EURO Greece

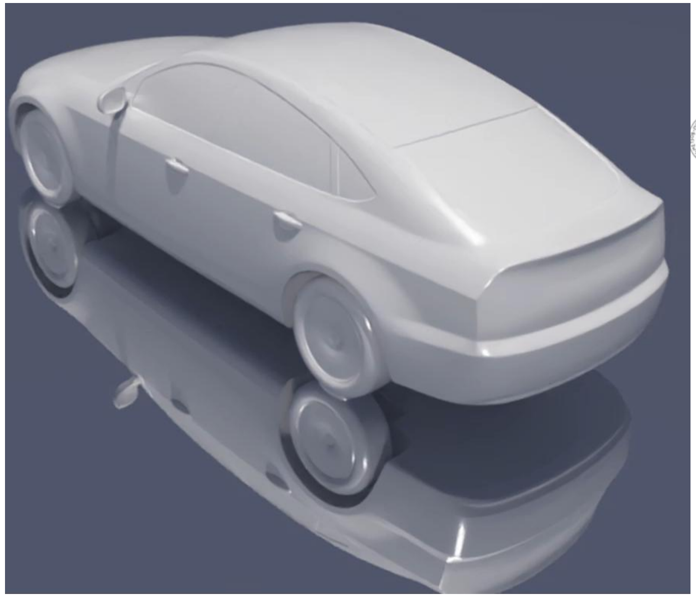
HPC Training Series

An introduction to adjoint methods and the shape and topology optimization workflow of OpenFOAM for CFD-based optimization

Dr. Evangelos (Vaggelis) Papoutsis-Kiachagias
Senior Researcher, NTUA

School of Mechanical Engineering, NTUA,
Parallel CFD & Optimization Unit
email: vpapout@mail.ntua.gr

What is Gradient-based Optimization?



Optimization Methods in CFD:
Improve the performance of an aerodynamic shape

Quantity describing the performance:
The objective function *J* (e.g. drag force exerted on a car) → computed through CFD

How are going to affect the objective function?:
By changing the values of the so-called design variables, $\vec{b}$.
For instance, control points affecting the shape of the car

How are we going to update the values of the design variables?:

Gradient-free Methods:
Require only the computation of *J*.
See the flow solver (or any evaluation tool) as a black box

Gradient-based Methods:
Require the computation of *J* and $dJ/d\vec{b}$
Potentially need access to the source code

How is this computed?

How can $dJ/d\vec{b}$ *be computed?*

The simplest way → Finite Differences

- ✓ Does not require the development of any additional s/w. Relies only on the flow solver
- × Its cost scales with the number of design variables *N*
- × Sensitive to the choice of the **ε** step

$$\frac{dJ}{db_i} = \frac{J(b_i + \varepsilon) - J(b_i)}{\varepsilon}$$

The hard way → The adjoint method

- × Requires a new mathematical development and programming if the flow problem or the objective function changes
- ✓ Has a cost that is independent of the number of the design variables. Ideal for expensive industrial problems with many design variables
- ✓ All components of $dJ/d\vec{b}$ are computed at the cost of only an additional set of PDEs → the adjoint equations

Review paper:
https://doi.org/10.1007/s11831-014-9141-9

The flow and adjoint PDEs

The flow PDEs

Continuity

$$R^p = \frac{\partial v_j}{\partial x_j} = 0$$

Momentum

$$R_i^v = v_j \frac{\partial v_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} = 0 \ , \quad i = 1,2,3$$

Spalart-Allmaras

$$R^{\widetilde{\nu}} = v_j \frac{\partial \widetilde{\nu}}{\partial x_j} - \frac{\partial}{\partial x_j}\left(\frac{(\nu + \widetilde{\nu})}{\sigma} \frac{\partial \widetilde{\nu}}{\partial x_j}\right) - \frac{c_{b2}}{\sigma}\left(\frac{\partial \widetilde{\nu}}{\partial x_j}\right)^2 + \widetilde{\nu}\left(D\left(\widetilde{\nu}, y\right) - P\left(\widetilde{\nu}, y\right)\right) = 0$$

The adjoint PDEs

Adjoint Continuity

$$R^q = \frac{\partial u_i}{\partial x_i} = 0$$

Adjoint Momentum

$$R_i^u = -\frac{\partial (v_j u_i)}{\partial x_j} + u_j \frac{\partial v_j}{\partial x_i} + \frac{\partial q}{\partial x_i} - \frac{\partial \tau_{ij}^{\alpha}}{\partial x_j} + \frac{\partial}{\partial x_j}\left(\frac{C_{\widetilde{S}}}{S} \widetilde{\nu_a} \widetilde{\nu}\left(\frac{\partial v_i}{\partial x_j} - \frac{\partial v_j}{\partial x_i}\right)\right) = 0 \ , \quad i = 1,2,3$$

Adjoint Spalart-Allmaras

$$R^{\widetilde{\nu_a}} = -\frac{\partial \left(v_j \widetilde{\nu_a}\right)}{\partial x_j} - \frac{\partial}{\partial x_j}\left(\frac{\nu + \widetilde{\nu}}{\sigma} \frac{\partial \widetilde{\nu}}{\partial x_j}\right) + \frac{1}{\sigma} \frac{\partial \widetilde{\nu}}{\partial x_j} \frac{\partial \widetilde{\nu_a}}{\partial x_j} + 2 \frac{c_{b2}}{\sigma} \frac{\partial}{\partial x_j}\left(\widetilde{\nu_a} \frac{\partial \widetilde{\nu}}{\partial x_j}\right) + C_{\widetilde{\nu}} \widetilde{\nu} \widetilde{\nu_a}$$
$$+ \widetilde{\nu_a}\left(D\left(\widetilde{\nu}, y\right) - P\left(\widetilde{\nu}, y\right)\right) + \frac{\partial u_i}{\partial x_j}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \frac{\delta \nu_t}{\delta \widetilde{\nu}} = 0$$

An Adjoint-based Optimization Loop

Define design variables b_n ⇨ **Effect on the flow equations?**

Solve primal equations $\rightarrow U$ ⇨ OpenFOAM

Compute J ⇨ **Forces, moments, losses**

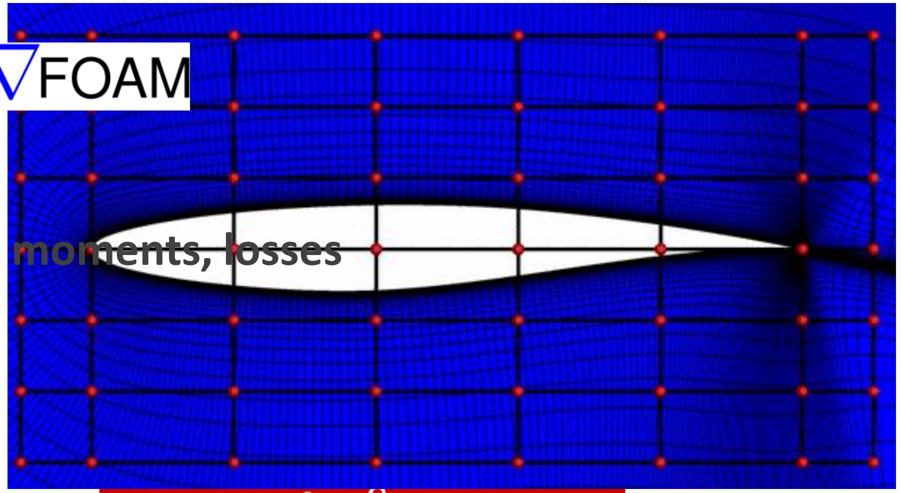
Solve adjoint equations $\rightarrow \Psi$

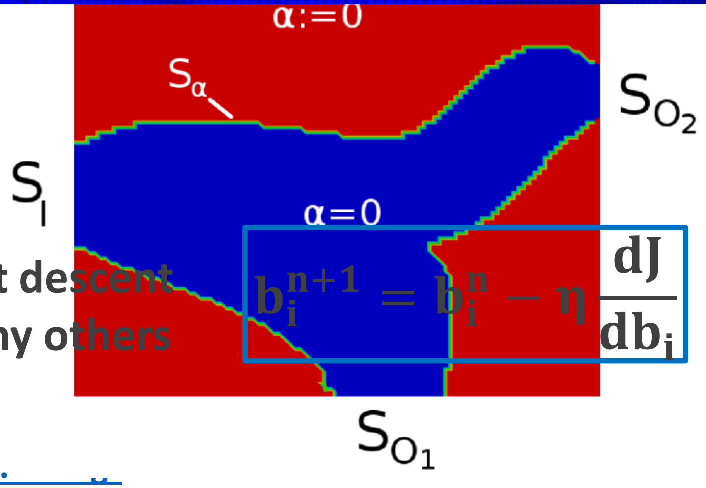
Compute $\delta J/\delta b_n\ (U, \Psi)$

Update design variables ⇨ **Steepest descent and many others**

$$b_i^{n+1} = b_i^n - \eta \frac{dJ}{db_i}$$

Shape optimization (ShpO): shape *and* mesh





adjointOptimisationFoam: an adjoint-based optimization framework in OpenFOAM

- An all-in-one OpenFOAM executable implementing an integrated, gradient-based optimization workflow
- Product of a 17 years of development at PCOpt/NTUA
- Integrated into the official OpenFOAM version in collaboration with OpenCFD in 2019
- User manual:
  https://www.openfoam.com/documentation/files/adjointOptimisationFoamManual_v2312.pdf
  Covers all functionality up until v2506

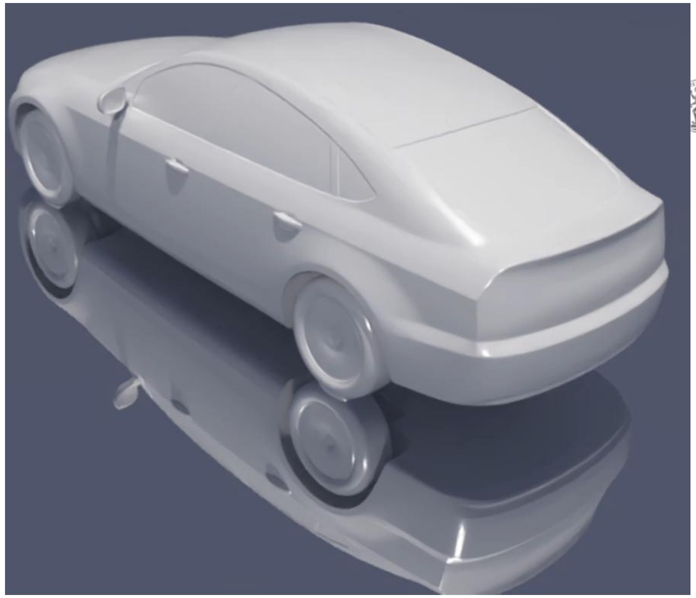
Acknowledgments:

Dr. Nikolaos Galanos
Dr Andreas Margetis
Dr Themis Skamagkis
Dr. Konstantinos Gkaragkounis
Dr. Ioannis Kavvadias
Dr. Alexandros Zymaris
James Koch
Dr. Andrew Heather

Current status of adjointOptimisationFoam

OpenFOAM version	Features
v1906	• Adjoint to incompressible, steady-state flows • Differentiation of the Spalart-Allmaras turbulence model • Computation of sensitivity maps
v1912	• Surface and volume parameterization using volumetric B-Splines • Automated shape optimization loops
v2006	• New objective function related to the qualitative evaluation and minimization of noise • Sensitivity contributions from rotating boundaries
v2112	• Smoothing of sensitivity maps
v2206	• Adjoint to the k-ω SST turbulence model
v2212	• Objective functions for internal aerodynamics (flow rate, flow rate distribution, uniformity, power losses)
v2312	• Topology optimization
2406-2506	• Incremental improvements

Sensitivity maps and Shape Optimization (ShpO)



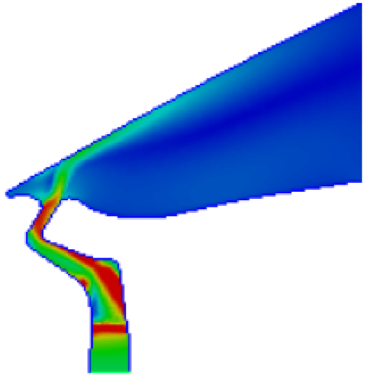
Sensitivity maps:

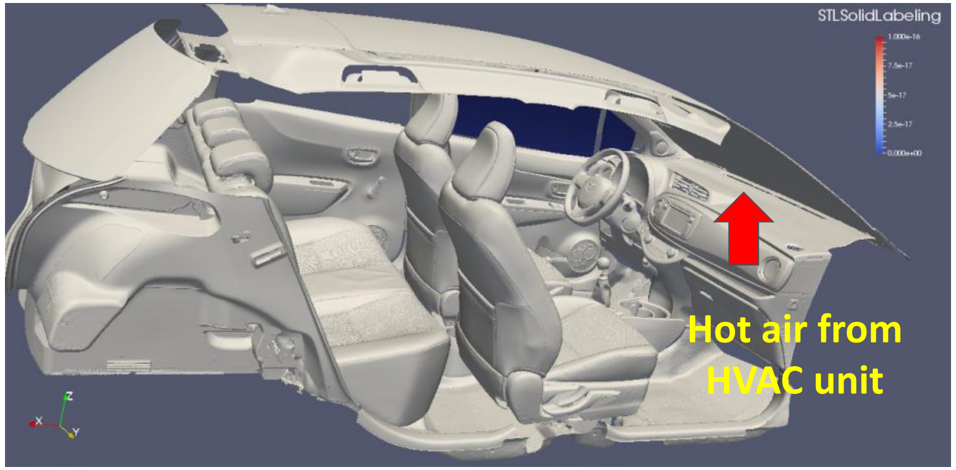
- The derivative of J w.r.t. the normal displacement of the boundary nodes
- No optimization loop; only 1 flow + 1 adjoint solution
- Identify the areas with a high optimization potential→ Intense colors
- Identify the favorable displacement direction → Blue: move surface inwards Red: move surface outwards

Shape optimization:

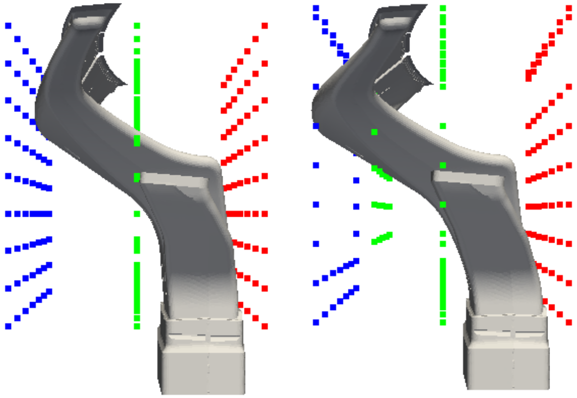
- An actual optimization loop is performed
- In each optimization cycle, the shape is updated, followed by the update of the internal grid nodes
- Each cycle has a cost of 1 flow + 1 adjoint solution
- Usually, a small number of cycles is required to reach convergence (< 20)

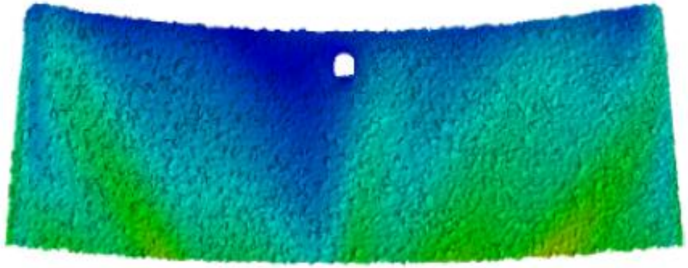
ShpO of the Defroster Nozzle of the HVAC unit of a Car

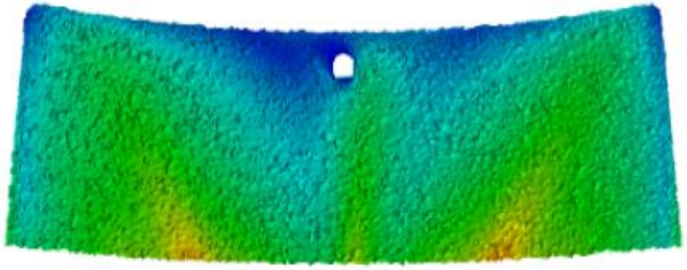




Initial Optimized







$$J_V = \frac{1}{2}\int_{\Omega'} \left(v - v^{target}\right)^2 d\Omega$$

Initial Optimized

ShpO of the defroster nozzle of the HVAC unit of a TOYOTA passenger car, to shorten the time for dispelling condensation or frost on the windshield in the most uniform way. To this end, a certain air velocity close to the windshield must be reached. The optimized geometry was manufactured (3D printing) and submitted to a defrost test in the TME's climate chamber (@ -20^o), leading to 15% less windshield defrost time.

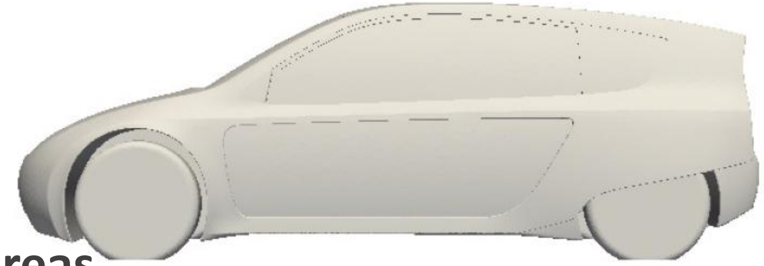
Green areas in the velocity isolines' plot on the windshield correspond to v^{target}.

Application funded by TOYOTA

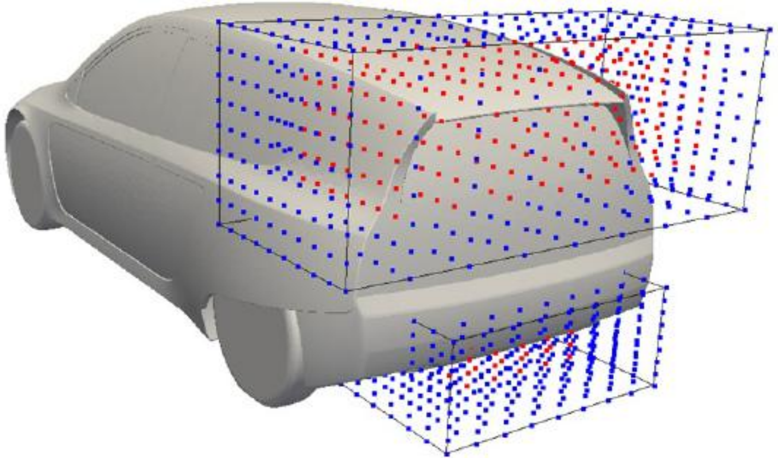
Multi-Point Aerodynamic ShpO of a Concept Car

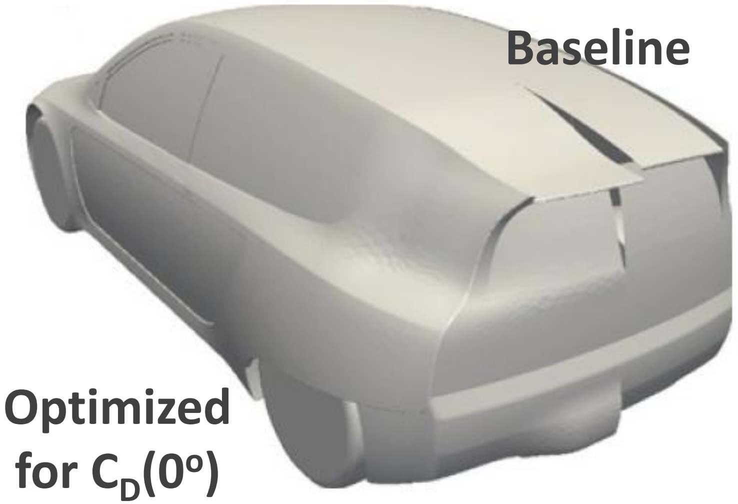
- **ShpO of an ultra-lightweight vehicle, designed by the Toyota Aerodynamic Dept., for making it less sensitive to side-wind (30^o from the port side; min. yaw moment), while maintaining a very low drag at 0^o (longitudinal wind).**
- **Objective function:**

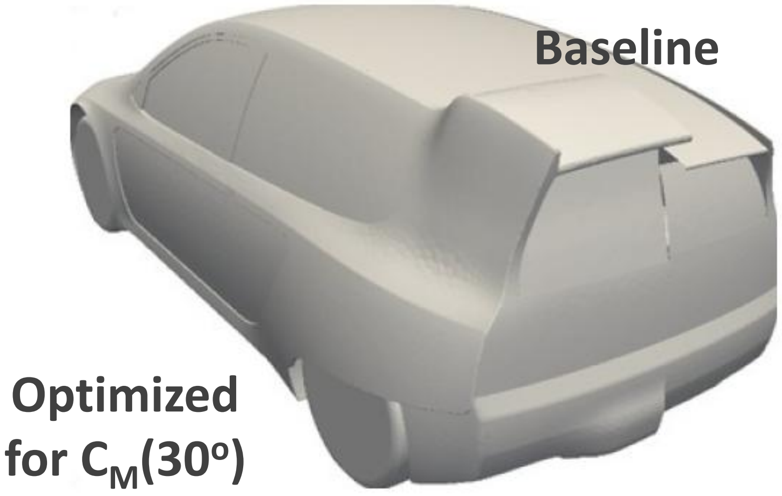
$$J = \omega_D C_D^{0^o} + \omega_M C_M^{30^o}$$



- **Pareto front computed by optimizing with different values of (ω_D,ω_M).**
- **Two simultaneously acting morphing boxes at the spoiler and diffuser areas.**

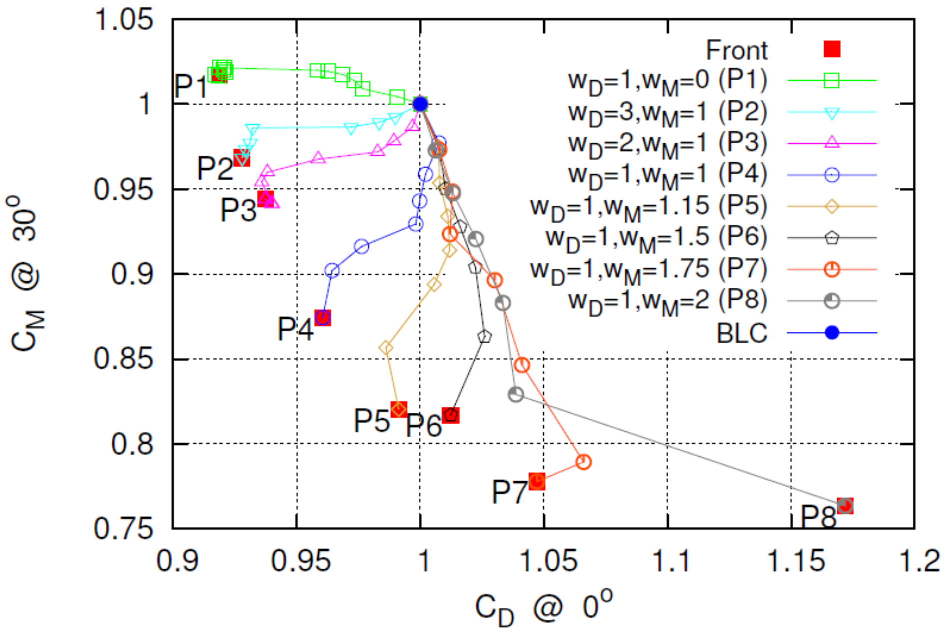


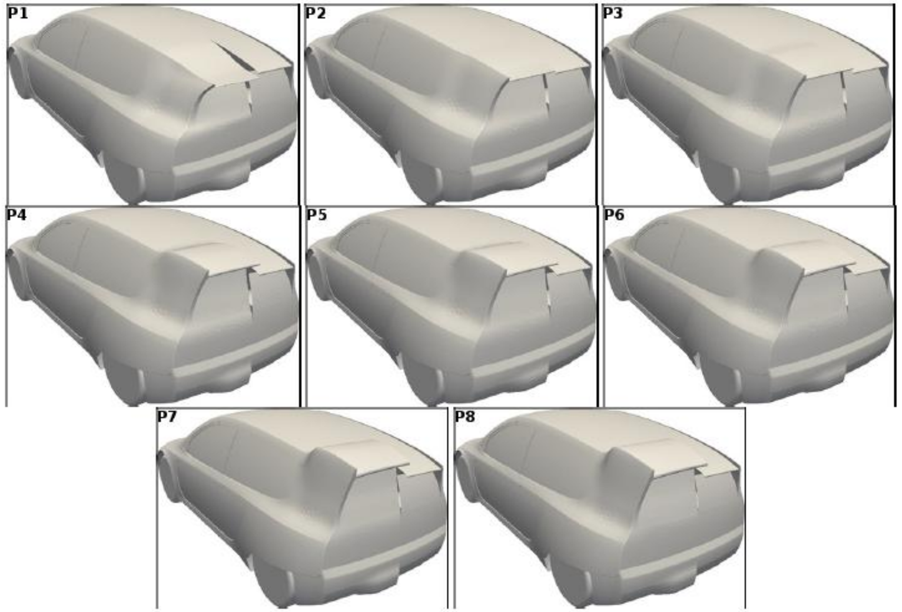




Application funded by TOYOTA

Multi-Point Aerodynamic ShpO of a Concept Car



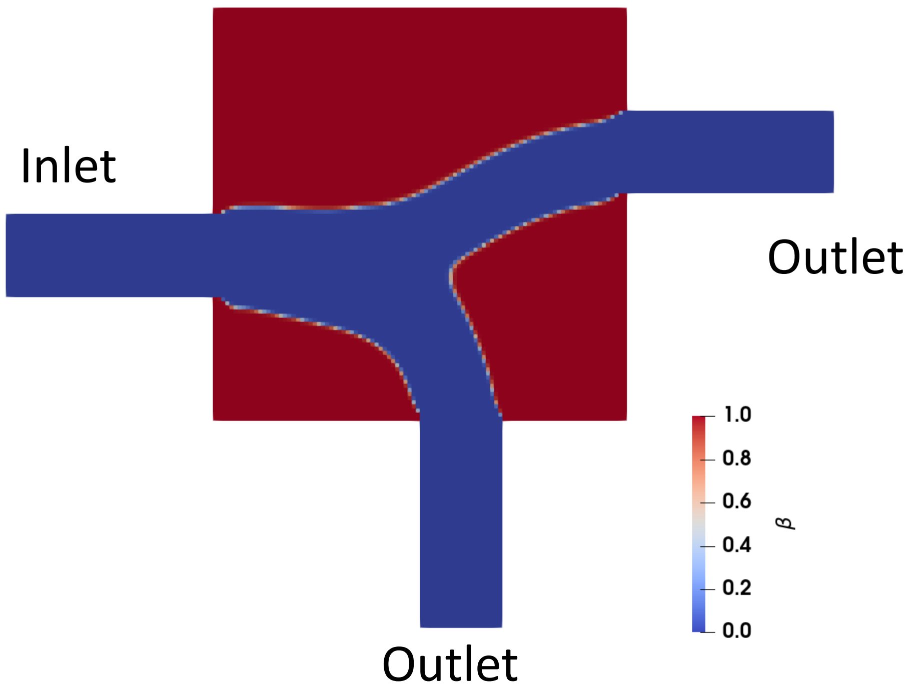


RANS-based ShpO using the adjoint to the Spalart-Allmaras model (with wall functions). Optimized geometries (port side) compared to the baseline (starboard side). C_D reduction at 0º results from a lowered spoiler, boat-tailing and a prolonged and widened diffuser. C_M reduction at 30º comes mainly from the increased spoiler height and the slight widening of the car; these increase pressure on the port side and decrease it on the starboard side to counter-balance the yaw moment due to side-wind.

Structural and Multidisciplinary Optimization, 59(2): 675–694, 2019.

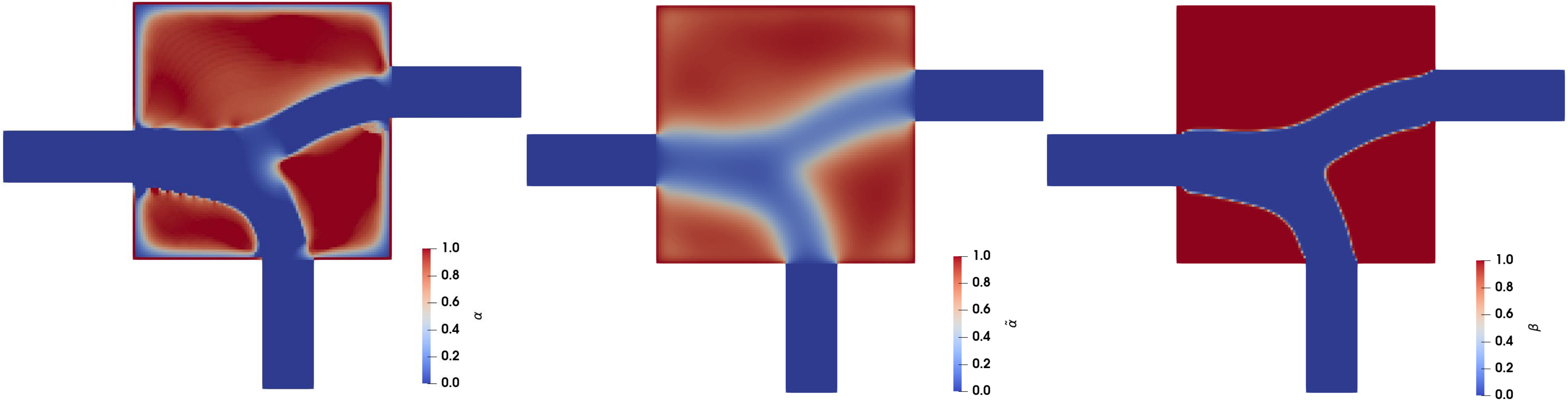
Application funded by TOYOTA

Topology Optimization (TopO)



- Primarily used for early design of duct systems with known inlets/outlets
- No shape parameterization
- Counter-productive cells are solidified through a source term in the flow equations
- $\boldsymbol{\beta \sim 1}$, solidified domain; theoretically, impermeable to flow
- $\boldsymbol{\beta = 0}$, flow domain
- Topology optimization: seeks optimal *β* fluid/solid identifier to minimize an objective function and satisfy the given constraints
- Number of design variables = Number of mesh cells
- Sensitivity derivatives computed with (continuous) adjoint

From design variables to Brinkman penalization terms



$$-r^2 \frac{\partial^2 \widetilde{\alpha}}{\partial x_j^2} + \widetilde{\alpha} = \alpha$$

Design variables → **Regularized field** → **Argument of Brinkman penalization terms**

Regularization

Projection/Sharpening

Primal Equations

- **Most general case examined: Navier-Stokes equations & the Spalart-Allmaras model for turbulent flows:**

Brinkman penalization terms

$$R^p = \frac{\partial v_i}{\partial x_i} = 0$$

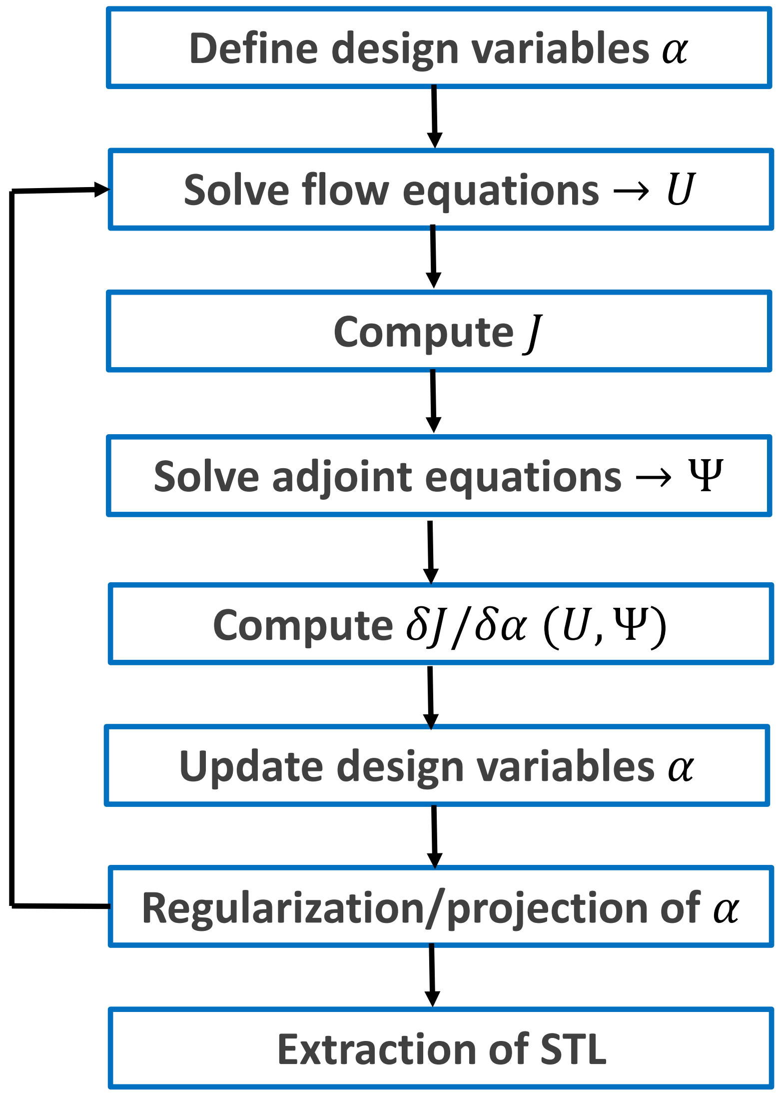
$$R_i^v = v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} + \frac{\partial p}{\partial x_i} + \underline{\beta_{max} I_v(\beta) v_i} = 0$$

$$R^{\widetilde{\nu}} = v_j \frac{\partial \widetilde{\nu}}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\widetilde{\nu}}{\sigma}\right)\frac{\partial \widetilde{\nu}}{\partial x_j}\right] - \frac{c_{b2}}{\sigma}\left(\frac{\partial \widetilde{\nu}}{\partial x_j}\right)^2 - \widetilde{\nu}\mathcal{P}(\widetilde{\nu}) + \widetilde{\nu}\mathcal{D}(\widetilde{\nu}) + \underline{\beta_{max} I_{\widetilde{\nu}}(\beta)\widetilde{\nu}} = 0$$

$$R^{\Delta} = \frac{\partial}{\partial x_j}\left(\Delta \frac{\partial \Delta}{\partial x_j}\right) - \Delta \frac{\partial^2 \Delta}{\partial x_j^2} - 1 + \underline{\beta_{max} I_{\Delta}(\beta)\Delta} = 0$$

- **β is related to the design variable field α**
- **β_{max} is a dimensioned constant ensuring that the variable computed by the PDE tends to zero when β is close to unity.**

Topology optimization loop

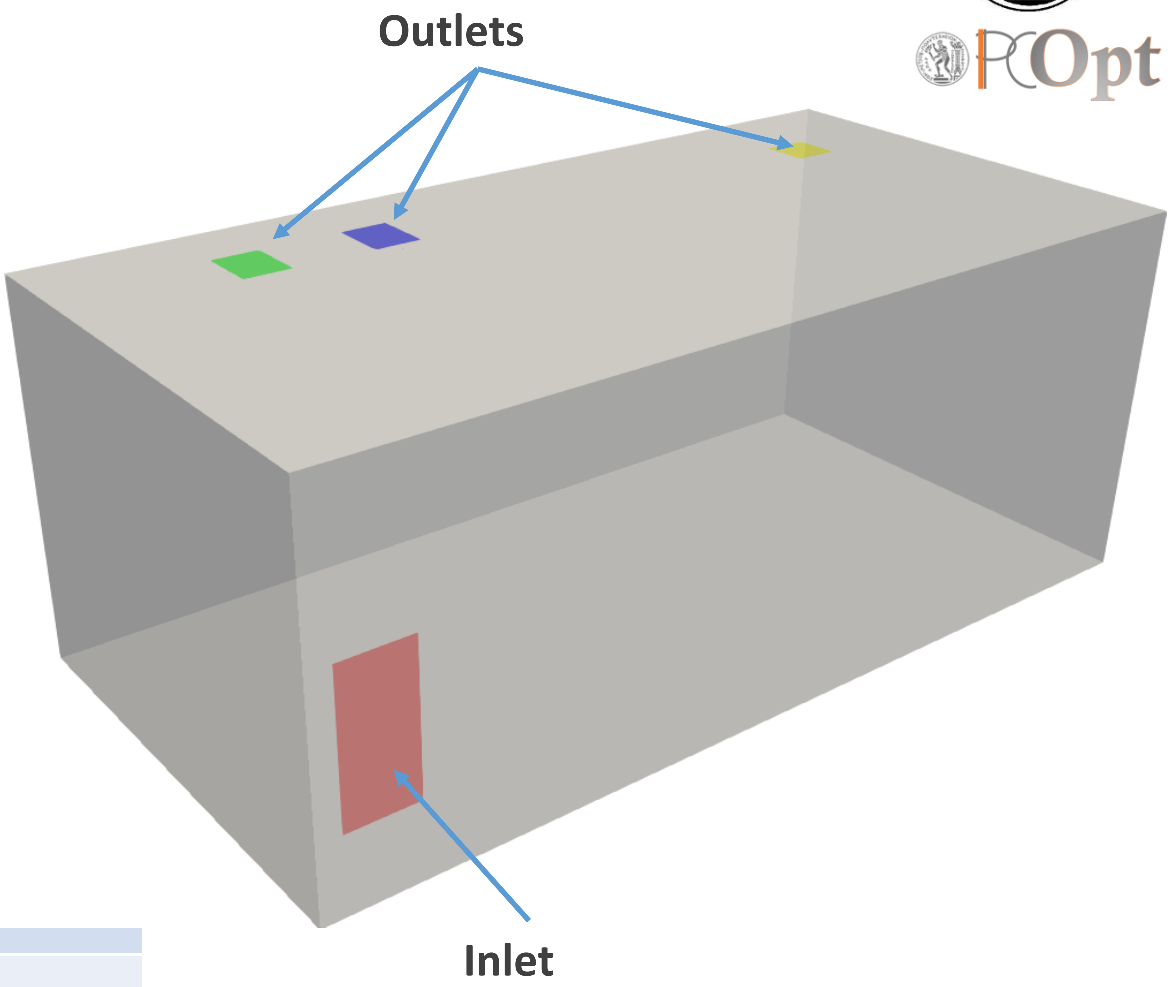


- Solve flow equations ⇨ **Augmented by the Brickman penalization terms**
- Compute J ⇨ **Quantity to be optimized (losses, uniformity, etc)**
- Compute $\delta J/\delta\alpha$ ⇨ **Drives the optimization loop**
- Update design variables ⇨ **Null space/MMA/SQP, allowing the utilization of inequality constraints**
- Regularization/projection ⇨ $\alpha \rightarrow \tilde{\alpha} \rightarrow \beta$: **Defines the next “geometry”**
- Extraction of STL ⇨ **Post-processing: make a body-fitted grid on the optimized geometry**

3D TopO, Foot channel HVAC duct

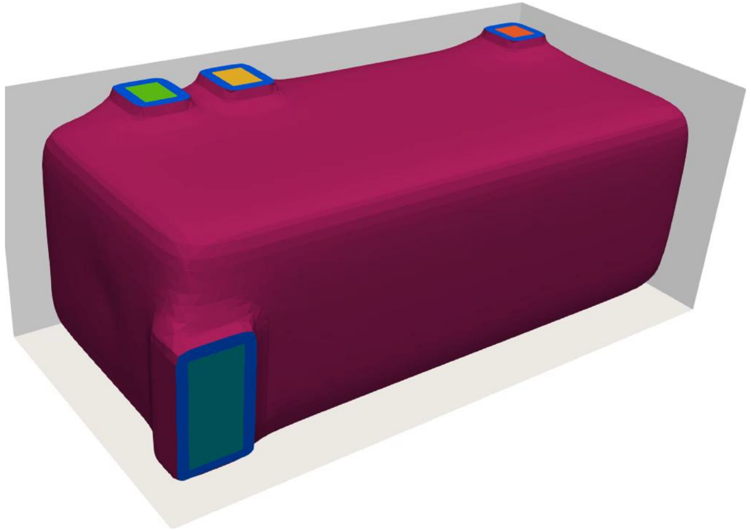
- $Re = 1.3 \times 10^5$ (turbulent flow, SA model)
- 1.1×10^5 design variables
- Multiple available objective functions

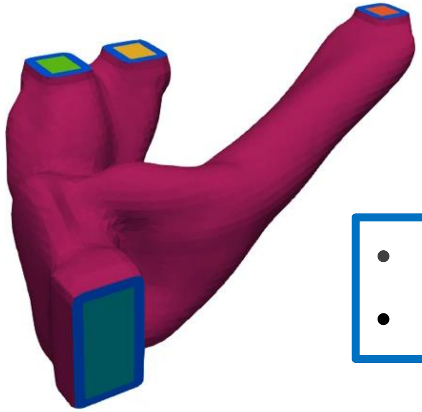
Objective	Formula
Total pressure losses	$J_{pt} = -\int_{S_{I,O}} p_t v_i n_i dS$
Flow rate partition	$J_m = 0.5 \sum_l (m_l - m_l^{tar})^2$ $m_l = -\int_{S_{O,l}} v_i n_i dS \, / \int_{S_I} v_i n_i dS$
Non-uniformity index	$J_u = 0.5 \sum_l \int_{S_l} (v_i - v_i^{mean})^2 dS$
Fluid volume	$V_F = \frac{\int_\Omega (1-\beta) d\Omega}{\int_\Omega d\Omega}$



3D TopO, Foot channel HVAC duct

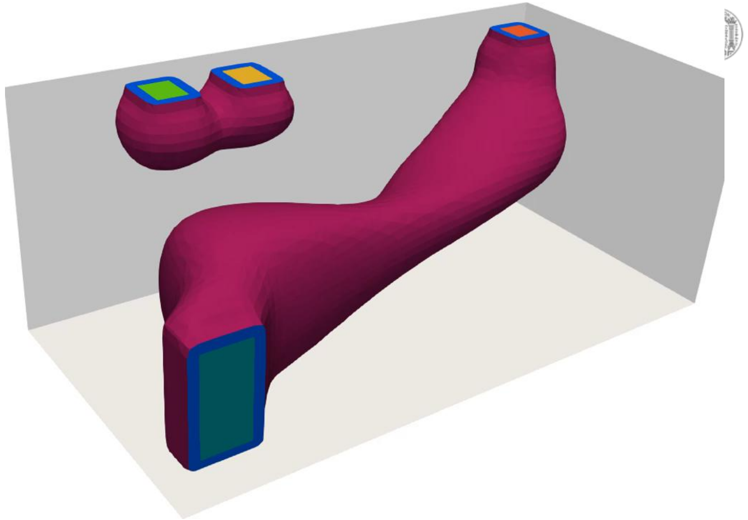
G1

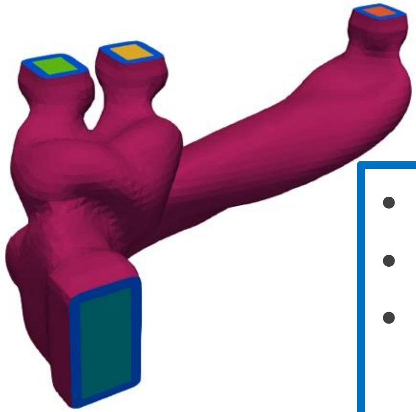




- Min. J_{p_t}
- $V_F < V_F^{tar}$

G2

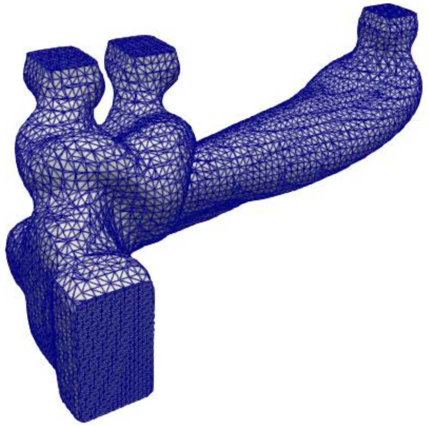




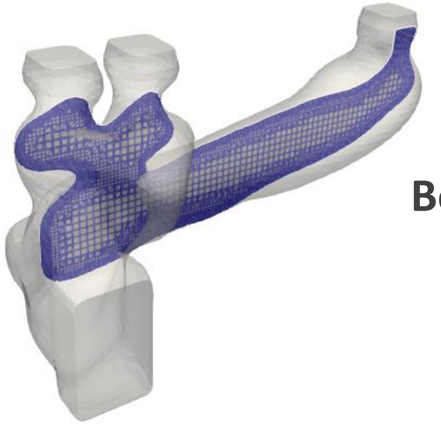
- Min. non – uniformity
- s.t.
- Equally-distribute flow rate
- $J_{p_t} < J_{p_t}^{tar}$
- $V_F < V_F^{tar}$

Foot channel HVAC duct: Re-evaluation on body-fitted meshes

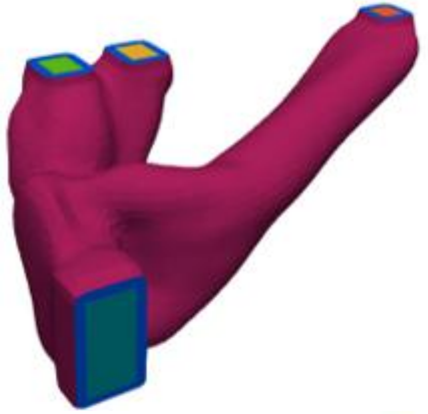
STL of the optimized geometry extracted by the topO code

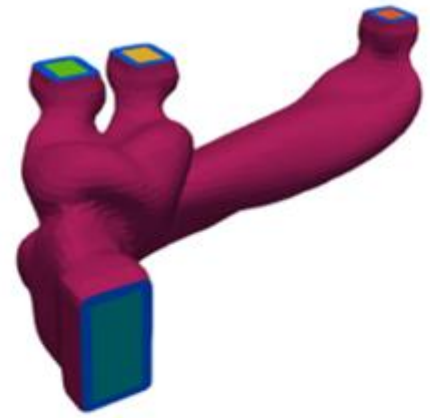


snappyHexMesh



Body-fitted evaluation





Geometry	J_{p_t}		Mass distribution		J_u	
	TopO	Body-fitted	TopO	Body-fitted	TopO	Body-fitted
G1	**15.43**	**13.69**	33/34.5/32.5	34/36/30	541	898
G2	22	16.7	**33/34/33**	**32/34/34**	**354**	**873**

Publicly Available Tools in OpenFOAM:

The latest version of the software can be downloaded from
https://develop.openfoam.com/Development/openfoam
The development branch can be found in
https://develop.openfoam.com/Development/openfoam/-/tree/develop
Extensive user-guide is available at
https://www.openfoam.com/documentation/files/adjointOptimisationFoamManual_v2312.pdf