYS Mechanical: same look & feel, same interaction logic
- Nested in the usual Mechanical tree as an added object, shares its scoping tools for **geometrical** and **mesh elements** selections
- Written in **python** and **xml**, uses external C++ RBF Morph core libraries
- **Cuda** and **OpenMP** acceleration
- Child **hierarchical logic** for complex morphings (two steps, three steps, …, n steps setups)



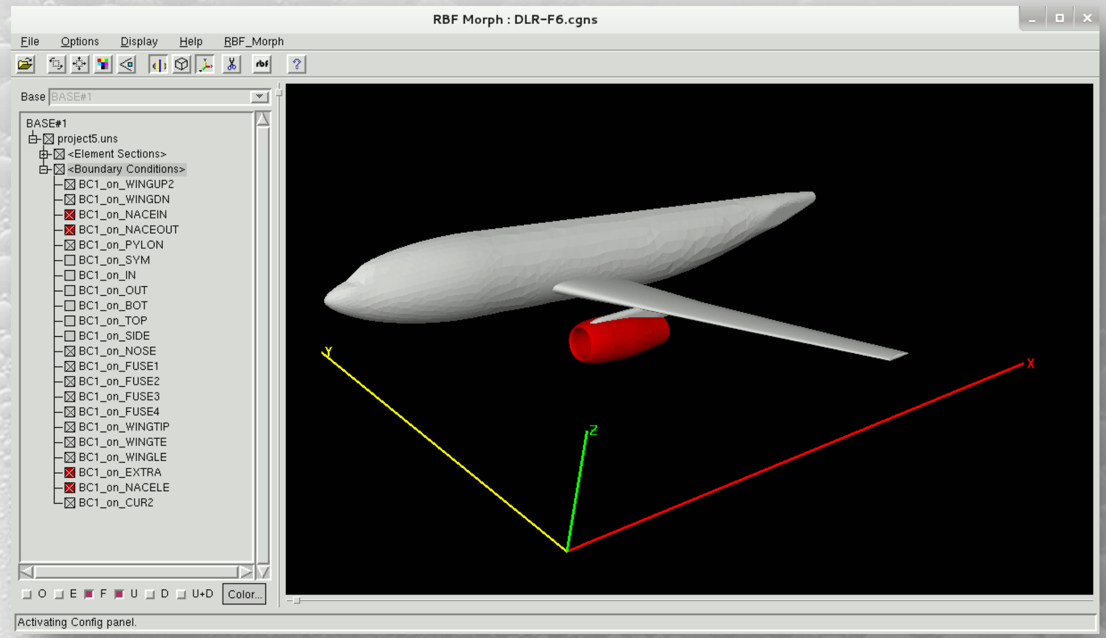




Node selection	
Scoping Method	Geometry Selection
Geometry	Apply / Cancel
Definition	
delta_x	0
delta_y	0
delta_z	0.4

RBF Morph Stand Alone

- RBF solutions are fully compatible and **exchangeable** between add-on and standalone versions
- Support for STL and CGNS file formats. Selected morphed surfaces can be exported in STL format and **back to CAD** is possible via STEP files
- **Add-on-like** interface
- **Solver independent** process currently supports many mesh formats
- Functions **scriptable** via tcl
- Global supported bi-harmonic functions and C^0, C^2, C^4 compact supported functions available



Mesh morphing with RBF

- A system of **radial functions** is used to **fit** a **solution** for the mesh movement/morphing, from a list of **source points** and their **displacements**.
- The RBF problem definition does not depend on the mesh
- Radial Basis Function interpolation is used to derive the displacement in **any location** in the space, each component of the displacement is interpolated:

$$\begin{cases} v_x = s_x(\mathbf{x}) = \sum_{i=1}^{N} \gamma_i^x \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k_i}\right\|\right) + \beta_1^x + \beta_2^x x + \beta_3^x y + \beta_4^x z \\ v_y = s_y(\mathbf{x}) = \sum_{i=1}^{N} \gamma_i^y \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k_i}\right\|\right) + \beta_1^y + \beta_2^y x + \beta_3^y y + \beta_4^y z \\ v_z = s_z(\mathbf{x}) = \sum_{i=1}^{N} \gamma_i^z \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k_i}\right\|\right) + \beta_1^z + \beta_2^z x + \beta_3^z y + \beta_4^z z \end{cases}$$

RBF Theory

- A radial basis fit exists if desired values are matched at source points with a null poly contribution
- The fit problem is associated with the solution of a linear system
- **M** is the interpolation matrix
- **P** is the constraint matrix
- **g** are the scalar values prescribed at source points
- γ and **β** are the fitting coefficients

$$s(\mathbf{x}_{k_i}) = g(\mathbf{x}_{k_i}) \quad 1 \le i \le N$$

$$0 = \sum_{i=1}^{N} \gamma_i q(\mathbf{x}_{k_i})$$

$$\begin{pmatrix} \mathbf{M} & \mathbf{P} \\ \mathbf{P}^{\mathbf{T}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{g} \\ \mathbf{0} \end{pmatrix}$$

$$M_{ij} = \phi\left(\left\| \mathbf{x}_{k_i} - \mathbf{x}_{k_j} \right\|\right) \quad 1 \le i \quad j \le N$$

$$\mathbf{P} = \begin{pmatrix} 1 & x_{k_1}^0 & y_{k_1}^0 & z_{k_1}^0 \\ 1 & x_{k_2}^0 & y_{k_2}^0 & z_{k_2}^0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{k_N}^0 & y_{k_N}^0 & z_{k_N}^0 \end{pmatrix}$$

RBF Theory

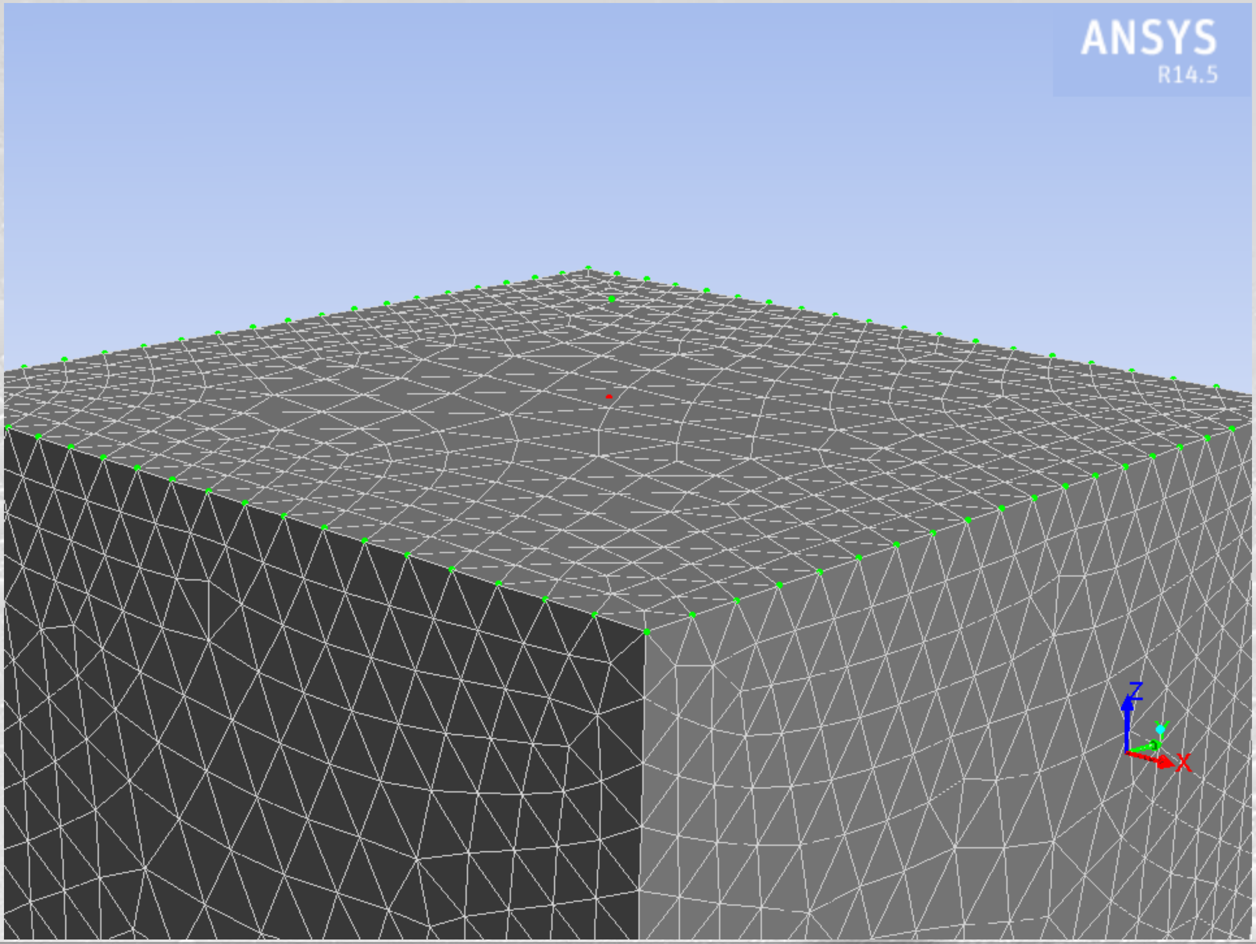
- The radial function can be fully or compactly supported. The bi-harmonic kernel fully supported gives the best results for smoothing.
- For the smoothing problem each component of the displacement prescribed at the source points is interpolated as a single scalar field.

Radial Basis Function	$\phi(r)$
Spline type (R_n)	$\lvert r\rvert^n$, n odd
Thin plate spline (TPS_n)	$\lvert r\rvert^n \log\lvert r\rvert$, 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}

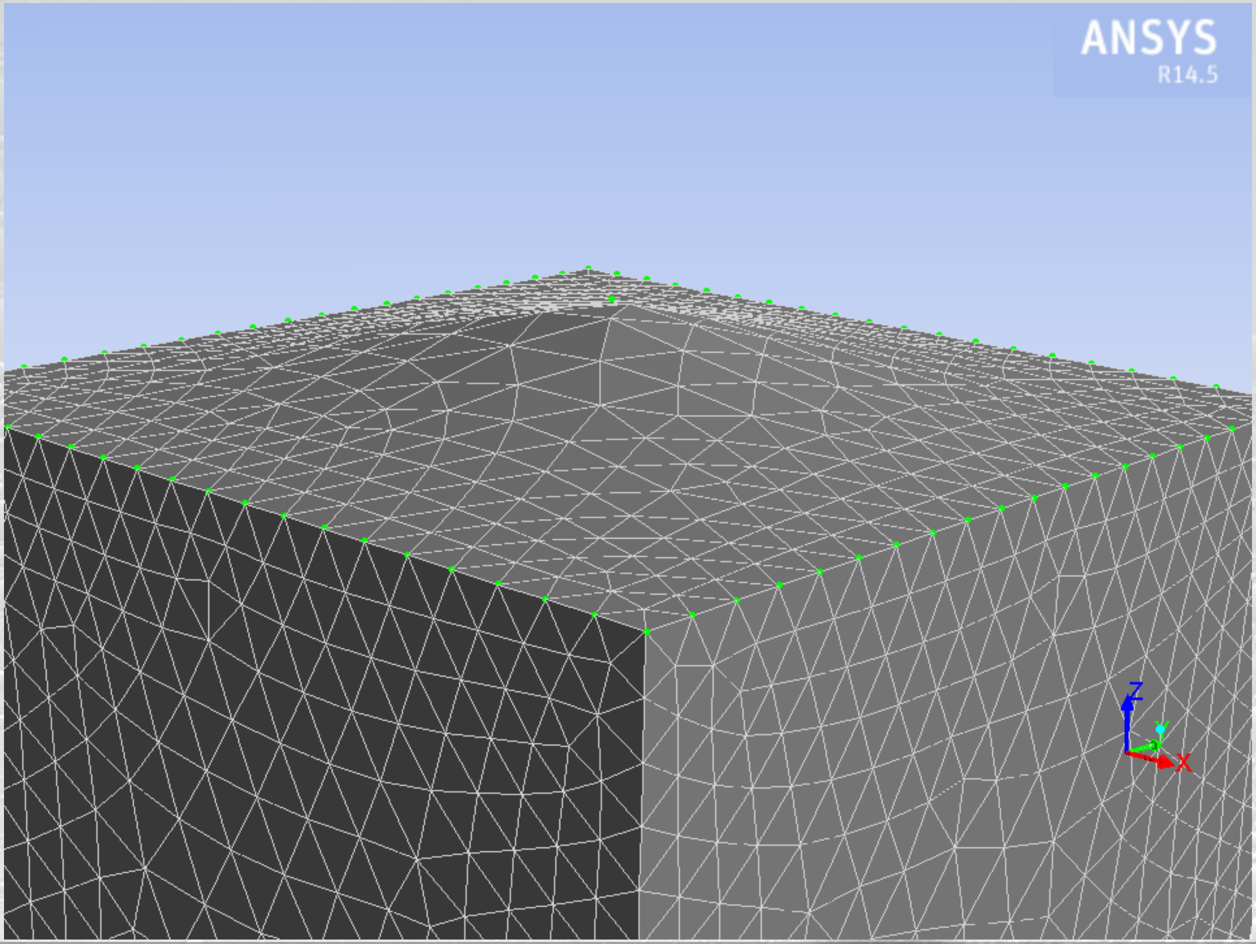
$$\begin{cases} v_x = s_x(\mathbf{x}) = \sum_{i=1}^{N} \gamma_i^x \phi\left(\left\|\mathbf{x}-\mathbf{x}_{k_i}\right\|\right) + \beta_1^x + \beta_2^x x + \beta_3^x y + \beta_4^x z \\ v_y = s_y(\mathbf{x}) = \sum_{i=1}^{N} \gamma_i^y \phi\left(\left\|\mathbf{x}-\mathbf{x}_{k_i}\right\|\right) + \beta_1^y + \beta_2^y x + \beta_3^y y + \beta_4^y z \\ v_z = s_z(\mathbf{x}) = \sum_{i=1}^{N} \gamma_i^z \phi\left(\left\|\mathbf{x}-\mathbf{x}_{k_i}\right\|\right) + \beta_1^z + \beta_2^z x + \beta_3^z y + \beta_4^z z \end{cases}$$

Source point control

One pt at center 80 pts at border

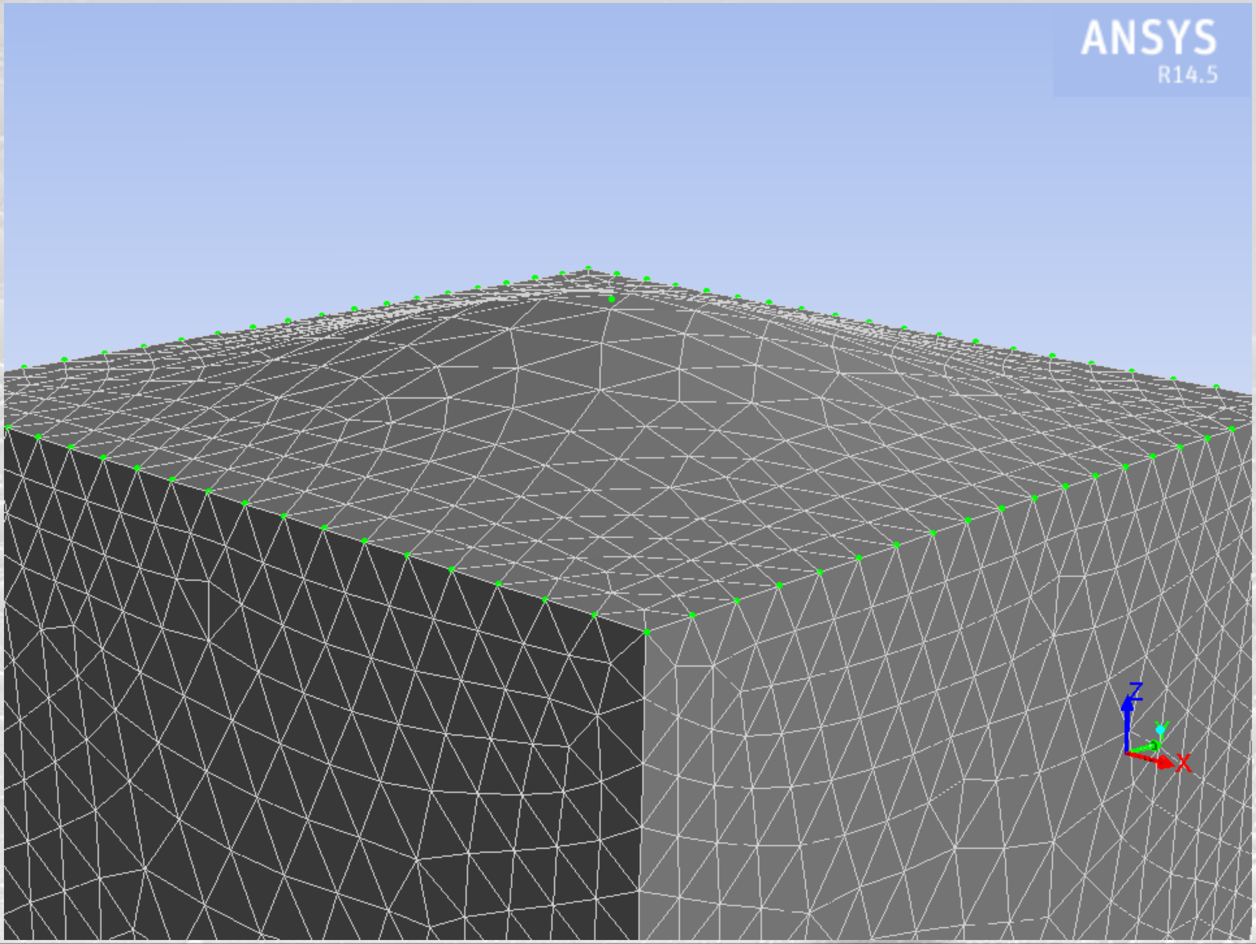


Effect on surface (gs-r)

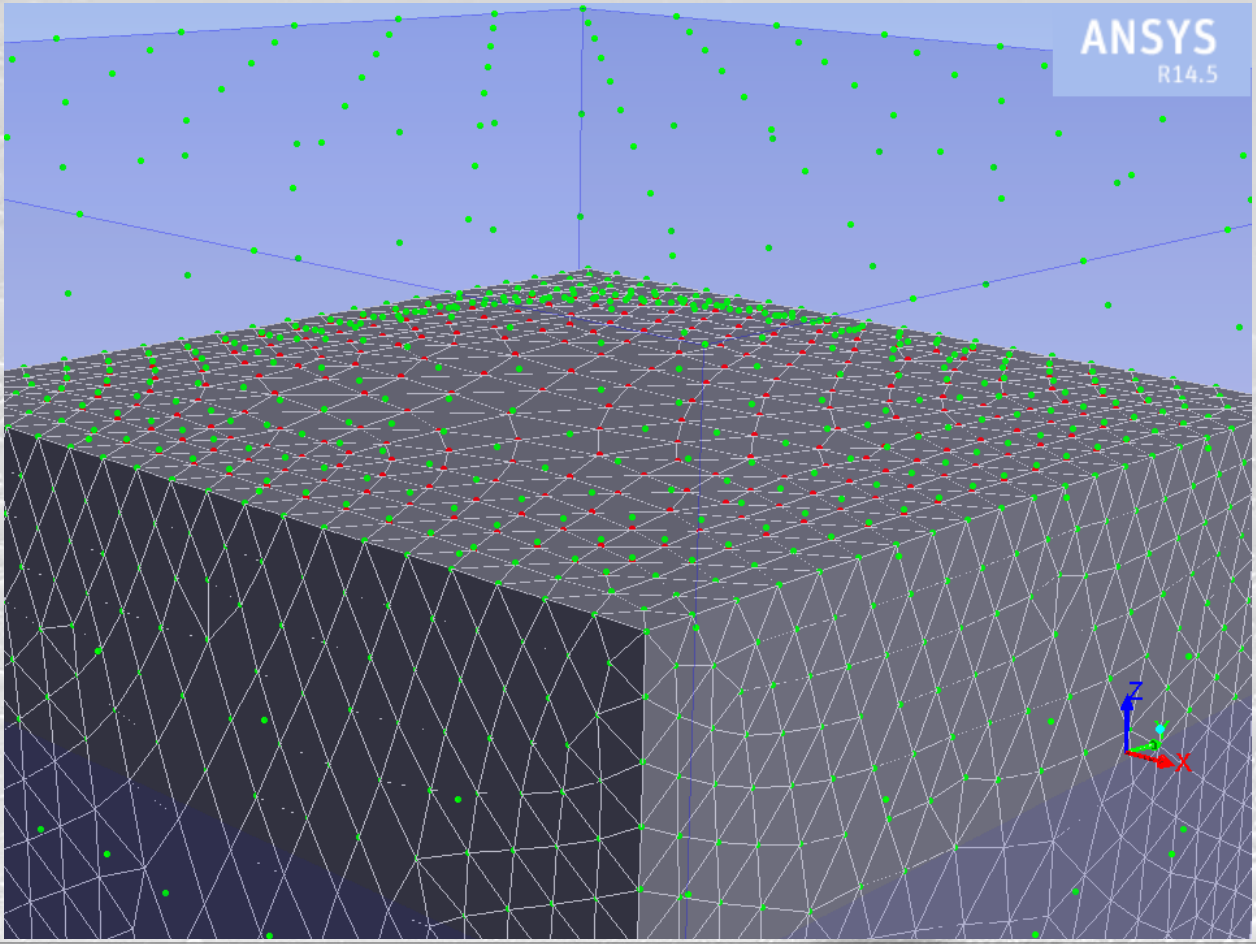


Source point control

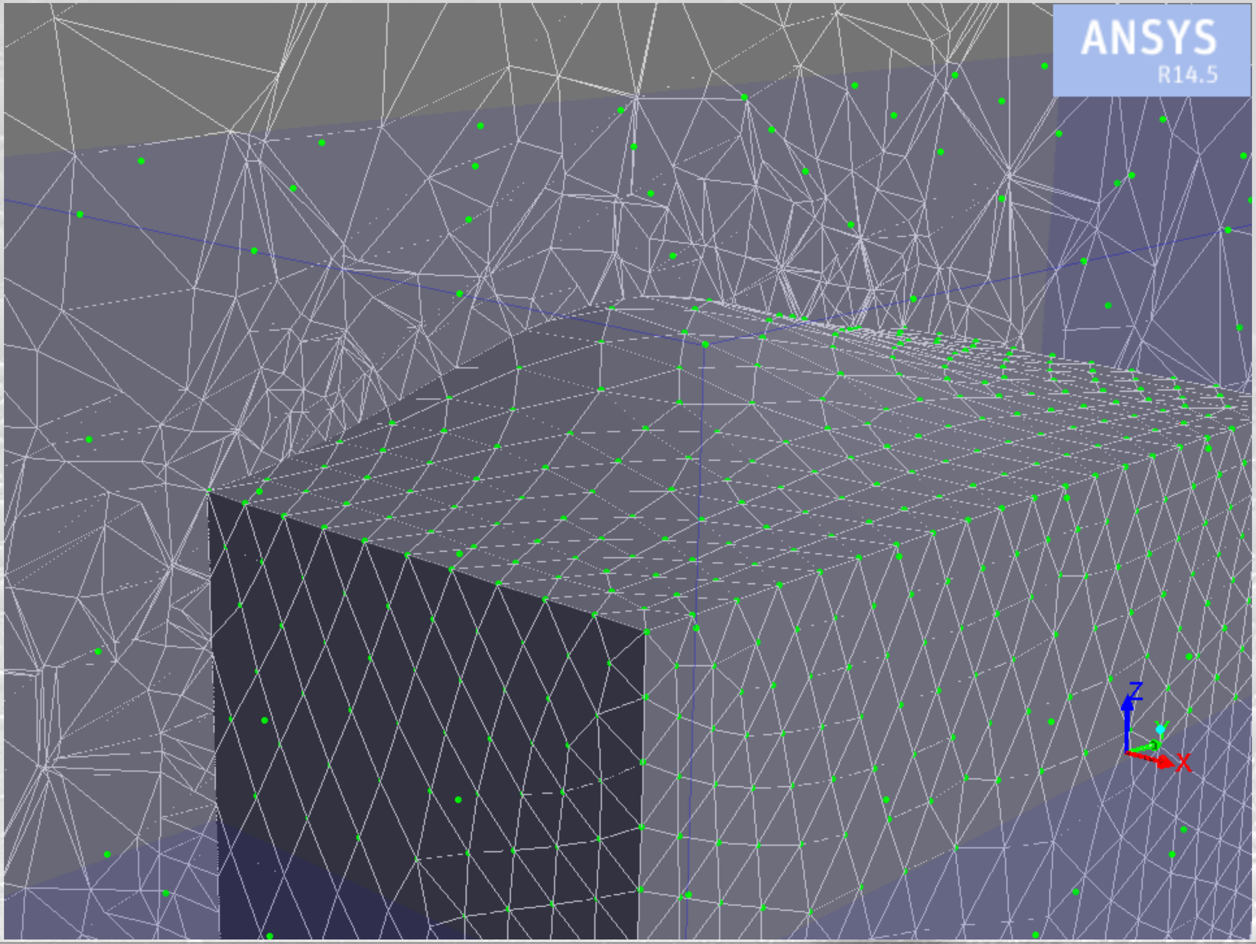
Effect on surface (cp-c4)



Control of volume mesh (1166 pts)



Morphing the volume mesh



RBF Theory

- The radial function can be fully or compactly supported. The bi-harmonic kernel fully supported gives the best results for smoothing.
- For the smoothing problem each component of the displacement prescribed at the source points is interpolated as a single scalar field.

Radial Basis Function	$\phi(r)$
Spline type (R_n)	$\lvert r\rvert^n$, n odd
Thin plate spline (TPS_n)	$\lvert r\rvert^n \log\lvert r\rvert$, 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}

$$\begin{cases} v_x = s_x(\mathbf{x}) = \sum_{i=1}^{N} \gamma_i^x \phi\left(\left\|\mathbf{x}-\mathbf{x}_{k_i}\right\|\right) + \beta_1^x + \beta_2^x x + \beta_3^x y + \beta_4^x z \\ v_y = s_y(\mathbf{x}) = \sum_{i=1}^{N} \gamma_i^y \phi\left(\left\|\mathbf{x}-\mathbf{x}_{k_i}\right\|\right) + \beta_1^y + \beta_2^y x + \beta_3^y y + \beta_4^y z \\ v_z = s_z(\mathbf{x}) = \sum_{i=1}^{N} \gamma_i^z \phi\left(\left\|\mathbf{x}-\mathbf{x}_{k_i}\right\|\right) + \beta_1^z + \beta_2^z x + \beta_3^z y + \beta_4^z z \end{cases}$$

accelerating the solver

- The evaluation of RBF at a point has a cost of order N
- The fit has a cost of order $\mathbf{N^3}$ for a direct fit (full populated matrix); this limit to ~10.000 the number of source points that can be used in a practical problem
- Using an iterative solver (with a good pre-conditioner) the fit has a cost of order $\mathbf{N^2}$; the number of points can be increased up to ~70.000
- Using also space partitioning to accelerate fit and evaluation the number of points can be increased up to ~**300.000**
- The method can be further accelerated using fast pre-conditioner building and FMM RBF evaluation…

Solver performances

- 10.000 RBF centers FIT
  - 120 minutes Jan 2008
  - 5 seconds Jan 2010
- Largest fit **2.600.000** 133 minutes
- Largest model morphed **300.000.000** cells
- Fit and Morph a **100.000.000** cells model using **500.000** RBF centers within **15 minutes**

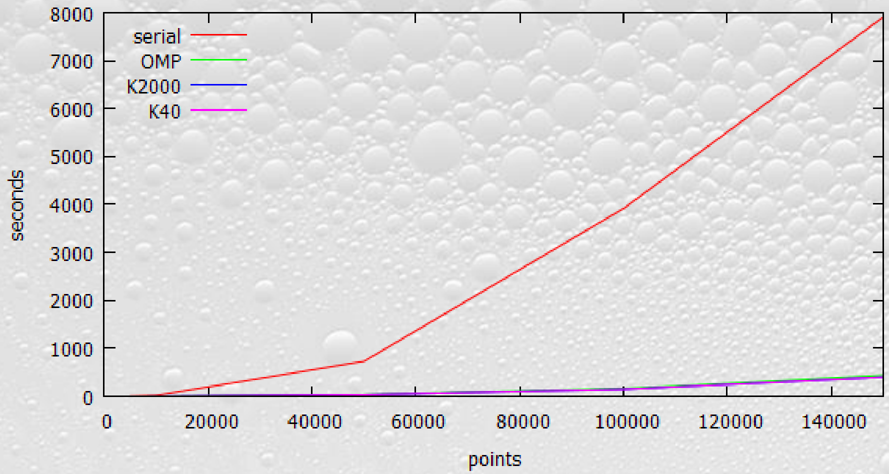
#points	2010 (Minutes)	2008 (Minutes)
3.000	0 (1s)	15
10.000	0 (5s)	120
40.000	1 (44s)	Not registered
160.000	4	Not registered
650.000	22	Not registered
2.600.000	133	Not registered

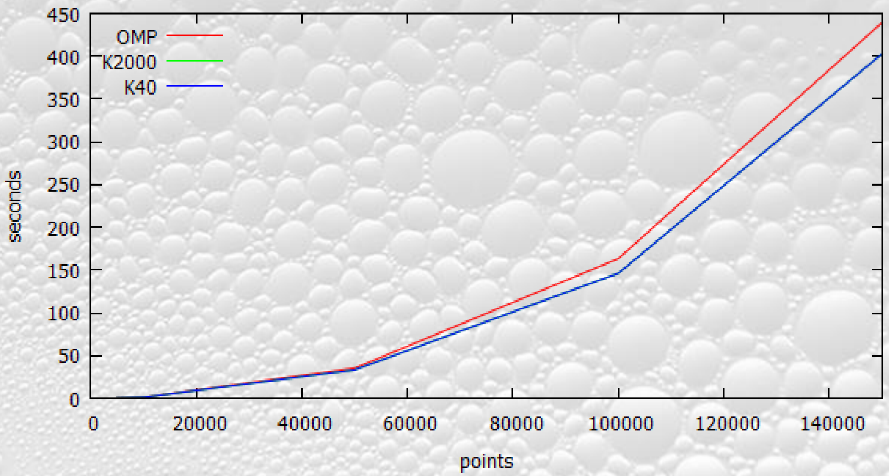
GPU acceleration!

- Single RBF complete evaluation
- Unit random cube
- **Kepler 40**: 2880 CUDA cores GPU Clock 0.75 GHz
- **Quadro K2000**: 384 CUDA cores GPU Clock 0.95 GHz
- **CPU**: Intel Xeon E5-2680 Clock 2.8 GHz 20 cores

GPU acceleration!

#points	CPU serial		CPU OMP		K2000		K40	
	Preconditioner build	iterations	Preconditioner build	iterations	Preconditioner build	iterations	Preconditioner build	iterations
5000	0.8	4.352	0.955	0.4770	0.94	0.491	2.194	0.060
10000	2.47	19.867	2.32	1.132	2.41	1.140	3.885	0.17
50000	47.07	733.463	46.463	35.20	47.510	33.13	48.95	5.12
100000	231.03	3922.31	238.68	163.375	238.82	146.20	239.007	21.365
150000	375.75	7915.24	379.923	439.546	400.12	402.95	457.597	52.568



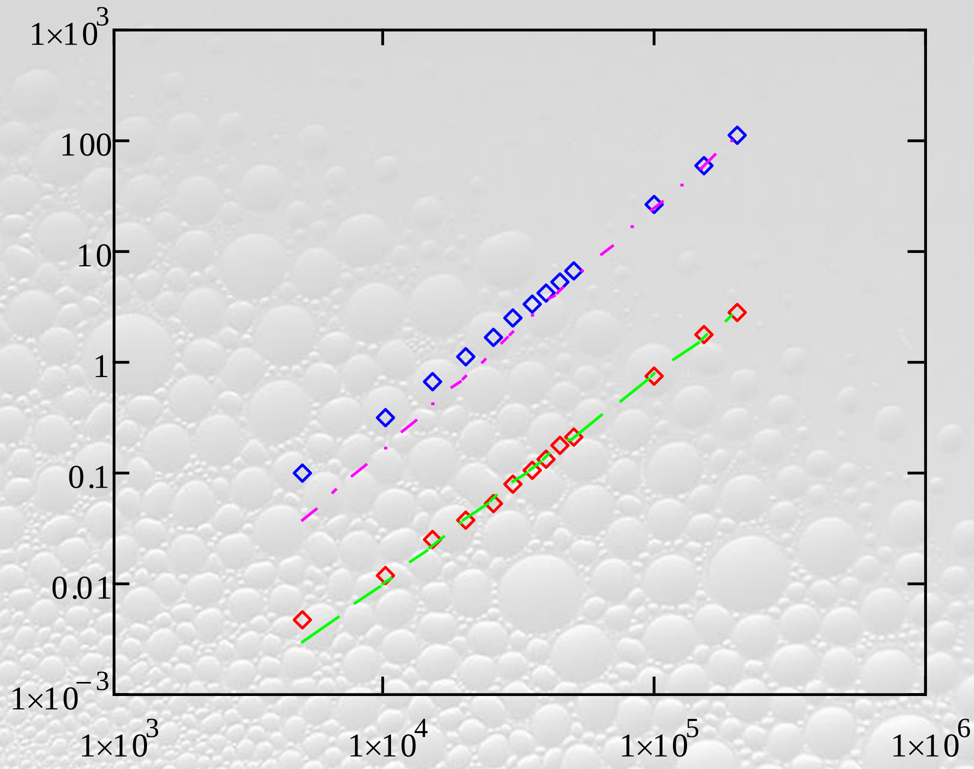


Scaling plot

- Complexity is expected to grow as **N ^{2}**
- GPU observed as **N $^{1.87}$**
- CPU observed as **N $^{2.174}$**
- Estimation at one million points:

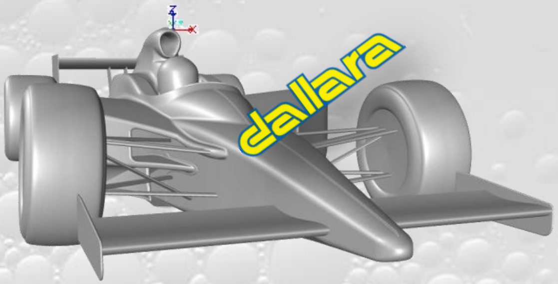
  GPU: 59 s

  CPU: 2783 s



Key benefits: performances

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



- Largest fitted cloud **2 mill**. points on **32 cpu** in **3 hours**.
- Largest model morphed (in our knowledge) **700.mill.** cells on **768 cpu** in **45 min**.

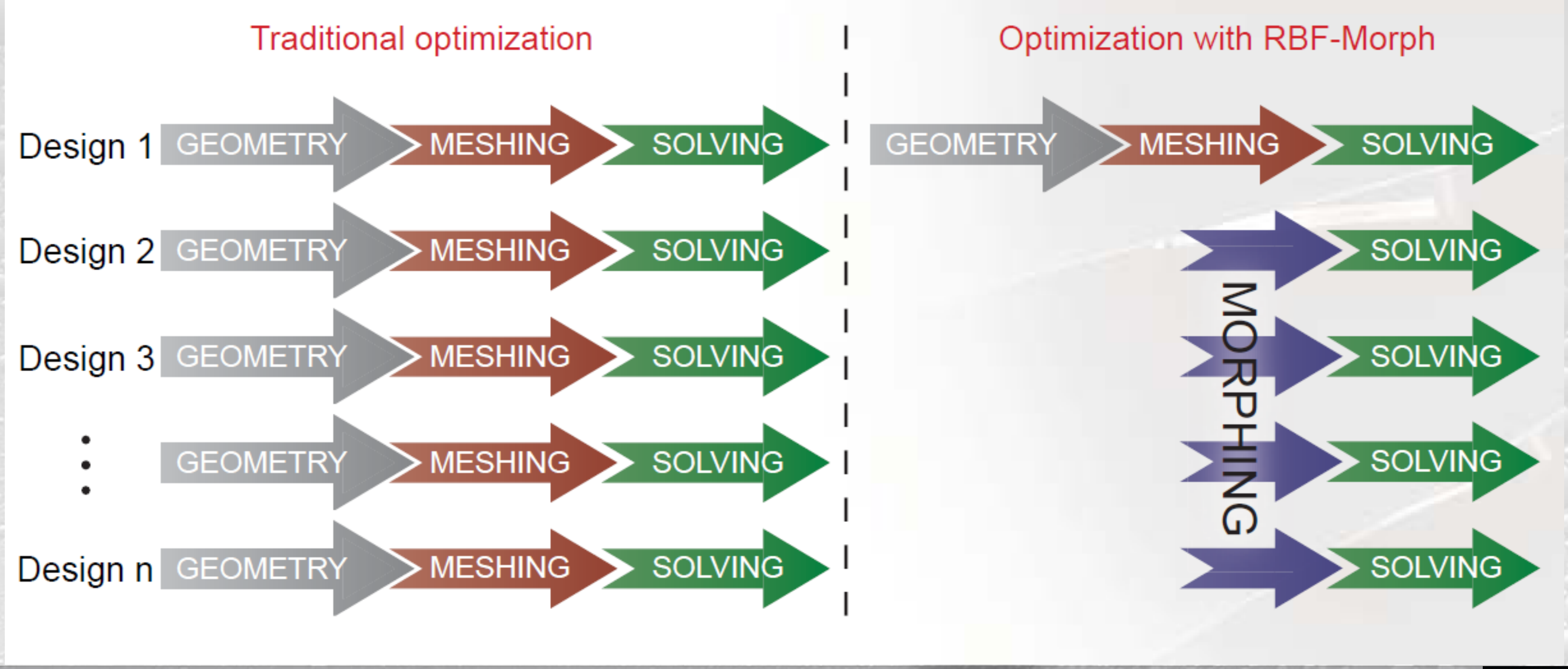
HPC Workflows

(rbf-morph)™ Welcome to the World of Fast Morphing!

Case	Motorbike windshield	Reference car	Sedan	Hull	**Volvo XC60**	Sails	DLR-F6	IR5
Organization	MRA/UTV	MIRA	ANSYS	Leeds	**ANSYS**	New Castle / UTV	MorphLab/ UTV	Dallara
Year	2009	2010	2011	2011	**2012**	2013	2013	2013
#Mcells	1,5	5,2	6	0,3	**50**	1,5	14	80
mesh type	tets	poly	tets	hexa	**tets**	hexa	tets	tets
#par	3	3	2	8	**4**	4	8	5
#design	45	27	9	45	**50**	100	81	1
RS Tool	modeF	Mathcad	DX	DX	**DX**	DX/ Mathcad	DX	FSI
ncores	4	2	12	4	**240**	16	16	256
RUN (hr)	48	300	24	45	**50**	26	102	1
Time to set-up one par (hr)	1,5	2,5	2	1	**2**	2	1	2
Time to set-up (hr)	4,5	7,5	4	8	**8**	8	8	8
Serial time one design (hr)	4,27	22,22	32,00	4,00	**240,00**	4,16	20,15	256,00
Serial time one design (hr/Mcells)	2,84	4,27	5,33	13,33	**4,80**	2,77	1,44	3,2

Key benefits: automated

- RBF Morph makes the Fluent model **parametric** with respect to the **shape**
- Works for **any size of mesh** (from small models managed with a WS up to huge Formula 1 meshes in an HPC environment)
- Exposed parameters can be steered with the **optimizer of choice** (DX, modeFRONTIER, Dakota,…)

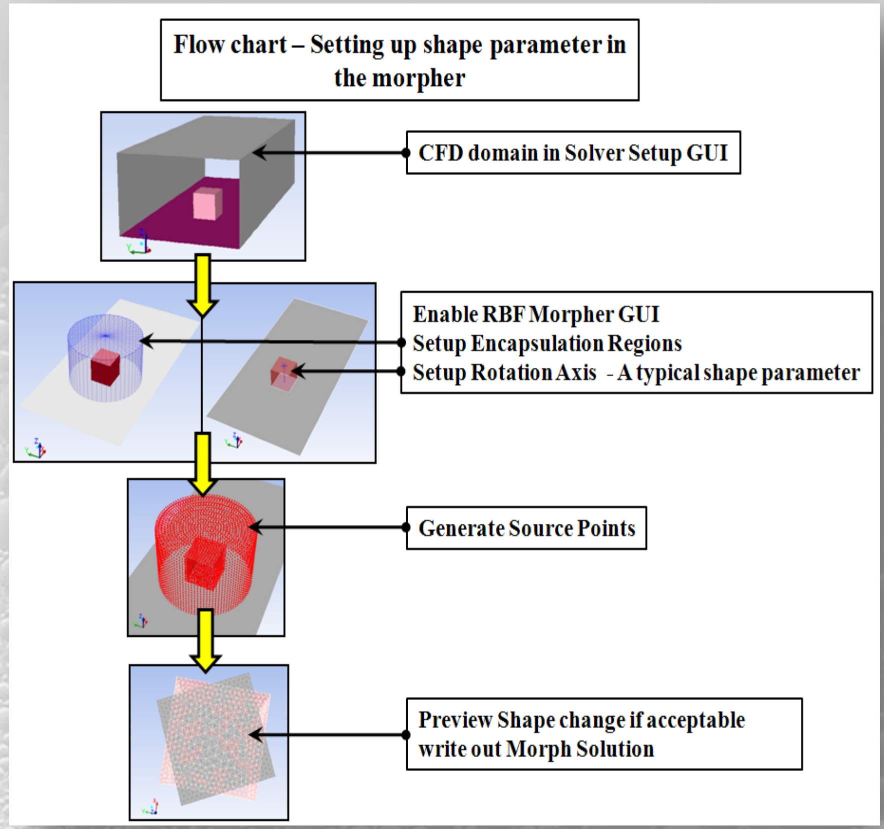


How it Works: the work-flow

- *RBF Morph* basically requires three different steps:
- **Step 1 setup** and definition of the problem (source points and displacements).
- **Step 2 fitting** of the RBF system (write out .rbf + .sol).
- **Step 3 [SERIAL or PARALLEL] morphing** of the surface and volume mesh (available also in the **CFD solution stage** it requires only baseline mesh and .rbf + .sol files).

How it Works: the problem setup

- The problem must describe correctly the **desired changes** and must **preserve exactly** the fixed part of the mesh.
- The prescription of the **source points** and their displacements fully defines the *RBF Morph* problem.
- Each problem and its fit define a mesh **modifier** or a **shape parameter**.



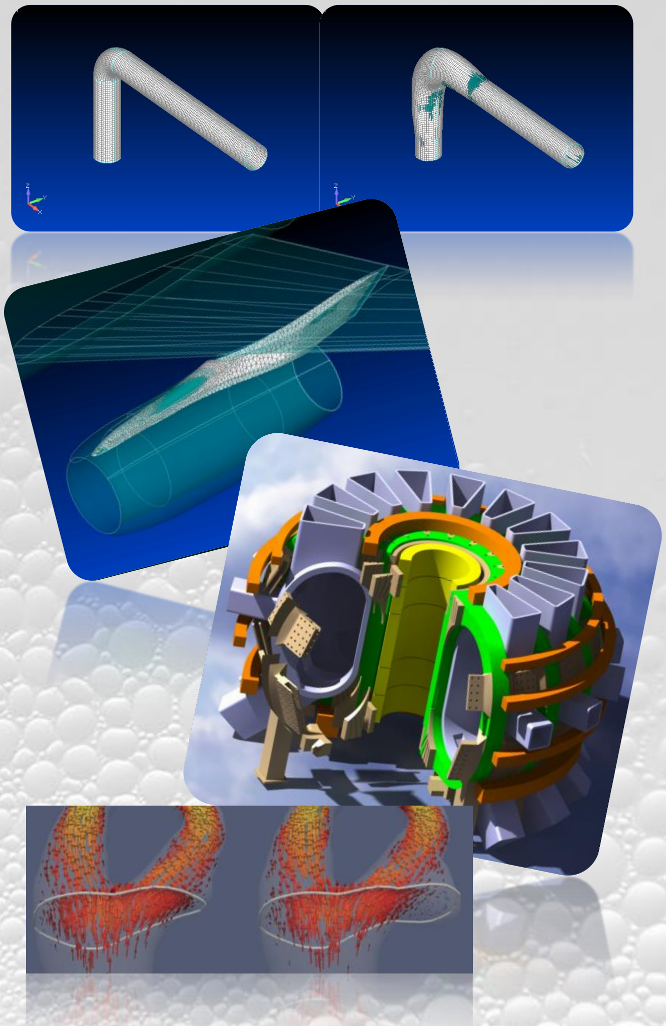
How it Works: Fluent parallel morphing

- **Interactive** update using the GUI **Multi-Sol** panel and the Morph/Undo commands.
- **Interactive** update using **sequential morphing** by the TUI command `(rbf-smorph)`.
- **Batch** update using the **single morphing** command `(rbf-morph)` in a journal file (the RBF Morph DOE tool allows to easily set-up a run).
- **Batch** update using several **sequential morphing** commands in a journal file.
- Link shape amplifications to **Fluent custom parameters** driven by **Workbench** (better if using **DesignXplorer**).
- More options (transient, FSI, modeFRONTIER, batch RBF fit …)

Ongoing RBF Morph Researches

Ongoing RBF researches

- RBF Morph and Adjoint coupling: Adjoint sculpting, Adjoint preview, Augmented DOE
- **STL** targeting, **CAD** controlled surfaces
- **Mesh to CAD** features
- Mapping of **magnetic** and **pressure** loads
- Interpolation of **hemodynamic** flow fields acquired *in vivo*
- Strain and **stress calculation** (experimental data, coarse FEM, isostatic lines)



RBF4AERO EU Project

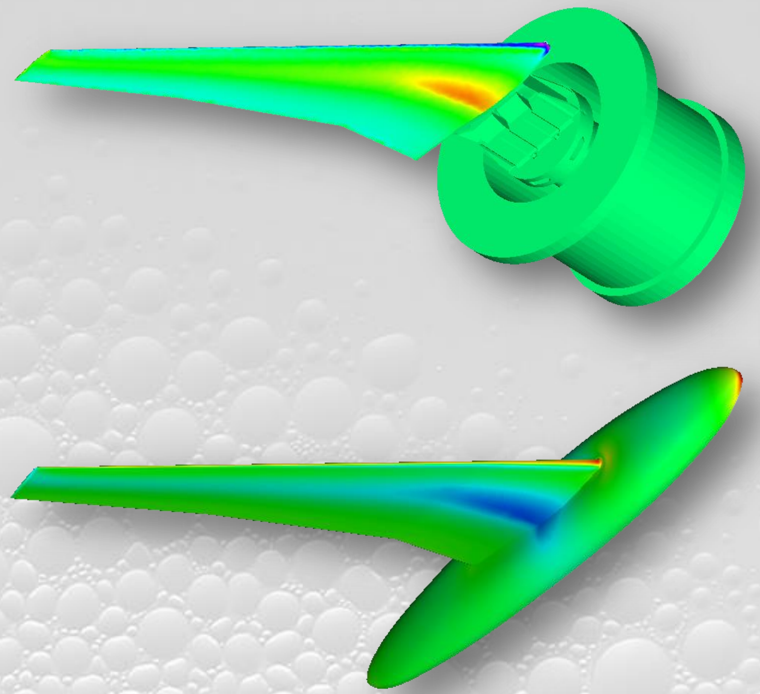
- “Innovative Benchmark Technology for Aircraft Engineering Design and Efficient Design Phase Optimisation” –

ACP3-GA-2013-605396

- **www.rbf4aero.eu**

RIBES

- **R**adial basis functions at fluid **I**nterface **B**oundaries to **E**nvelope flow results for advanced **S**tructural analysis



JTI-CS-2013-GRA-01-052

RIBES

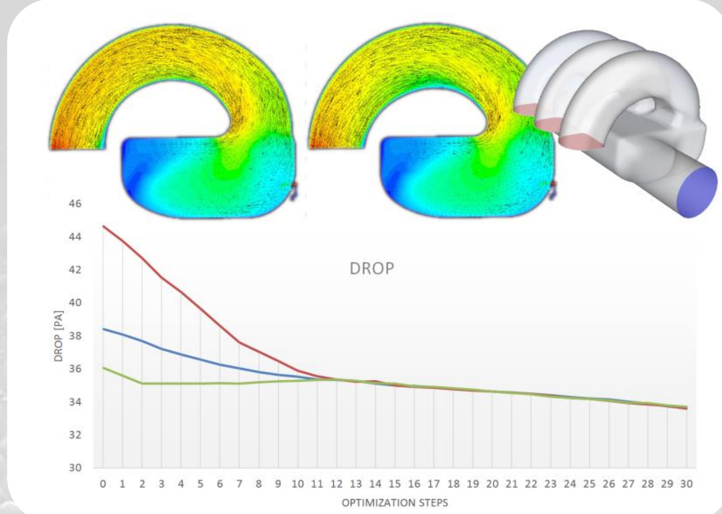
CLEANSKY

Tor Vergata

Fortissimo EU Project

- **F**actories **O**f the Future **R**esources, **T**echnology, **I**nfrastructure and **S**ervices for **SI**mulation and **MO**delling

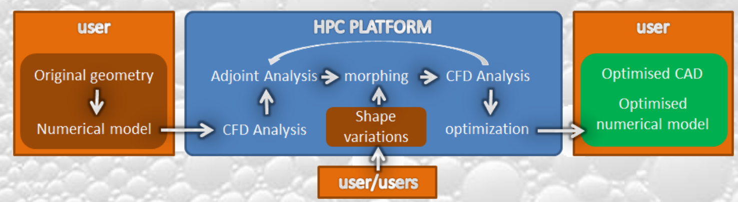
- Approved experiment: "Virtual Automatic Rapid Prototyping Based on Fast Morphing on HPC Platforms"



FORTISSIMO

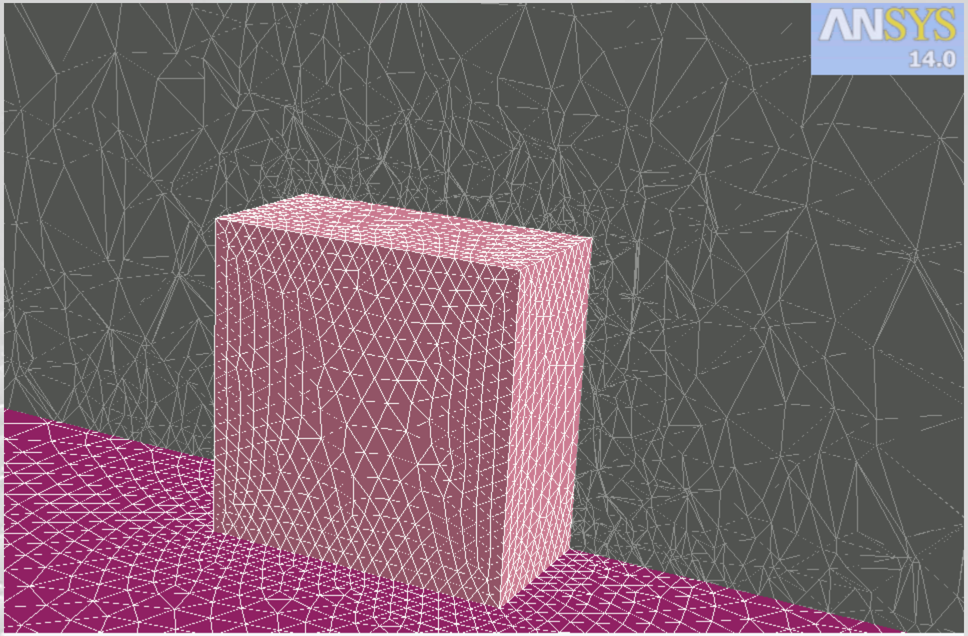
Tor Vergata

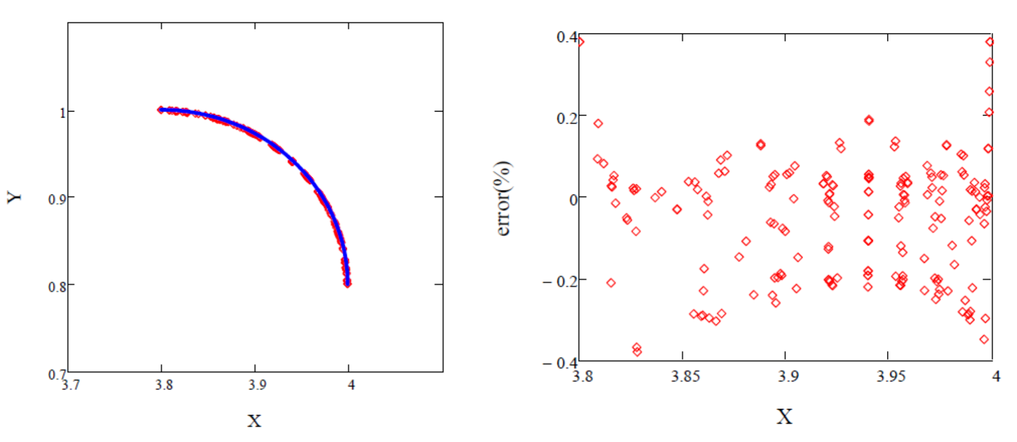
CINECA SCAI
SuperComputing Applications and Innovation



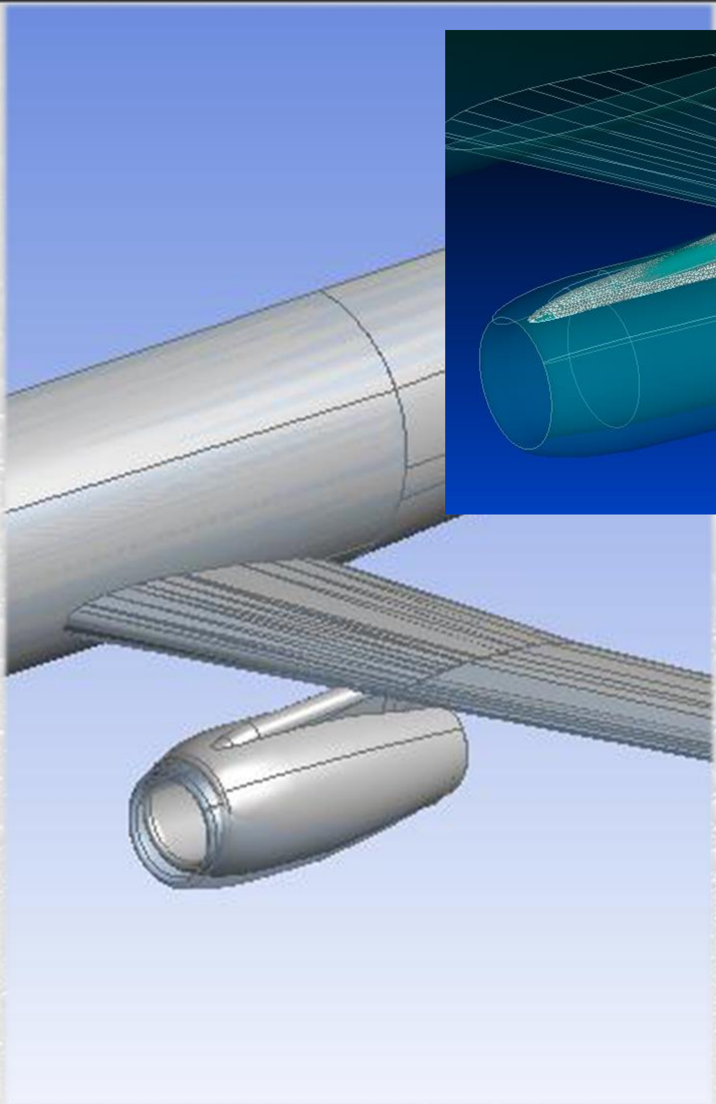
CAD controlled surfaces

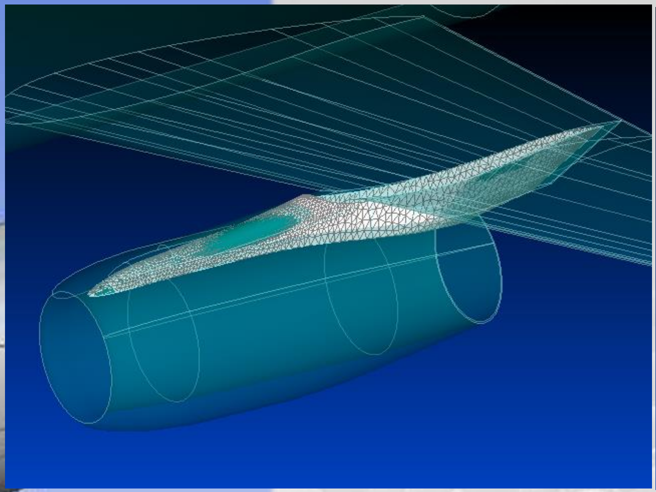
- A new shape known in advance can be inserted using an **STL target**
- In the example a **fillet** with radius in the range 20-30 mm is applied to one edge of the 1000 mm side cube
- Shape **blending** allows a continuous variation
- Accuracy of implicit surface is validated against reference analytic geometry

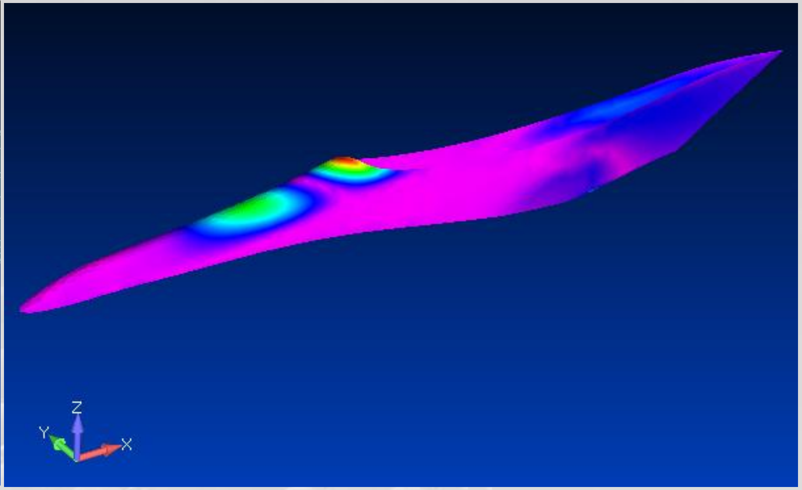


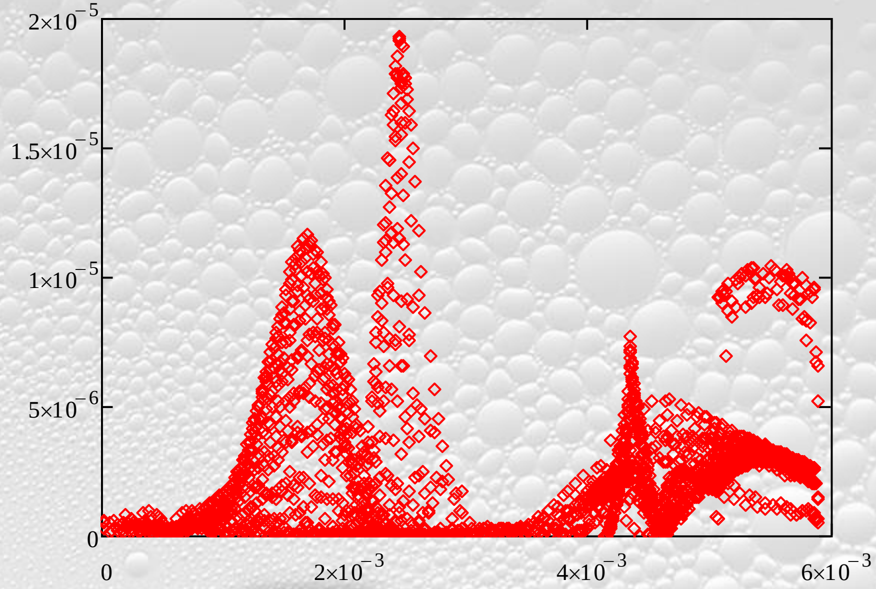


MESH2CAD - NURBS

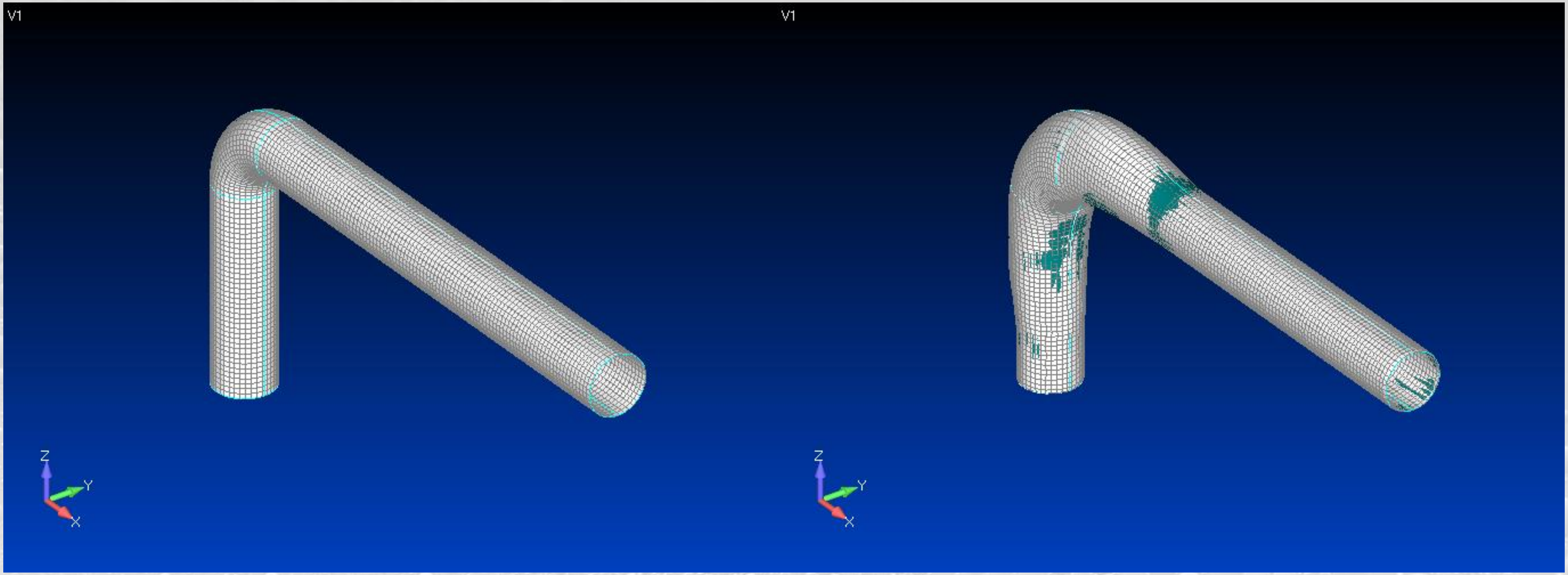






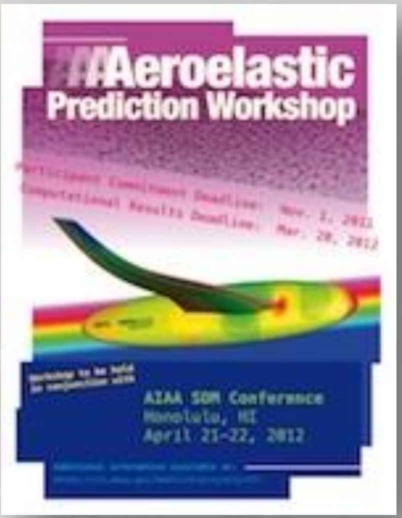


Adjoint Sculpting



Fluid Structure Interaction

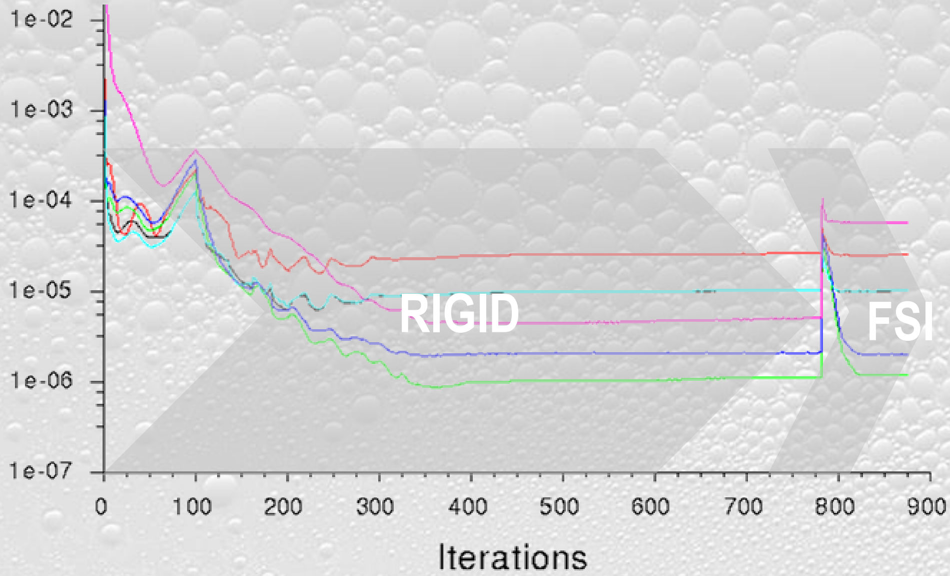
- Active project with ANSYS Germany & ANSYS Italy focused on HIRENASD case of benchmark of Aeroelastic Prediction Workshop (Thorsten Hansen, Angela Lestari, Benjamin Duda & Domenico Caridi)
- Flexible CFD model allows to do a steady aeroelastic run at the same cost of a rigid one
- Flexible CFD model can be used for transient FSI

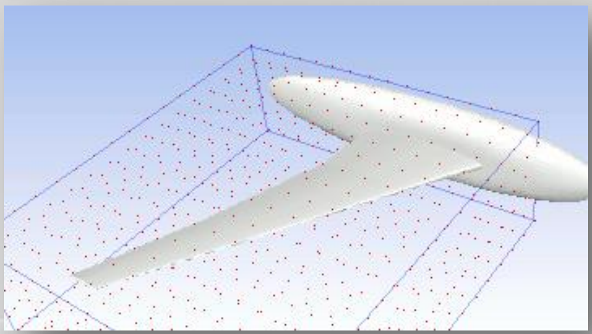


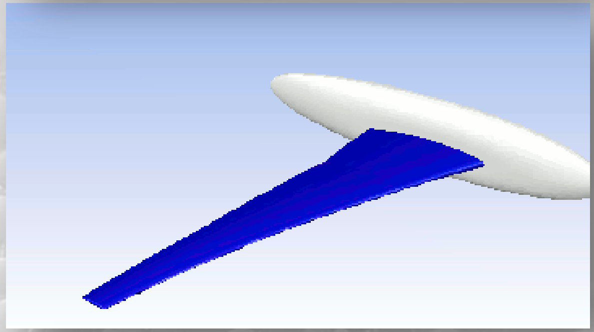
- Actual modal coordinates can be linked to FEM for stress recovery
- Modal Forces are integrated within Fluent over the CFD surface mesh with actual pressure data
- FSI commands to fast update the mesh using current modal coordinates (steady & transient)

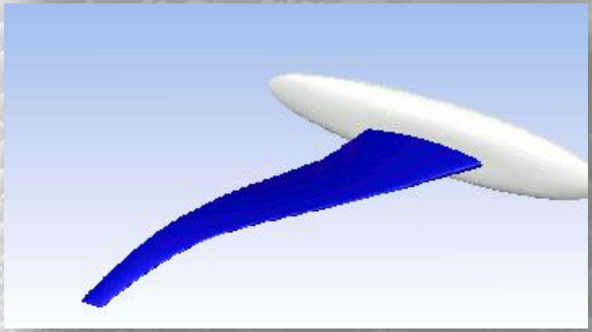
Fluid Structure Interaction

- Modal basis is validated within Fluent thanks to Preview panel
- Update command defined using a scheme function and invoked each 25 iterations as a Fluent calculation activity



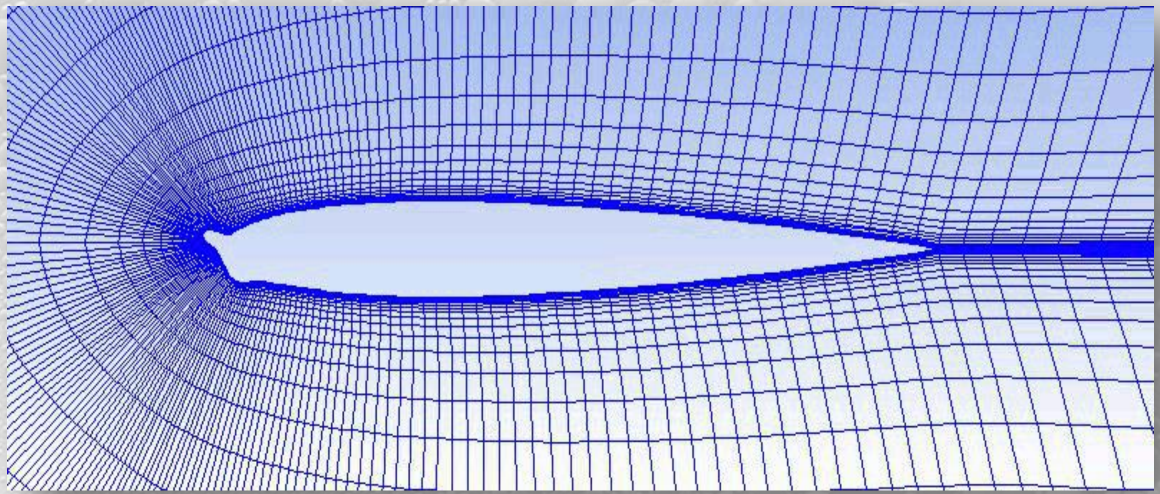






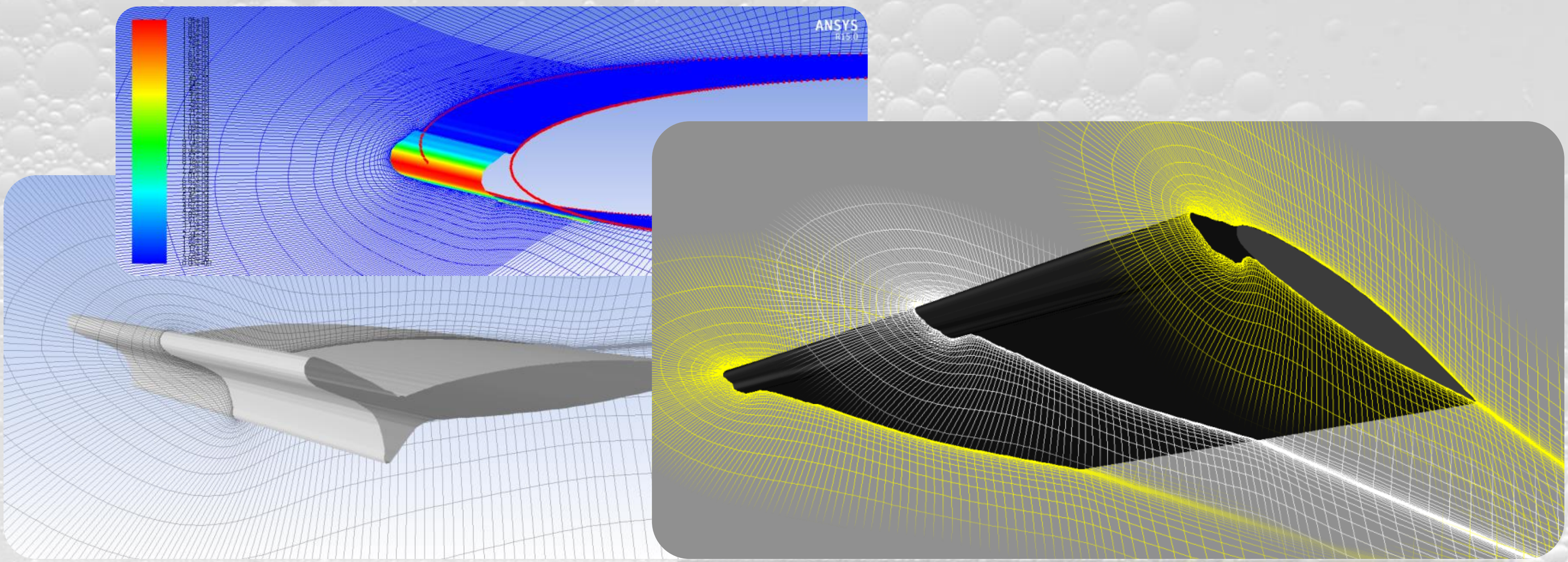
Icing accretion morphing

- Any Ice accretion model that exposes results can be coupled with RBF Morph
- Thickness are applied as normal displacements and imported as meshless points
- At each step the mesh is automatically modified allowing to be used for the subsequent accretion calculation
- The method is valid for 2D and 3D cases
- Capability tested on geometries from LEWICE 2.0 validation manual



Icing accretion morphing

- The workflow can be automated by using scheme and UDF
- y+ is preserved after morphing
- Active project with ANSYS india coupling EWF + custom accretion model + RBF Morph to automatically update the mesh during calculation



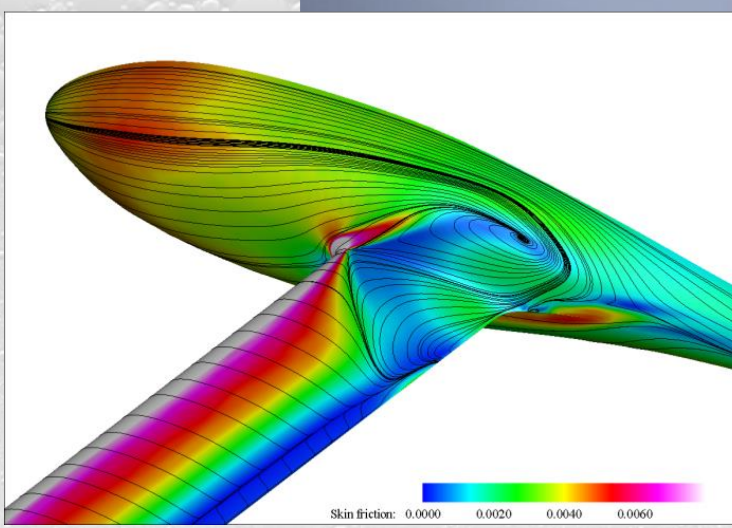
Test case : Glider optimisation

Aerodynamic improvement

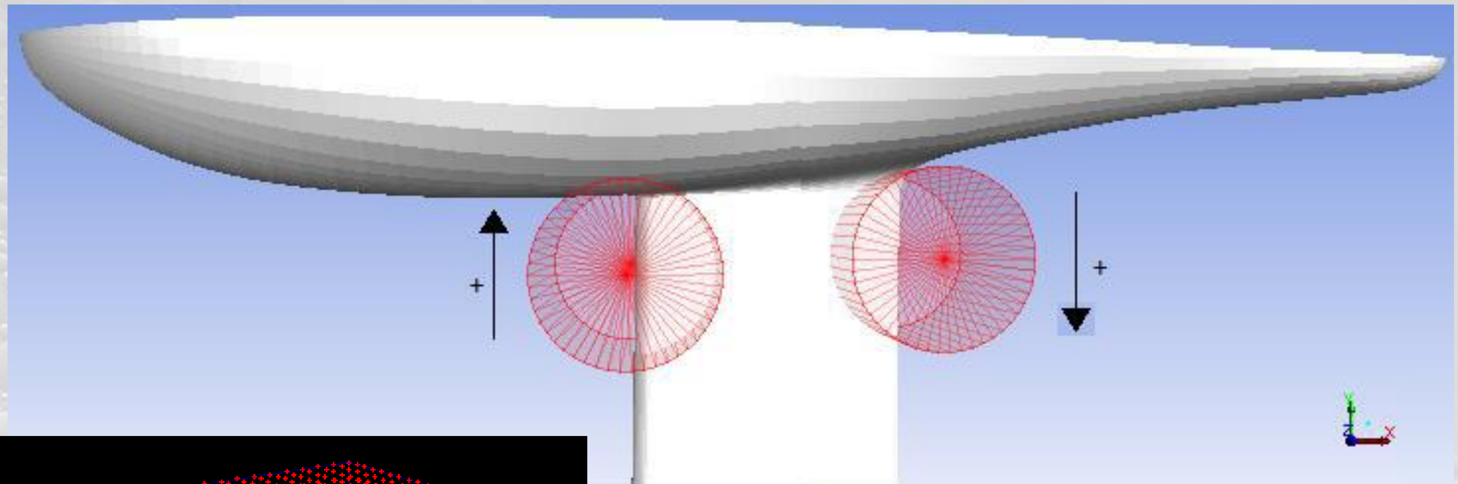
(rbf-morph)™
Welcome to the World of Fast Morphing!

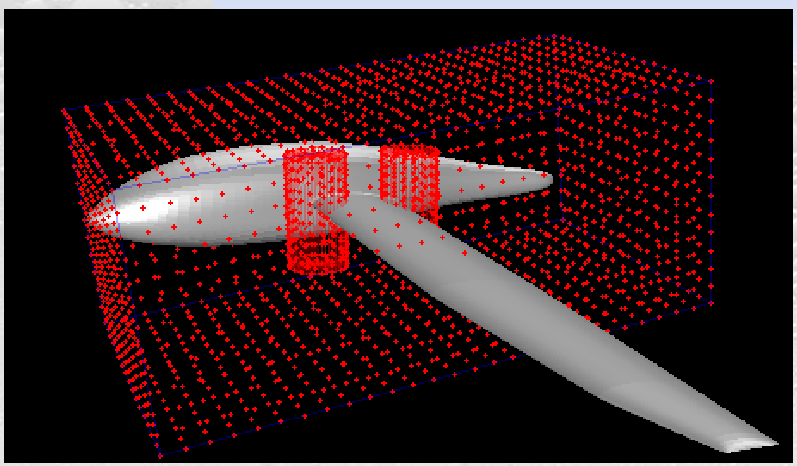
Reynolds 1.24 mill. (c 0.8 m)
Alpha 8 deg

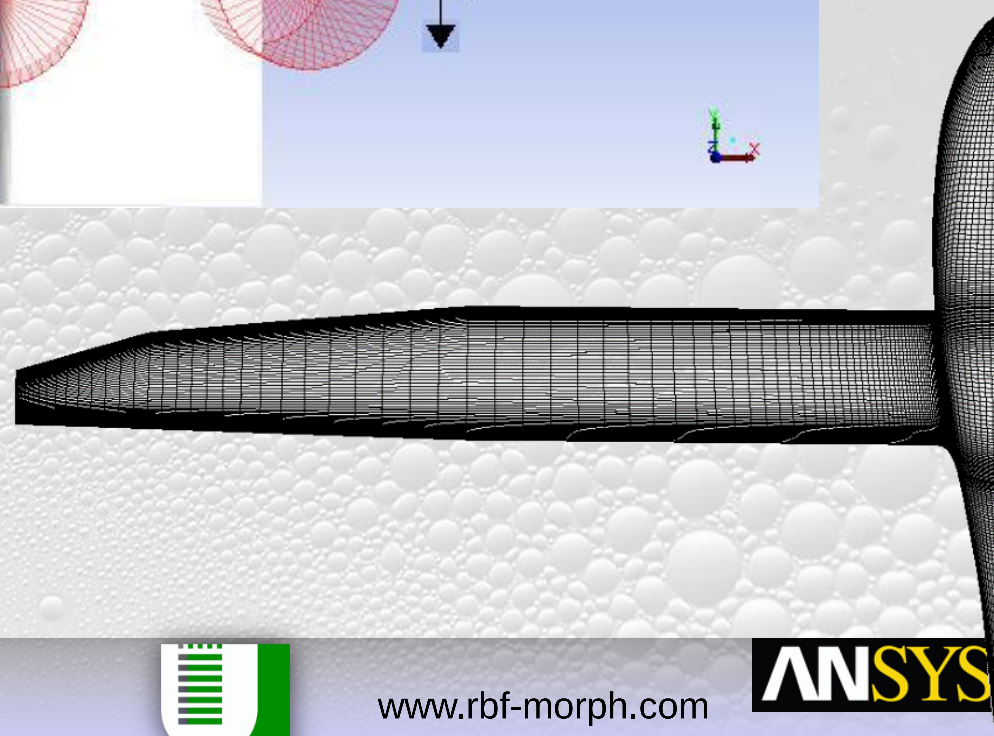
RBF4AERO
PIPISTREL
RINA 1861 D'APPOLONIA
consulting, design, operation & maintenance engineering



Problem setup

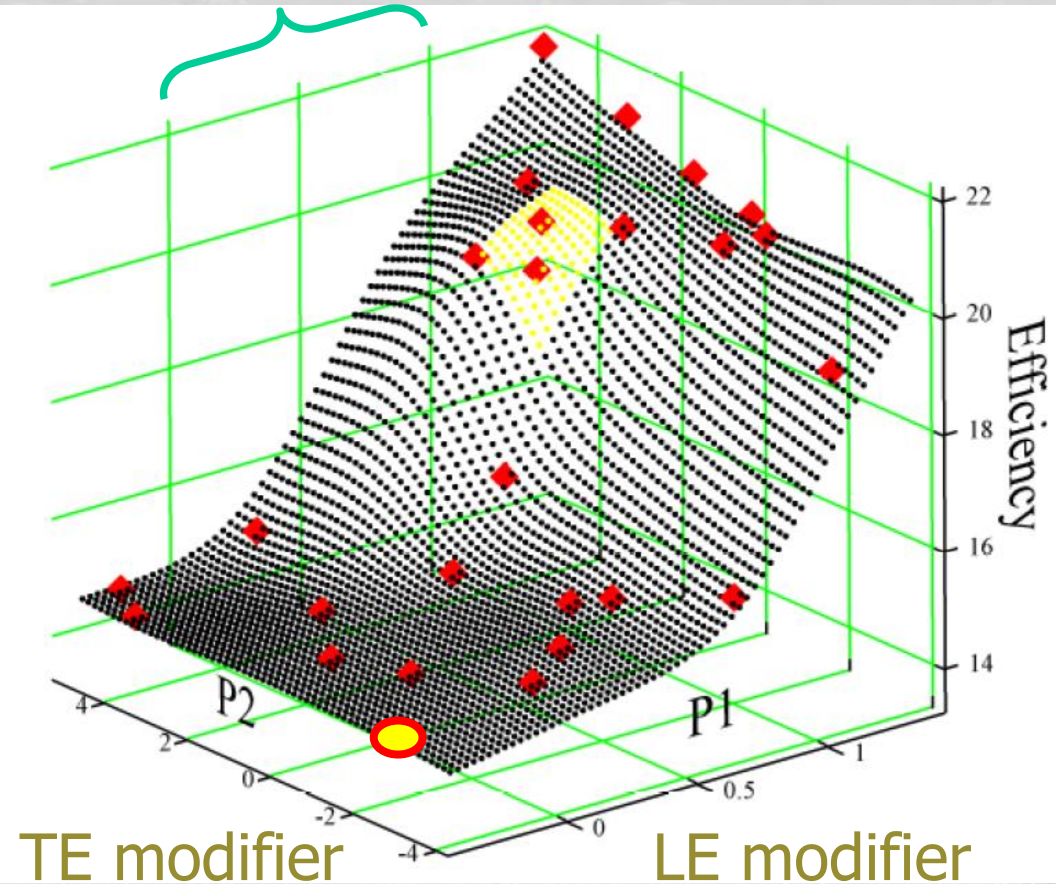






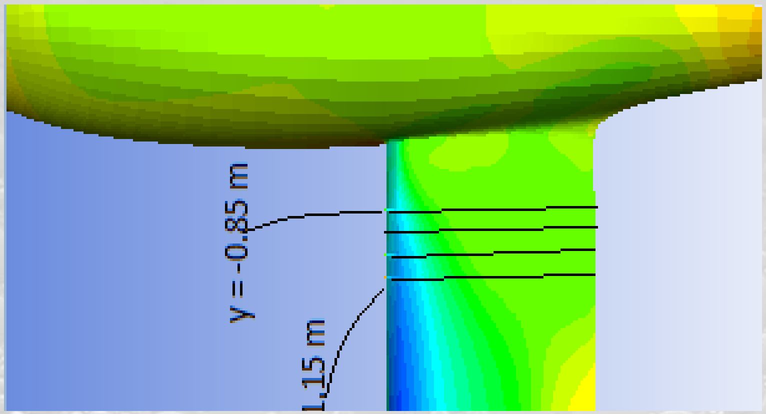
Optimization solution

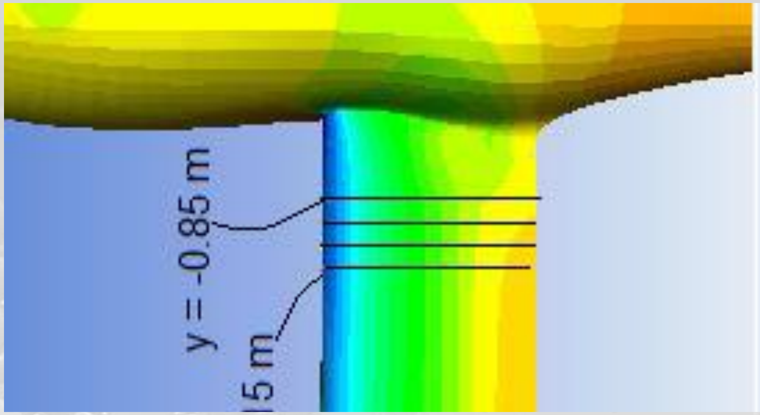
10 cm

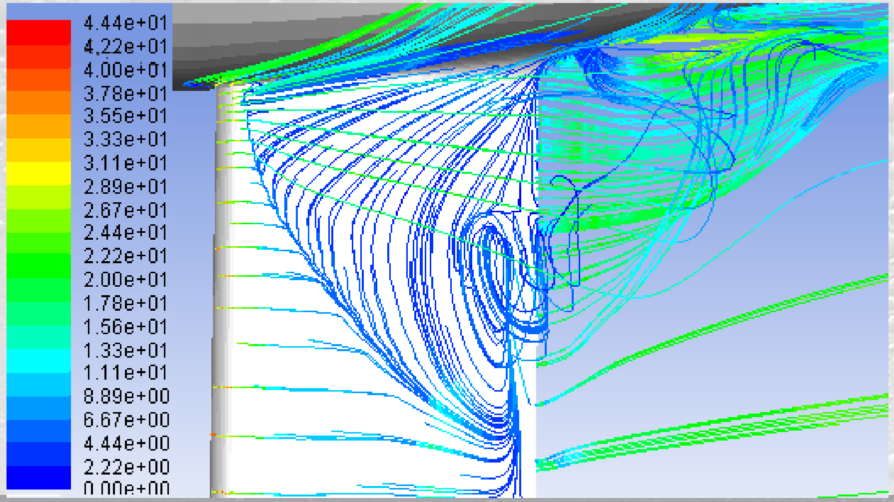


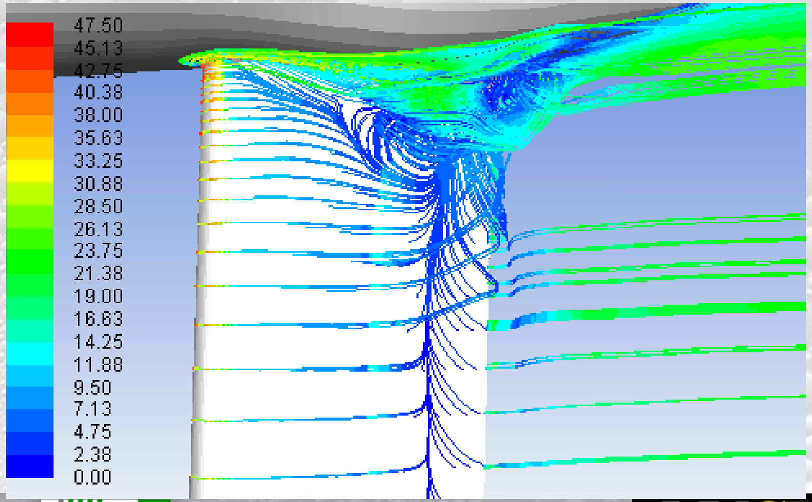
	C_L	C_D	E
Baseline	1.131	760 dc	14.9
Optimised	1.216	605 dc	20.1
Variation	**+7.5 %**	**-20.4 %**	**+35.0 %**

Improvement

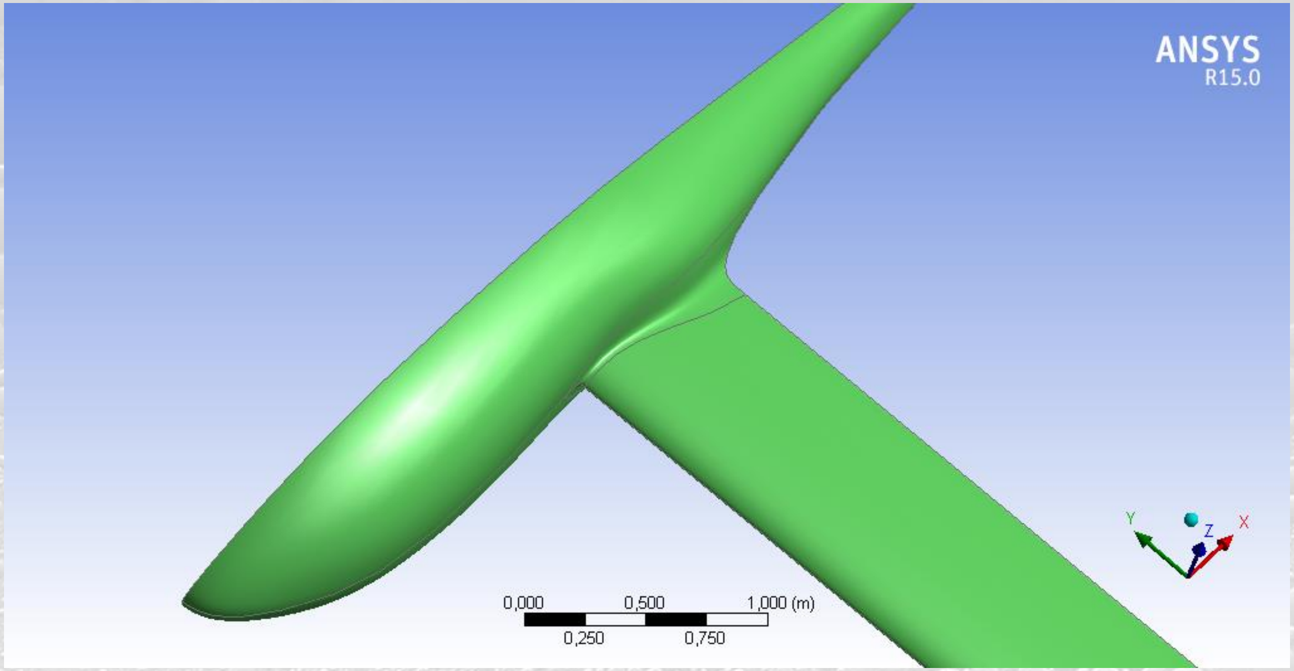








Back To CAD

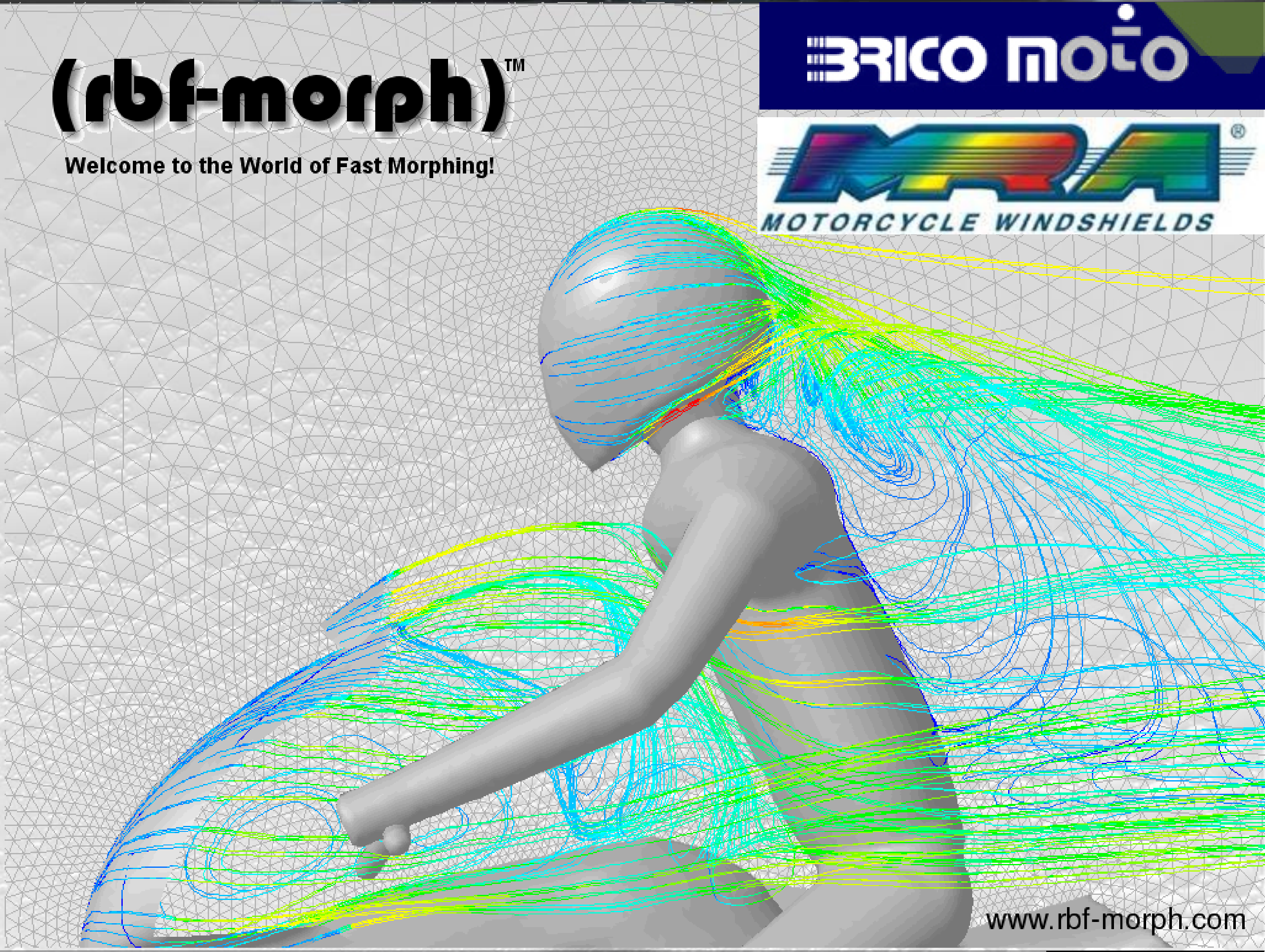


Optimized morphing action to the baseline CAD model

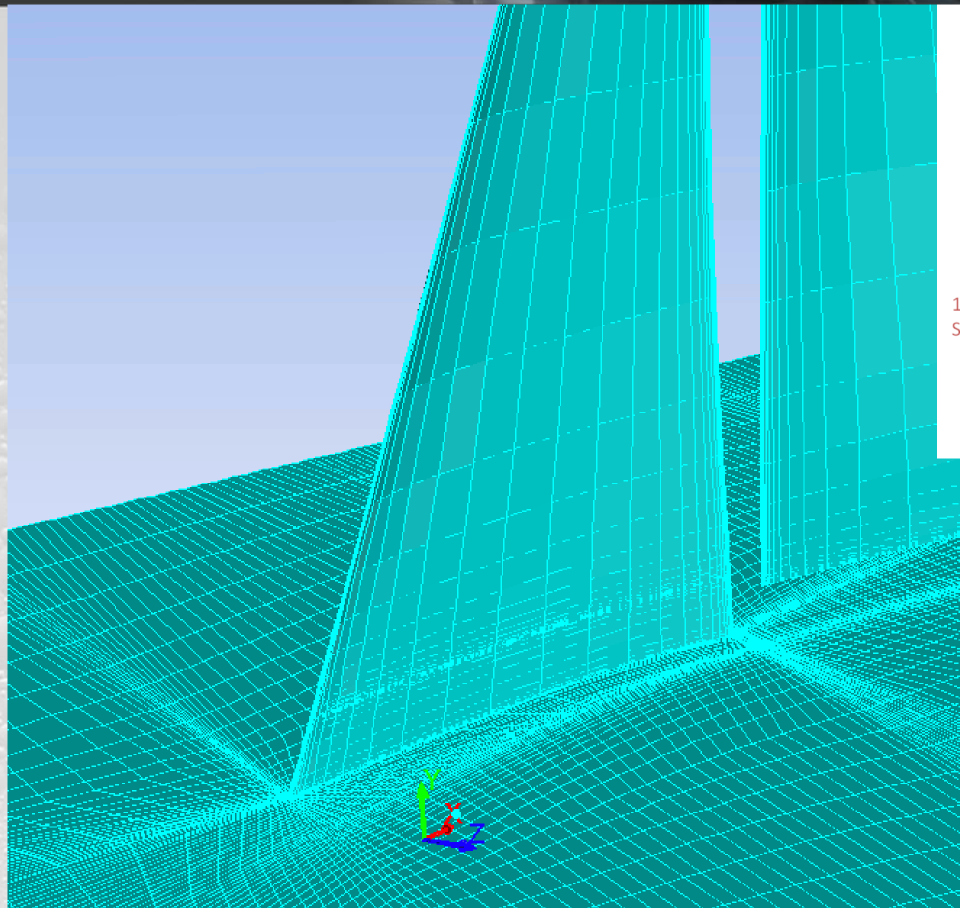
Hands on!

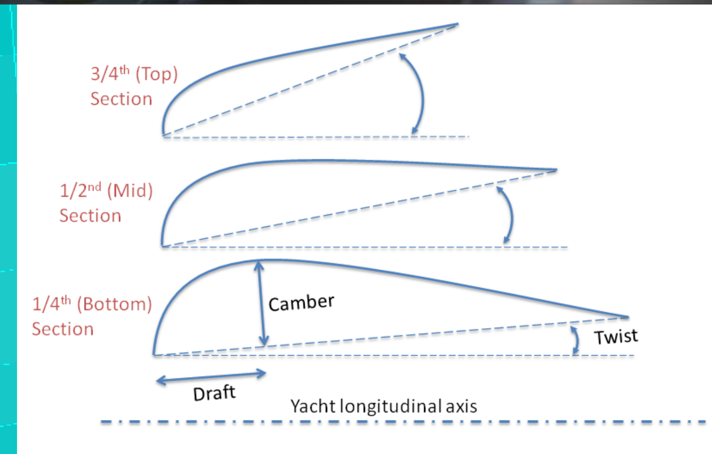
Industrial Applications

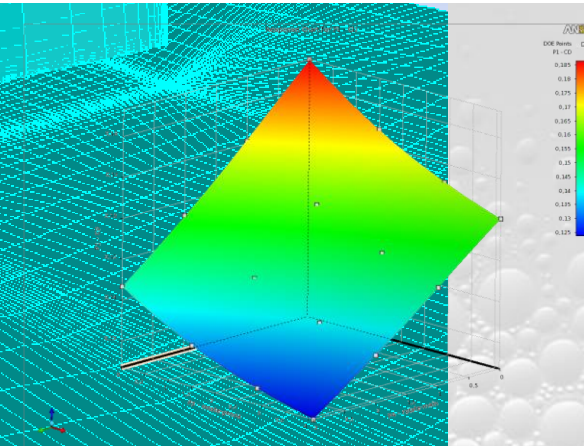
Motorbike Windshield (Bricomoto, MRA)



Sails Trim (Ignazio Maria Viola, University of Newcastle)





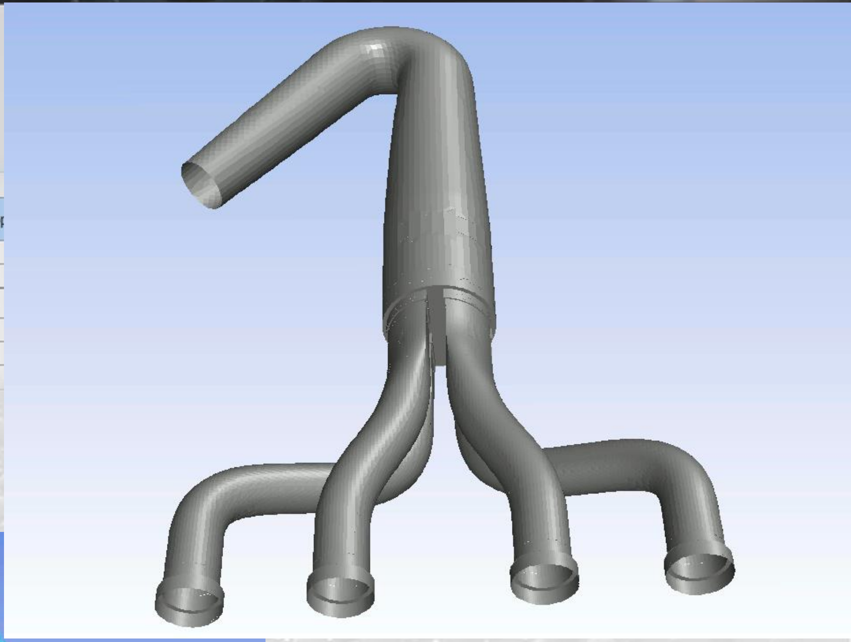


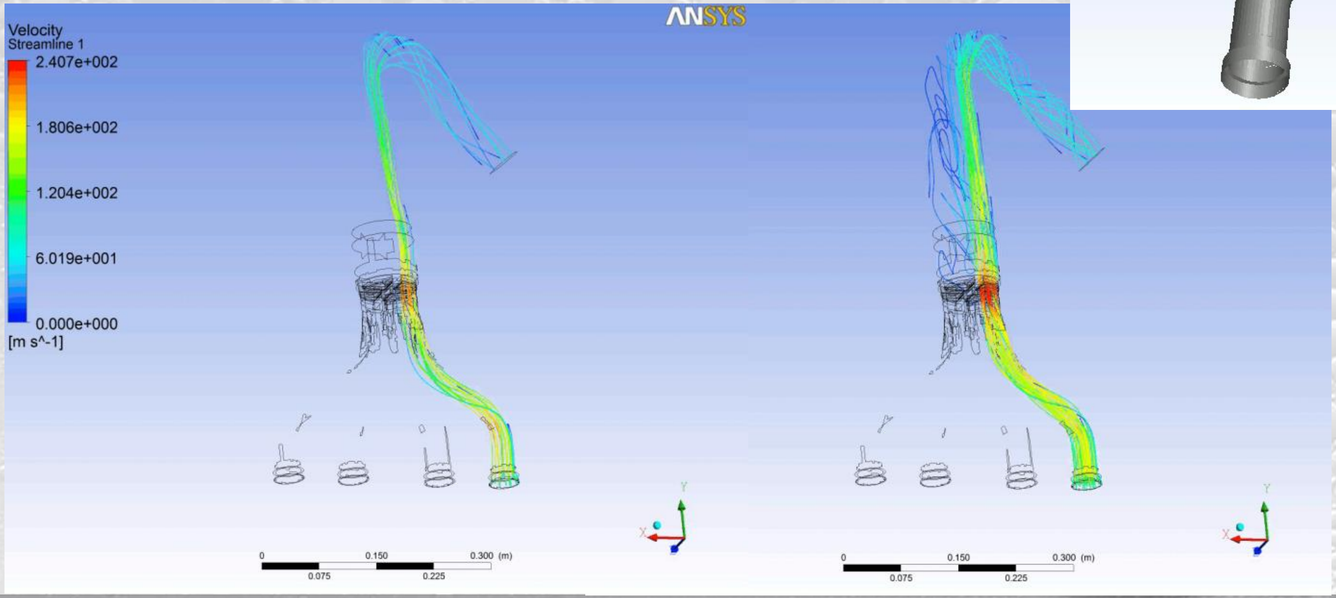
Morphing Preview (A=0)

Newcastle University
Yacht and superyacht consultancy and research
school of marine science and technology
ignazio.viola@ncl.ac.uk

Exhaust manifold

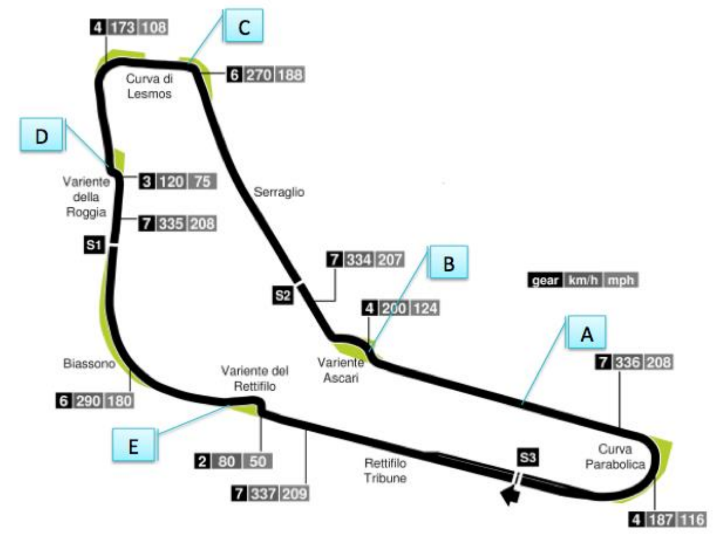
	A	B	C	D	E	F	G	H	I
1	Name	P5 - Pipe1Curve1	P6 - Pipe2	P7 - Pipe4Curve1	P8 - Pipe3	P1 - PressureDrop1	P2 - PressureDrop2	P3 - PressureDrop3	P4 - PressureDrop
2						Pa	Pa	Pa	Pa
3	Current	4	4	4	4	12892	11366	13028	16619
4	DP 1	3	3	3	3	12882	11247	13487	16731
5	DP 2	2	2	2	2	12897	11546	13554	16911
6	DP 3	1	1	1	1	13403	11477	13920	17666
7	DP 4	0	0	0	0	13555	11750	13967	17718

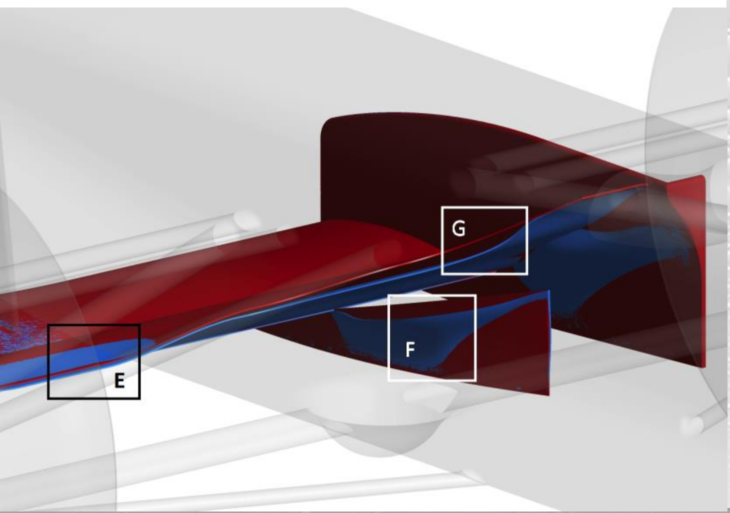


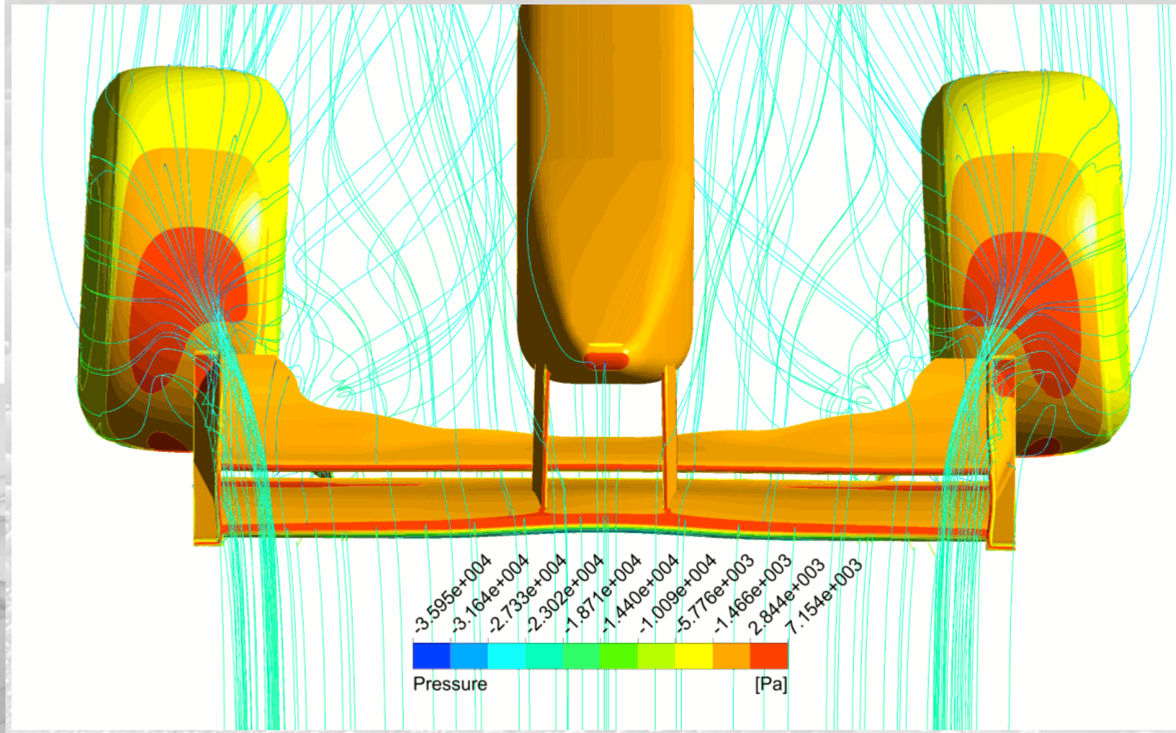


UNIVERSITA' degli STUDI di ROMA
TOR VERGATA

Steering wheels – lap time optimisation

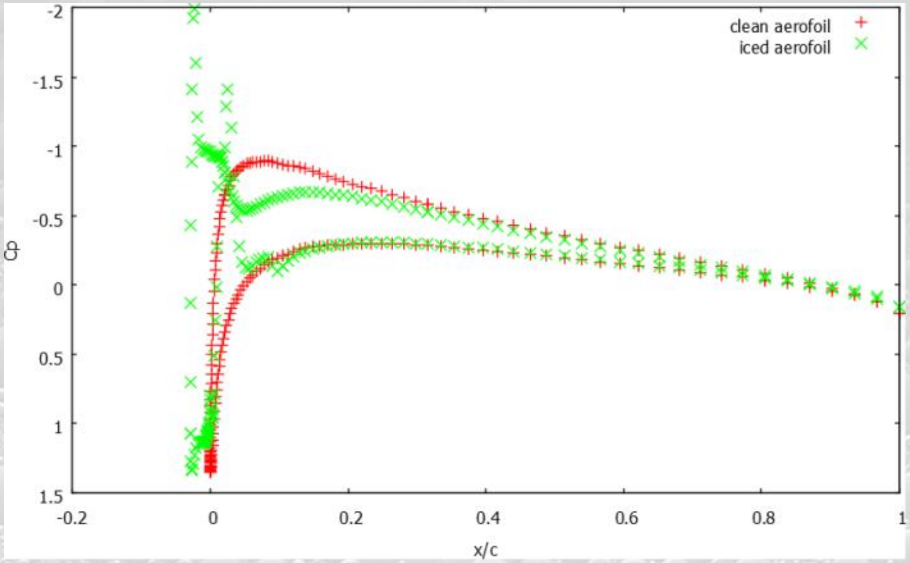


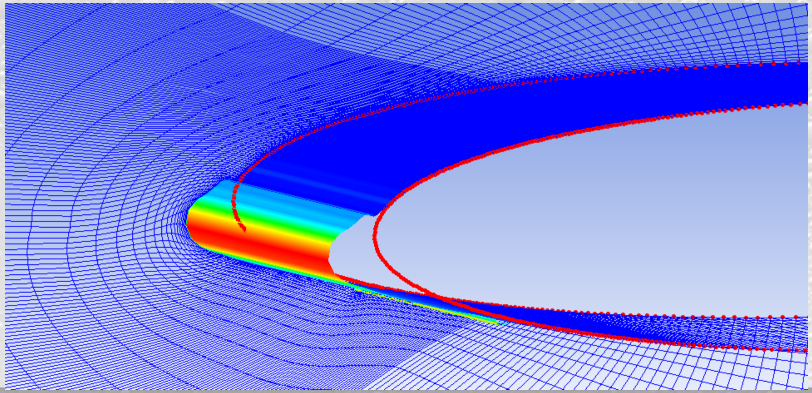


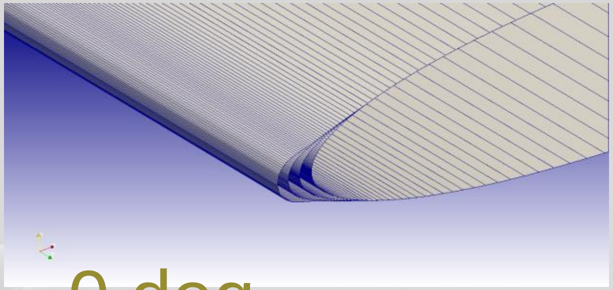


2D - NACA 0012

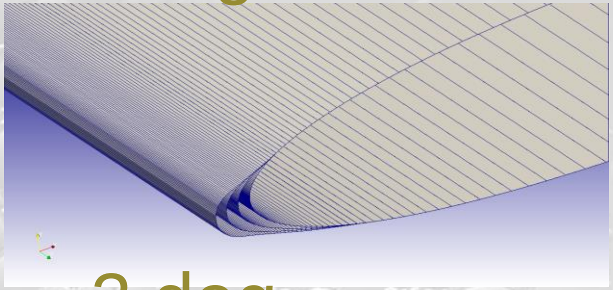
7, 14, 21 min.



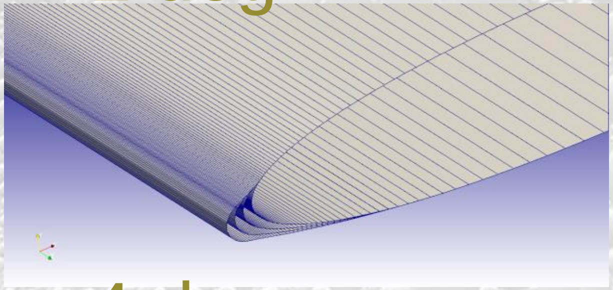




0 deg



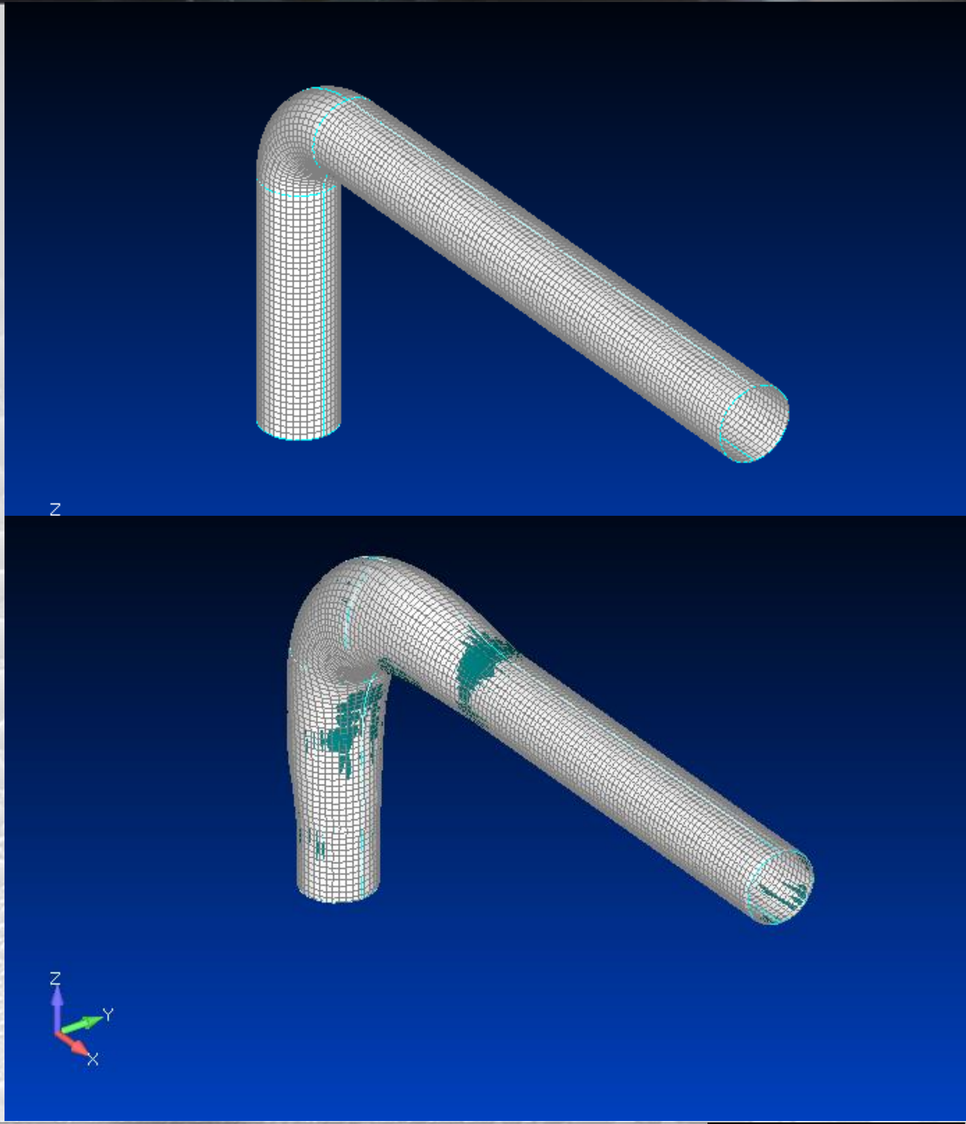
2 deg



4 deg

Adjoint Self Sculpting

- 90 deg bend optimization
- New shape is sculpted using **adjoint data**
- Original geometry (2 cylinders and a torus) is transformed in **NURBS**
- **NURBS** are morphed using the **back to CAD** tool of RBF Morph

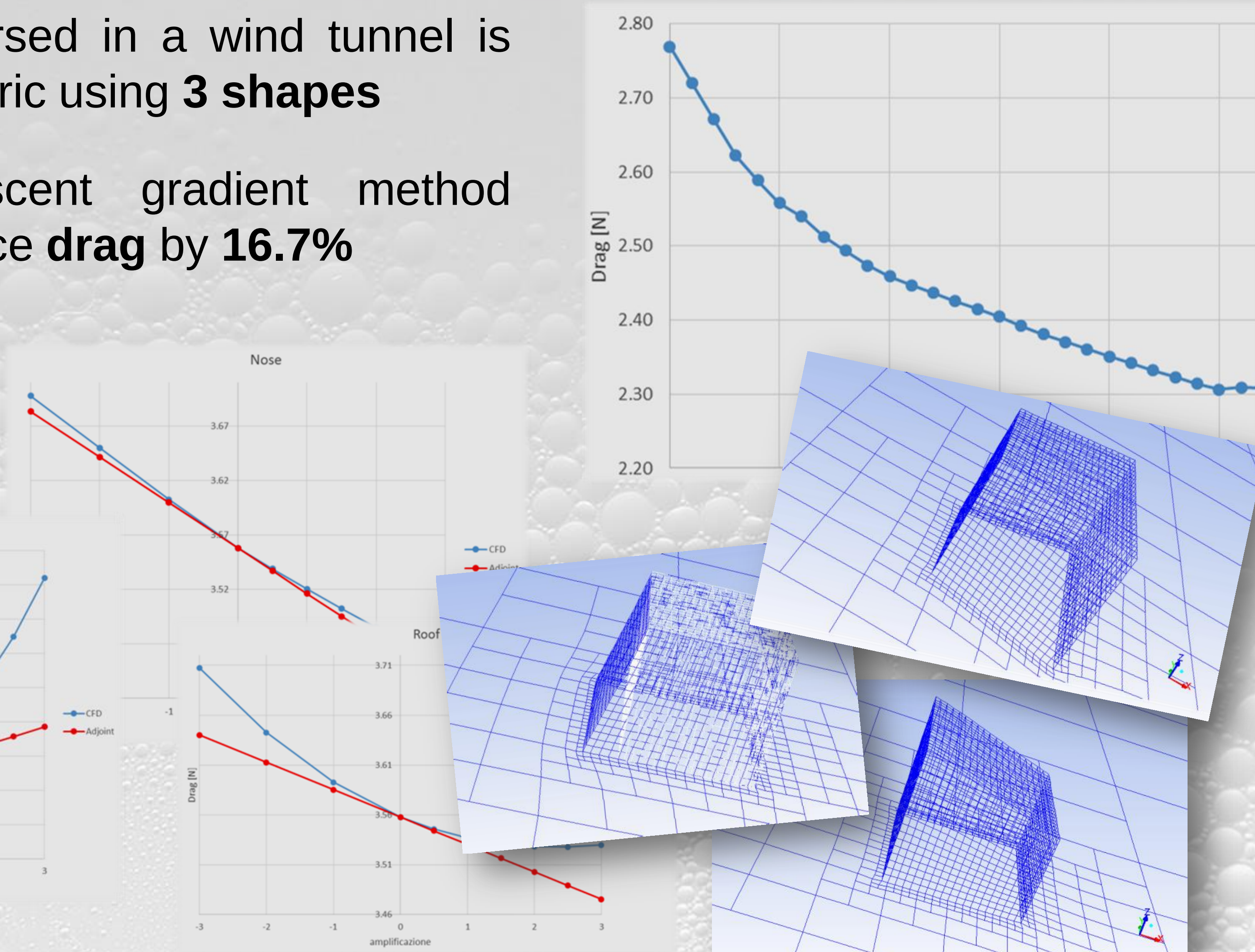


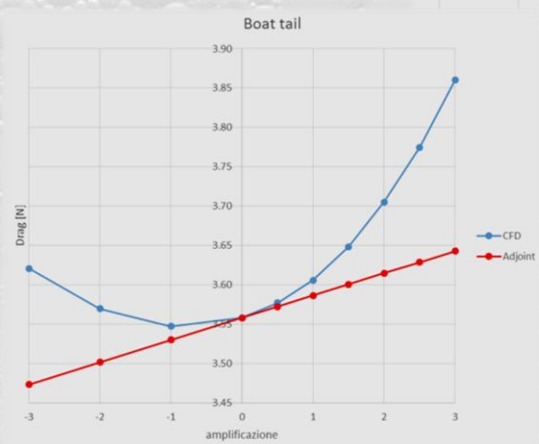
Adjoint preview

(rbf-morph)™
Welcome to the World of Fast Morphing!

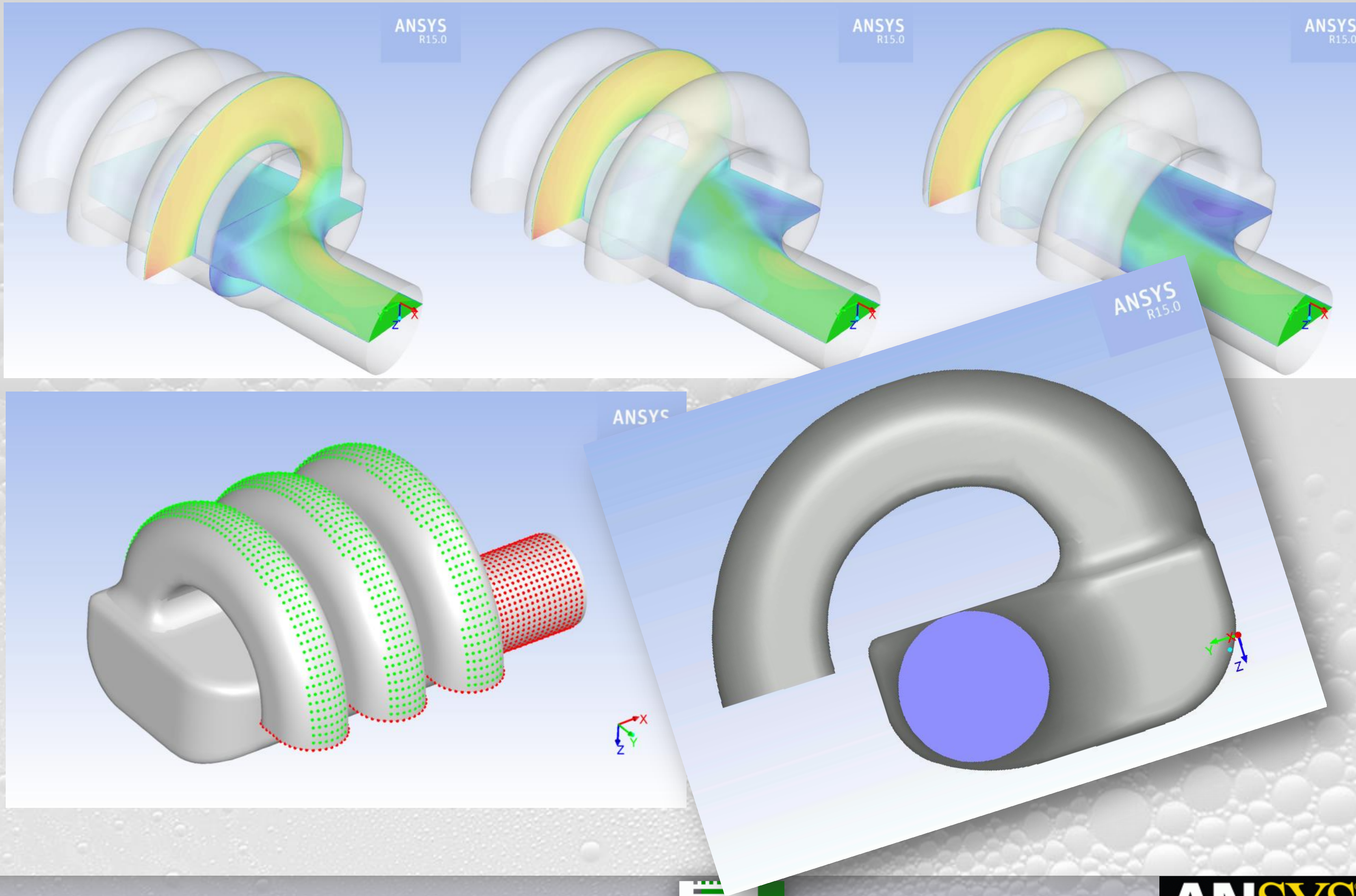
- A cube immersed in a wind tunnel is made parametric using **3 shapes**
- Steepest descent gradient method allows to reduce **drag** by **16.7%**

$$\alpha_j = -\beta \frac{dI}{dx_j}$$

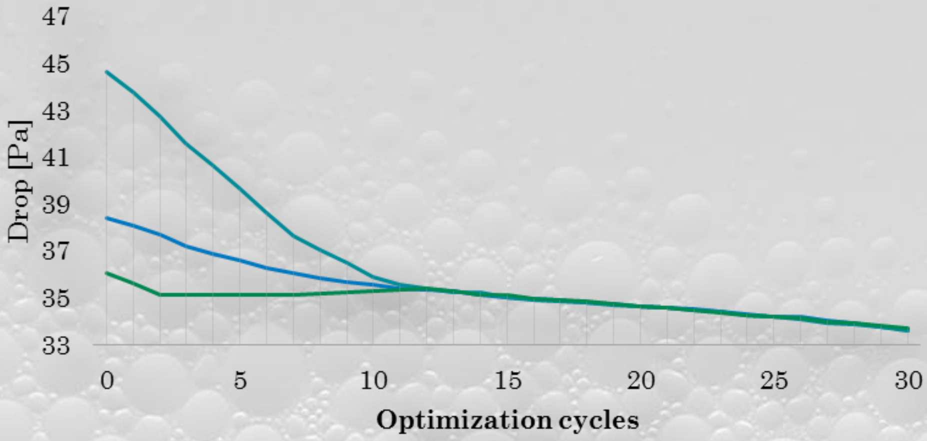




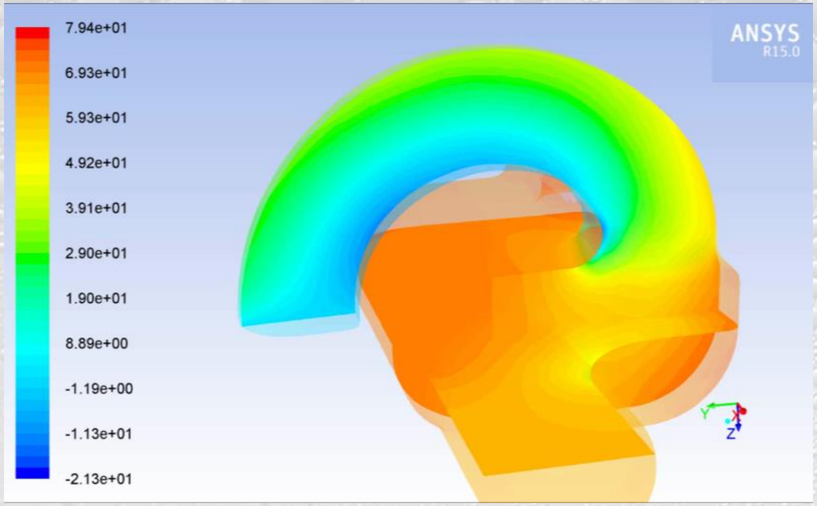
Engine Air box Adjoint shape optimization

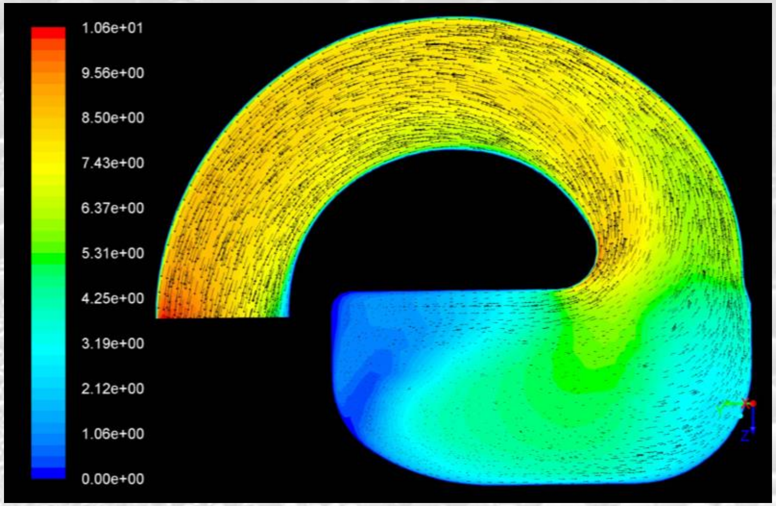


Engine Air box Adjoint shape optimization



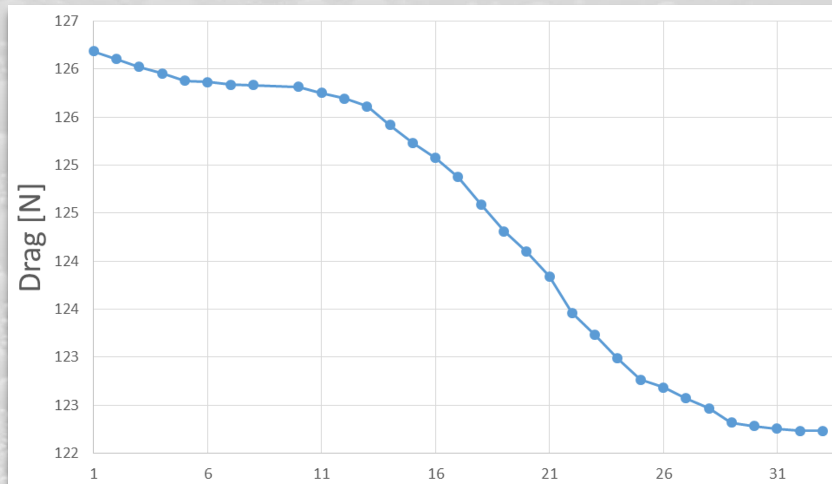
	Mean pressure Drop [Pa]	Unbalance
Baseline	39.7	12.45%
Optimized	33.635	0.12%
Reduction	15.3%	99.0%

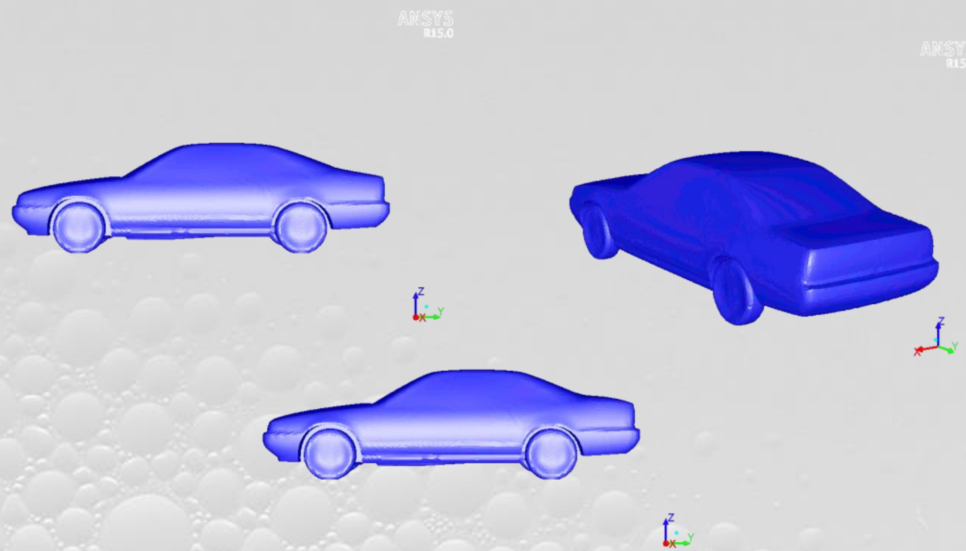




Sedan car

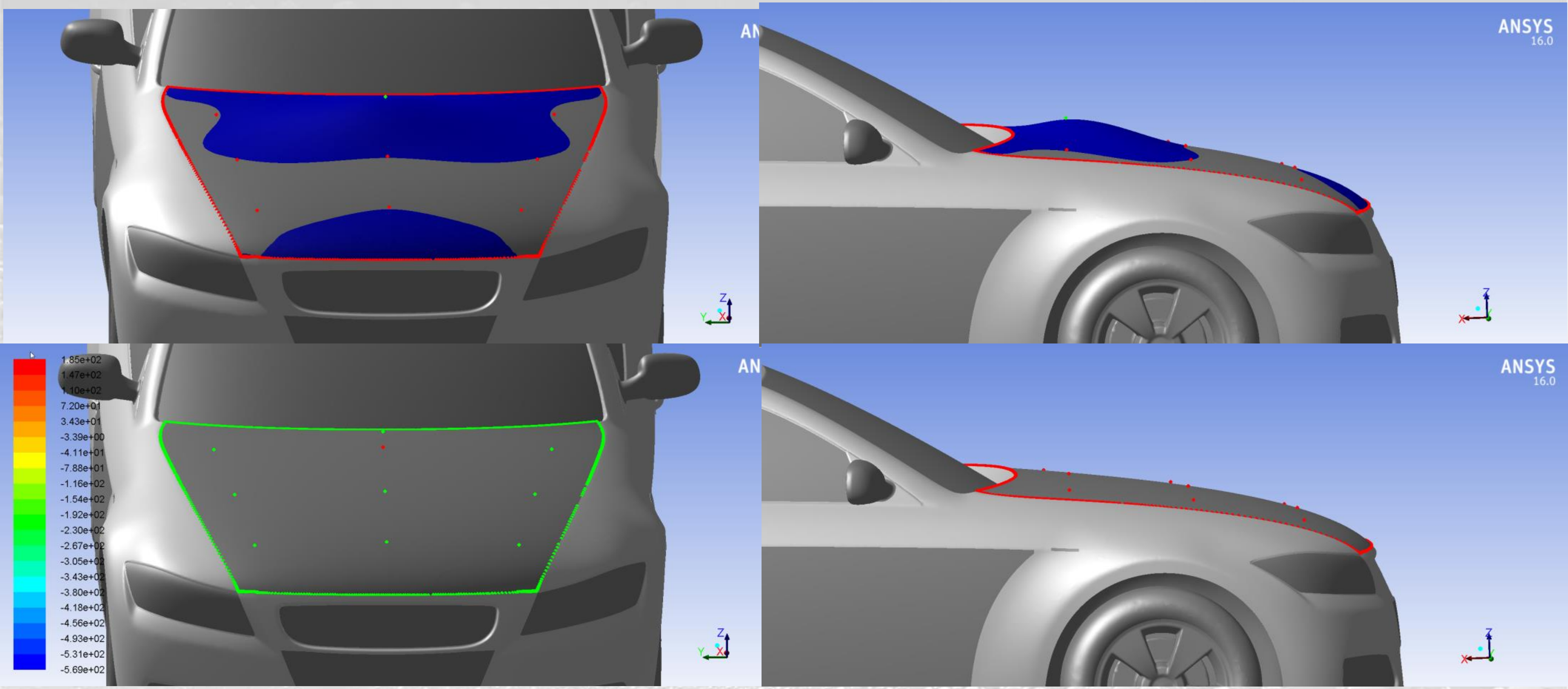
- A **3.13%** drag reduction is achieved after 33 cycles





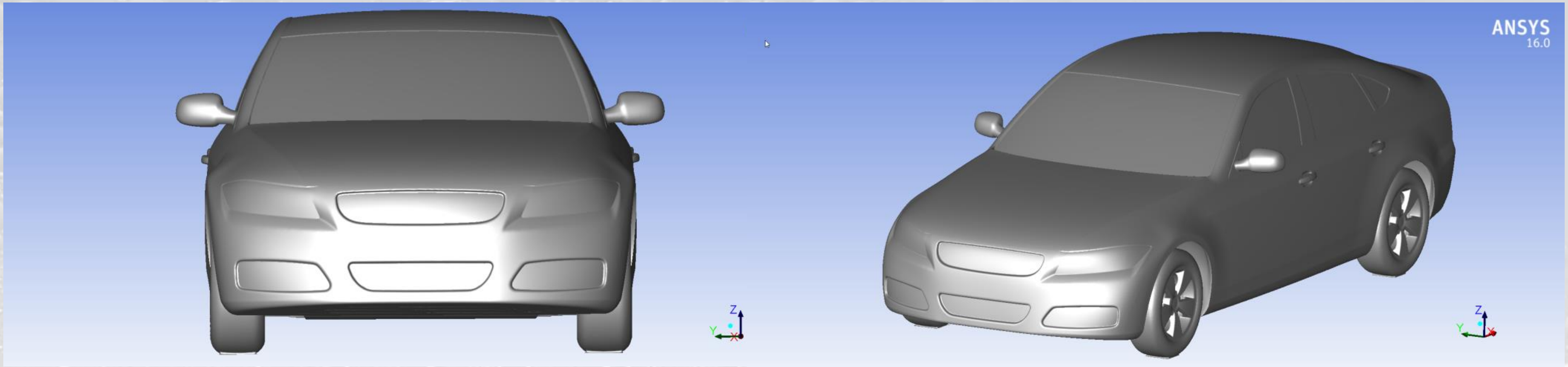
$$\alpha_j = -\beta \frac{dI}{dx_j}$$

DrivAer adjoint sensitivity

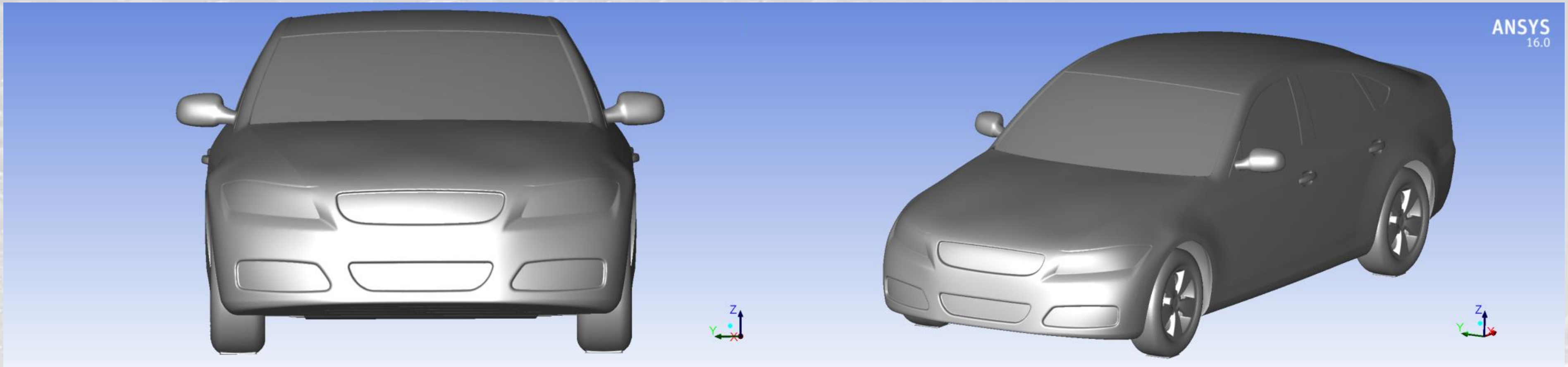


DrivAer adjoint sensitivity

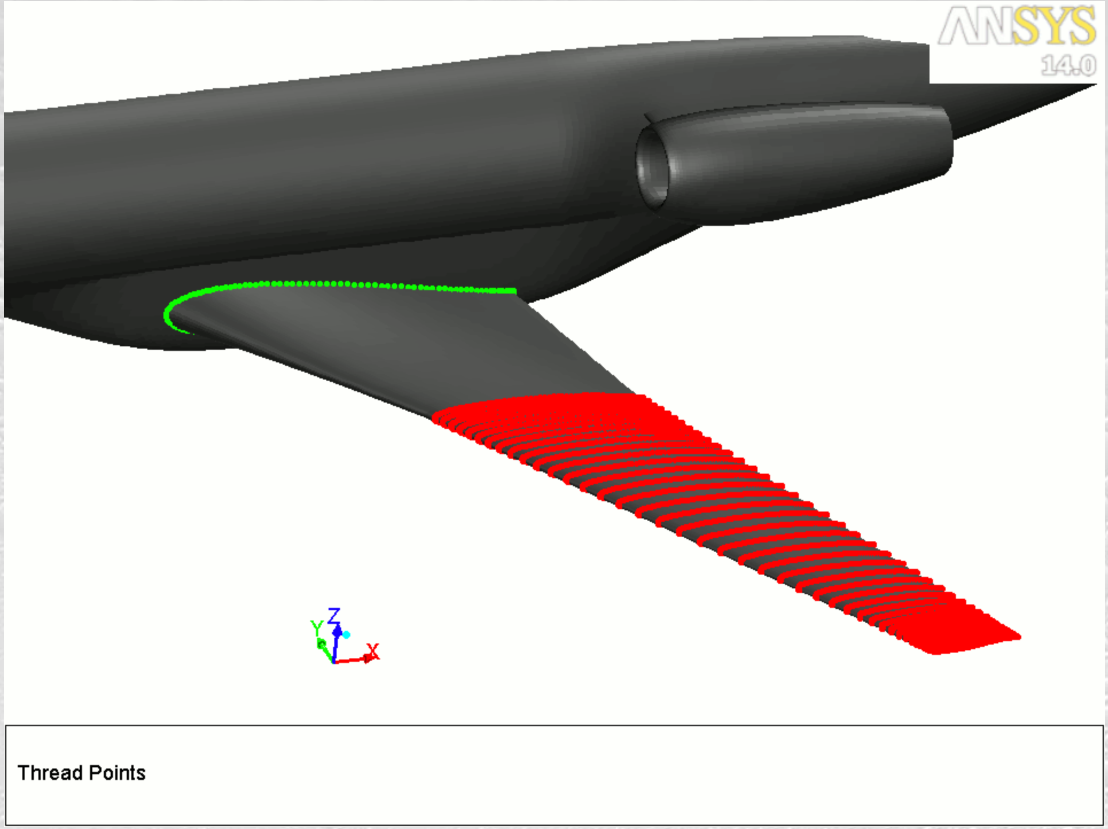
- Adopting a maximum displacement of 5 mm and updating the surface according to gradient data a 0.5% drag reduction is expected.



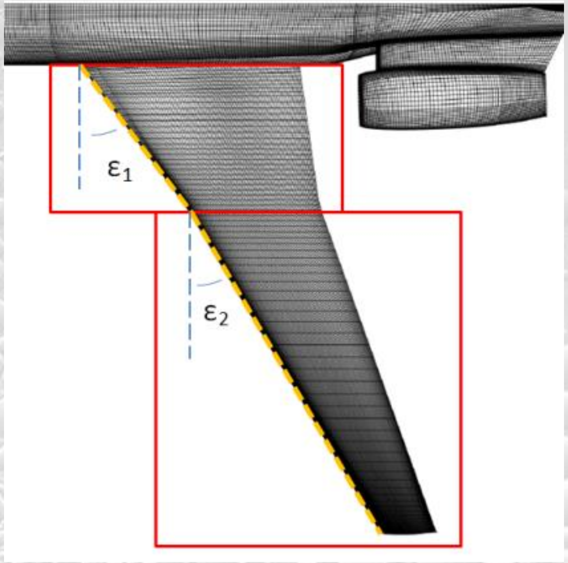
DrivAer adjoint sensitivity



Wing sweep parameterization

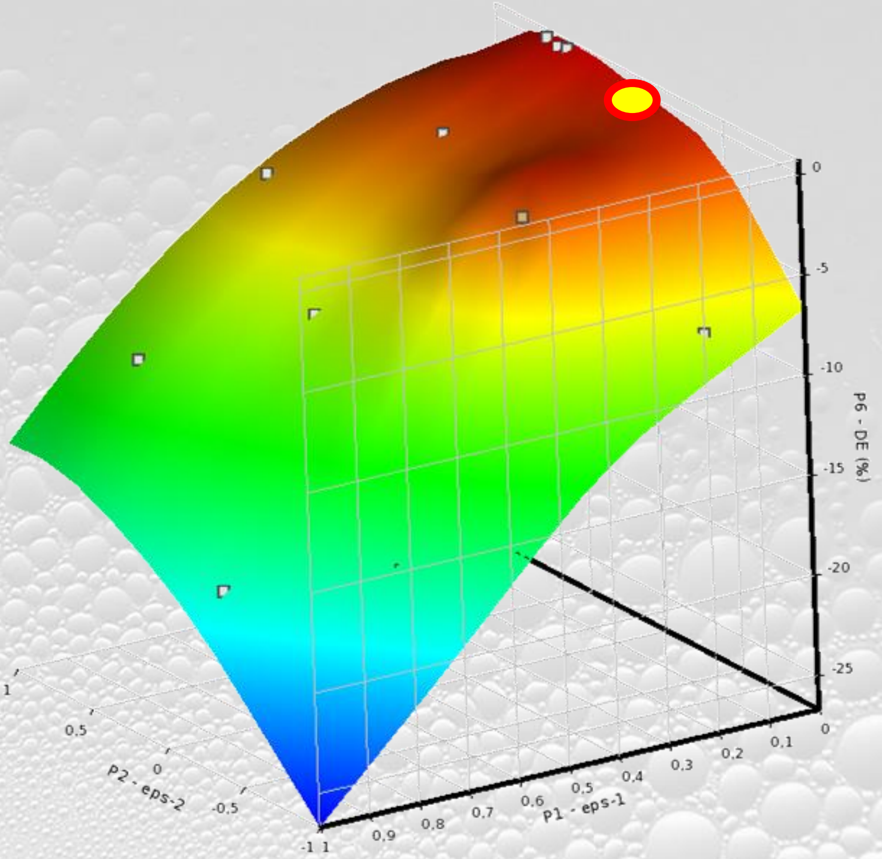


PIAGGIO AEROSPACE

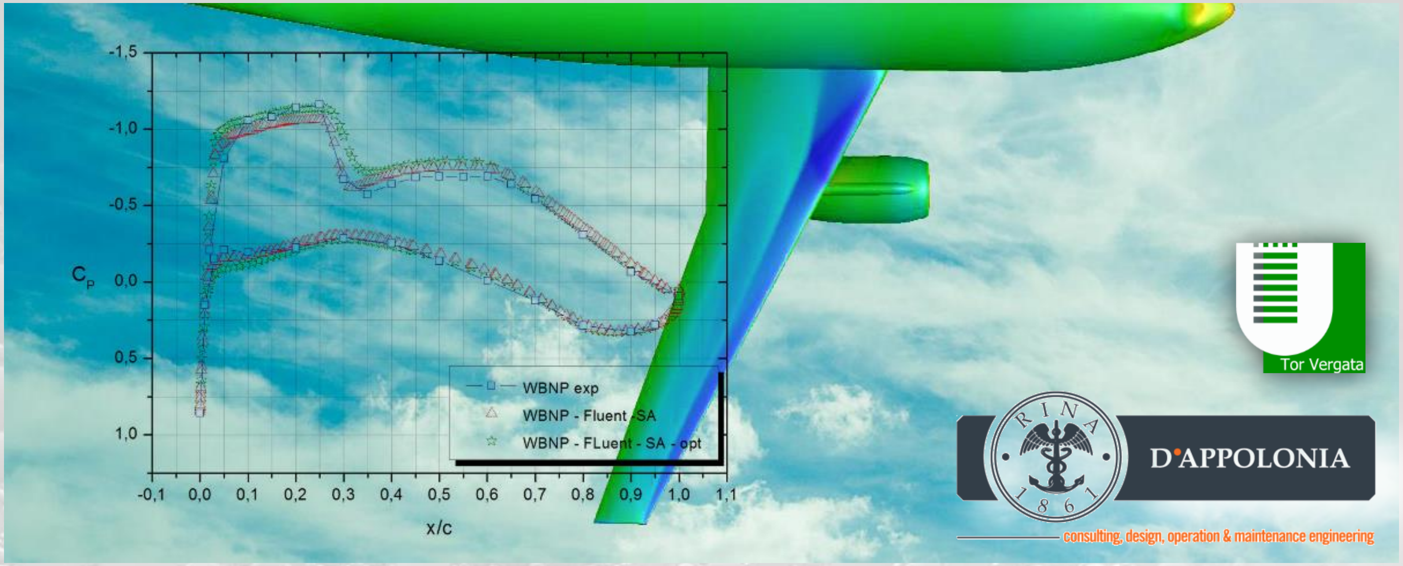


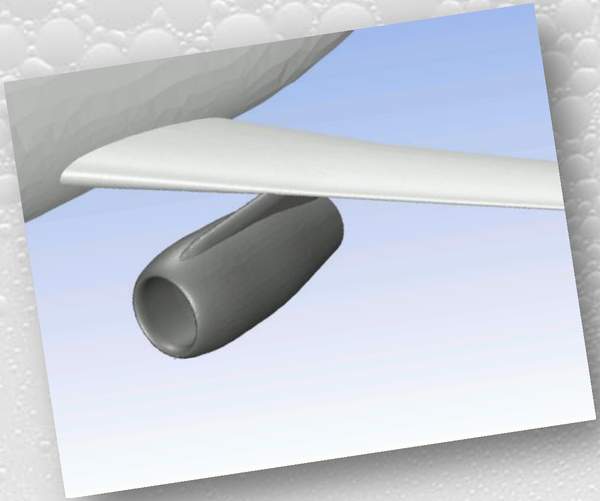
Sweep angles optimization

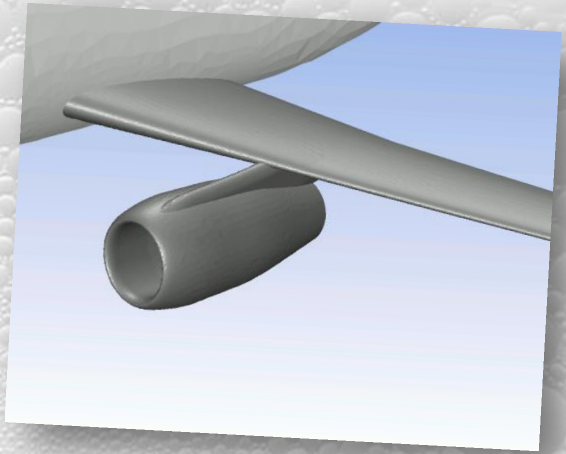
- **DOE + RSM** approach
- **2** design variables
- Maximization of **Efficiency** in cruising conditions
- Slight improvement (less than **1%**).



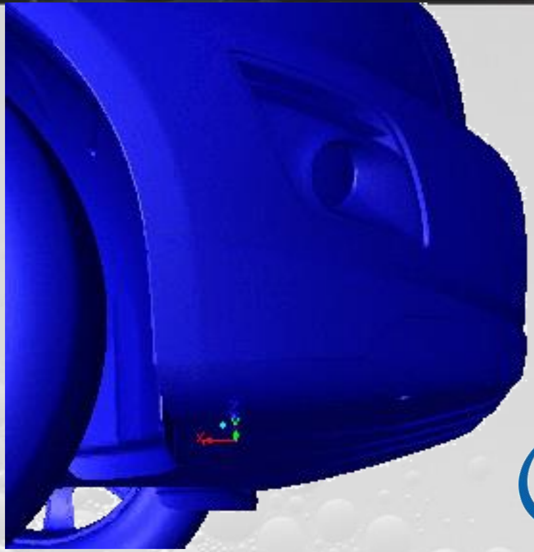
Optimization of nacelle

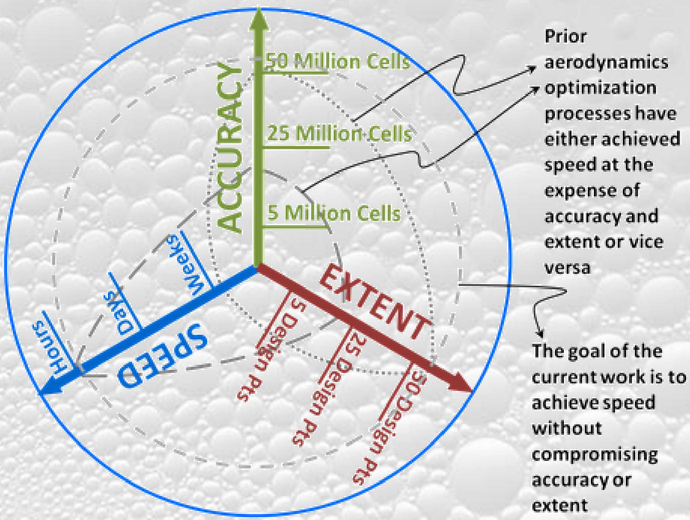




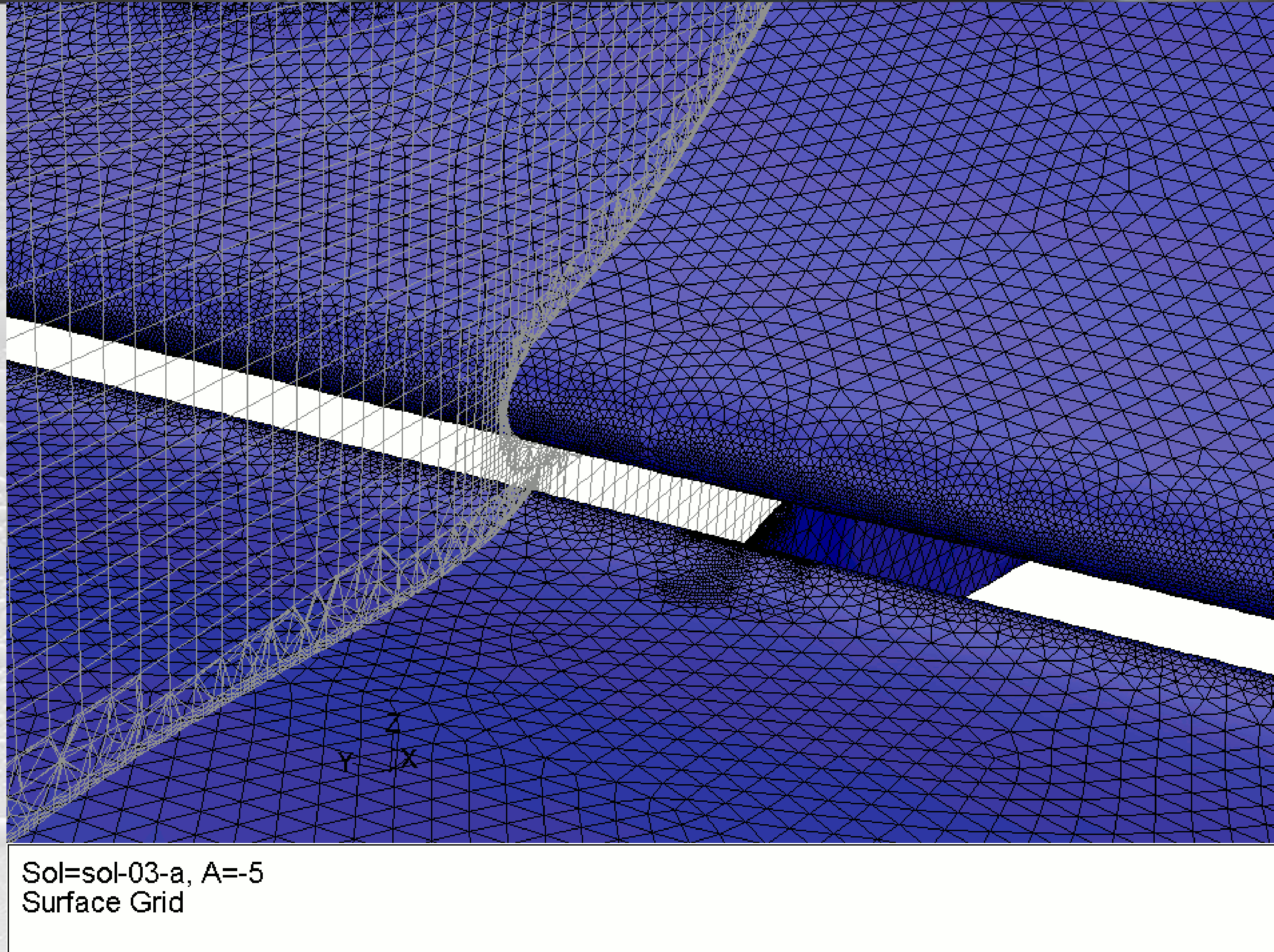


50:50:50 Project Volvo XC60 (Ansys, Intel, Volvo)

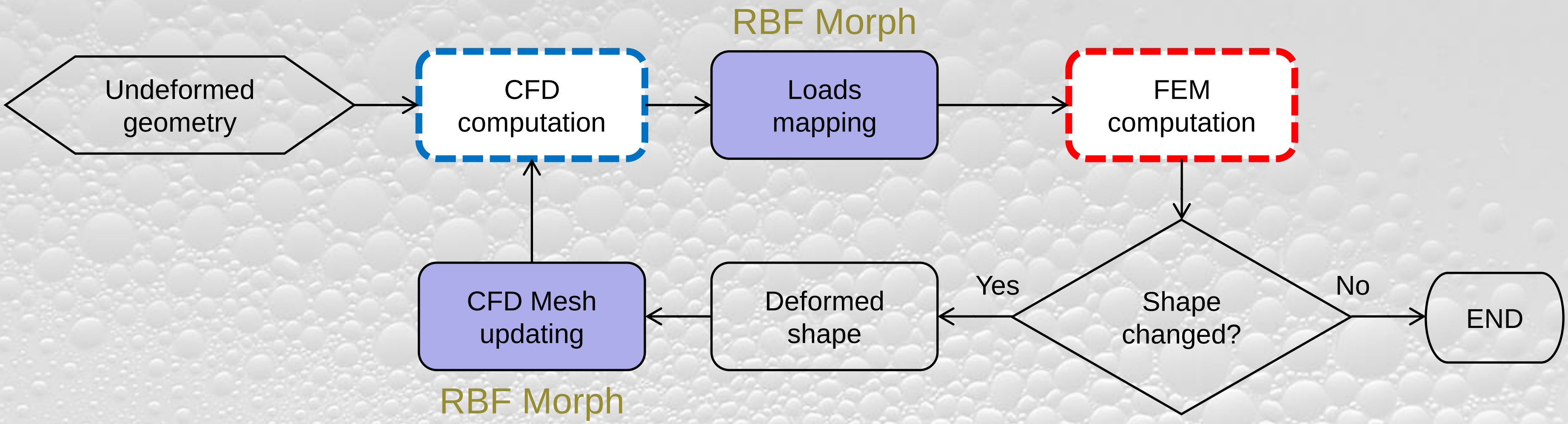




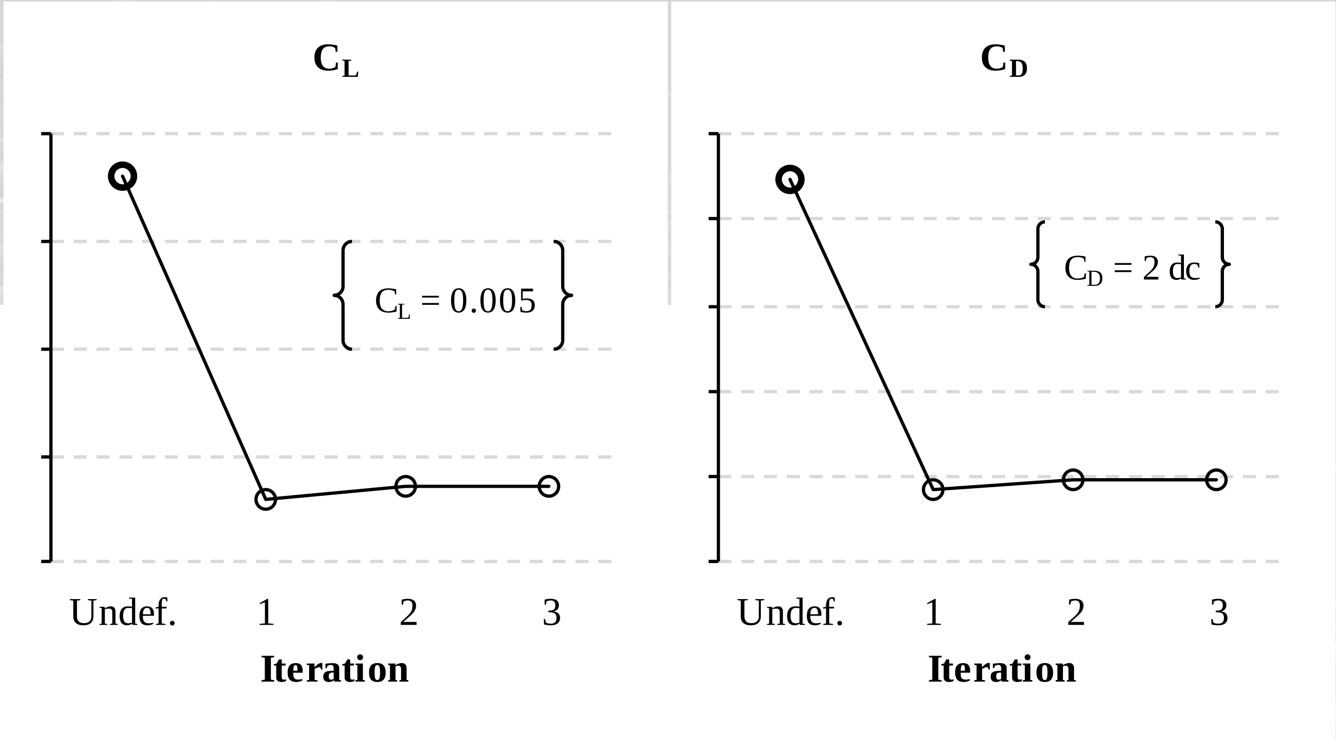
Generic Formula 1 Front End

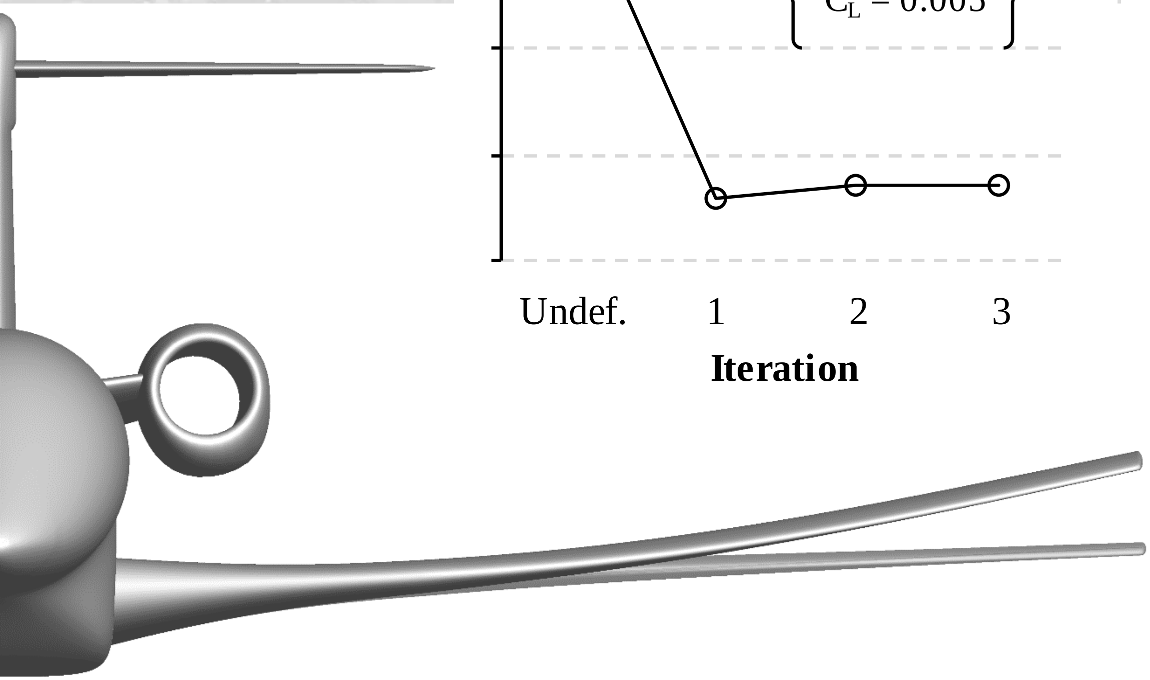


2 ways FSI procedure



Convergence histories



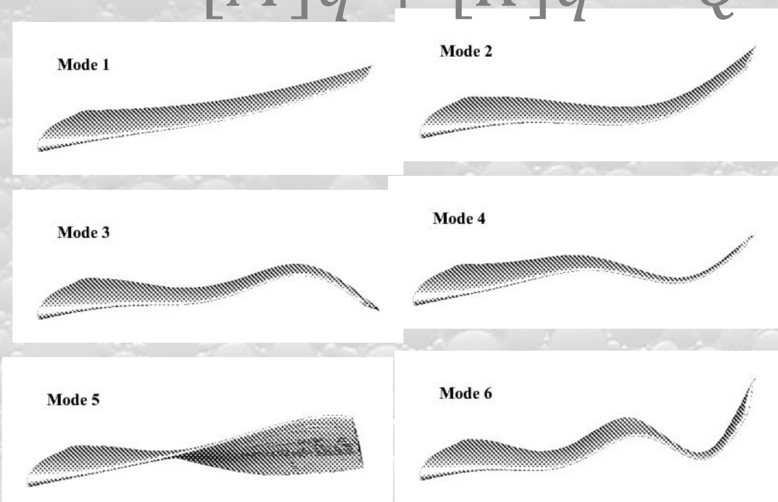


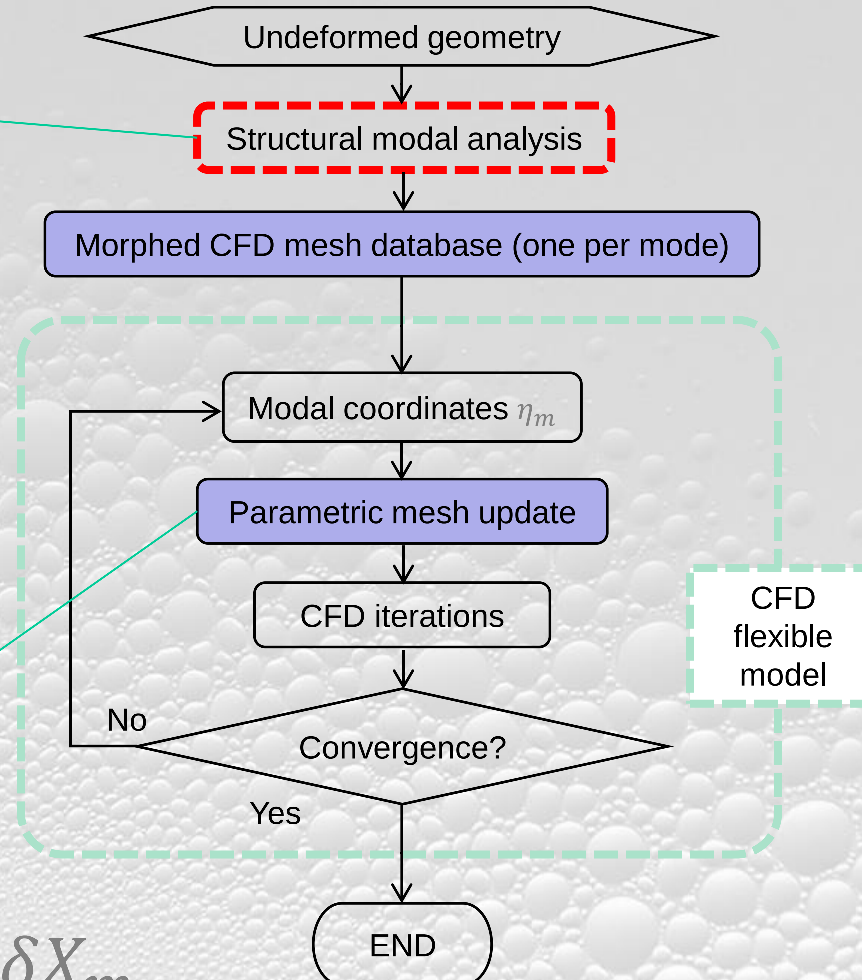
~1% of the wing exposed semispan (figure amplified)

FSI modal approach

undamped vibration modes

$$[M]\ddot{q} + [K]q = Q$$

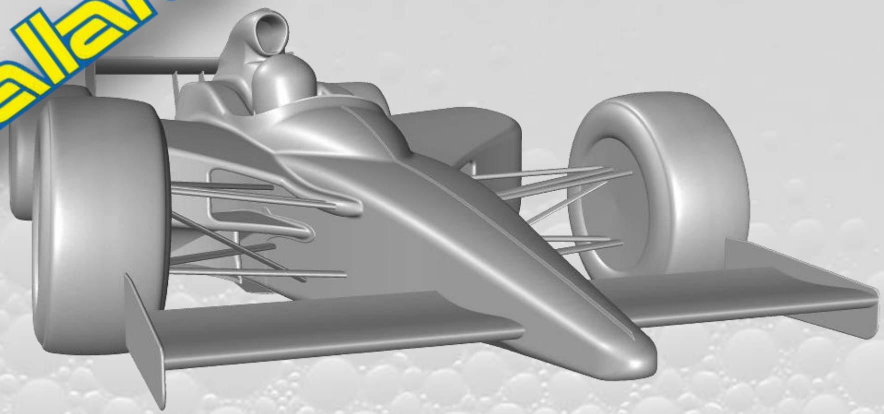


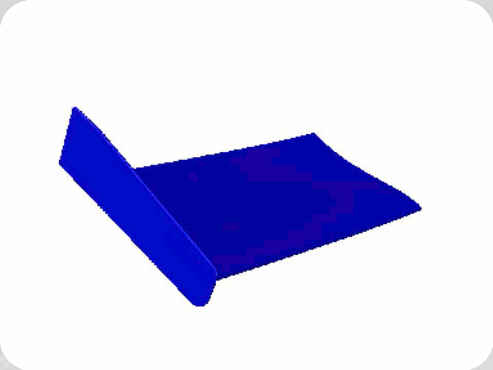


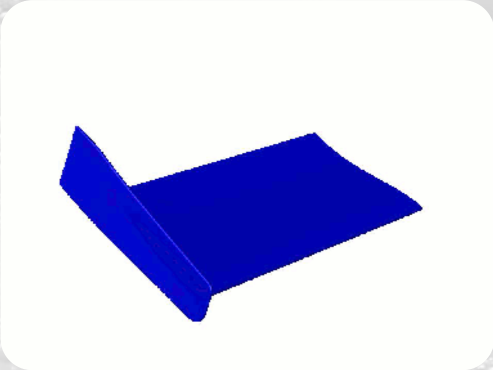
Parametric mesh formulation

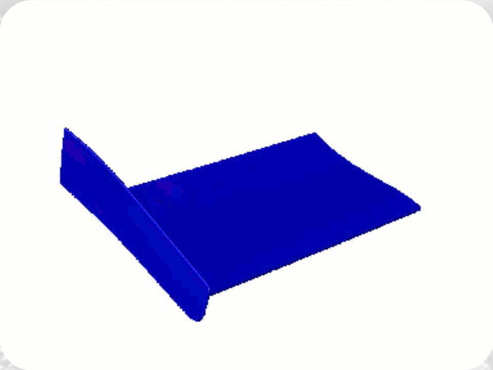
$$X_{CFD} = X_{CFD0} + \sum_{m=1}^{n_{modes}} \eta_m \delta X_m$$

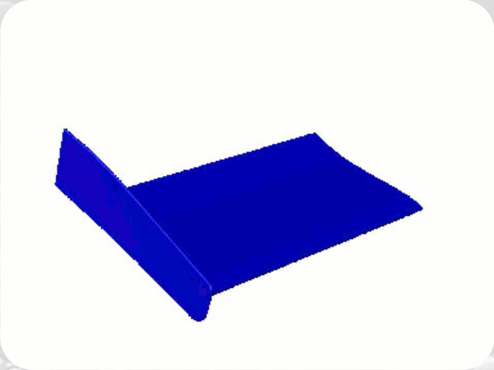
FSI analysis on a indy race car



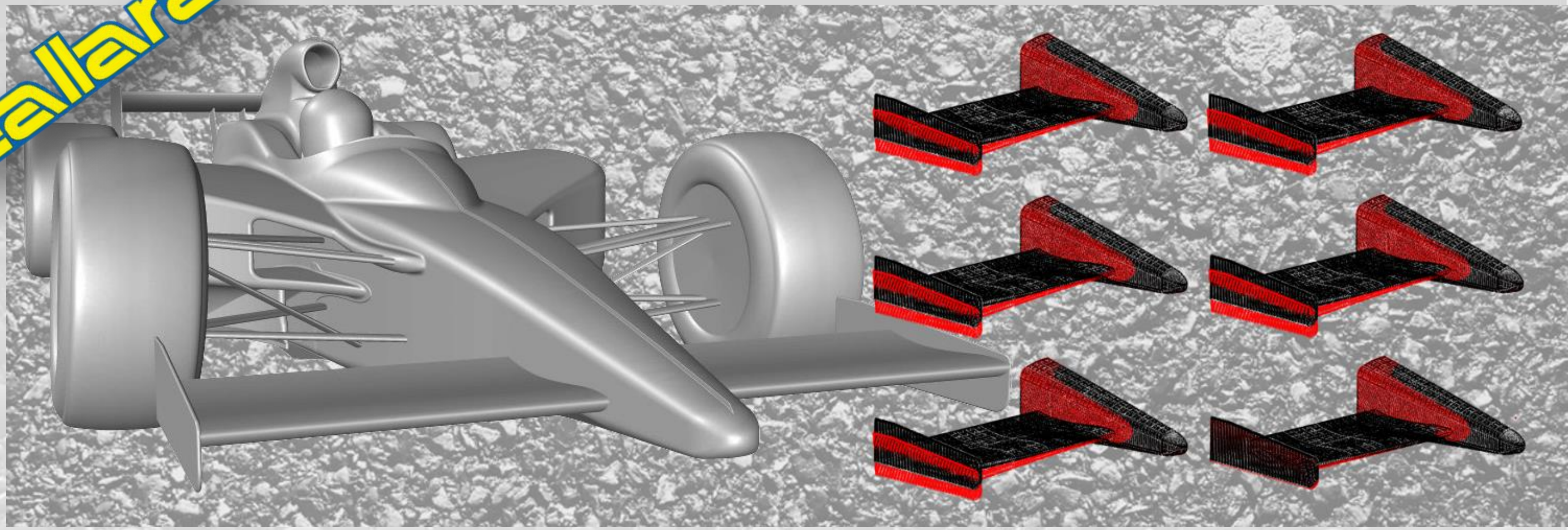






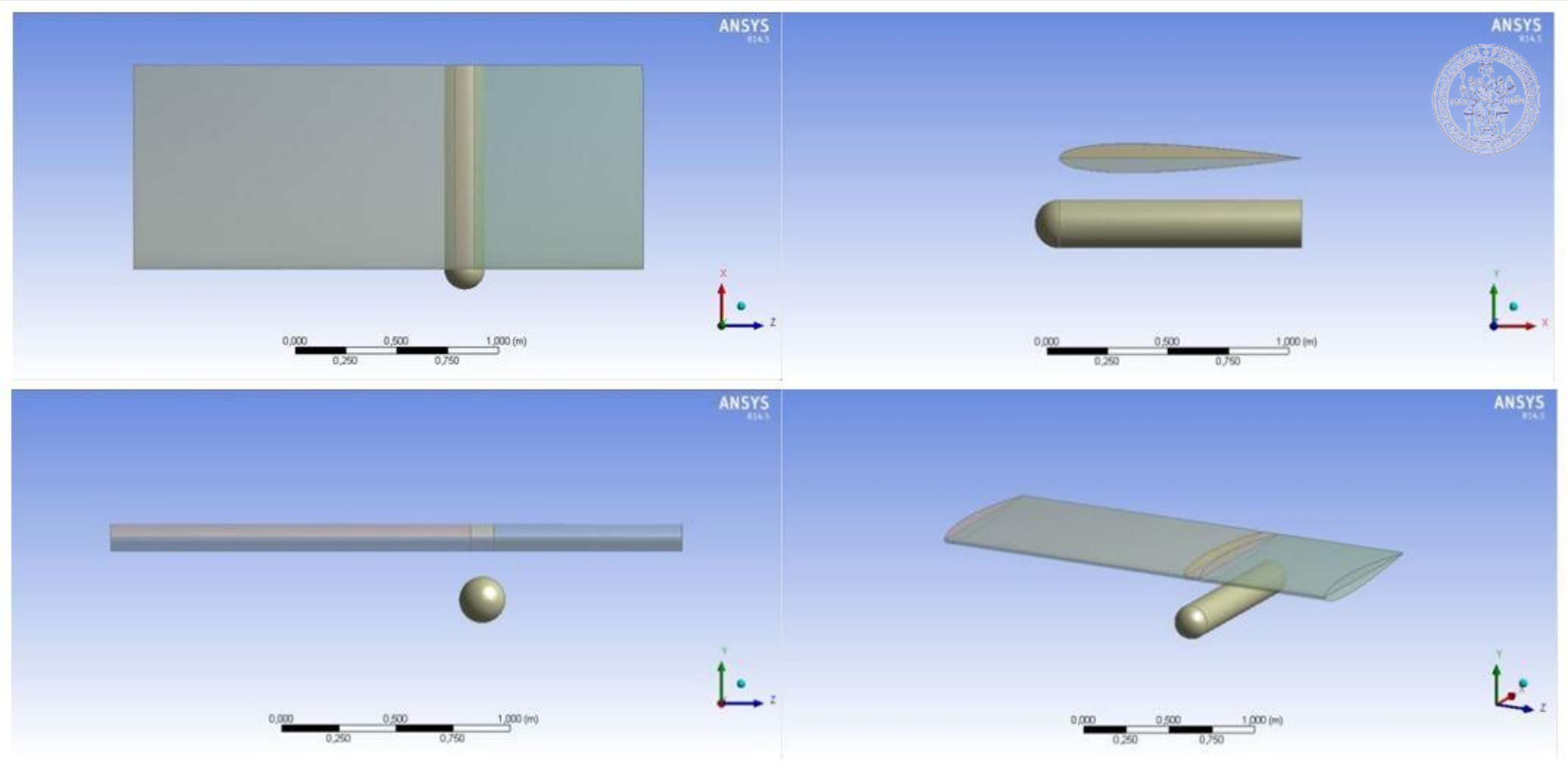


FSI analysis on a indy race car

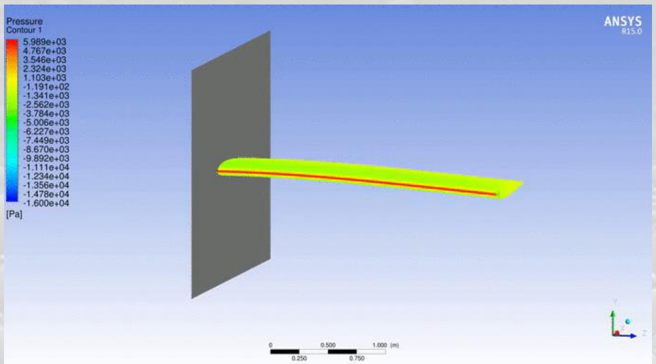


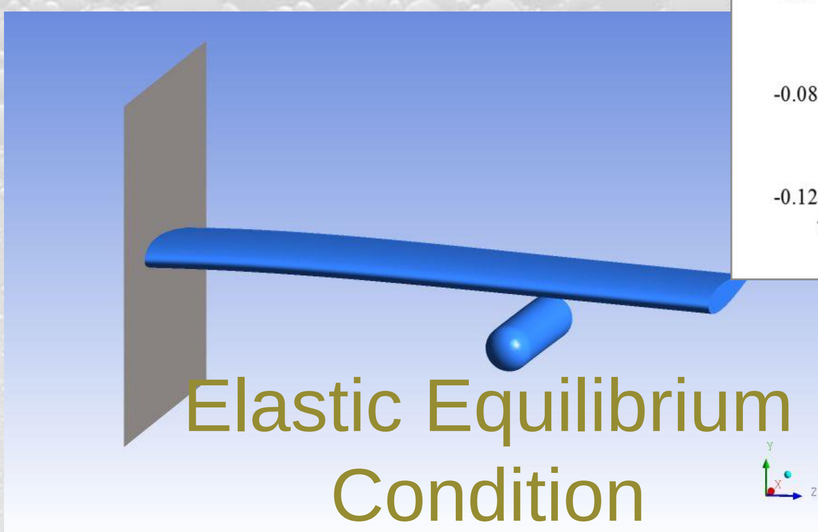
Modes used	Maximum displacement (mm)	Maximum difference (mm)	Maximum error (%)
1	5.941	4.946	8.3
2	5.898	3.817	6.5
3	5.584	1.483	2.7
4	5.56	7.722	1.4
5	5.555	0	0

Unsteady FSI modal analysis

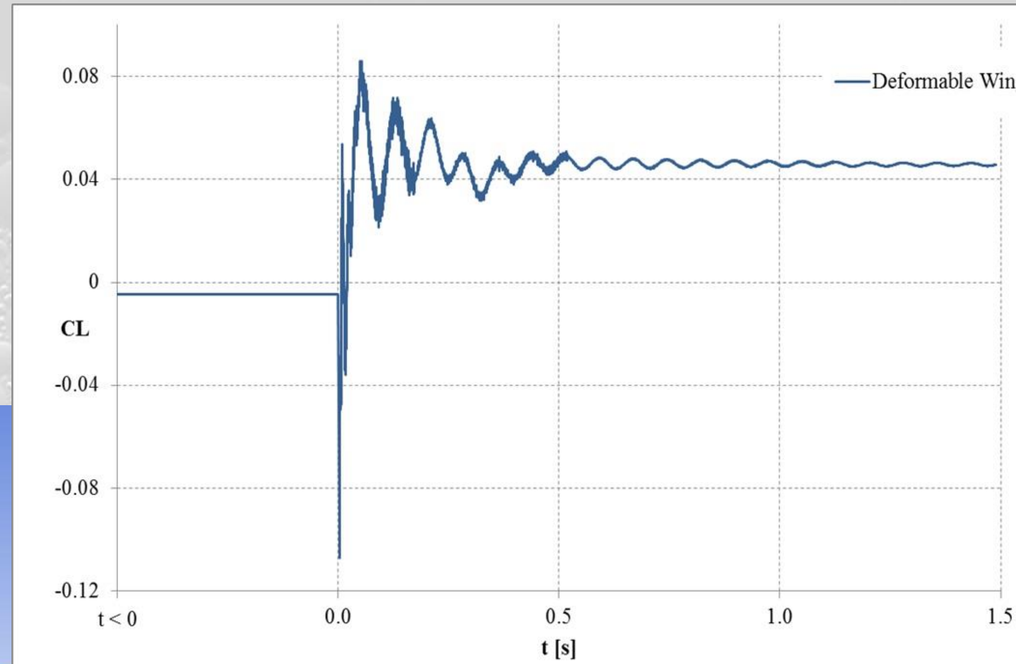


Lift time history





Elastic Equilibrium Condition



Thank you very much!

Dr. Corrado Groth

E-mail: info@rbf-morph.com

Web: www.rbf-morph.com

YouTube: www.youtube.com/user/RbfMorph