ESTECO USERS' MEETING
um 2020
30 SEPT - 1 OCT

A Gradient Based Optimization Workflow Based on CFD Adjoint Solver and Advanced RBF Mesh Morphing

Marco Evangelos Biancolini
Ubaldo Cella

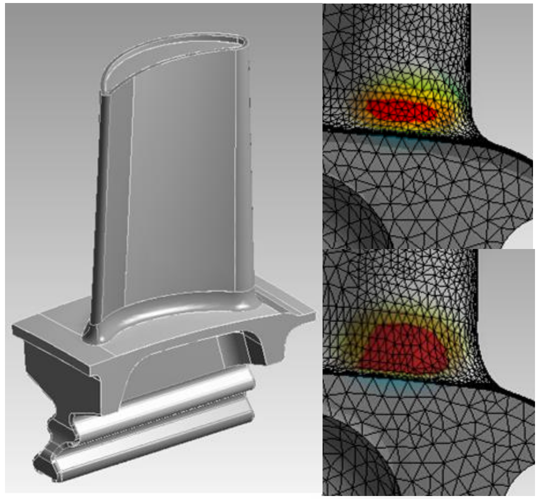
RBF Morph S.r.l.

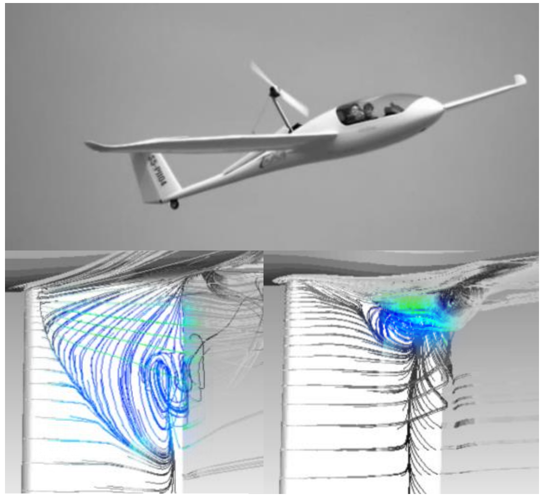
Alessandro Nizzoli

Automobili Lamborghini S.p.a.

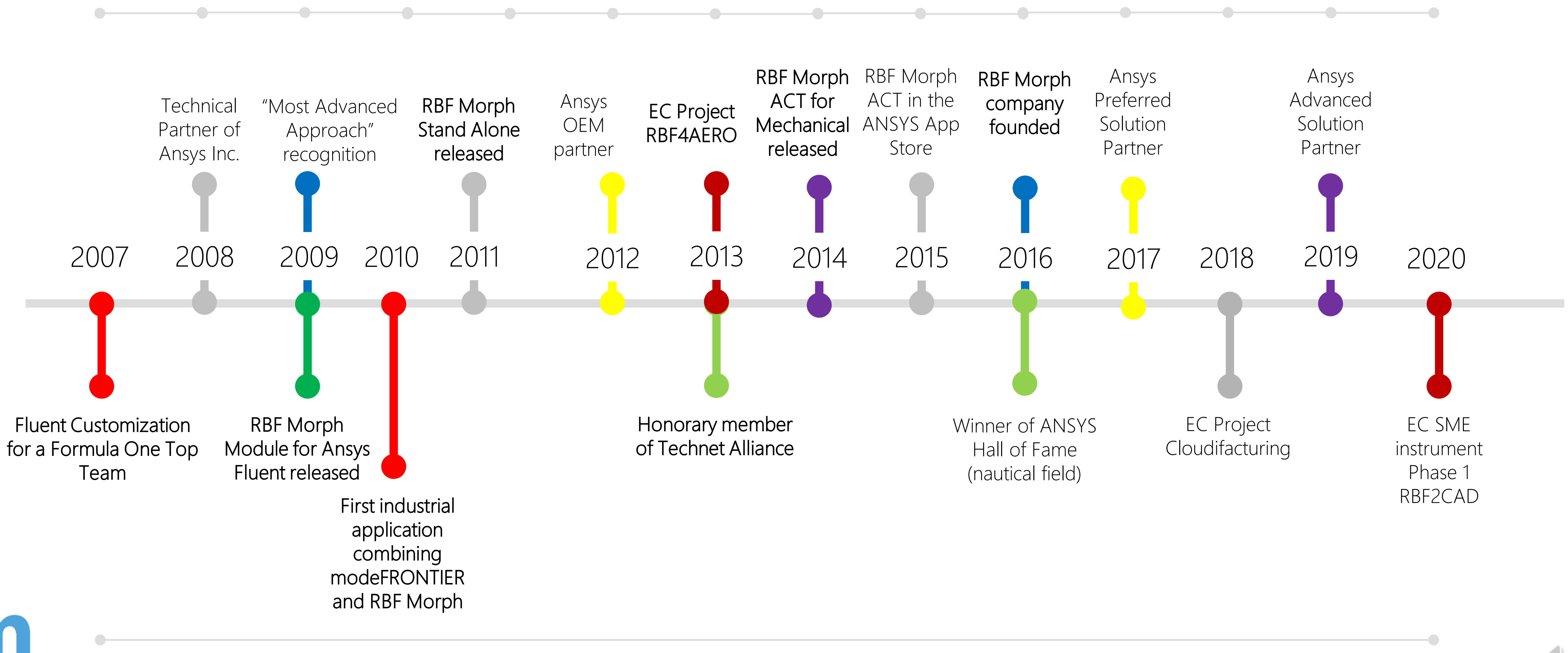
Outline

- RBF Morph technology and software solutions for shape optimization
- Optimization-driven design adopting MDO
- Gradient based method based on adjoint solver
- Proposed workflow with modeFRONTIER, Fluent and RBF Morph
- Industrial example: optimal shape of an airbox
- Conclusions





10+ years of RBF Morph



Industries served (100+ institutions)

- Automotive
- Aerospace & Defence
- Nautical & Marine
- Healthcare & Medical
- Energy
- Oil & Gas

Radial Basis Functions mesh Morphing

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

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

Biancolini, M. E. (2017). *Fast radial basis functions for engineering applications*. Springer International Publishing.

2020

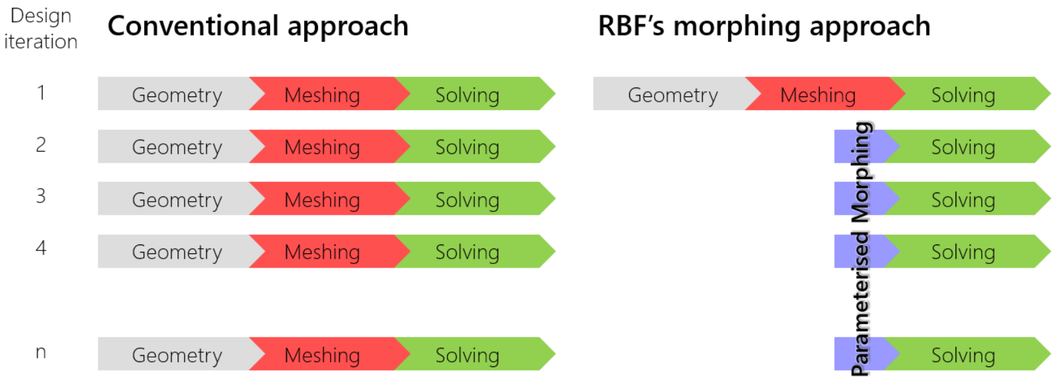
Parametric CAE models

CAE models supported includes flow analysis (CFD) and structural analysis (FEM)

RBF Morph makes the CAE model parametric with respect to the shape.

Works for any size of the mesh.

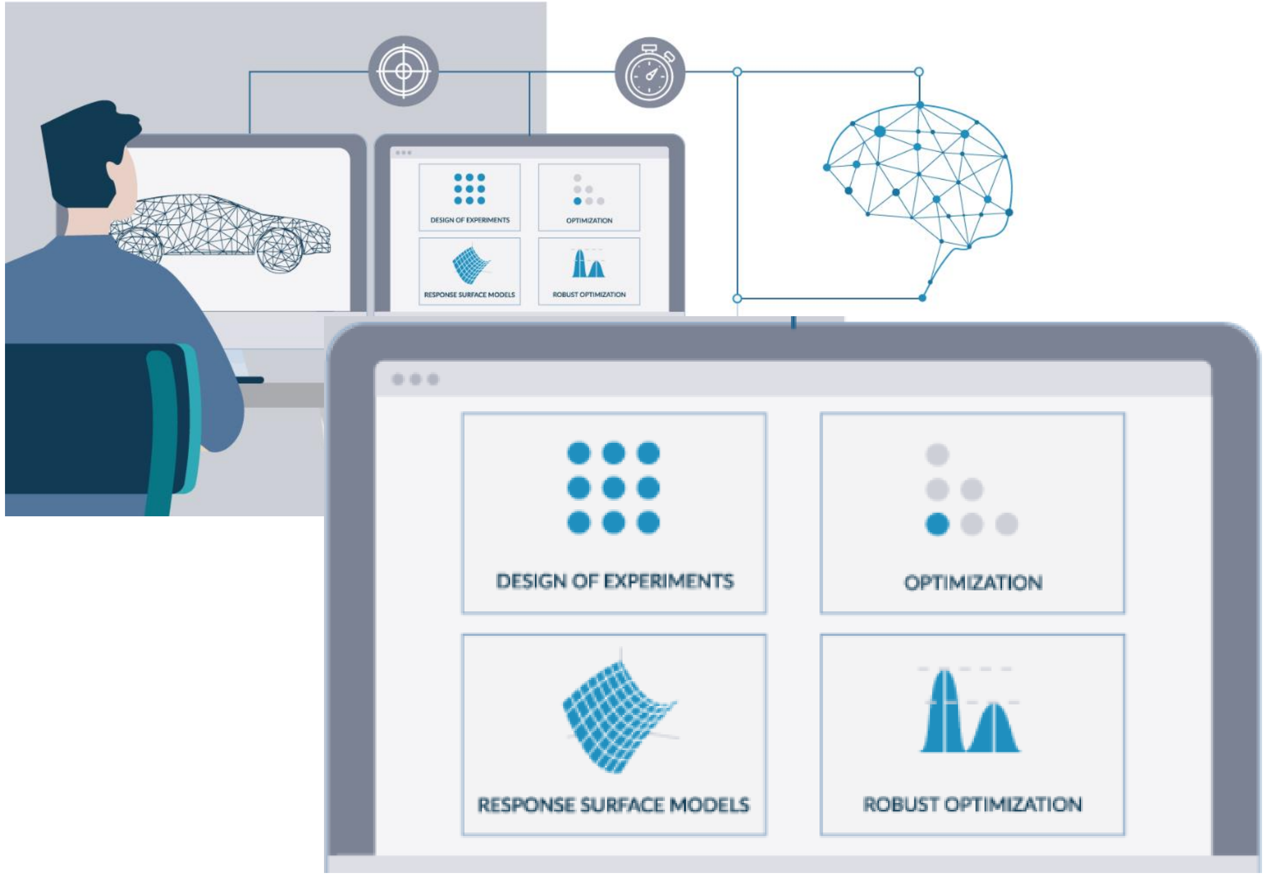
Shape parameters can be steered with the optimizer of choice.



- It's **easy and fast**: shape parameters are defined in the CAE GUI. No need to iterate the CAD.
- The turnaround time of the optimization is usually **reduced by a factor five** (weeks become days)

Optimization-driven design adopting MDO*

- Multi-physics design optimization allows to control at the same time different KPIs of a new system/part
- High fidelity simulations (FEM/CFD) are adopted but the cost is very high!
- The optimization time turnaround can be reduced by
  - Intense use of HPC and fast CAE solvers
  - Faster optimization algorithms



*Courtesy of Esteco. https://www.esteco.com/technology/optimization-driven-design

Optimization-driven design adopting MDO - Algorithms

- MDO can be based on
  - Global methods
  - Local methods based on the gradient
- Hybrid methods get the benefit of both
- The access to derivatives is a key enabler for gradient based methods and for hybrid ones

			MANUAL	SELF-INITIALIZING	AUTONOMOUS
GRADIENT-BASED			✓		
GLOBAL SEARCH	SINGLE-OBJECTIVE	simplex, powell	✓	✓	
GLOBAL SEARCH	MULTI-OBJECTIVE	moga II, nsga II, ES, armoga, mopso, mosa, hybrid, fast, mogt, sangea, MEGO	✓	✓	
pilOPT				✓	✓

modeFRONTIER

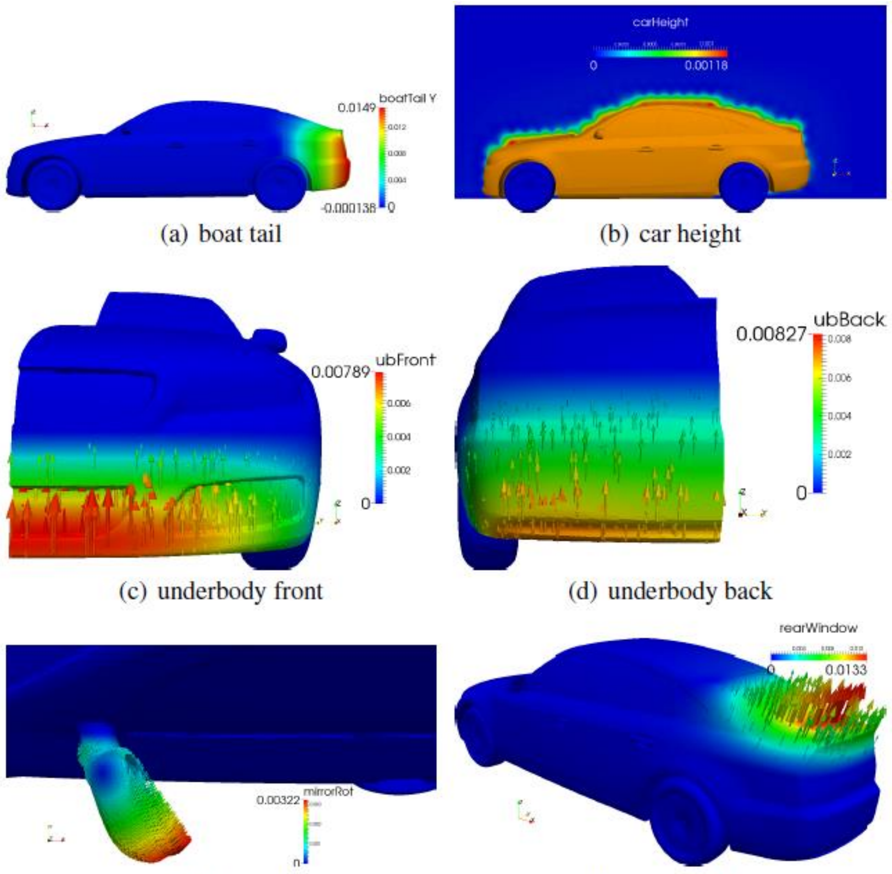
Adjoint-based shape optimization

- The adjoint formulation provides the gradient of an objective function with respect to surface displacements.

$$\frac{\delta F}{\delta \vec{b}} = \frac{\delta F}{\delta x_\kappa} \frac{\delta x_\kappa}{\delta \vec{b}}$$

rbf™

- RBF Morph provides the deformation velocity (adjoint preview).

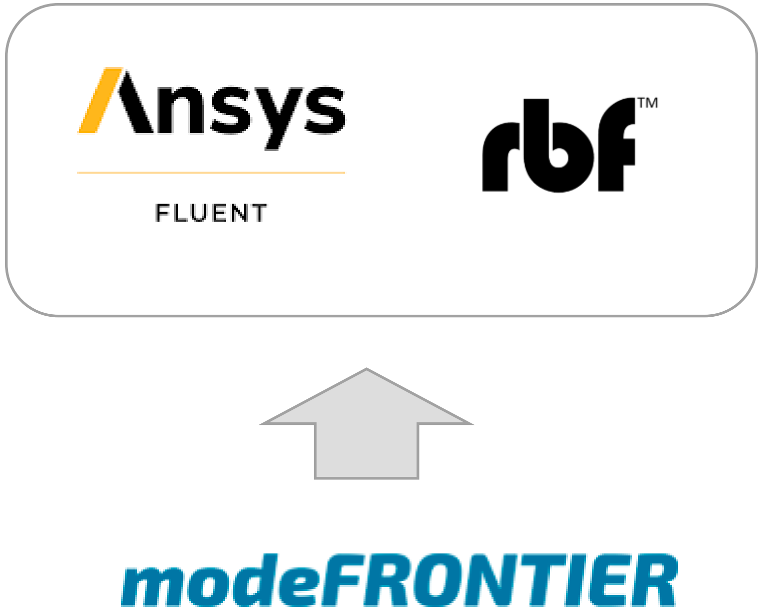


(a) boat tail (b) car height (c) underbody front (d) underbody back (e) mirror rotation (f) rear window

Papoutsis-Kiachagias, E. M., Giannakoglou, K. C., Porziani, S., Groth, C., Biancolini, M. E., Costa, E., & Andrejašič, M. (2019). Combining an OpenFOAM®-Based Adjoint Solver with RBF Morphing for Shape Optimization Problems on the RBF4AERO Platform. In *OpenFOAM®* (pp. 65-75). Springer, Cham.

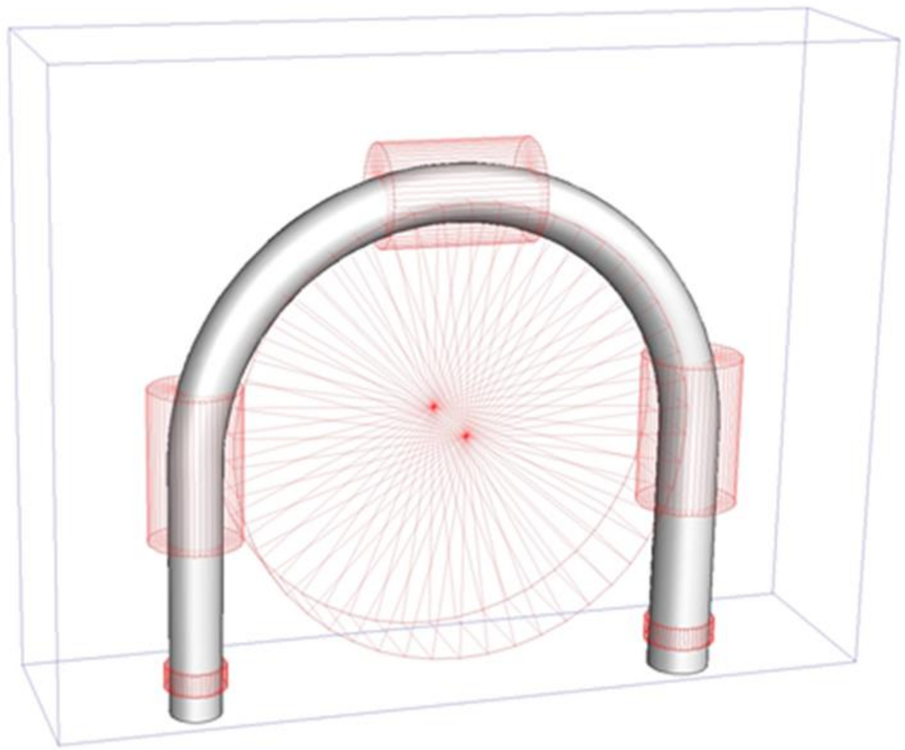
Proposed workflow

- High fidelity CAE solver adopted in this study is **Ansys Fluent** (CFD+adjoint)
- Advanced mesh morphing is provided by combining the CFD solver with the **RBF Morph Fluent module**
- The optimization process is controlled by **modeFRONTIER**



First example a U-bend*

- Set-up the CFD and adjoint run in Ansys Fluent. Prescribed velocity at the inlet. Pressure drop as observed output
- Define 6 morphing parameters (two transversal scaling factor of three sections) by RBF Morph
- Save morphing files (.rbf and .sol)



*Clarich, A., Battaglia, L., Nobile, E., Biancolini, M. E., & Cella, U. Adjoint Optimization combined with mesh morphing for CFD applications.

um 2020

modeFRONTIER workflow for adjoint optimization

um 2020

Define adjoint script by a few steps

```
;
;    Design variables setting
;
(define inpScaleX-sez1 2)    ; Input Scale X on sez 1
(define inpScaleY-sez1 2)    ; Input Scale Y on sez 1
(define inpScaleX-sez2 2)    ; Input Scale X on sez 2
(define inpScaleY-sez2 2)    ; Input Scale Y on sez 2
(define inpScaleZ-sez2 2)    ; Input Scale Z on sez 2
(define inpScaleX-sez3 2)    ; Input Scale X on sez 3
(define inpScaleY-sez3 2)    ; Input Scale Y on sez 3
```

1

```
;
;    Run parameters
;
(define numiter 400)          ; Number of iterations
(define numiteradj 150)       ; Number of iterations for adjoint
```

2

```
;
;    Load case and disable wornings
;
(Ti-menu-load-string (format #f "rc utube-adj.cas"))
(Ti-menu-load-string (format #f "solve/set flow-warnings no"))
(Ti-menu-load-string (format #f "file/set-batch-options no no no"))
```

3

```
;
;    Mesh morphing
;
(my-open-udf-library rbf-library)  ; Load RBF libraries
```

4

```
;
(rbf-morph
    (list
        (list "ScaleX-sez1" inpScaleX-sez1)
        (list "ScaleY-sez1" inpScaleY-sez1)
        (list "ScaleX-sez2" inpScaleX-sez2)
        (list "ScaleY-sez2" inpScaleY-sez2)
        (list "ScaleZ-sez2" inpScaleZ-sez2)
        (list "ScaleX-sez3" inpScaleX-sez3)
        (list "ScaleY-sez3" inpScaleY-sez3)
    )
```

```
;    Run the CFD computation
;
(Ti-menu-load-string (format #f "/solve/initialize/hyb-initialization ok"))        ; Initialize the flow field
(Ti-menu-load-string (format #f "it -a" numiter))                                  ; Run the computation
;
;    Run the Adjoint computation
;
(Ti-menu-load-string (format #f "/adjoint/run/initialize"))                        ; Initialize the Adjoint solution
(Ti-menu-load-string (format #f "/adjoint/run/iterate numiteradj"))                ; Run the Adjoint computation
```

6

```
;
;    Compute the sensitivities of the observable to the RBF solutions application
;
(Ti-menu-load-string (format #f "/adjoint/reporting/report inner"))                ; Workaround to let Fluent update shape sensitivity data
;
(rbf-smorph-init)  ; Initialize sequential morph command
;
(rbf-smorph-adjoint '((ScaleX-sez1 1)))
(define adj-ScaleX-sez1 (%rpgetvar 'rbf/smorph-adjoint-eval))
(rbf-smorph-adjoint '((ScaleY-sez1 1)))
(define adj-ScaleY-sez1 (%rpgetvar 'rbf/smorph-adjoint-eval))
(rbf-smorph-adjoint '((ScaleX-sez2 1)))
(define adj-ScaleX-sez2 (%rpgetvar 'rbf/smorph-adjoint-eval))
(rbf-smorph-adjoint '((ScaleY-sez2 1)))
(define adj-ScaleY-sez2 (%rpgetvar 'rbf/smorph-adjoint-eval))
(rbf-smorph-adjoint '((ScaleZ-sez2 1)))
(define adj-ScaleZ-sez2 (%rpgetvar 'rbf/smorph-adjoint-eval))
(rbf-smorph-adjoint '((ScaleX-sez3 1)))
(define adj-ScaleX-sez3 (%rpgetvar 'rbf/smorph-adjoint-eval))
(rbf-smorph-adjoint '((ScaleY-sez3 1)))
(define adj-ScaleY-sez3 (%rpgetvar 'rbf/smorph-adjoint-eval))
```

7

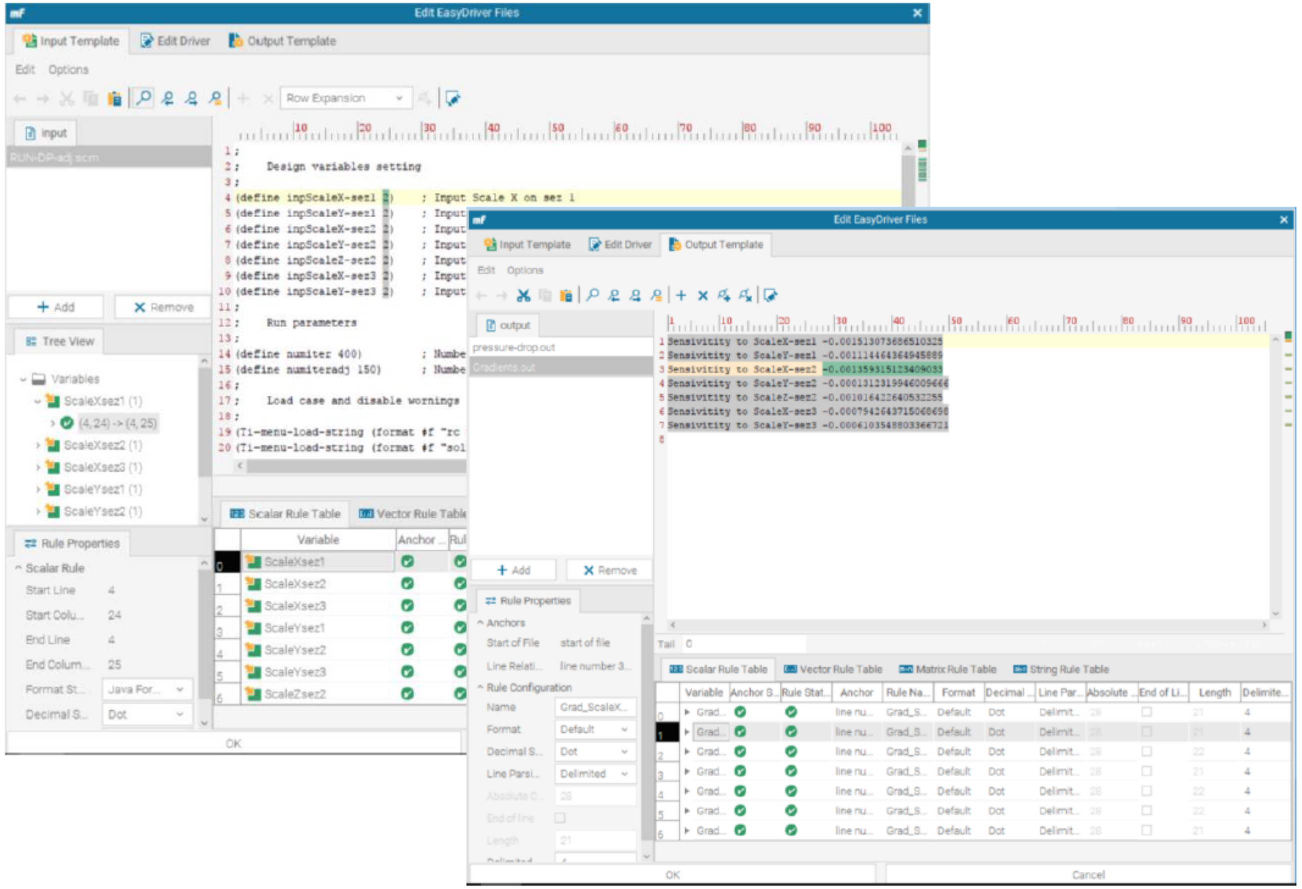
```
;
;    Save sensitivities to a file
;
(define gradfile (open-output-file "Gradients.out"))
(display "Sensivitity to ScaleX-sez1 " gradfile) (write adj-ScaleX-sez1 gradfile) (newline gradfile)
(display "Sensivitity to ScaleY-sez1 " gradfile) (write adj-ScaleY-sez1 gradfile) (newline gradfile)
(display "Sensivitity to ScaleX-sez2 " gradfile) (write adj-ScaleX-sez2 gradfile) (newline gradfile)
(display "Sensivitity to ScaleY-sez2 " gradfile) (write adj-ScaleY-sez2 gradfile) (newline gradfile)
(display "Sensivitity to ScaleZ-sez2 " gradfile) (write adj-ScaleZ-sez2 gradfile) (newline gradfile)
(display "Sensivitity to ScaleX-sez3 " gradfile) (write adj-ScaleX-sez3 gradfile) (newline gradfile)
(display "Sensivitity to ScaleY-sez3 " gradfile) (write adj-ScaleY-sez3 gradfile) (newline gradfile)
(close-output-port gradfile)
;
;    Save observable value to a file
;
(Ti-menu-load-string (format #f "/define/parameters/output-parameters/write-to-file pressure-drop-op pressure-drop.out"))
;
;    EXIT FLUENT
;
(Ti-menu-load-string (format #f "exit"))
```

8

1. Define values of morphing **parameters**
2. Define number of **iterations** for Fluent and for adjoint
3. Load .cas **model** file
4. Load rbf **libraries**

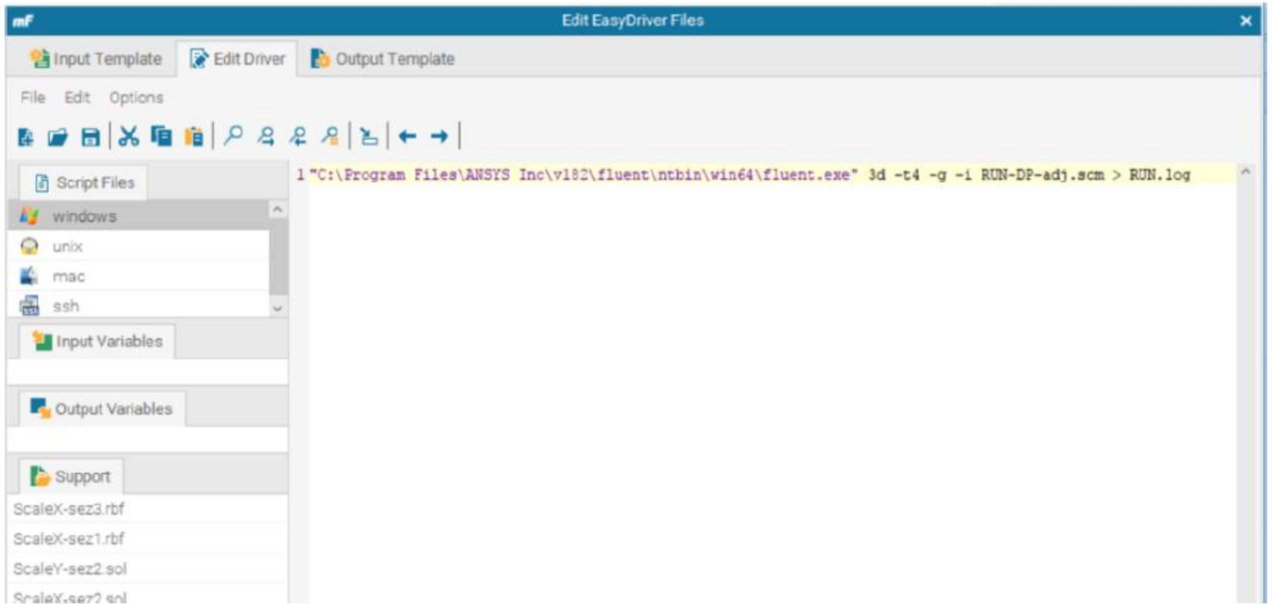
6. **Run** fluent and after that run adjoint evaluation
7. Compute **sensitivity** in function of morphing parameters
8. **Write** sensitivities and observabe to file

RBF Morph integration by Easydriver: I/O templates



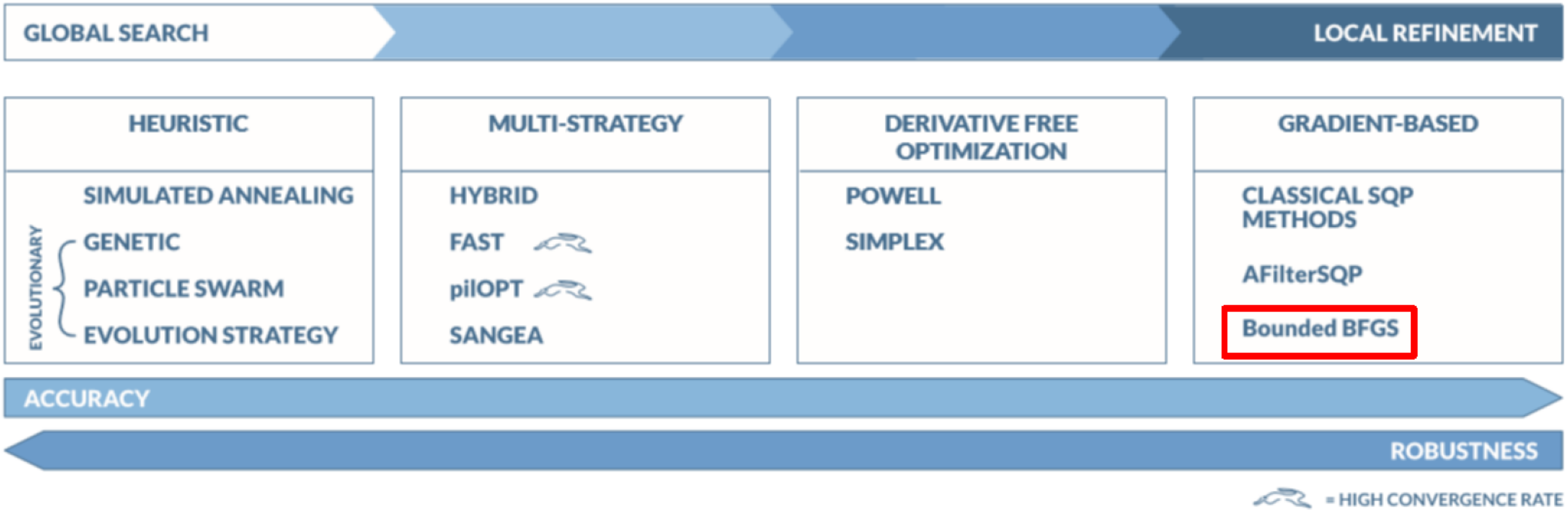
- Link each workflow variable to the corrisponding I/O

RBF Morph integration by Easydriver: driver



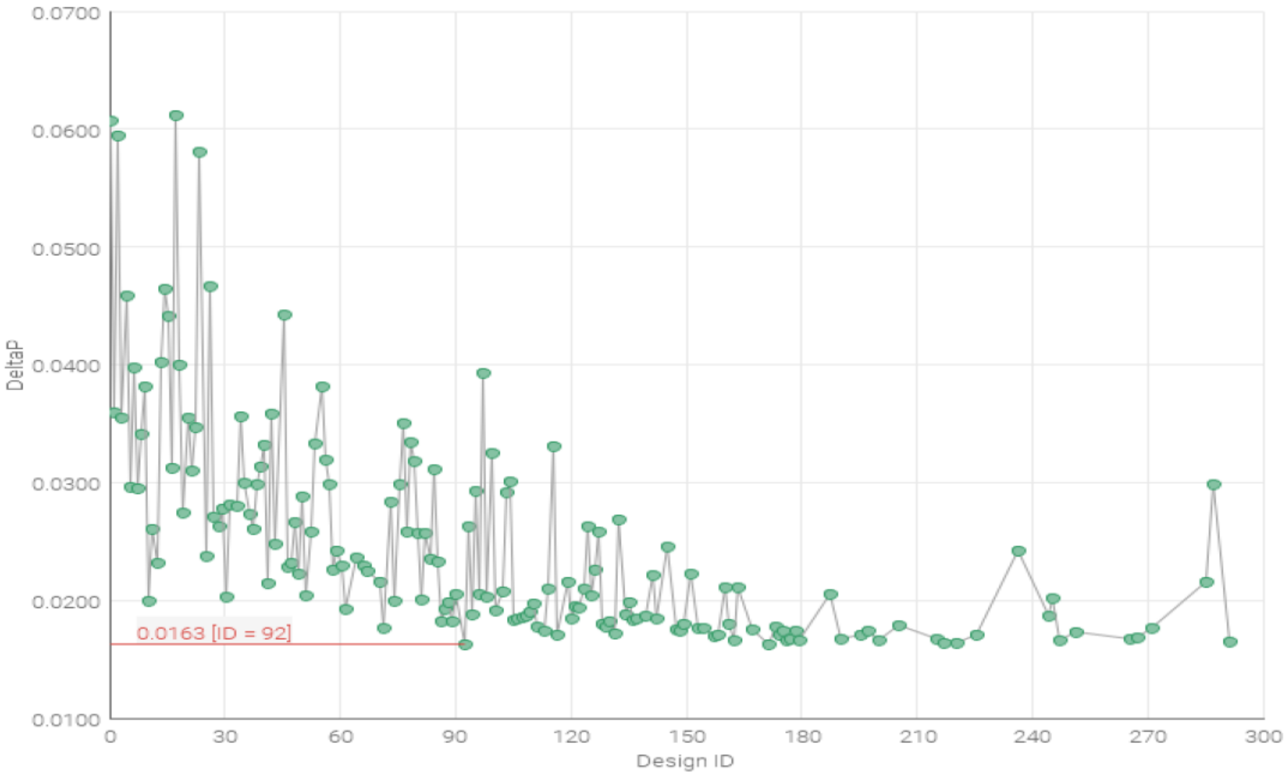
- Command to run Fluent in batch

Optimization Algorithms in modeFRONTIER



um 2020

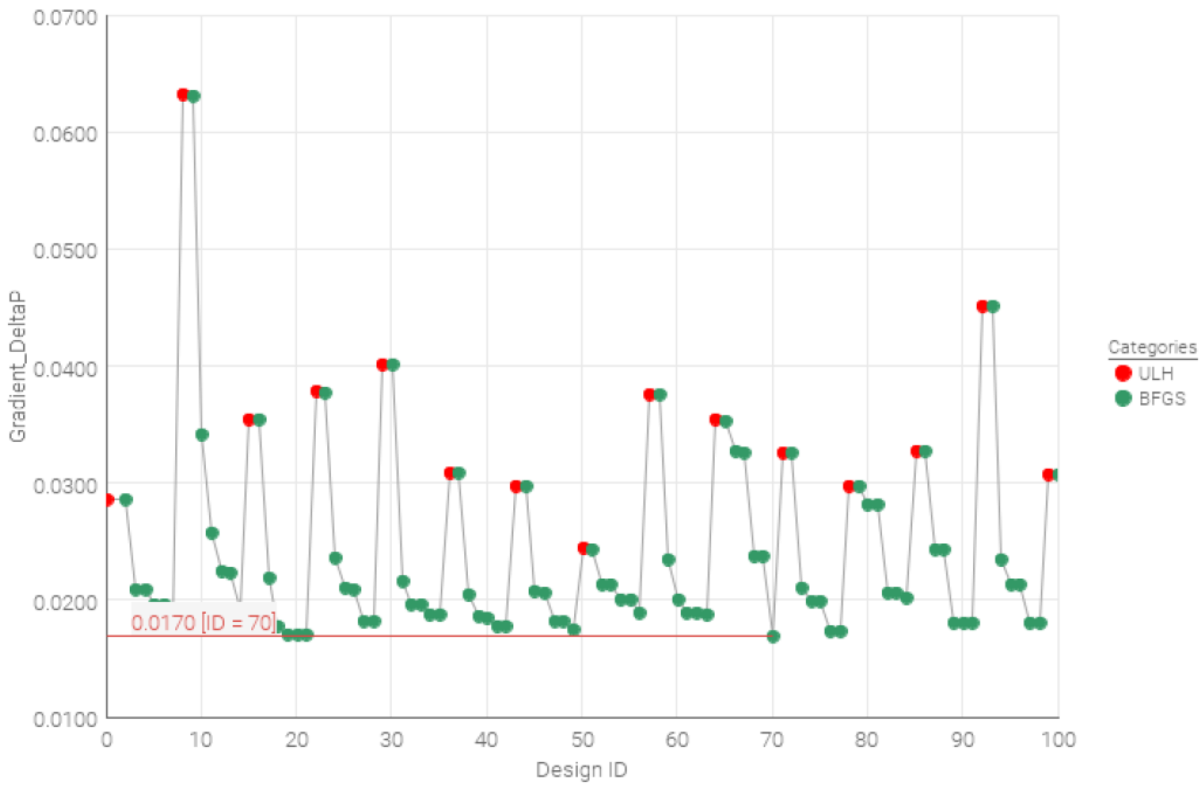
Optimization approach without gradient (traditional)



- 1 simulation (400 Fluent iterations): **4 mins**
- Starting DOE: 15 points by ULH – 300 total simulations by MOGAII (20 generations)
- Time to reach optimal results: 90 simulations **(360 mins)**

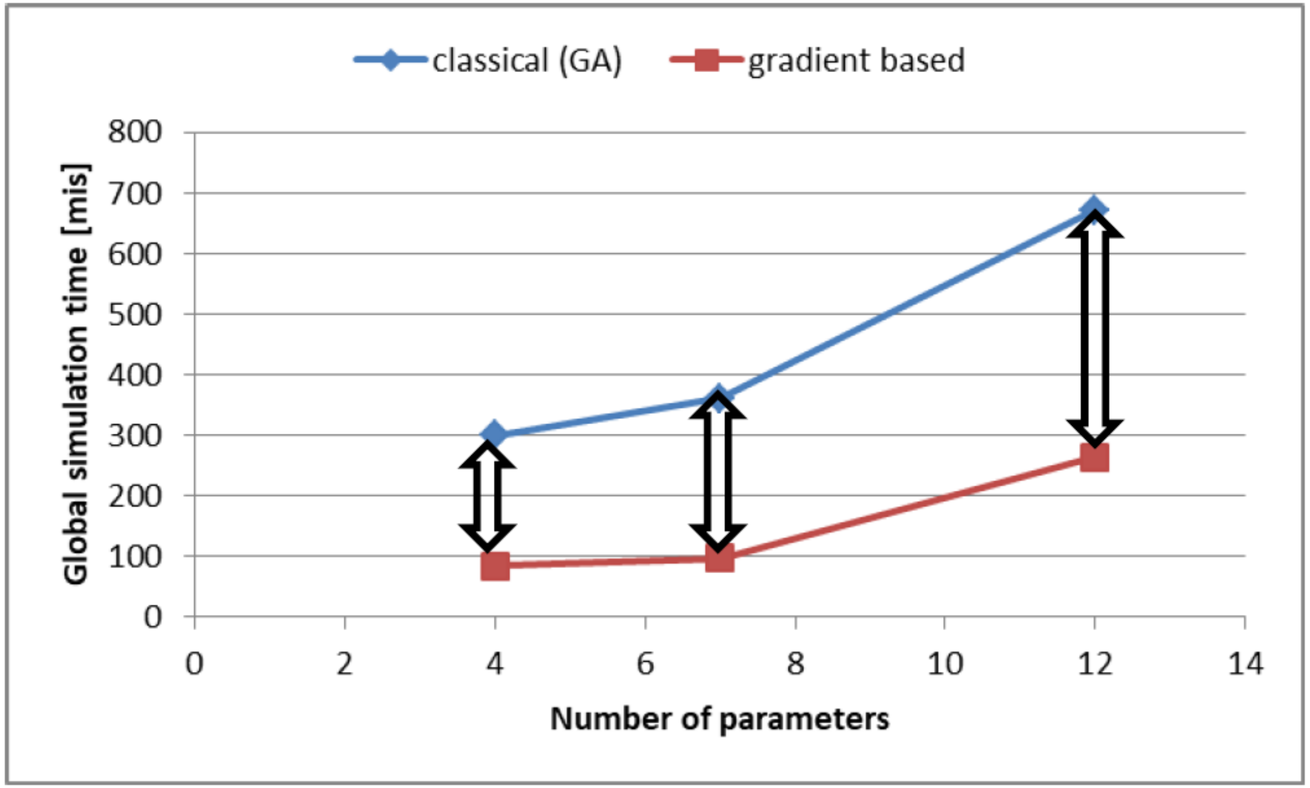
um 2020

Optimization approach with Adjoint method



- 1 simulation (400 Fluent + 150 adjoint iterations): **13 mins**
- Simulation budget equivalent to no-adjoint case: >100 simulations (same DOE)
- Time to get optimal results: only 7 simulations (**91 mins**)!

Adjoint method vs number of parameters



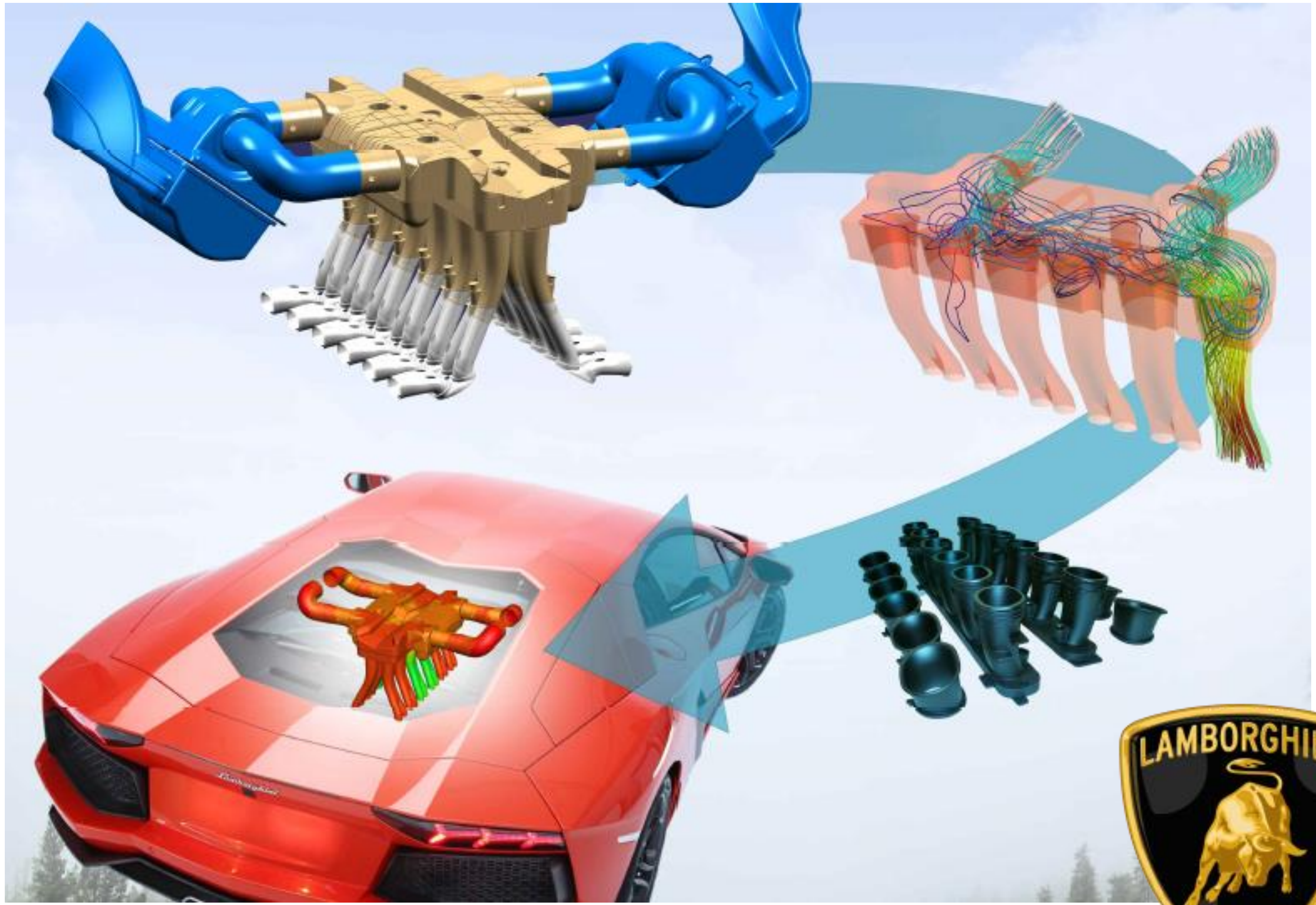
- Different parameterization on the same model (two approaches compared)
- Adjoint advantages increases with **number of parameters**

um 2020

Industrial application

Ansys FLUENT

rbf™



modeFRONTIER

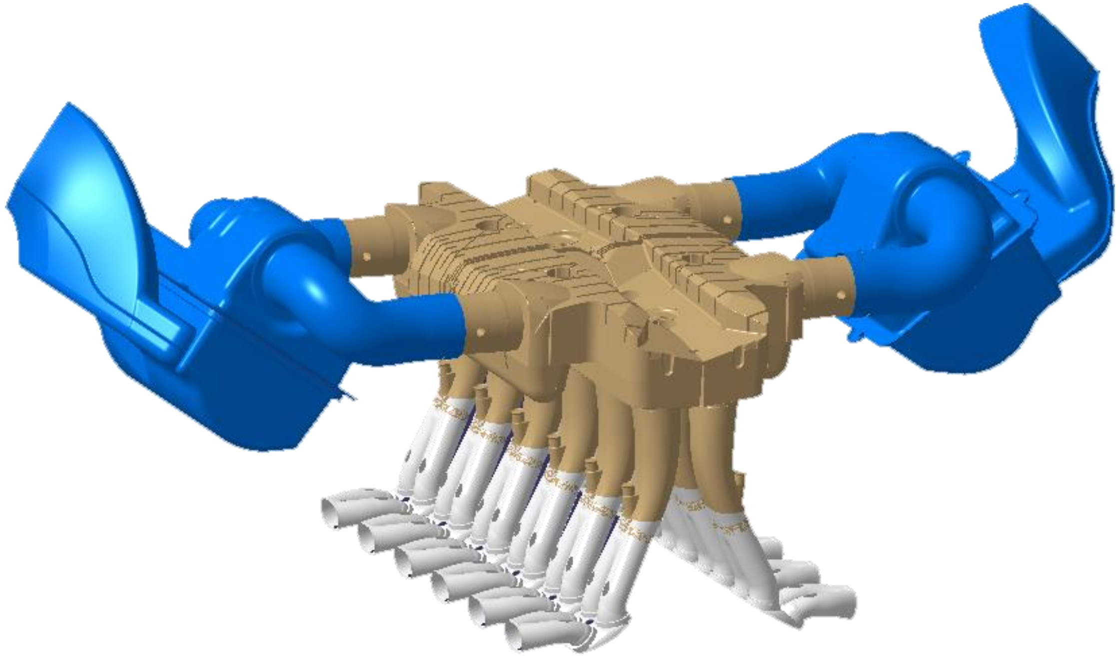
- The workflow tuned for a simple example is here demonstrated for an industrial application

LAMBORGHINI

um 2020

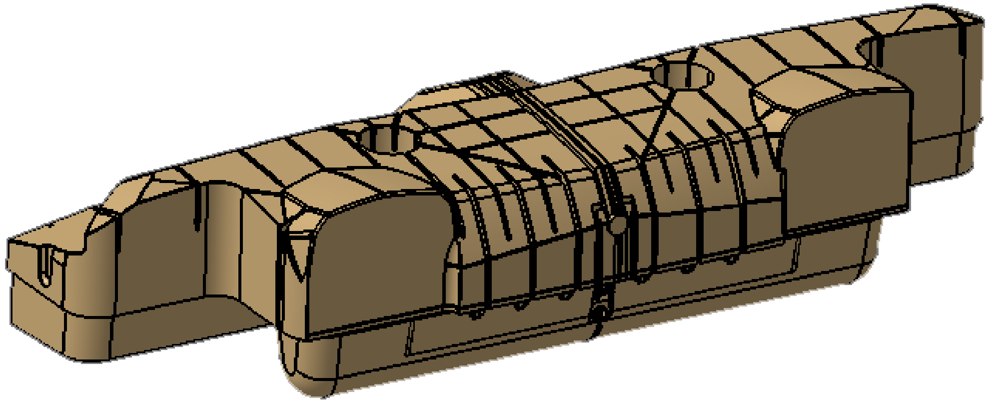
Case study

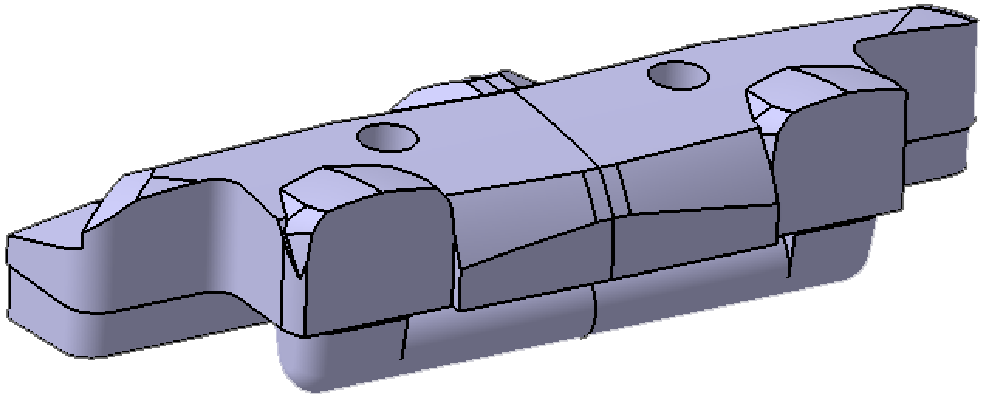
- Airbox of the **Lamborghini Aventador**
- V12 700Hp@8250RPM
- Detailed CFD analyses of intake runners pressure drops
- Define a new shape for charging efficiency maximization



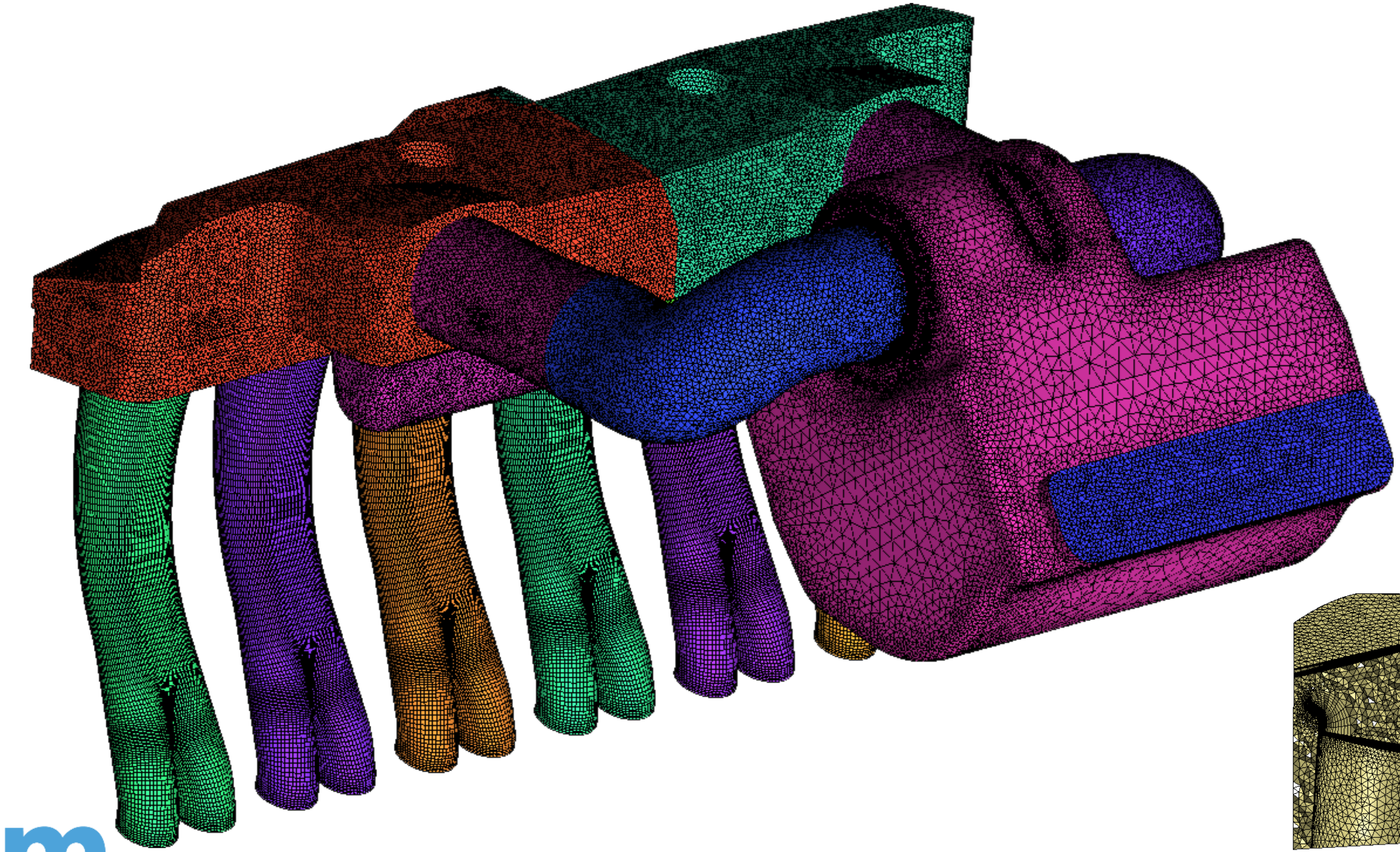
CAD model preparation

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

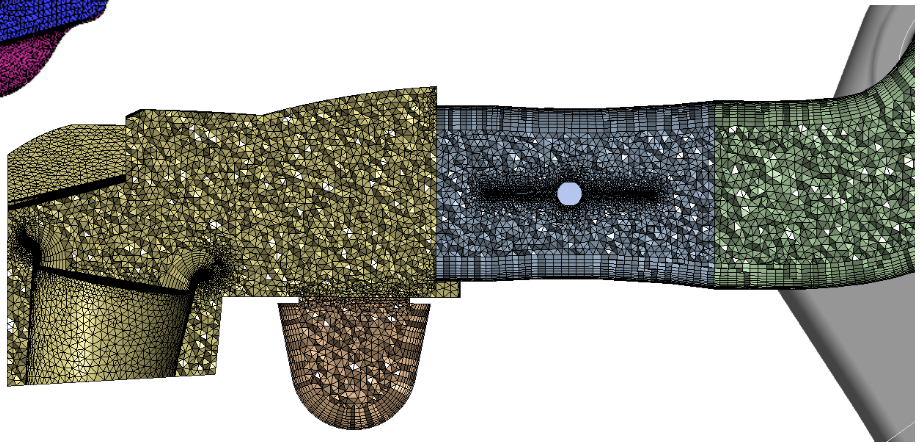




Mesh assembly

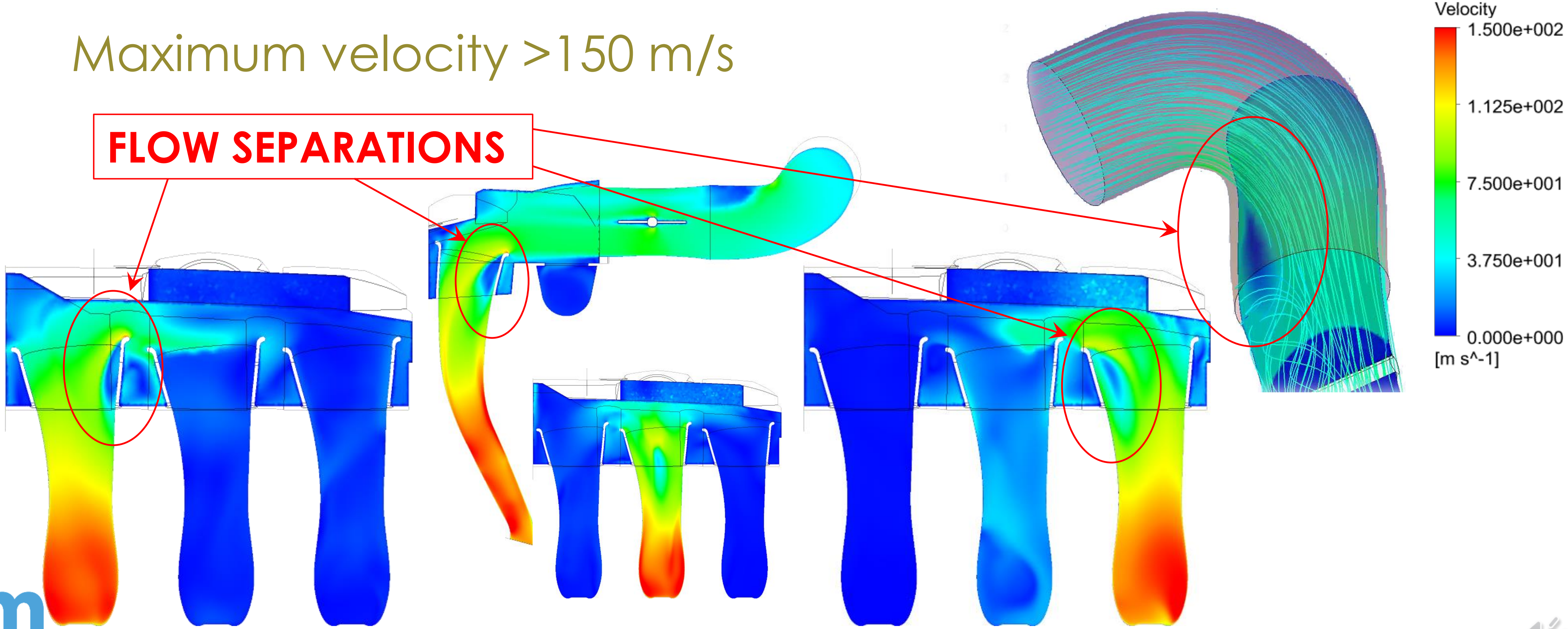


Maximum dimension
9.5 millions

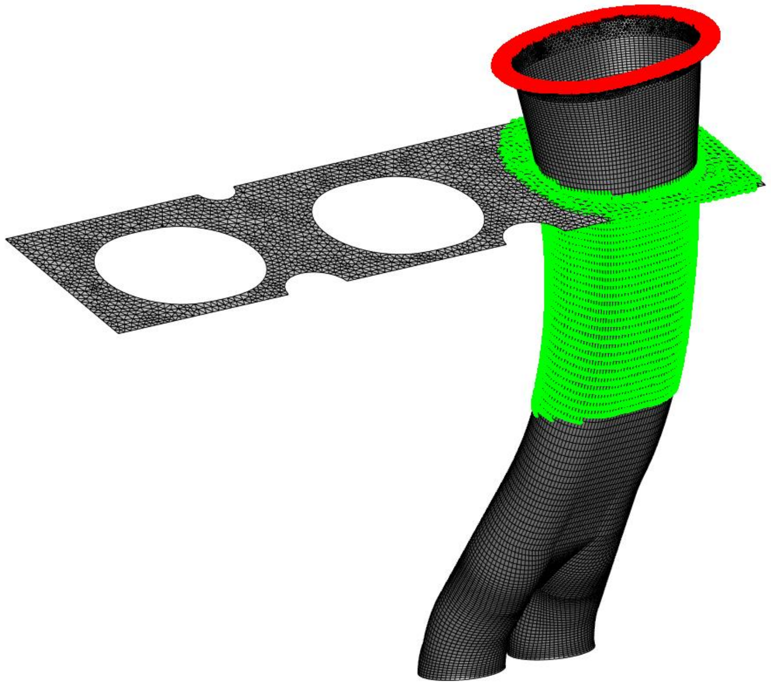


Critical regions

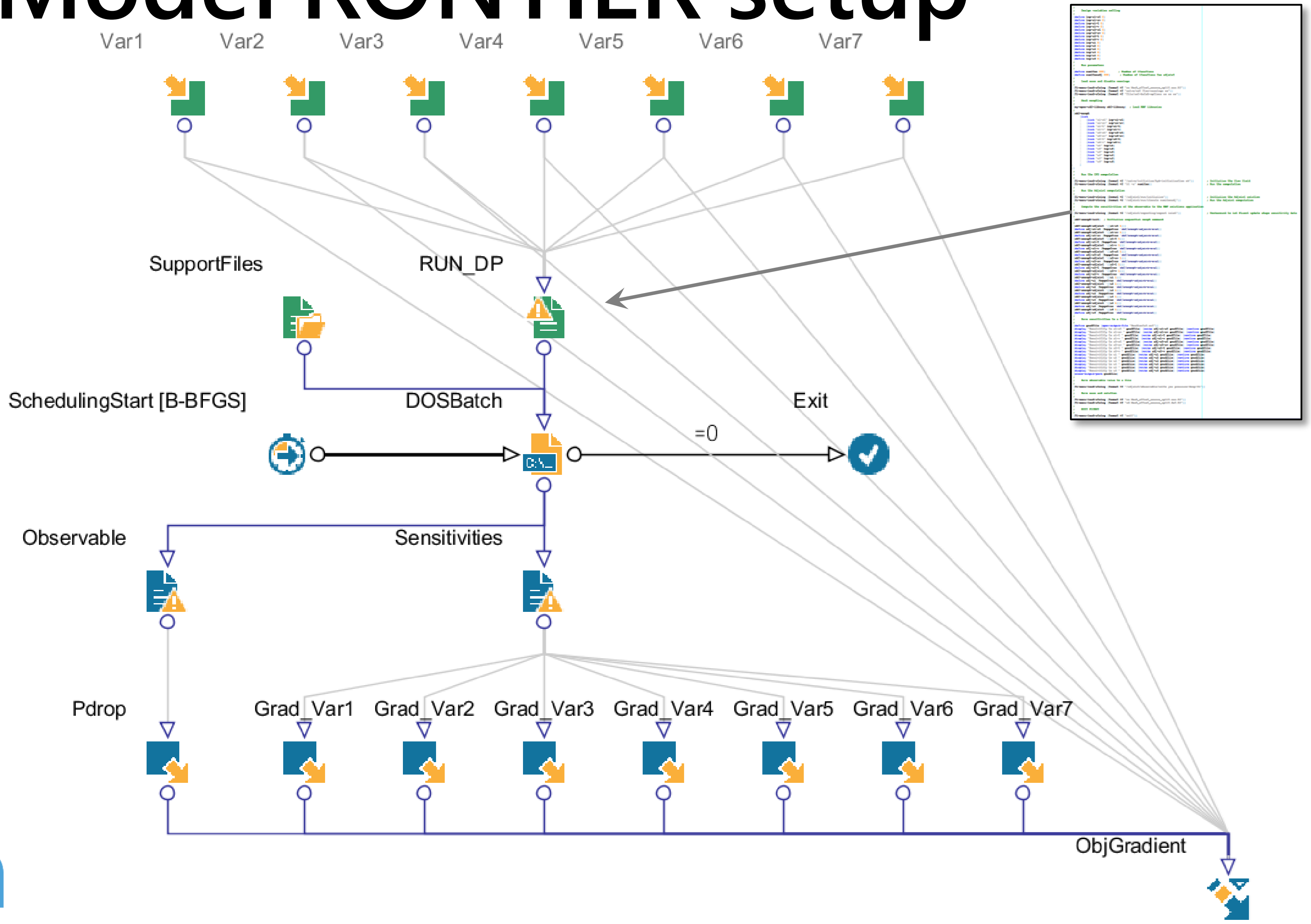
Maximum velocity >150 m/s



Shape modifiers



ModeFRONTIER setup

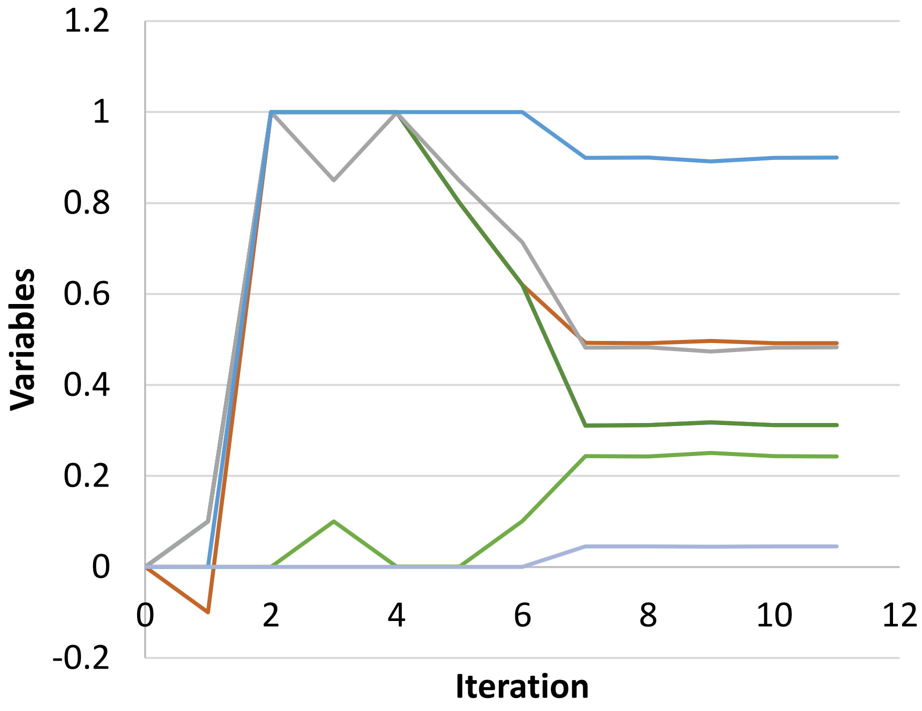


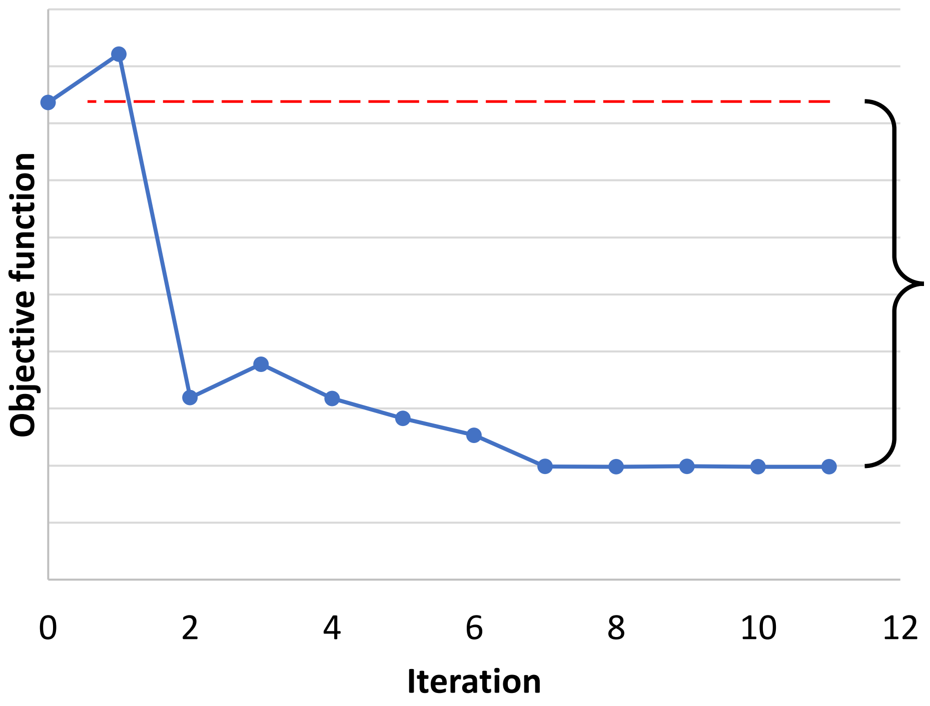
Scheme script that drives the design point evaluation.

Tasks:

- Case loading
- Mesh morphing
- Flow solution
- Adjoint computation
- Pressure drop saving in an ASCII file
- Gradients saving in an ASCII file

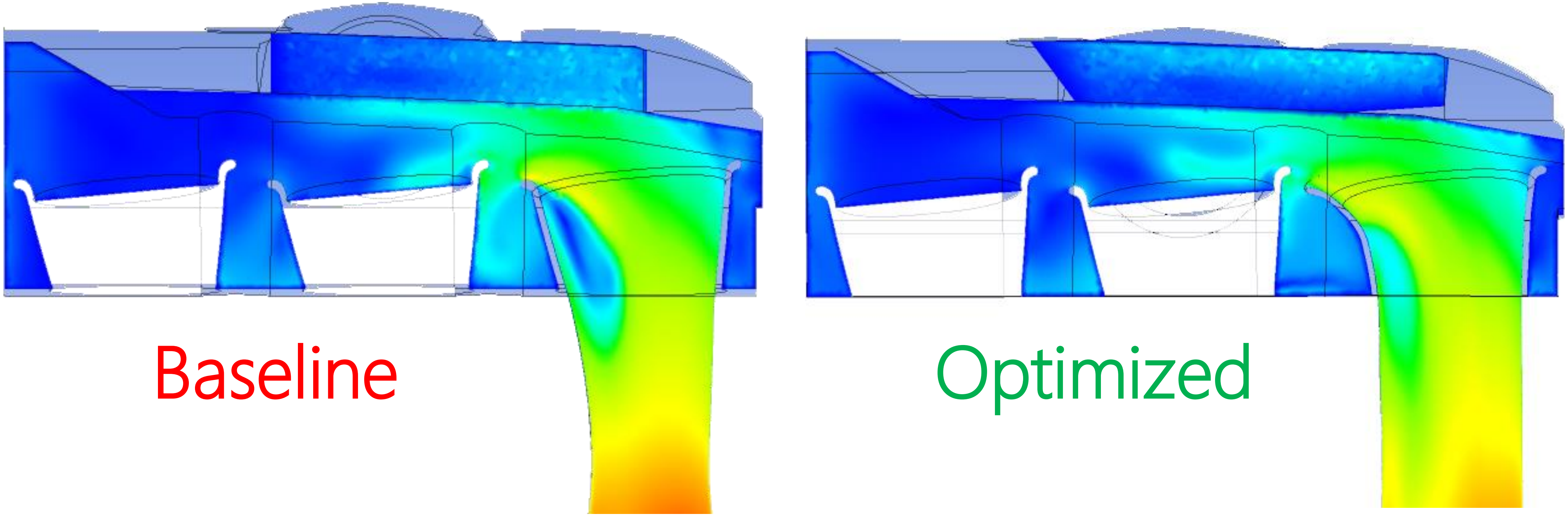
Convergence history





Pressure drop reduced by ~6.2%

Solution



Conclusions

- Optimization-driven design adopting MDO requires intense use of high fidelity simulations and fast optimization algorithms are needed to reduce the optimization turnaround time
- Gradient based methods are a possible answer but derivatives are needed
- When **shape sensitivities** are available (adjoint solution) we can compute derivatives of performance vs. parameters
- In this study we presented a workflow based on Ansys Fluent, RBF Morph and modeFRONTIER that on a u-bend optimization is 4 time faster than traditional approach
- The proposed method was applied to reduce the pressure drop of the **Lamborghini Aventador** engine airbox runners gaining a **6%**

ESTECO

ESTECO USERS' MEETING 2020 30 SEPT - 1 OCT

Thank you!

marco.biancolini@rbf-morph.com

linkedin.com/company/rbf-morph

youtube.com/user/RbfMorph

rbf-morph.com