Available online at www.eccomasproceedia.org
Eccomas Proceedia EUROGEN (2023) 189-205

ECCOMAS
Proceedia

EUROGEN 2023
15th ECCOMAS Thematic Conference on
Evolutionary and Deterministic Methods for Design, Optimization and Control
N. Gauger, K. Giannakoglou, M. Papadrakakis, J. Periaux (eds.)
Chania, Crete Greece, 1–3 June 2023

AN ADJOINT-BASED, PARAMETERIZATION-FREE FRAMEWORK FOR AERODYNAMIC SHAPE OPTIMIZATION IN OPENFOAM

Evangelos M. Papoutsis-Kiachagias[1], **and Kyriakos C. Giannakoglou**[1]

[1] Parallel CFD & Optimization Unit (PCOpt), School of Mechanical Engineering,
National Technical University of Athens (NTUA)
Zografou Campus, 9, Iroon Polytechniou str, 15772 Zografou, Athens, Greece
e-mail: {vpapout,kgianna}@mail.ntua.gr

Abstract. *This paper presents a parameterization-free framework for aerodynamic shape optimization based on the continuous adjoint method; the optimization controls a field of virtual boundary displacements and requires a regularization which acts on the aforementioned (likely noisy) field to transform them into a smooth displacement field, used to update the baseline geometry. Two different regularization approaches, based either on a Laplace-Beltrami equation solved on the surface of the designed geometry or a p-Laplacian equation solved in the entire computational domain, are utilized and assessed. To smoothly fade out the computed displacements close to the fixed part of the body shape (if any) and enforce a seamless synthesis of the optimized and fixed parts of the geometry, a proximity smoothing technique is also applied. The whole chain is differentiated, introducing the adjoint to the regularization equations and leading to the computation of accurate sensitivity derivatives of the objective function. The tool is implemented and tested on a $2D$ duct case as well as the optimization of the DrivAer car model. The framework is developed by extending the* adjointOptimisationFoam *tool-set of OpenFOAM, developed and made publicly available by the group of authors.*

Keywords: Parameterization-free Optimization, Node-based Shape Optimization, Laplace-Beltrami, p-Laplacian, Continuous Adjoint, OpenFOAM

Peer-review under responsibility of the organizing committee of EUROGEN 2023.
doi: 10.7712/140123.10199.18949

1 INTRODUCTION

Shape optimization usually employs some kind of parameterization to control the shape being designed, either based on a CAD model, [12], or by directly controlling the CFD grid (such as with a Free Form Deformation, FFD, tool [11]). These approaches have the advantage of producing smooth geometries controlled by a relatively small number of design variables. However, such a treatment may lead to sub-optimal solutions since these parameterizations restrict the design space. With an FFD tool, in particular, keeping certain parts of the designed geometry fixed can become quite challenging as all nodes laying within the morphing lattice(s) move when the positions of the control points change.

Alternatively, one could use the displacement of the boundary wall nodes or faces as the design variables. However, the sensitivity derivatives of the objective function with respect to (w.r.t.) the latter are often noisy, resulting to non-smooth optimized shapes. To deal with this issue, some kind of regularization of the sensitivity derivatives, computed either explicitly, [15, 3] or implicitly, [4, 13, 8] must be employed; a schematic representation of such an approach is given in fig. 1. The present work follows an implicit approach but formulates the optimization problem in a way that does not regularize the sensitivity derivatives ad hoc. Instead, a new set of design variables is introduced, which can be seen as the non-smooth vectorial displacement of the boundary to be optimized. Then, regularization is applied to the design variables and converts them to a smooth displacement field which can be used to update the shape. Adding the adjoint to the regularization process into the (continuous) adjoint method, [10, 5], leads to the computation of accurate sensitivity derivatives of the objective function w.r.t. these design variables. This allows the utilization of state-of-the art update methods for the design variables, like BFGS and, more importantly, allows the treatment of constraints (for instance through SQP) as with any other parameterization; the latter is challenging with approaches that regularize only the sensitivity derivatives [8].

The developed shape optimization framework is demonstrated with two regularization methods, one based on a Laplace-Beltrami equation solved only on the designed surface, [4], and one based on a p-Laplacian equation solved in the entire computational domain, [8]. The two variants are initially tested in a $2D$ internal aerodynamics case and, then, used for the drag minimization of the DrivAer car model, [2]. The developed method is implemented within the in-house variant of the publicly available *adjointOptimisationFoam* tool-set, programmed by the group of authors within the OpenFOAM environment.

2 DESIGN VARIABLES AND SHAPE UPDATE

Let b_i^f, $i \in [1, d]$ with $d = 2(3)$ for $2D(3D)$ problems and $f \in [1, N_f]$ be a vector defined on the N_f faces of the part of the geometry to be designed (active part of the geometry). In what follows, superscript f will be used to indicate that a quantity is computed/stored on the aforementioned boundary faces. The $N = N_f d$ components of b_i^f stand as the design variables of the optimization problem and can be seen as the (possibly non-smooth) displacement of each boundary face from its original position $x_i^{f,0}$. If b_i^f were used to directly update the positions of the boundary faces through a gradient-based optimization algorithm, non-smooth geometries might emerge, [15]; these can be attributed to the fact that the derivatives of the objective function J w.r.t. each b_i^f contain information about this face only, with no mechanism to enforce the regularity of the final shape. Hence, some kind of regularization should be applied to obtain a smooth boundary displacement, denoted by $m_i, i \in [1, d]$ in the continuous sense (or m_i^f, as seen from a discrete perspective). In this article, the m_i field is computed through the solu-

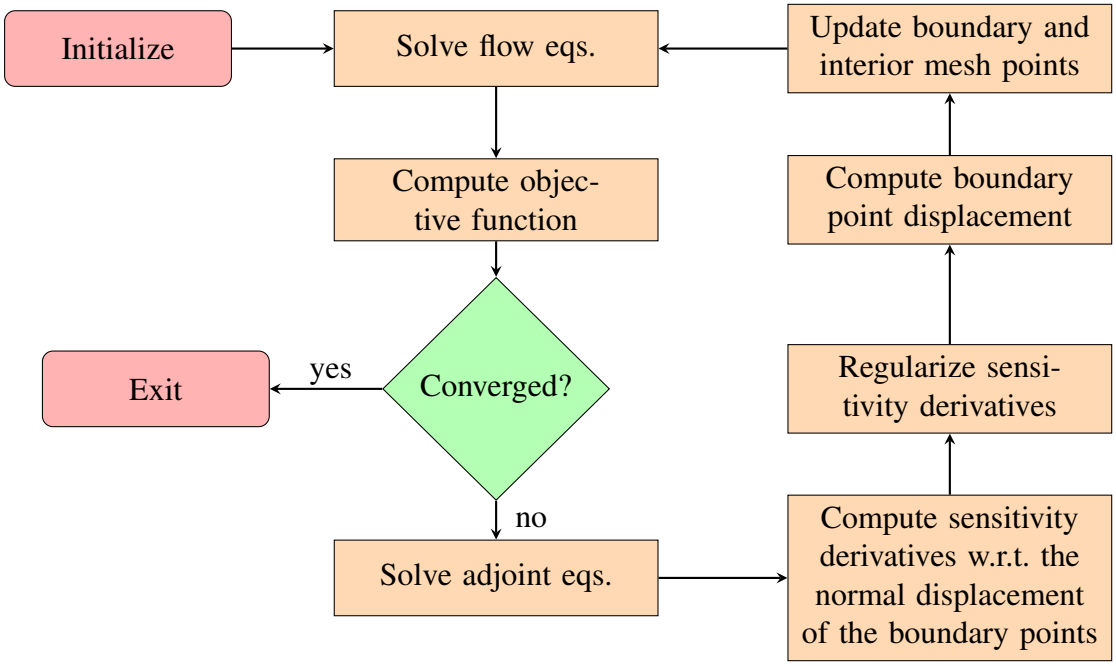


Figure 1: Flow chart of a typical optimization workflow based on the regularization of the sensitivity derivatives (as in [4, 15, 13, 8].)

tion of a regularization Partial Differential Equation (PDE) R_i^r, solved either on the designed geometry or throughout the entire domain, see section 2.1. We, additionally, introduce the $\widetilde{m_i}$ displacement field as

$$\widetilde{m_i} = f(r_g) m_i, \quad i \in [1, d] \tag{1a}$$

$$f(r_g) = min(-2r_g^3 + 3r_g^2, 1) \tag{1b}$$

where r_g are geodesic distances from the interface of the active and fixed parts of the geometry, normalized by r_p, to be referred to as the proximity threshold. The purpose of eq. 1a is to smoothly fade out m_i values close to the fixed part of the geometry, maintaining C^0 and C^1 continuity, fig. 3b; any other function with similar attributes could be used instead of eq. 1b. We will hereafter refer to the process described by eqs. 1 as the proximity smoothing.

Finally, the update of the coordinates of point p is given by

$$\Delta x_i^p = \sum_{f \in (p)} w^{fp} \widetilde{m_i}^f \tag{2}$$

where $f \in (p)$ denotes all faces having p as a vertex and w^{fp} are weights, computed using the inverse distance of the face center of f to point p and normalized so that $\sum_{f \in (p)} w^{fp} = 1$. It should be noted that, in order to fully define the updated geometry using only the initial one and the design variables field, the displacement field is always added to the coordinates of the initial geometry. For the same reason, R_i^r and $f(r_g)$ are solved/computed on the initial geometry at any optimization cycle. The workflow of the proposed shape optimization framework is given in fig. 2, after briefly explaining the rest of its constituents.

2.1 The regularization PDEs

Two regularization PDEs are used in this paper, one based on a Laplace-Beltrami operator and solved on the active parts of the geometry and one based on a p-Laplacian PDE solved in

the entire computational domain.

2.1.1 Laplace-Beltrami

The Laplace-Beltrami PDEs, [4], read

$$R_i^{r,LB} = -r_{LB}^2 \frac{\partial^2 m_i}{\partial x_{s,j}^2} + m_i - b_i = 0,\ i \in [1,d] \tag{3}$$

where $\frac{\partial(\cdot)}{\partial x_{s,j}}$ is the gradient operator constrained on the active part of the geometry S; zero Neumann boundary conditions are imposed on its boundary, ∂S. The solution of eq. 3 is the field that minimizes an integral objective function comprising the smoothness of m_i and its deviation from b_i, weighted through the regularization radius r_{LB}. We choose to regularize the vectorial displacement b_i, instead of its normal component only, to also take in-plane regularization into account. Eq. 3 is numerically solved using the Finite Area Method (FAM, [16]), which resembles the Finite Volume Method (FVM) but is dedicated in solving PDEs which are constrained on curved surfaces.

2.1.2 p-Laplacian

Inspired by [8], the p-Laplacian regularization PDE is defined through the computational domain as

$$R_i^{r,pL} = \frac{\partial}{\partial x_j}\left[|\nabla \vec{m}|^{p-2}\,\frac{\partial m_i}{\partial x_j}\right] = 0 \tag{4}$$

with boundary conditions expressed by

$$r_{pL}\,|\nabla \vec{m}|^{p-2}\,\frac{\partial m_i}{\partial x_j} n_j + m_i - b_i = 0 \tag{5}$$

where $|\nabla \vec{m}| = \sqrt{\frac{\partial m_k}{\partial x_l}\frac{\partial m_k}{\partial x_l}}$ and r_{pL} is a regularization radius. The non-linear diffusivity in eq. 4, namely $|\nabla \vec{m}|^{p-2}$, penalizes mesh areas with high deformation gradients, with higher p values increasing this penalization. If $p=2$, eq. 4 corresponds to a Laplace equation with a Robin-type boundary condition. According to eq. 5, regularization is mainly conducted in the normal to the boundary direction, with in-plane regularization taking place due to the indirect connection of adjacent boundary values through eq. 4. Eq. 4 is solved using typical infrastructure existing in any FVM software. The solution of eq. 4 can also be used to update the positions of both boundary and interior mesh nodes or, in other words, eq. 4 can also be used as the grid displacement PDE.

3 COMPUTATION OF SENSITIVITY DERIVATIVES

The flow problem is governed by the Navier-Stokes equations for incompressible fluids, coupled with the Spalart-Allmaras turbulence model, [14]. The detailed development of the continuous adjoint equations can be found in [10] and the computation of the sensitivity derivatives of the objective function J w.r.t. the coordinates of a boundary point x_i^p is conducted using the the so-called E-SI continuous adjoint approach, introduced in [5]. Here, we focus exclusively on differentiating the chain presented in section 2 to compute $\delta J/\delta b_i^f$, starting from $\delta J/\delta x_i^p$.

By applying the chain rule, we get

$$\frac{\delta J}{\delta b_i^f} = \sum_{l=1}^{N_f} \sum_{j=1}^{d} \frac{\delta J}{\delta m_j^l} \frac{\delta {m_j}^l}{\delta b_i^f} \tag{6}$$

$$\frac{\delta J}{\delta m_j^l} = \sum_{k=1}^{d} \frac{\delta J}{\delta \widetilde{m}_k^l} \frac{\delta \widetilde{m}_k^l}{\delta {m_j}^l} \tag{7}$$

The derivative of J w.r.t. the face displacement field $\widetilde{m_k}$ at a face l can be computed after taking eq. 2 into consideration as

$$\frac{\delta J}{\delta \widetilde{m_k}^l} = \sum_{j=1}^{d} \sum_{p} \frac{\delta J}{\delta x_j^p} \frac{\delta x_j^p}{\delta \widetilde{m_k}^l} = \sum_{j=1}^{d} \sum_{p} \frac{\delta J}{\delta x_j^p} \sum_{f \in (p)} w^{fp} \frac{\delta x_j^f}{\delta x_k^l} = \sum_{p \in (l)} w^{lp} \frac{\delta J}{\delta x_k^p} \tag{8}$$

where $p \in (l)$ are the points that belong to face l. It is interesting to note that sensitivities w.r.t. the face displacement are computed using a point-to-face interpolation using the same weights used for the face-to-point interpolation of the boundary displacement, eq. 2. Another interesting observation is that, in the case of the point-to-face interpolation of the sensitivities, the interpolation weights do not sum up to one. Additionally, according to eq. 1a, $\frac{\delta \widetilde{m}_k^l}{\delta {m_j}^l} = f(r_g^l)\delta_{kj}$, where δ_{kj} is the Kronecker symbol.

The computation of $\delta m_j / \delta b_i^f$ in eq. 6 would require the differentiation of eq. 3 or 4 w.r.t. b_i^f which requires the solution of N differentiated regularization PDEs. To reduce this cost, the adjoint to the regularization PDEs can be devised. This is developed separately for $R_i^{r,LB}$ and $R_i^{r,pL}$ in subsections 3.1 and 3.2.

3.1 Adjoint to the Laplace-Beltrami (LB) PDE

Let us consider the Lagrangian function $L = J - \int_S \Psi_k^{LB} R_k^{r,LB} dS$, where Ψ_i^{LB} is the adjoint to m_i, as computed by eq. 3. Its derivative w.r.t. b_i^f reads

$$\begin{aligned}\frac{\delta L}{\delta b_i^f} = &\int_S \left(\frac{1}{\Delta S} \frac{\delta J}{\delta m_k} + r_{LB}^2 \frac{\partial^2 \Psi_k^{LB}}{\partial x_{s,j}^2} - \Psi_k^{LB} \right) \frac{\delta m_k}{\delta b_i^f} dS + \int_S \Psi_k^{LB} \frac{\delta b_k}{\delta b_i^f} dS \\ &+ \int_{\partial S} r_{LB}^2 \Psi_k^{LB} \frac{\partial}{\partial x_{s,j}} \left(\frac{\delta m_k}{\delta b_i^f} \right) n_{s,j} \partial(dS) - \int_{\partial S} r_{LB}^2 \frac{\partial \Psi_k^{LB}}{\partial x_{s,j}} n_{s,j} \frac{\delta m_k}{\delta b_i^f} \partial(dS)\end{aligned} \tag{9}$$

where $n_{s,j}$ are the components of the normal vectors to ∂S, constrained on S. To facilitate the derivation of the continuous adjoint to $R_i^{r,LB}$, the discrete form of eq. 6 was converted into the first term in the first integral on the r.h.s. of in eq. 9, with ΔS being the local face area. To avoid the computation of $\delta m_k / \delta b_i^f$, its multiplier in the surface integral of eq. 9 is set to zero, formulating the adjoint Laplace-Beltrami PDEs as

$$R_{\Psi_k}^{LB} = -r_{LB}^2 \frac{\partial^2 \Psi_k^{LB}}{\partial x_{s,j}^2} + \Psi_k^{LB} - \frac{1}{\Delta S} \frac{\delta J}{\delta m_k} = 0 \, , k \in [1, d] \tag{10}$$

with zero Neumann conditions on ∂S, which nullify the last term on the r.h.s. of eq. 9. The third term on the r.h.s. of the same equation vanishes automatically due to the Neumann conditions imposed on m_i. After satisfying the adjoint Laplace-Beltrami PDEs and their boundary

conditions, the sensitivity derivatives are given by

$$\frac{\delta L}{\delta b_i^f} = \int_S \Psi_k^{LB} \frac{\delta b_k}{\delta b_i^f} dS = \left. \Psi_i^{LB} \right|^f \Delta S^f \tag{11}$$

i.e. they are computed as the solution of the adjoint Laplace-Beltrami PDEs, multiplied by the local face area. It is interesting to note that eq. 10 is the equivalent to eq. 3, with the sensitivity derivative w.r.t. the face displacement being regularized, instead of the face displacement.

3.2 Adjoint to the p-Laplacian (pL) PDE

Following a similar process, we define $L = J + \int_\Omega \Psi_k^{pL} R_k^{r,pL} d\Omega$ and differentiate it w.r.t. b_i^f to get

$$\begin{aligned}
\frac{\delta L}{\delta b_i^f} = & \int_S \frac{1}{\Delta S} \frac{\delta J}{\delta m_k} \frac{\delta m_k}{\delta b_i^f} dS + \int_S (p-2) \left|\nabla \vec{m}\right|^{p-4} \Psi_k^{pL} \frac{\partial m_n}{\partial x_l} \frac{\partial m_k}{\partial x_j} n_j \frac{\partial}{\partial x_l} \left(\frac{\delta m_n}{\delta b_i^f} \right) dS \\
& - \int_S (p-2) \left|\nabla \vec{m}\right|^{p-4} \frac{\partial \Psi_k^{pL}}{\partial x_j} \frac{\partial m_n}{\partial x_l} n_l \frac{\partial m_k}{\partial x_j} \frac{\delta m_n}{\delta b_i^f} dS + \int_S \left|\nabla \vec{m}\right|^{p-2} \Psi_k^{pL} \frac{\partial}{\partial x_j} \left(\frac{\delta m_k}{\delta b_i^f} \right) n_j dS \\
& - \int_S \left|\nabla \vec{m}\right|^{p-2} \frac{\partial \Psi_k^{pL}}{\partial x_j} n_j \frac{\delta m_k}{\delta b_i^f} dS + \int_\Omega (p-2) \frac{\partial}{\partial x_l} \left[\left|\nabla \vec{m}\right|^{p-4} \frac{\partial \Psi_n^{pL}}{\partial x_j} \frac{\partial m_n}{\partial x_j} \frac{\partial m_k}{\partial x_l} \right] \frac{\delta m_k}{\delta b_i^f} d\Omega \\
& + \int_\Omega \frac{\partial}{\partial x_j} \left[\left|\nabla \vec{m}\right|^{p-2} \frac{\partial \Psi_k^{pL}}{\partial x_j} \right] \frac{\delta m_k}{\delta b_i^f} d\Omega
\end{aligned} \tag{12}$$

After differentiating eq. 5, the sum of the second and fourth integrals on the r.h.s. of eq. 12 is transformed into integrals that contain only $\delta m_k / \delta b_i^f$ and eq. 12 becomes

$$\begin{aligned}
\frac{\delta L}{\delta b_i^f} = & \int_S \left[\frac{1}{\Delta S} \frac{\delta J}{\delta m_k} - \frac{\Psi_k^{pL}}{r_{pL}} - (p-2) \left|\nabla \vec{m}\right|^{p-4} \frac{\partial \Psi_n^{pL}}{\partial x_j} \frac{\partial m_n}{\partial x_j} \frac{\partial m_k}{\partial x_l} n_l - \left|\nabla \vec{m}\right|^{p-2} \frac{\partial \Psi_k^{pL}}{\partial x_j} n_j \right] \frac{\delta m_k}{\delta b_i^f} dS \\
& + \int_\Omega \left[\frac{\partial}{\partial x_j} \left(\left|\nabla \vec{m}\right|^{p-2} \frac{\partial \Psi_k^{pL}}{\partial x_j} \right) + (p-2) \frac{\partial}{\partial x_l} \left(\left|\nabla \vec{m}\right|^{p-4} \frac{\partial \Psi_n^{pL}}{\partial x_j} \frac{\partial m_n}{\partial x_j} \frac{\partial m_k}{\partial x_l} \right) \right] \frac{\delta m_k}{\delta b_i^f} d\Omega \\
& + \int_S \frac{\Psi_k^{pL}}{r_{pL}} \frac{\delta b_k}{\delta b_i^f} dS
\end{aligned} \tag{13}$$

To avoid the computation of $\delta m_k / \delta b_i^f$, its multipliers in eq. 13 are set to zero, resulting to the adjoint p-Laplacian PDEs

$$R_{\Psi_k}^{pL} = \frac{\partial}{\partial x_j} \left(\left|\nabla \vec{m}\right|^{p-2} \frac{\partial \Psi_k^{pL}}{\partial x_j} \right) + (p-2) \frac{\partial}{\partial x_l} \left(\left|\nabla \vec{m}\right|^{p-4} \frac{\partial \Psi_n^{pL}}{\partial x_j} \frac{\partial m_n}{\partial x_j} \frac{\partial m_k}{\partial x_l} \right) = 0, \; k \in [1, d] \tag{14}$$

and their boundary conditions

$$\left|\nabla \vec{m}\right|^{p-2} \frac{\partial \Psi_k^{pL}}{\partial x_j} n_j + \frac{\Psi_k^{pL}}{r_{pL}} + (p-2) \left|\nabla \vec{m}\right|^{p-4} \frac{\partial \Psi_n^{pL}}{\partial x_j} \frac{\partial m_n}{\partial x_j} \frac{\partial m_k}{\partial x_l} n_l - \frac{1}{\Delta S} \frac{\delta J}{\delta m_k} = 0 \tag{15}$$

Finally, the sensitivity derivatives are given by

$$\frac{\delta L}{\delta b_i^f} = \int_S \frac{\Psi_k^{pL}}{r_{pL}} \frac{\delta b_k}{\delta b_i^f} dS = \left. \Psi_i^{pL} \right|^f \frac{\Delta S^f}{r_{pL}} \tag{16}$$

It is interesting to note that for $p \neq 2$ and for the initial geometry with $m_i = 0$ everywhere, eq. 4 is not differentiable due to the presence of $|\nabla \vec{m}|^{p-2}$. One can, however, start the optimization with $p = 2$, in which case eqs. 14 and 15 simplify to

$$R_{\Psi_k}^{pL} = \frac{\partial^2 \Psi_k^{pL}}{\partial x_j^2} = 0 \tag{17a}$$

$$\frac{\partial \Psi_k^{pL}}{\partial x_j} n_j + \frac{\Psi_k^{pL}}{r_{pL}} - \frac{1}{\Delta S} \frac{\delta J}{\delta m_k} = 0 \tag{17b}$$

and increase p in the subsequent optimization cycles.

A flow chart of the complete shape optimization framework is given in fig. 2.

4 APPLICATIONS

The shape optimization workflow summarized in fig. 2 is initially applied to the optimization of a $2D$ S-shaped duct and, then, to the DrivAer car model.

4.1 S-shaped duct

The shape optimization of the S-shaped duct of fig. 3a, targets the minimization of total pressure losses between its inlet and outlet. The flow is laminar with $Re = 1000$, a structured grid of $24K$ cells is used and the active part of the geometry contains 226 faces located on the two curved parts of the boundaries, leading to 452 design variables. L-BFGS, [6], is used to update the design variables, coupled with a line-search approach that satisfies the strong Wolfe conditions, [9], and the boundary displacement is propagated to the interior grid nodes using the Inverse Distance Weighting method, [7]. Optimizations run for 20 cycles at most, unless the relative reduction in the J value between two optimization cycles is smaller than 10^{-4}.

The workflow of fig. 2 is applied using both the Laplace-Beltrami (LB) and the p-Laplacian (pL) as regularization PDEs. In all cases, $r_p = 0.5m$ (see section 2), compared to the length of the active part of the geometry which is $2m$. The distribution of the proximity filter, $f(r_g)$ in eq. 1b, along the length of the initial geometry is shown in fig. 3b.

Concerning the LB regularization, it can be observed that $r_{LB} \in [0.25, 1]m$ lead to optimized geometries with almost the same J value, fig. 4a, even though the resulting geometries differ noticeably from each other, fig. 4b. From the same figure, we can observe that larger r_{LB} values lead to geometries with larger wavelength features, as expected. Additionally, increasing r_{LB} beyond a certain threshold ($4m$ in this case) does not seem to have any impact on the optimized geometry or the convergence of the optimization (in figs. 4a and 4b , curves corresponding to $r_{LB} = 4m$ and $5m$ practically coincide). This can be interpreted by viewing r_{LB} as the equivalent smoothing radius of an explicit regularization method using a Gaussian kernel, [15], above which all nodes of the surface are taken into consideration when computing $\delta J / \delta b_i^f$ for a specific face. Fig. 5 compares the values of $\delta J / \delta b_i^f$, for various r_{LB} radii, with the values of $\delta J / \delta \widetilde{m_i}^f$ (i.e. the regularized and non-regularized sensitivities). It can be observed that larger r_{LB} values gradually smooth out the picks of the non-regularized sensitivities, up to a point where increasing r_{LB} further has no impact on the result. Finally, the virtual and actual

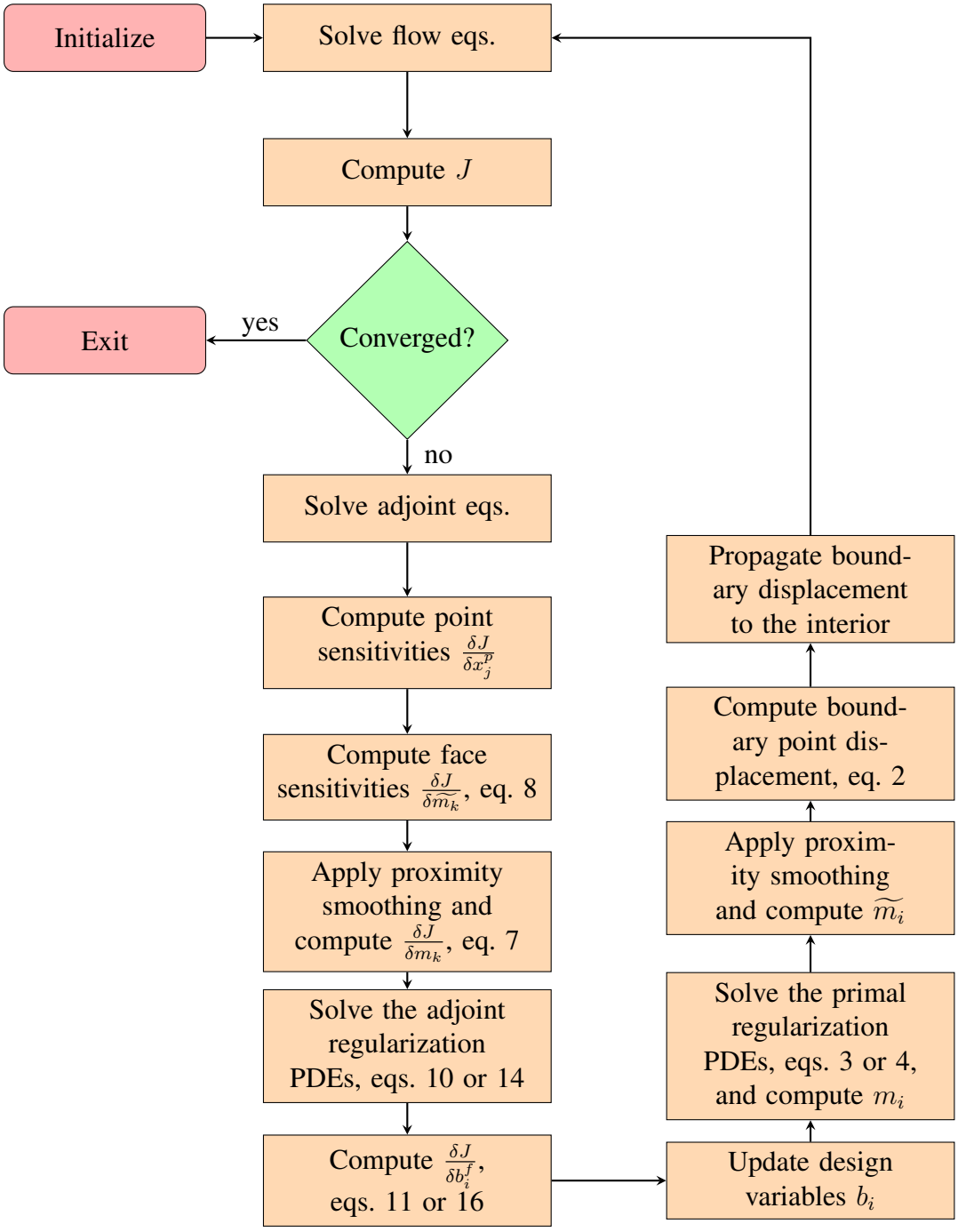
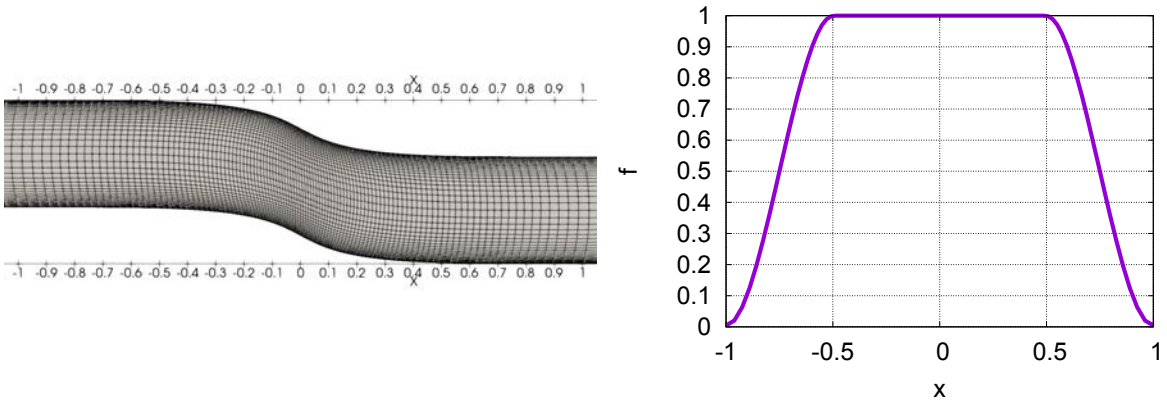


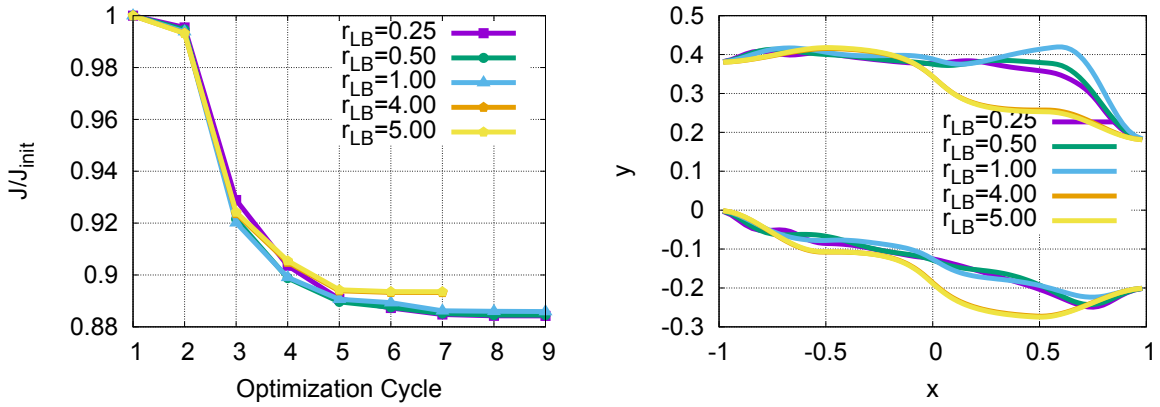
Figure 2: Flow chart of the proposed optimization workflow.

displacements corresponding to the optimized duct geometry computed with $r_{LB} = 0.2m$ are plotted in fig. 6. The impact of proximity smoothing close to the boundaries of the duct is evident, with regularization seemingly playing a smaller role in this case.



(a) (b)

Figure 3: S-shaped duct: (a) active part of the geometry and mesh, focused around the S-shaped part of the duct (the mesh is further extended upstream and downstream) and (b) the distribution of the proximity smoothing function of eq. 1b.



(a) (b)

Figure 4: S-shaped duct: (a) convergence history, normalized with the first J value, and (b) shapes (not in scale) of the optimized ducts obtained using LB regularization with different regularization radii, r_{LB}.

The convergence history and optimized geometries obtained using the pL regularization with different r_{LP} values and $p = 2$ are presented in figs. 7a and 7b, respectively. Smaller r_{pL} values tend to produce geometries with smaller J values. Additionally, geometries obtained using different r_{pL} values do not differ significantly from each other, even for large r_{pL} values. This can be justified by examining the $\delta J/\delta b_i^f$ values for various r_{pL} in fig. 8. From there, it can be observed that increasing r_{pL} beyond a certain value seems to only scale the sensitivity derivatives with $1/r_{pL}$, without changing their spatially distribution significantly. This indicates

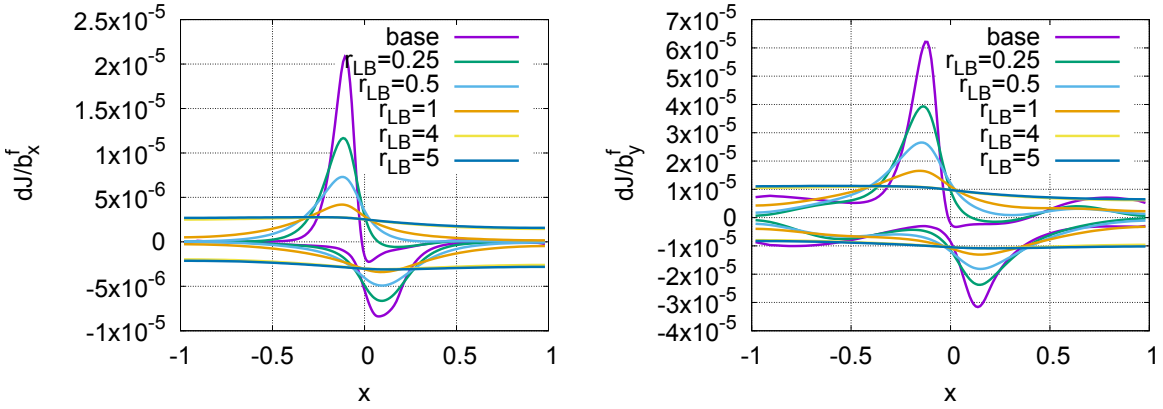


Figure 5: S-shaped duct: sensitivity derivatives of J obtained using LB regularization with different r_{LB} values. Curves labeled with "base" correspond to the non-regularized sensitivities, $\delta J/\delta \widetilde{m_i}^f$. Curves corresponding to $r_{LB}=4m$ and $5m$ are practically indistinguishable.

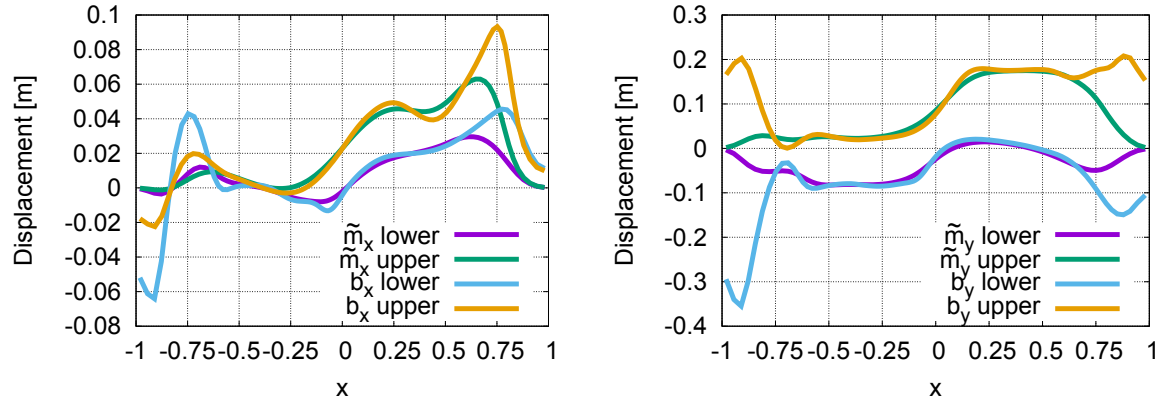
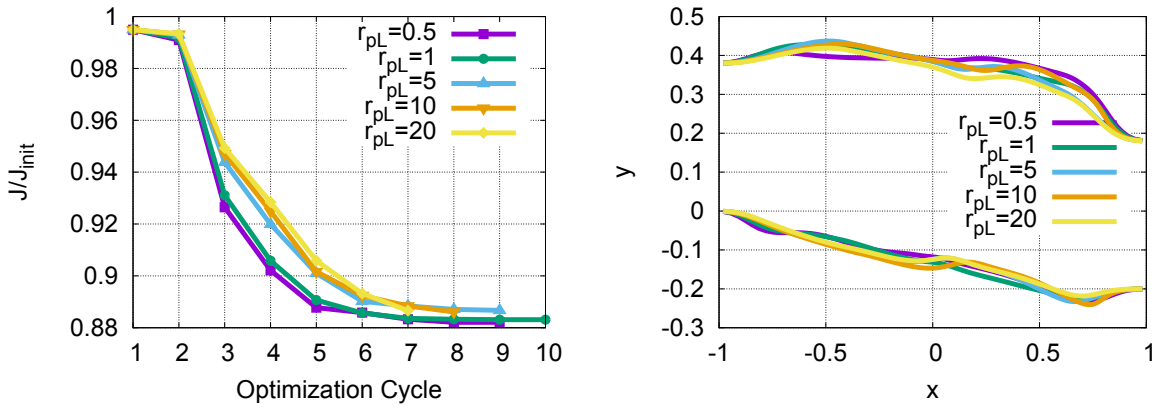


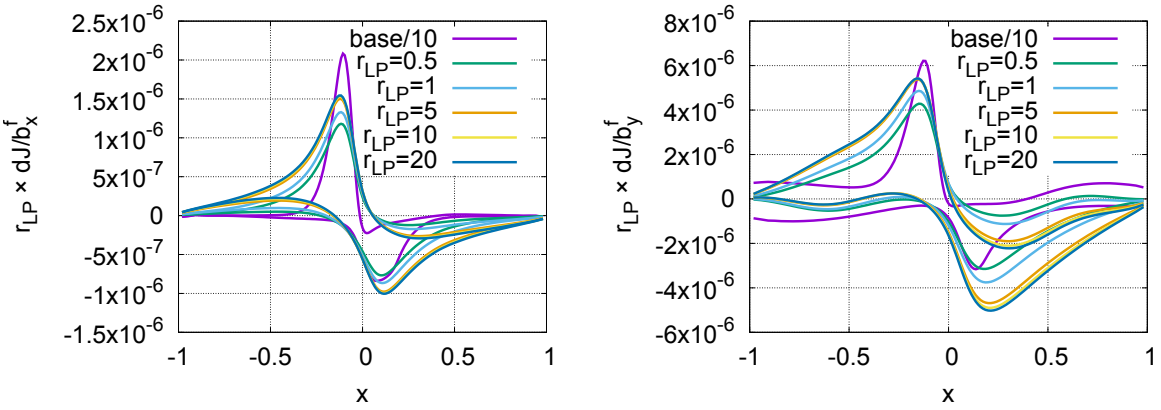
Figure 6: S-shaped duct: x (left) and y (right) components of the virtual ($\vec{b}$) and actual ($\vec{\widetilde{m}}$) face displacement fields, plotted over the lower and upper walls of the LB optimized geometry, with $r_{LB}=0.2m$.

that the p-Laplacian regularization cannot smooth out certain features of $\delta J/\delta \widetilde{m_i}^f$; the impact of this will become apparent in the optimization of the DrivAer car model.



(a) (b)

Figure 7: S-shaped duct: (left) convergence history and (right) shapes of the optimized ducts obtained using pL regularization with different r_{LP} values and $p=2$.



(a) x component of the sensitivity derivatives (b) y component of the sensitivity derivatives

Figure 8: S-shaped duct: sensitivity derivatives of J obtained using pL regularization with different r_{pL} values and $p=2$; all regularized sensitivities are multiplied with their corresponding r_{pL} value. Curves labeled with "base" correspond to the non-regularized sensitivities, $\delta J/\delta \widetilde{m_i}^f$, and have been divided by 10 to fit into the scale of the plot.

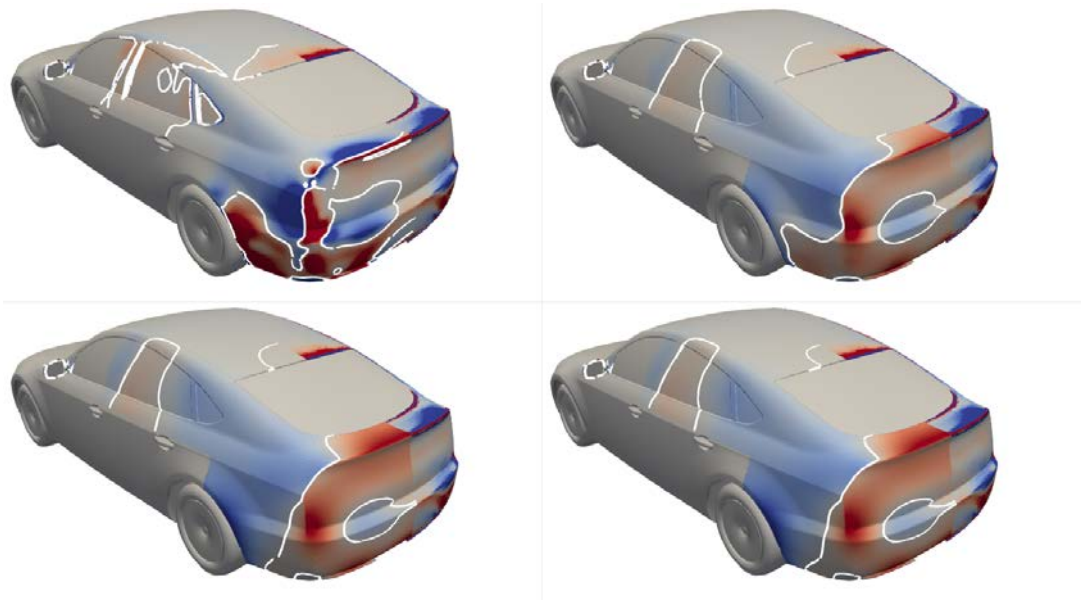
4.2 DrivAer Car Model

The shape optimization framework described in section 2 is applied to the drag minimization of the fast-back configuration of the DrivAer car model, with a smooth underbody, with mirrors and wheels (F_S_wm_ww). A CFD grid of approximately 5.3 million cells is used to model half of the car, the simulation includes a moving road, rotating wheels (through an appropriate boundary condition, since the wheels are closed) and turbulence is modeled by means of the Spalart-Allmaras model with wall functions. Since a steady state primal solver is used, J cannot reach a constant value within each optimization cycle but oscillates around a "mean" value. Each flow evaluation performs 5500 iterations and the objective function results from the averaging of the last 3500 of them.

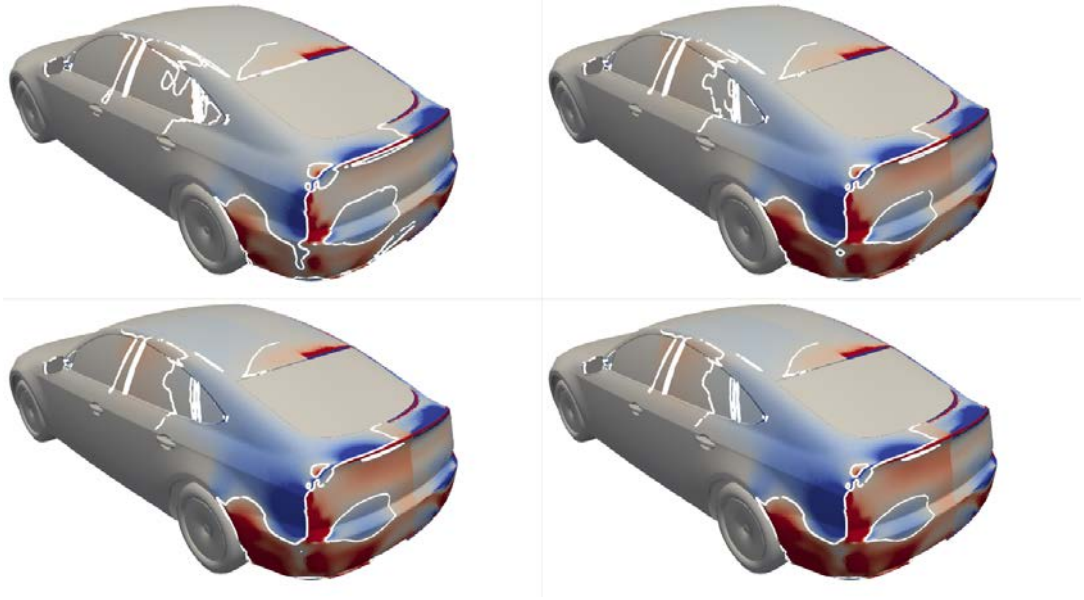
The active part of the geometry includes the back side of the car, excluding the rear window, and the diffuser, fig. 9. Before performing an optimization, an investigation is first made on the impact of the regularization radii in eq. 10 and 15 on $\delta J/\delta b_i^f$. For this purpose, the latter is computed for various r_{LB} and r_{PL} values and its projection to the local unit normal vector (the so-called sensitivity map) is compared with the non-regularized sensitivity field, $(\delta J/\delta \widetilde{m_i}^f)n_i^f$, in figs. 10 and 11. Sensitivities regularized with LB do not change significantly above $r_{LB} = 2m$ and all small wavelength features have been smoothed out. This is the r_{LB} value used in the optimization. On the other hand, and in agreement with what was observed in fig. 5 for the S-shaped duct, pL cannot smooth out the small wavelength features of the sensitivities, irrespective of the r_{pL} value. The sensitivity maps obtained using pL and $r_{pL} > 0.5m$ appear similar to the ones obtained with LB and $r_{LB} = 0.1m$, i.e. the smallest value from the ones tested herein. For the pL-based optimization that follows, $r_{pL} = 10m$ is used.

Figure 9: DrivAer: The active part of the geometry, i.e. the part allowed to move during the optimization, is colored in red. Only half of the geometry is simulated, using symmetry conditions. The geometry is then, mirrored, for visualization.

Having picked a proper value for the regularization radius, two optimizations were performed based on the two regularization approaches. Twenty optimization cycles were performed using the Conjugate Gradient method, [1], to update the design variables; the two optimized geometries will be referred to as gLB and gpL, from the initials of the corresponding regularization methods used to design them. Irrespective of the regularization method used, the boundary displacement is propagated to the interior using eq. 4 with $p=4$ and Dirichlet conditions computed through eq. 2. The gLB geometry led to a 5.1% drag reduction while the gpL one had a 14.7% reduction. The large difference in drag reduction can be attributed to the fact that LB filtered all small-scale features of the sensitivities leading to a much smoother shape while pL retained some of these features, leading to better performance but also to a less aesthetically pleasing and potentially non-manufacturable solution. Despite this, the complex gpL geometry showcases the robustness of the shape optimization framework. The geometries obtained from the



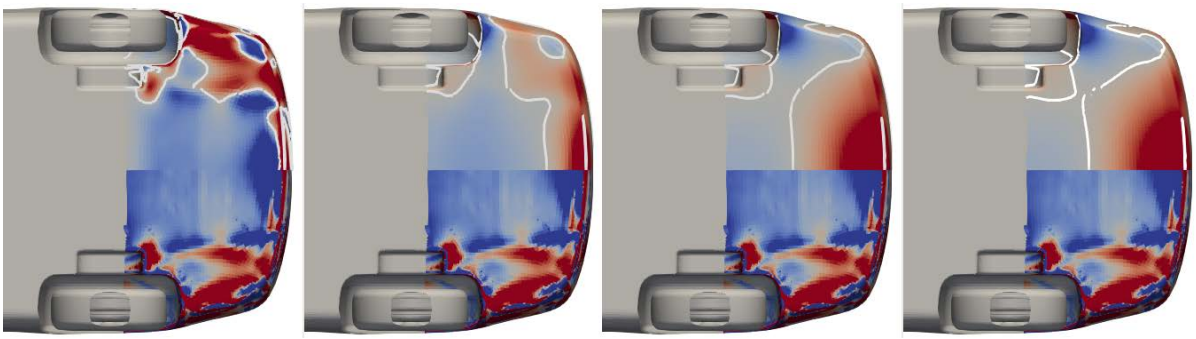
(a) Regularization based on the adjoint Laplace-Beltrami and different r_{LB}



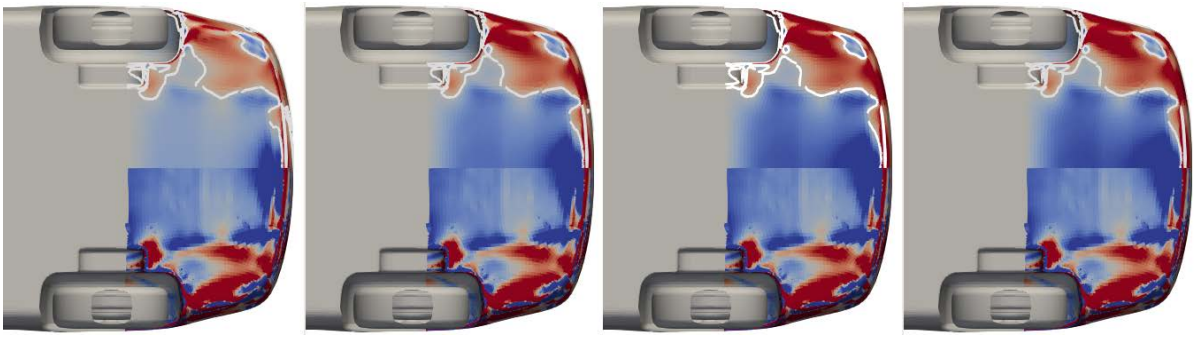
(b) Regularization based on the adjoint p-Laplacian and different r_{pL}

Figure 10: DrivAer: Non-regularized (starboard side) and regularised (port side) sensitivity derivatives of the baseline geometry, projected on the unit normal vector; the regularization involves the adjoint LB and adjoint pL PDEs, for various r_{LB} and r_{pL} values ($0.1m$ (top-left), $0.5m$ (top-right), $2m$ (bottom-left), $10m$ (bottom-right) in each set of figures); sensitivity values computed using p-Laplacian have been multiplied with their corresponding r_{pL}. Blue colour (negative normal derivatives) indicates areas that should be pushed inwards to reduce drag while red areas (positive normal derivatives) should be pulled outwards. Zero sensitivity isolines are depicted in white.

two optimizations are shown in fig. 12 while fig. 13 depicts the cumulative normal displacement from the baseline geometry. Finally, the comparison of the cumulative normal displacement and the projection of the design variables field to the unit normal vector of the baseline geometry is



(a) Regularization based on the adjoint Laplace-Beltrami and different r_{LB}



(b) Regularization based on the adjoint p-Laplacian and different r_{pL}

Figure 11: DrivAer optimization: Effect of the regularization radius on the sensitivity derivatives. Notation and regularization radii as in fig. 10 (from left to right, $0.1m, 0.5m, 2m$ and $10m$).

given in fig. 14. The effect of the regularization PDEs on the displacement field can mostly be spotted on the bottom and back sides of the optimized geometries.

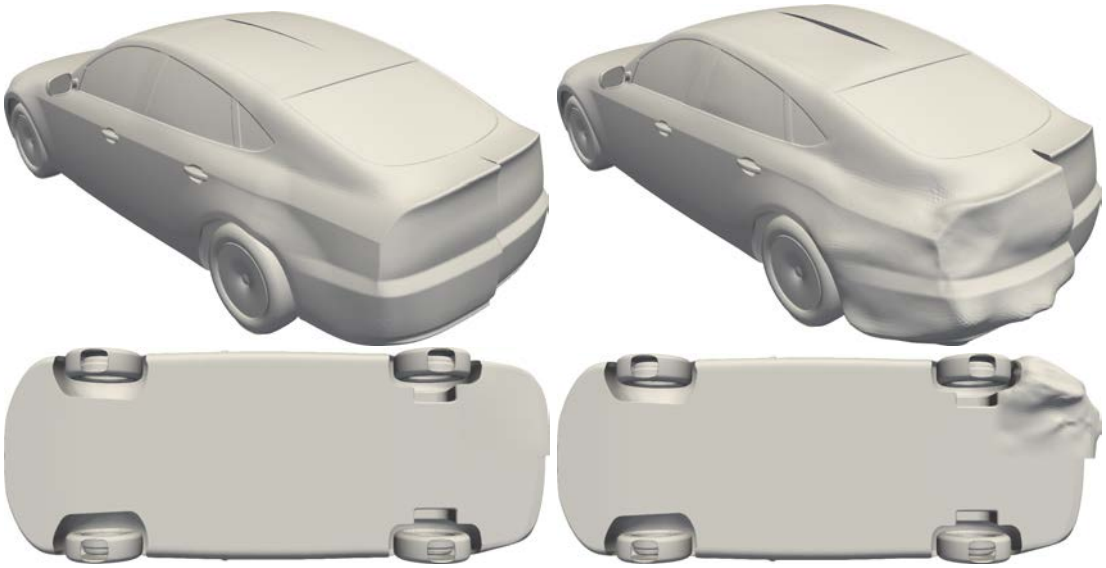


Figure 12: DrivAer: Comparison of the optimized geometries obtained with LB (left column) and pL (right column) with the baseline one. In all figures, optimized geometries are shown on the port side and the baseline on the starboard side.

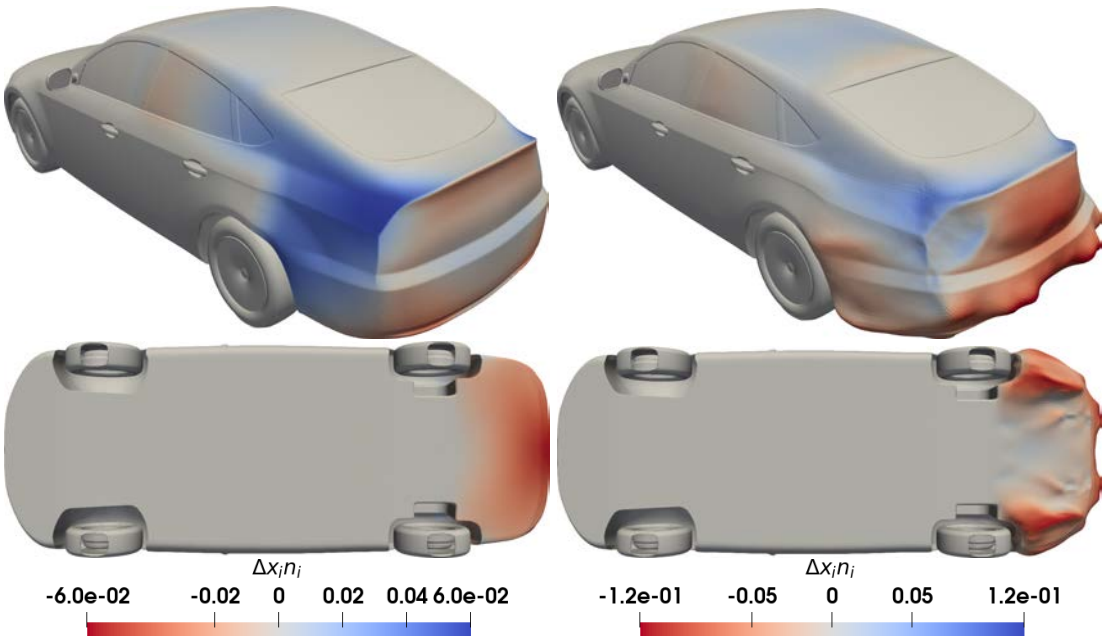


Figure 13: DrivAer: Cumulative normal displacement of the LB-based (left) and pL-based (right) optimized geometries from the baseline one. Blue/red areas have been displaced inwards/outwards.

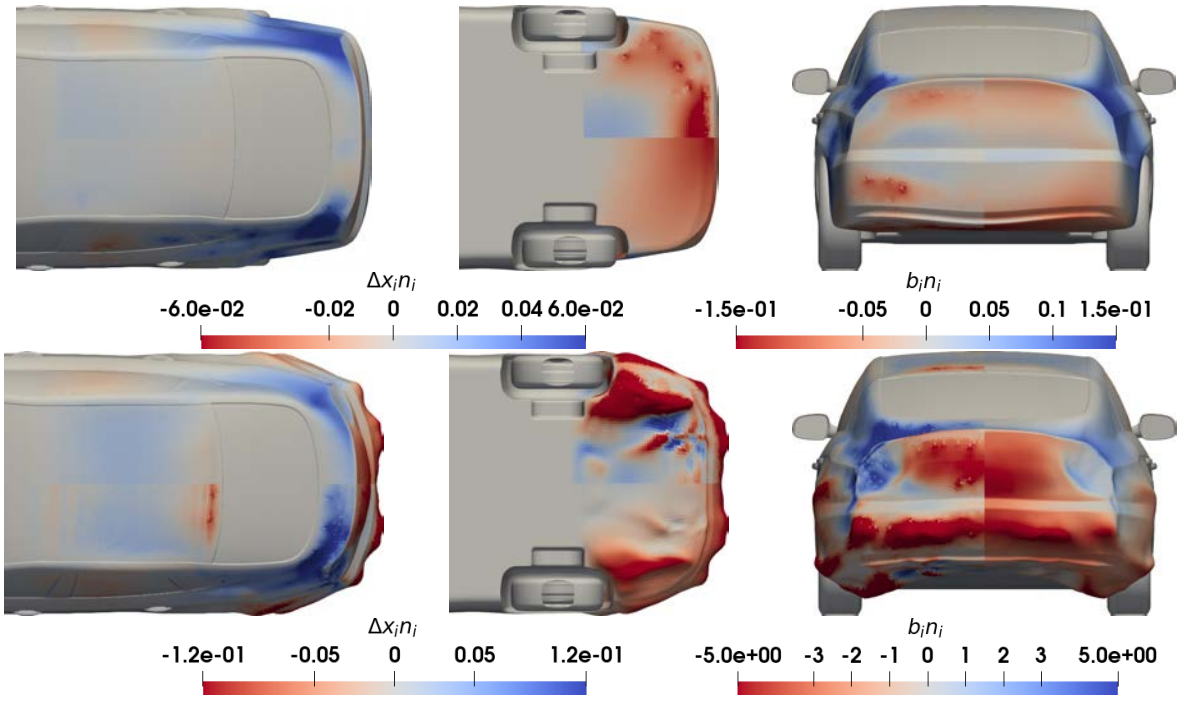


Figure 14: DrivAer: Cumulative normal displacement (starboard side) and projection of the design variables field to the unit normal vector of the baseline geometry ($b_i^f n_i^f$, port side) of the LB-based (top) and pL-based (bottom) optimized geometries. Blue/red areas have been displaced inwards/outwards.

5 SUMMARY - CONCLUSIONS

A parameterization-free shape optimization framework was developed within *adjointOptimisationFoam* through a chain of tools that link a field of design variables, defined as one vector per boundary face, to the boundary displacement from the baseline geometry, using a regularization phase and a proximity smoothing. Regularization was performed using either the Laplace-Beltrami or the p-Laplacian PDEs. Defining such a chain and differentiating it introduces the adjoint to the regularization PDEs and leads to the computation of accurate sensitivity derivatives w.r.t. the design variables, which allows the utilization of any method to update them, such as L-BFGS; this is an advantage over methods that regularize the sensitivity derivatives ad hoc. This framework was applied to the optimization of a $2D$ S-shaped duct and the DrivAer car model. In both cases, regularization based on the Laplace-Beltrami PDE proved capable of removing small wavelength features from the sensitivity derivatives and the optimized shapes while the p-Laplacian PDE, with $p = 2$, allowed such features to appear in the optimized geometry, even when using a high regularization radius. Despite the p-Laplacian regularization leading to an irregular optimized geometry in the case of the DrivAer model, the fact that its performance in terms of drag reduction was significantly better than that obtained with the Laplace-Beltrami approach ($\sim 15\%$ reduction compared to $\sim 5\%$) indicates that such small wavelength features can indeed be desirable from the aerodynamics point of view.

REFERENCES

[1] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *Computer Journal*, 7:149–154, 1964.

[2] A. Heft, T. Indinger, and N. Adams. Experimental and numerical investigation of the DrivAer model. In *ASME 2012, Symposium on Issues and Perspectives in Automotive Flows*, pages 41–51, Puerto Rico, USA, 8-12 July 2012.

[3] M. Hojjat, E. Stavropoulou, and KU. Bletzinger. The Vertex Morphing method for node-based shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 268:494 – 513, 2014.

[4] A. Jameson and J. Vassberg. Studies of alternative numerical optimization methods applied to the brachistochrone problem. *Computational Fluid Dynamics Journal*, 9(3), 2000.

[5] I.S. Kavvadias, E.M. Papoutsis-Kiachagias, and K.C. Giannakoglou. On the proper treatment of grid sensitivities in continuous adjoint methods for shape optimization. *Journal of Computational Physics*, 301:1–18, 2015.

[6] DC. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 1(3):503–528, 1989.

[7] E. Luke, E. Collins, and E. Blades. A fast mesh deformation method using explicit interpolation. *Journal of Computational Physics*, 231(2):586–601, 2012.

[8] PM. Müller, N. Kühl, M. Siebenborn, K. Deckelnick, M. Hinze, and T. Rung. A novel p-harmonic descent approach applied to fluid dynamic shape optimization. *Structural and Multidisciplinary Optimization*, 64:3489–3503, 2021.

[9] J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer, New York, 1999.

[10] E.M. Papoutsis-Kiachagias and K.C. Giannakoglou. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: industrial applications. *Archives of Computational Methods in Engineering*, 23(2):255–299, 2016.

[11] E.M. Papoutsis-Kiachagias, N. Magoulas, J. Mueller, C. Othmer, and K.C. Giannakoglou. Noise reduction in car aerodynamics using a surrogate objective function and the continuous adjoint method with wall functions. *Computers & Fluids*, 122:223–232, 2015.

[12] T.T. Robinson, C.G. Armstrong, H.S. Chua, C. Othmer, and T. Grahs. Optimizing parameterized CAD geometries using sensitivities based on adjoint functions. *Computer-Aided Design & Applications*, 9(3):253–268, 2012.

[13] S. Schmidt, C. Ilic, V. Schulz, and NR. Gauger. Three-dimensional large-scale aerodynamic shape optimization based on shape calculus. *AIAA Journal*, 51(11):2615–2627, 2013.

[14] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In *AIAA Paper 1992-0439, 30th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 6-9 January 1992.

[15] A. Stück and T. Rung. Adjoint RANS with filtered shape derivatives for hydrodynamic optimisation. *Computers & Fluids*, 47(1):22–32, 2011.

[16] Z. Tukovic and H. Jasak. Simulation of free-rising bubble with soluble surfactant using moving mesh finite volume/area method. In *6th International Conference on CFD in Oil & Gas, Metallurgical and Process Industries*, Trondheim, Norway, 10-12 August 2008.