Structural and Multidisciplinary Optimization (2023) 66:111
https://doi.org/10.1007/s00158-023-03548-2

RESEARCH PAPER

The implicit bulk-surface filtering method for node-based shape optimization and a comparison of explicit and implicit filtering techniques

Reza Najian Asl[1] · **Kai-Uwe Bletzinger**[1]

Received: 1 August 2022 / Revised: 9 March 2023 / Accepted: 10 March 2023 / Published online: 20 April 2023

Abstract
This work studies shape filtering techniques, namely the convolution-based (explicit) and the PDE-based (implicit), and introduces the implicit bulk-surface filtering method to control the boundary smoothness and preserve the internal mesh quality simultaneously in the course of bulk (solid) shape optimization. To that end, the volumetric mesh is governed by the pseudo-solid smoothing equations, which are stiffened by the mesh-Jacobian and endowed with the Robin boundary condition, which involves the Laplace-Beltrami operator on the mesh boundaries. Its superior performance from the non-simultaneous (sequential) treatment of boundary and internal meshes is demonstrated for the shape optimization of complex solid structures. Well-established explicit filters, namely Gaussian and linear, and the Helmholtz/Sobolev-based (implicit) filter are critically examined for shell optimization in terms of consistency (rigid-body-movement production), geometric characteristics, and computational cost. It is demonstrated that implicit filtering is more numerically efficient and robustly enforces fixed boundaries compared to explicit filtering. Supported by numerical experiments, a regularized Green's function is introduced as an equivalent explicit form of the Helmholtz/Sobolev filter. Furthermore, we give special attention to deriving mesh-independent filtered sensitivities for node-based shape optimization with non-uniform meshes. It is shown that mesh-independent filtering can be achieved by scaling discrete sensitivities with the inverse of the mesh mass matrix.

Keywords Implicit bulk-surface filtering method · Node-based shape optimization · Explicit and implicit filters · Consistency and mesh-independency

1 Introduction

Filtering is widely recognized as a successful technique in discrete topology and shape optimization. For a given filter length scale, it should regularize the optimization problem independent of the discretization of the underlying geometry. Together with adjoint-based sensitivity analysis to determine the discrete gradients, filtering or regularization has become a very successful procedure for large-scale optimization problems in the industry. Techniques for filtering are divided into two main categories whose similarities and differences are going to be discussed in this paper. In the first approach, smoothing is applied implicitly by solving elliptic PDEs whose inverse operator is a local smoother, the so-called implicit filtering. In the second category, the raw field is smoothed by its convolution with a kernel function, the so-called explicit filtering. Both techniques are conventionally used to smooth either design variables or design gradients. The former is more consistent with the optimization problem formulation, whereas the latter may confuse the optimizer and disturb the convergence due to discrepancies between the filtered and true (raw) gradients.

Explicit filtering was introduced by Sigmund (1994) in order to eliminate two well-known problems in structural topology optimization, namely the checkerboard problem and the mesh-dependency problem. Since then, the method has become a standard tool in structural design and optimization, mainly due to its simple formulation and robustness. In shape optimization, the explicit filtering has been applied to structural (Bletzinger et al. 2005, 2010; Le et al. 2011; Firl and Bletzinger 2012) as well as fluid and aerodynamic

Responsible Editor: Jianbin Du

✉ Reza Najian Asl
reza.najian-asl@tum.de

[1] Lehrstuhl für Statik, Technische Universität München, Arcisstr. 21, 80333 Munich, Germany

(Stück and Rung 2011; Hojjat et al. 2014; Kröger and Rung 2015; Najian Asl et al. 2017; Ghantasala et al. 2021b; Antonau et al. 2022) applications. Furthermore, Bletzinger (2014) has established a perfect analogy between the explicit filtering and parameterization techniques used for shape optimization, especially CAD-based techniques.

The traction method (Azegami and Wu 1996) and the Sobolev-gradient smoothing (Jameson and Vassberg 2000) were pioneering works on PDE-based or implicit filtering. The main idea is to project the gradients into a smoother design space, the Hilbert or Sobolev space, by letting the gradients be the solution of an elliptic equation. Schmidt et al. (2008) and Eppler et al. (2009) have demonstrated mathematically that the Sobolev smoothing can be interpreted as a reduced shape Hessian operator, which turns the steepest descent into a quasi-Newton step. Following this interpretation, Dick et al. (2022) have used the Sobolev smoothing as a preconditioner for discrete shape sensitivities to accelerate the convergence in CAD-based shape optimization, even though the parameterization is smooth itself. Mohammadi and Pironneau (2009) also observed that the conditioning of the optimization problem is more favorable with the Sobolev-based filtering than with the CAD-based filtering. Recently, implicit filtering has come to the attention of the topology optimization community after the early works by Lazarov and Sigmund (2011) and Kawamoto et al. (2011). The primary motivation behind this is that it has better performance profiles than classical or explicit filtering.

This work studies the explicit and implicit filters in the context of node-based shape optimization. They are discussed qualitatively and quantitatively in terms of the consistency between shape gradients and shape updates, interpolation of the domain edges (fixed boundaries), and the sensitivity of filtered fields to the discretization of the underlying geometry, i.e., mesh dependency. Furthermore, we present the bulk-surface filtering method designed to control the boundary smoothness and simultaneously prevent internal mesh distortion. This is similar to the enhanced traction method (Azegami and Takeuchi 2006), which computes shape updates by solving a pseudo-elastic linear problem that is stiffened by springs on the domain boundary and is loaded by discrete shape gradients. In this work, to maintain mesh quality and reduce the frequency of remeshing during the shape optimization, the so-called Jacobian-based stiffening (Tezduyar et al. 1992; Stein et al. 2003; Tonon et al. 2021) is applied to the pseudo-elastic solid model. Moreover, to ensure smooth boundary geometry shapes, the model is endowed with the Robin boundary condition, which involves the Laplace-Beltrami operator on the boundary mesh. Furthermore, here we use the filter in a consistent manner; hence it is applied to discrete shape control points rather than the sensitivities. Consistent filtering was first introduced for topology optimization (Bourdin 2001; Bruns and Tortorelli 2001), and since then, it has been widely used. It should be mentioned that, unlike topology optimization, in shape optimization, the control field is unknown a priori, and it can be calculated from the discrete shape by inverse filtering (deconvolution). Doing so introduces a design space in parallel to the geometry space where shape updates are applied. Mathematically speaking, the actual optimization problem is defined in the control space, and control sensitivities should be derived consistently by the chain rule. Last but not least, since discrete sensitivities are consistent nodal values that show size effects in the gradients, the inverse of the mesh mass matrix is used as a preconditioner for the discrete control sensitivities to avoid the mesh dependency issue.

This paper is structured as follows: In Sect. 2, the explicit shape filtering is revisited, and its discrete form is derived. Therein, consistency and mesh-dependency issues are elaborated, and the mesh mass matrix preconditioner is introduced. In Sect. 3, implicit shape filtering is presented for surface geometries and bulk domains using the previously described bulk-surface formulation. Finally, Sect. 4 critically compares the explicit and implicit filtering techniques regarding geometric characteristics and the computational cost. Therein, the performance of the developed bulk-surface filter is studied for the shape optimization of complex solid structures, and comparisons are made against the non-simultaneous (sequential) treatment of boundary and internal meshes.

2 Explicit shape filtering

This technique generates the three-dimensional geometry at point $\boldsymbol{x}_0 = (x_0^1, x_0^2, x_0^3)$ of the design surface Γ from the control field $\boldsymbol{s} = (s^1, s^2, s^3)$ via a convolutional filtering operation:

$$\boldsymbol{x}_0 = \int_{\Gamma} F(\boldsymbol{x}, \boldsymbol{x}_0)\, \boldsymbol{s}(\boldsymbol{x})\, d\Gamma = \int_{\Sigma} F(\boldsymbol{x}, \boldsymbol{x}_0)\, \boldsymbol{s}\, d\Sigma \tag{1}$$

where F is the filter (kernel) function which controls the properties of filtering and subsequently the produced shapes, Σ is the portion of Γ which lies within the filter's support span. In free-form shape optimization, unlike topology and CAD-based shape optimization, the control field is unknown a priori, and it can be calculated by inverse filtering (deconvolution). Gaussian and linear hat functions are the most commonly used kernels in shape and topology optimization and they are defined respectively as

$$F^G(\boldsymbol{x}, \boldsymbol{x}_0) = \frac{1}{r_\Gamma^E \sqrt{2\pi}}\, e^{\frac{-1}{2}(\|\boldsymbol{x}-\boldsymbol{x}_0\|/r_\Gamma^E)^2} \tag{2a}$$

$$F^L(\boldsymbol{x},\boldsymbol{x}_0) = \max\left(0, \frac{r_\Gamma^E - \|\boldsymbol{x}-\boldsymbol{x}_0\|}{r_\Gamma^E}\right) \tag{2b}$$

where the surface filter radius of explicit filtering is denoted by r_Γ^E. Let $\boldsymbol{x} = (x^1, x^2, x^3)$ represent the coordinates of a point on the surface. Then, $\|\boldsymbol{x}-\boldsymbol{x}_0\|$ denotes the Euclidean distance from this point to the center of the filter, $\boldsymbol{x}_0$. (Fig. 1).

In shape optimization, the design surface is often open and limited by adjacent non-design areas, which must not be modified. In other words, the edges of the design surface are fixed or must be interpolated during the optimization. As a matter of fact, edge interpolation is a Dirichlet boundary condition to the optimization problem and must be satisfied strongly and point-wise. In the explicit shape filtering, damping (Kröger and Rung 2015; Baumgartner 2020) is used to suppress modification of the fixed boundaries and smoothly transit between design and not-design areas. The implementation of damping is straightforward, and one only needs to multiply the filter operator with a damping function which is computed based on the distance to the non-design domain. Then Eq.1 is updated as

$$\boldsymbol{x}_0 = D(\boldsymbol{x}_0,\boldsymbol{x}_{\text{cpp}}) \int_\Sigma F(\boldsymbol{x},\boldsymbol{x}_0)\, s\, d\Sigma \tag{3a}$$

$$D(\boldsymbol{x}_0,\boldsymbol{x}_{\text{cpp}}) = 1 - F(\boldsymbol{x}_0,\boldsymbol{x}_{\text{cpp}})/F_{max} \tag{3b}$$

where D is the damping function, $\boldsymbol{x}_{\text{cpp}}$ is determined based on the so-called closest point projection (CPP) of $\boldsymbol{x}_0$ onto the design surface boundary $\partial\Gamma$, and F_{max} is defined as the maximum value of the filter function over its evaluated domain. It this way, it is ensured that non-design domains, such as the edges of the design surface, are interpolated by the control variables, allowing for the generation of smooth shapes without modifying these non-design domains.

For numerical analysis, the shape governing equation of the design surface should be discretized. Although one can use different resolutions to discretize the left- and right-hand sides of Eq. 1, in this work the same grid is used for the sake of simplicity. Applications of non-matching grids can be found in Najian Asl (2019). We use piecewise linear finite element functions to approximate the geometry and data fields, i.e., isoparametric finite elements. Therefore, the actual geometry and its control field within each element are approximated as $x^i \approx \boldsymbol{N}^i\boldsymbol{x}^i, \quad s^i \approx \boldsymbol{N}^i\boldsymbol{s}^i; \quad i \in \{1,2,3\}$, where $\boldsymbol{N}^i$ is the vector of element shape functions in the ith Cartesian direction, $\boldsymbol{x}^i$ and $\boldsymbol{s}^i$ are the ith components of nodal coordinate vector and nodal control point vector, respectively. Then, the geometry at node i reads

$$\boldsymbol{x}_{\Gamma,i} = D(\boldsymbol{x}_{\Gamma,i}) \sum_{k\in\mathbb{E}_{e,i}} \int_{\Gamma^e} F(\boldsymbol{x}_{\Gamma,i},\boldsymbol{x}_{\Gamma,k})\, \boldsymbol{N}_\Gamma^T \boldsymbol{N}_\Gamma\, \boldsymbol{s}_{\Gamma,k}\, d\Gamma \tag{4}$$

where $\mathbb{E}_{e,i}$ is the set of elements in the filter support domain of node i, and subscript Γ indicates that the quantity belongs to the surface. Note that the filter function at node i should be ideally evaluated at the Gauss points of the associated elements $\mathbb{E}_{e,i}$; however since the neighbor search operation is cost prohibitive, it is evaluated at the mesh points, i.e., it is discretized. In the vector-matrix format, the explicit shape filtering reads

$$\begin{aligned} \boldsymbol{x}_\Gamma &= \boldsymbol{D}_\Gamma \cdot \boldsymbol{W}_\Gamma \cdot \boldsymbol{M}_\Gamma \cdot \boldsymbol{s}_\Gamma = \boldsymbol{A}_\Gamma^E \cdot \boldsymbol{s}_\Gamma \\ \boldsymbol{M}_\Gamma &= \sum \int_{\Gamma^e} \boldsymbol{N}_\Gamma^T \boldsymbol{N}_\Gamma d\Gamma \end{aligned} \tag{5}$$

$\boldsymbol{x}_\Gamma \in \mathbb{R}^{3n_\Gamma\times 1}$ is the nodal coordinates vector of the design boundary mesh, $\boldsymbol{A}_\Gamma^E \in \mathbb{R}^{3n_\Gamma\times 3n_\Gamma}$ is the explicit filtering matrix composed of the surface mesh mass matrix $\boldsymbol{M}_\Gamma$, the weighting matrix $\boldsymbol{W}_\Gamma$ whose entries are $\boldsymbol{W}_\Gamma(i,j) = F(\boldsymbol{x}_{\Gamma,i},\boldsymbol{x}_{\Gamma,j}) \cdot \boldsymbol{I}_{3\times3}$ and the diagonal damping matrix $\boldsymbol{D}_\Gamma(i,i) = (1 - F(\boldsymbol{x}_{\Gamma,i},\boldsymbol{x}_{\text{cpp}})/F_{\max}) \cdot \boldsymbol{I}_{3\times3}$. A drawback of such convolutional filtering is that the *smoothing* effect becomes mesh dependent, especially on unstructured meshes and finite domains with open boundaries. It is clear that the discrete surface $\boldsymbol{x}_\Gamma$ is constructed based on a weighted sum of the control values at the mesh points, where the weighting is purely a function of the distances to mesh points. The standard remedy found in literature is to rescale the filter function so that it has a unit volume everywhere in the domain, i.e.,

$$\int_\Gamma F(\boldsymbol{x},\boldsymbol{x}_0)\, d\Gamma = 1.0, \quad \forall \boldsymbol{x}_0 \quad \text{on} \quad \Gamma \tag{6}$$

This can be achieved easily by dividing the filter function by the inverse of its volume, i.e., $F(\boldsymbol{x},\boldsymbol{x}_0)/ = \int_\Gamma F(\boldsymbol{x},\boldsymbol{x}_0)\, d\Gamma$. This scaling has two effects. On one hand, it results in the so-called *consistency* property for the filtering, meaning that a constant variation in the control field $\delta\boldsymbol{s} = [1,1,1]$ results in a constant update in the geometry $\delta\boldsymbol{x} = [1,1,1]$. On the other hand, it destroys the self-adjoint property of the filter

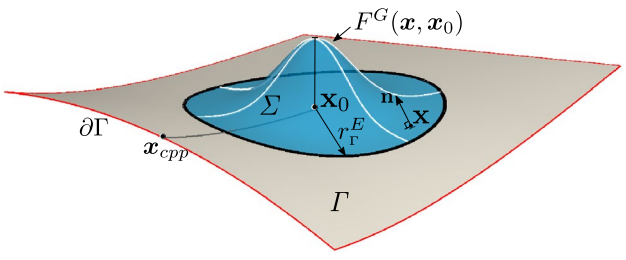


Fig. 1 Notional schematic of the design surface (Γ), the Gaussian filter function (F), the integration area (Σ) and the closest point projection of $\boldsymbol{x}_0$ ($\boldsymbol{x}_{\text{cpp}}$) onto the design surface boundary ($\partial\Gamma$)

function and, accordingly, the symmetry property of the weighting matrix. The consistent explicit filtering matrix reads

$$\boldsymbol{A}_{\Gamma}^{E} = \boldsymbol{D}_{\Gamma} \cdot \boldsymbol{V}_{\Gamma}^{-1} \cdot \boldsymbol{W}_{\Gamma} \cdot \boldsymbol{M}_{\Gamma} \tag{7}$$

where $\boldsymbol{V}_{\Gamma} \in \mathbb{R}^{3n_{\Gamma} \times 3n_{\Gamma}}$ is a diagonal matrix and its entries are basically the volume under nodal filter functions, i.e., $\boldsymbol{V}_{\Gamma}(i,i) = \sum_{k \in \mathbb{E}_{e,i}} \int_{\Gamma^e} F(\boldsymbol{x}_{\Gamma,i})\, N_{\Gamma}\, d\Gamma \cdot \boldsymbol{I}_{3\times 3}$. Replacing $\boldsymbol{D}_{\Gamma}$ by the identity matrix, it is noted that the *consistency* property is reflected as unit row sums in the filtering matrix. Also, note that the matrix is a large and dense matrix; however, it can be applied in a matrix-free mode by computing the weights in advance, storing them inside each node and cycling through the nodes. At first glance, the lack of the self-adjoint property does not seem relevant; however, a closer look at the response sensitivity analysis and shape update rule that will follow reveals some inconsistency issues.

2.1 Sensitivity analysis and update rule

In this work, filtering is applied to the discrete shape rather than the discrete shape sensitivities, meaning that parallel to the geometry space x where shape sensitivities are evaluated, the design space s lives where the optimization problem is defined. Applying the chain rule of differentiation, control gradients of the response J read

$$\begin{aligned} \frac{dJ}{d\boldsymbol{s}_{\Gamma}} &= \frac{dJ}{d\boldsymbol{x}_{\Gamma}} \cdot \frac{d\boldsymbol{x}_{\Gamma}}{d\boldsymbol{s}_{\Gamma}} = (\boldsymbol{A}_{\Gamma}^{E})^{T} \cdot \frac{dJ}{d\boldsymbol{x}_{\Gamma}} \\ &= \boldsymbol{M}_{\Gamma} \cdot \boldsymbol{W}_{\Gamma} \cdot \boldsymbol{V}_{\Gamma}^{-1} \cdot \boldsymbol{D}_{\Gamma} \cdot \frac{dJ}{d\boldsymbol{x}_{\Gamma}} \end{aligned} \tag{8}$$

where $dJ/d\boldsymbol{s}_{\Gamma}$ and $dJ/d\boldsymbol{x}_{\Gamma}$ are the discrete control and shape sensitivities of the response, respectively. It is important to note that the derivation above makes use of the fact that the submatrices of the filter matrix exhibit symmetry, even though the filter matrix ($\boldsymbol{A}_{\Gamma}^{E}$) itself may not be symmetric. This implies that when the filter matrix is transposed, it only reverses the order of the submatrices.

In fluid and structural optimization, response functions are typically smooth differentiable integral functional whose continuous variation reads

$$\delta J(\boldsymbol{x}) = \int_{\Gamma} \delta j(\boldsymbol{x})\, d\Gamma \tag{9}$$

where the spatial variation of the integral is neglected. Then, consistent or discrete gradients can be derived by basis functions of finite elements (Oden and Reddy 1972) as

$$\frac{dJ}{d\boldsymbol{x}_{\Gamma}} = \boldsymbol{M}_{\Gamma} \cdot \frac{dj}{d\boldsymbol{x}_{\Gamma}} \tag{10}$$

$dj/d\boldsymbol{x}_{\Gamma}$ is the continuous gradient field sampled at mesh points and associated with the discrete/consistent field $dJ/d\boldsymbol{x}_{\Gamma}$ via the mesh mass matrix $\boldsymbol{M}_{\Gamma}$. Since the control field $\boldsymbol{s}_{\Gamma}$ is also a continuous field sampled at mesh points, it must be updated with a field of the same type in the direction of the steepest descent. Therefore, inspired by Eq. 10 it seems reasonable to develop the following control update rule:

$$\begin{aligned} \Delta\boldsymbol{s}_{\Gamma} &= -\alpha \cdot \frac{dj}{d\boldsymbol{s}_{\Gamma}} = -\alpha \cdot \boldsymbol{M}_{\Gamma}^{-1} \cdot \frac{dJ}{d\boldsymbol{s}_{\Gamma}} \\ &= -\alpha \cdot \boldsymbol{M}_{\Gamma}^{-1} \cdot \boldsymbol{M}_{\Gamma} \cdot \boldsymbol{W}_{\Gamma} \cdot \boldsymbol{V}_{\Gamma}^{-1} \cdot \boldsymbol{D}_{\Gamma} \cdot \frac{dJ}{d\boldsymbol{x}_{\Gamma}} \\ &= -\alpha \cdot \boldsymbol{W}_{\Gamma} \cdot \boldsymbol{V}_{\Gamma}^{-1} \cdot \boldsymbol{D}_{\Gamma} \cdot \frac{dJ}{d\boldsymbol{x}_{\Gamma}} \\ &= -\alpha \cdot \boldsymbol{W}_{\Gamma} \cdot \boldsymbol{V}_{\Gamma}^{-1} \cdot \boldsymbol{D}_{\Gamma} \cdot \boldsymbol{M}_{\Gamma} \cdot \frac{dj}{d\boldsymbol{x}_{\Gamma}} \end{aligned} \tag{11}$$

where α is a constant step size, $dj/d\boldsymbol{s}_{\Gamma}$ is the scaled/continuous control sensitivities which are linked to the consistent ones $dJ/d\boldsymbol{s}_{\Gamma}$ via the inverse mass matrix. One can interpret the proposed update rule as a quasi-Newton step with the diagonal approximation of the Hessian matrix by the scaled mass matrix. A careful look at Eqs. 8,11 shows that the desirable consistency property (unit row sum) in the explicit shape filtering (Eqs. 5,7) does not hold for the sensitivity filtering. This means that a uniform continuous gradient field, e.g., $dj/d\boldsymbol{x}_{\Gamma} = [1]_{3n_{\Gamma}\times 1}$, does not result in a uniform gradient field in the control space and subsequently a uniform geometry update, i.e.,

$$\begin{aligned} \Delta\boldsymbol{x}_{\Gamma} &= \boldsymbol{A}_{\Gamma}^{E} \cdot \Delta\boldsymbol{s}_{\Gamma} \\ &= -\alpha \cdot \boldsymbol{A}_{\Gamma}^{E} \cdot \boldsymbol{W}_{\Gamma} \cdot \boldsymbol{V}_{\Gamma}^{-1} \cdot \boldsymbol{M}_{\Gamma} \cdot [1]_{3n_{\Gamma}\times 1} \\ &\neq -\alpha \cdot [1]_{3n_{\Gamma}\times 1} \end{aligned} \tag{12}$$

where for the sake of simplicity, no damping or design boundary interpolation is considered, i.e., $\boldsymbol{D}_{\Gamma}(i,i) = \boldsymbol{I}_{3\times 3}$. The above holds in particular for open design surfaces, independent of the mesh type.

2.2 Consistency check

In gradient-based shape optimization, the *consistency* property means rigid-body-movement production for a given translational (uniform) sensitivity field, independent of the mesh discretization. Kröger and Rung (2015) have discussed the consistency aspects of explicit filtering in great detail. They strictly require the filter to compute constant and linear filtered fields, respectively, for constant and linear distributions of the continuous sensitivity field. In shape and topology optimization, the filter matrix can also be only applied to the design sensitivities rather than the design variables themselves, the so-called sensitivity filtering technique. However, in shape and density filtering, the matrix is used

twice, once in the design update rule (Eq. 12) and once in the sensitivity analysis using a transpose operation (Eq. 8).

We perform consistency tests with a uniform continuous sensitivity field of $dj/d\boldsymbol{x}_\Gamma = [(0,0,1)]_{n_\Gamma \times 1}$ over a perforated plate which is discretized non-uniformly, Fig. 2. It should be emphasized that although the continuous shape sensitivity field $dj/d\boldsymbol{x}_\Gamma$ is uniformly distributed, the consistent/discrete nodal sensitivities $dJ/d\boldsymbol{x}_\Gamma$ are not necessarily uniform, and indeed they are scaled by the mesh mass matrix, see Eq. 10. Control sensitivities obtained after explicit filtering with a linear hat function are demonstrated in Fig. 3. As could be expected, the consistent/discrete control sensitivities are non-smooth and, indeed, mesh-dependent. In contrast, the ones scaled by the inverse mass matrix are certainly smooth and mesh-independent. As a matter of fact, any consistent discrete field is mesh-dependent and noisy. Therefore, scaling the consistent sensitivities $dJ/d\boldsymbol{s}_\Gamma$ with the inverse mass matrix is absolutely necessary to avoid mesh-dependency and introducing irrelevant noises into the design. We also notice that the scaled sensitivities are non-uniform close to the boundary edges, indicating a lack of consistency in mapping/filtering sensitivity with the explicit approach.

3 Implicit shape filtering

Filtering can also be performed implicitly by elliptic PDEs whose inverse operator is a local smoother. A well-established and commonly used implicit filter is the so-called Helmholtz/Laplace-Beltrami operator (Lazarov and Sigmund 2011; Kawamoto et al. 2011), which is also labeled as the Sobolev smoothing (Jameson and Vassberg 2000; Schmidt et al. 2008; Mohammadi and Pironneau 2009; Dick et al. 2021); it reads

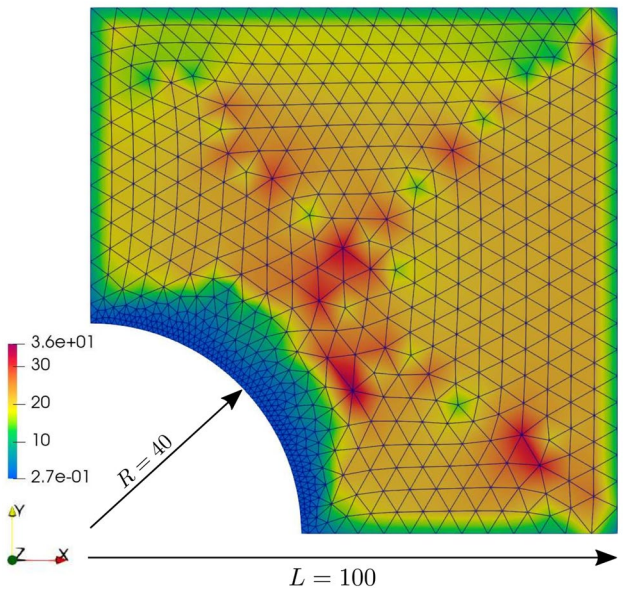
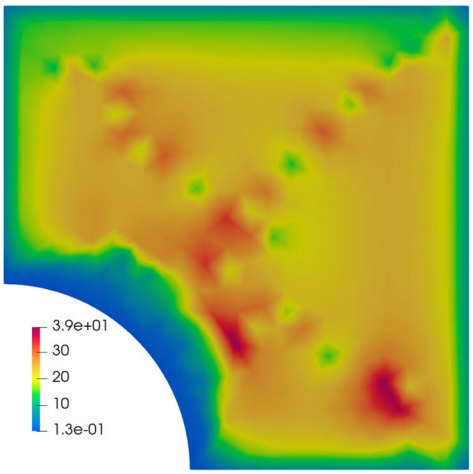
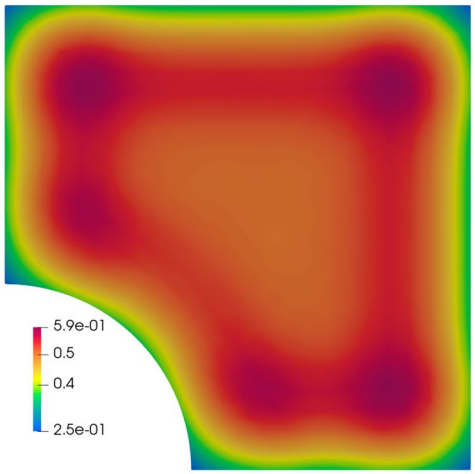


Fig. 2 A perforated plate meshed non-uniformly. Consistent/discrete nodal sensitivities $dJ/d\boldsymbol{x}_\Gamma$ associated with the uniform continuous sensitivity field $dj/d\boldsymbol{x}_\Gamma = [(0,0,1)]_{n_\Gamma \times 1}$



(a) Consistent/discrete control sensitivities $dJ/d\boldsymbol{s}_\Gamma$.



(b) Scaled control sensitivities $dj/d\boldsymbol{s}_\Gamma = \boldsymbol{M}_\Gamma^{-1} \cdot dJ/d\boldsymbol{s}_\Gamma$.

Fig. 3 Control/filtered sensitivities computed with explicit filtering for the consistent/discrete sensitivities in Fig. 2

$$
\begin{aligned}
(I - \epsilon \cdot \Delta)\, \widetilde{d} &= d \\
\boldsymbol{n} \cdot \nabla \widetilde{d} &= 0, \quad \text{along external boundaries.} \\
\widetilde{d} &= d, \quad \text{along fixed/Dirichlet boundaries.}
\end{aligned}
\tag{13}
$$

where is $\widetilde{d}$ is the filtered field, d is the raw field and ϵ is a scalar penalizing high spatial variations in $\widetilde{d}$. As mentioned previously, the so-called traction method (Azegami 1994; Azegami and Takeuchi 2006; Riehl et al. 2014) is also an implicit filtering scheme. This method determines the filtered field by solving a pseudo-elastic problem with

the loading of the raw field. In this work, we develop surface and bulk-surface implicit smoothers based on the Helmholtz operator in Eq. 13. They are respectively applied to the shell and solid geometries to ensure C^1 shape variations during the optimization iterations. It must be mentioned that the bulk-surface filtering treats boundary and internal domains simultaneously. Therefore, there is no need for the inclusion of any mesh motion technique in shape optimization for volumetric domains.

3.1 Surface filtering

The Sobolev/Helmholtz operator in Eq. 13 can be cast on the design surface as

$$-(r_{\Gamma}^{H})^2 \, \nabla_{\Gamma} \cdot \nabla_{\Gamma} \, x^i + x^i = s^i, \quad i \in \{1,2,3\}; \quad \text{on} \quad \Gamma \tag{14}$$

$\nabla_{\Gamma} \cdot \nabla_{\Gamma}$ is the Laplace-Beltrami operator. ∇_{Γ} is the tangential gradient operator and can be calculated for a function w defined on some open neighborhood of Γ with a unit normal $\boldsymbol{n}$ as $\nabla_{\Gamma} w = \nabla w - (\boldsymbol{n}.\nabla w)\, \boldsymbol{n}$, r_{Γ}^{H} is the filter radius/length parameter of Helmholtz-based filtering which plays a similar role as r_{Γ}^{E} in the explicit filtering and as it approaches zero, $r_{\Gamma}^{H} \to 0$, smoothing effect disappears, i.e., $\boldsymbol{x} = \boldsymbol{s}$. The smoothing property of Eq. 14 in every spatial direction can be clearly explained by the minimization of the associated potential Π as

$$\Pi\left(x^i\right) = \frac{1}{2} \, (r_{\Gamma}^{H})^2 \int_{\Gamma} |\nabla x^i|^2 \, d\Gamma + \frac{1}{2} \int_{\Gamma} \left(x^i - s^i\right)^2 \, d\Gamma \tag{15}$$

where the first and second integrals respectively measure the noisiness (spatial variation) of geometry and the difference between control and actual geometries. Here, r_{Γ}^{H} can be interpreted as the weight/importance of smoothness in the minimization of the multi-objective function Π. It should be noted that Eq. 14 is an isotropic in-surface filtering that does not depend on the direction.

3.2 The implicit bulk-surface filtering method

In node-based shape optimization with volumetric domains, we need to treat not only the surface/boundary nodes but also the internal nodes to avoid losing the mesh quality for numerical computations. In the literature so far, surface and internal nodes have mainly been treated separately in a master-follower manner, meaning that surface nodes are the master and internal nodes follow them using mesh motion techniques. The main issue with this approach is that not every smooth shape mode of the surface mesh can be tolerated by the volume mesh in terms of quality measures. Therefore, remeshing may be needed in the course of the optimization process. To enforce the surface smoothness and maintain the volume mesh quality in one shot, we use an elliptic partial differential equation, posed on the solid domain with a smooth boundary, endowed with a generalized Robin boundary condition which involves the Laplace-Beltrami operator on the boundary surface as

$$-(r_{\Omega}^{H})^2 \, \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{x} = \boldsymbol{s}, \quad \text{in} \quad \Omega \tag{16a}$$

$$-(r_{\Gamma}^{H})^2 \, \nabla_{\Gamma} \cdot \nabla_{\Gamma} \boldsymbol{x} = \boldsymbol{0}, \quad \text{on} \quad \Gamma \tag{16b}$$

where r_{Ω}^{H} is the Helmholtz bulk filter radius, ∇ is the conventional spatial gradient operator, $\boldsymbol{\sigma}$ is a second order tensor similar to the Cauchy stress tensor in continuum mechanics and defined as

$$\boldsymbol{\sigma} = \lambda \, \mathrm{tr}(\boldsymbol{\epsilon}(\boldsymbol{x}))\, \boldsymbol{I} + 2 \, \mu \, \boldsymbol{\epsilon}(\boldsymbol{x}) \tag{17a}$$

$$\boldsymbol{\epsilon}(\boldsymbol{x}) = \frac{1}{2}\left(\nabla \boldsymbol{x} + (\nabla \boldsymbol{x})^T\right) \tag{17b}$$

where tr() is the trace operator, λ and μ are the Lamé constants, $\boldsymbol{I}$ is the identity tensor, $\boldsymbol{\epsilon}$ is the strain tensor acting on geometry. Once again, one should note that r_{Ω}^{H} and r_{Γ}^{H} are devised to control the surface and bulk filtering properties of the presented shape parameterization. To reveal these properties and their relation, we formulate Eqs. 16 as the minimization of associated potential

$$\begin{aligned} \Pi(\boldsymbol{x}) = & \frac{1}{2} \, (r_{\Omega}^{H})^2 \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\epsilon} \, d\Omega \\ & + \frac{1}{2} \, (r_{\Gamma}^{H})^2 \int_{\Gamma} \nabla_{\Gamma} \boldsymbol{x}^T . \nabla_{\Gamma} \boldsymbol{x} \, d\Gamma \\ & + \frac{1}{2} \int_{\Omega} (\boldsymbol{x} - \boldsymbol{s})^T . (\boldsymbol{x} - \boldsymbol{s}) \, d\Omega \end{aligned} \tag{18}$$

Here, we minimize a weighted sum of the total strain energy of the geometry occupying Ω, the integrated spatial variation of the boundary geometry (noisiness) and the totaldeviation between the control and actual geometries. The corresponding weights are respectively $(r_{\Omega}^{H})^2$, $(r_{\Gamma}^{H})^2$ and 1, assuming uniform distribution over the integration domains. It should be noted that this is a multi-criteria optimization that may be posed to well-known challenges such as the conflict between criteria and dominance over each other. Since smoothness of the design boundary Γ (second term in Eq. 18) is of greater importance and to prevent it from being dominated by the volumetric strain (first term in Eq. 18), the Helmholtz bulk filter radius is calculated as follows

$$(r_{\Omega}^{H})^2 = \beta \, \frac{(r_{\Gamma}^{H})^2 \int_{\Gamma} \nabla_{\Gamma} \boldsymbol{x}^T . \nabla_{\Gamma} \boldsymbol{x} \, d\Gamma}{\int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\epsilon} \, d\Omega}, \quad 0 < \beta \leq 1. \tag{19}$$

β is the weighting factor that directly controls the importance of the volumetric strain minimization versus the boundary noisiness minimization. $\beta = 1$ means that the volumetric and surface criteria are treated equally.

3.3 FEM discretization

To solve PDEs, we use piecewise linear finite element functions to approximate the geometry and data fields, i.e., isoparametric finite elements. Therefore, actual geometry and its control field within each element are approximated as $x^i \approx \boldsymbol{N}^i \boldsymbol{x}^i, \quad s^i \approx \boldsymbol{N}^i \boldsymbol{s}^i; \quad i \in \{1,2,3\}$, where $\boldsymbol{N}^i$ is the vector of element shape functions in the ith Cartesian direction, $\boldsymbol{x}^i$ and $\boldsymbol{s}^i$ are the ith component of nodal coordinate vector and nodal control point vector, respectively. Furthermore, the traces of isoparametric bulk finite element functions on the boundary are used as surface finite elements. The same approach has been followed by Edelmann (2021); Dziuk and Elliott (2013) in the context of bulk-surface and surface PDEs. We derive a matrix-vector formulation of the discretized PDEs using a standard Galerkin-based finite element formulation. Then we can define the following bulk and surface mass and stiffness matrices:

$$\boldsymbol{M}_{\Omega} = \sum \int_{\Omega^e} \boldsymbol{N}^T \boldsymbol{N} d\Omega, \tag{20a}$$

$$\boldsymbol{M}_{\Gamma} = \sum \int_{\Gamma^e} \boldsymbol{N}_{\Gamma}^T \boldsymbol{N}_{\Gamma} d\Gamma \tag{20b}$$

$$\boldsymbol{K}_{\Omega} = \sum \int_{\Omega^e} (r_{\Omega^e}^H)^2 \boldsymbol{B}^T \boldsymbol{C} \boldsymbol{B} \, d\Omega \tag{20c}$$

$$\boldsymbol{K}_{\Gamma} = (r_{\Gamma}^H)^2 \sum \int_{\Gamma^e} (\nabla_{\Gamma} \boldsymbol{N}_{\Gamma})^T \nabla_{\Gamma} \boldsymbol{N}_{\Gamma} d\Gamma \tag{20d}$$

where $\boldsymbol{B}$ contains the spatial gradients of the bulk shape functions $\boldsymbol{N}$, $\boldsymbol{C}$ is the linear-elastic isotropic constitutive matrix, $\boldsymbol{N}_{\Gamma}$ is the traces of bulk shape functions on the boundary, $r_{\Omega^e}^H$ is the elemental Helmholtz bulk filter radius which is assumed to be spatially varying. In this work, with the aim to maintain the mesh quality and reduce the frequency of remeshing during shape optimization, the bulk filter radius is selected element-wise, based on element sizes. Among others, "Mesh-Jacobian-based stiffening" (MJBS) (Tezduyar et al. 1992; Stein et al. 2003) has been widely used to selectively stiffen or soften elements against shape changes. It can be implemented simply by dropping Jacobian from the finite element formulation of the mesh governing equations, resulting in the smaller elements being stiffened more than the larger ones. That can be realized by choosing the elemental Helmholtz filter radius as $r_{\Omega^e}^H = \frac{J^0}{J^e}$, where J^e is the Jacobian for element e and J^0 is a scaling parameter which is calculated based on Eq. 19 as follows

$$J^0 = \beta \frac{(r_{\Gamma}^H)^2 (\boldsymbol{x}_{\Gamma})^T \boldsymbol{K}_{\Gamma} \boldsymbol{x}_{\Gamma}}{(\boldsymbol{x})^T \left(\sum \int_{\Omega^e} (\frac{1}{J^e}) \boldsymbol{B}^T \boldsymbol{C} \boldsymbol{B} \, d\Omega \right) \boldsymbol{x}} \tag{21}$$

Having evaluated the bulk and surface mass and stiffness matrices, the shape governing equations for shells and solids (Eqs. 14,16) read respectively as

$$\boldsymbol{x}_{\Gamma} = (\boldsymbol{A}_{\Gamma}^I)^{-1} \cdot \boldsymbol{s}_{\Gamma}; \quad \boldsymbol{A}_{\Gamma}^I = (\boldsymbol{K}_{\Gamma} + \boldsymbol{M}_{\Gamma})^{-1} \cdot \boldsymbol{M}_{\Gamma} \tag{22a}$$

$$\boldsymbol{x} = (\boldsymbol{A}^I)^{-1} \cdot \boldsymbol{s}; \qquad \boldsymbol{A}_{\Gamma}^I = (\boldsymbol{K}_{\Gamma} + \boldsymbol{K}_{\Omega} + \boldsymbol{M}_{\Omega})^{-1} \cdot \boldsymbol{M}_{\Omega} \tag{22b}$$

where the superscript I refers to the implicit filtering, $\boldsymbol{A}_{\Gamma}^I$ and $\boldsymbol{A}^I$ are the implicit filter matrix of surface and bulk geometries, respectively. As the corresponding tangent matrices are symmetric and positive definite, the solution of the linear system can be achieved by utilizing efficient iterative solvers such as Conjugate Gradient.

3.4 Sensitivity analysis and update rule

Straight forward, applying the chain rule of differentiation, the derivative of the response function with respect to the surface and volumetric control points is given respectively as

$$\frac{dJ}{d\boldsymbol{s}_{\Gamma}} = \boldsymbol{M}_{\Gamma} \cdot (\boldsymbol{K}_{\Gamma} + \boldsymbol{M}_{\Gamma})^{-1} \cdot \frac{dJ}{d\boldsymbol{x}_{\Gamma}} \tag{23a}$$

$$\frac{dJ}{d\boldsymbol{s}} = \boldsymbol{M}_{\Omega} \cdot (\boldsymbol{K}_{\Gamma} + \boldsymbol{K}_{\Omega} + \boldsymbol{M}_{\Omega})^{-1} \cdot \frac{dJ}{d\boldsymbol{x}} \tag{23b}$$

It should be reminded that discrete sensitivities in the above equations are consistent nodal values that show size effects in the gradients. To avoid mesh dependency, the discrete control sensitivities are scaled by the inverse mass matrix, which basically means that we reconstruct the control gradient field sampled at mesh points from the consistent nodal sensitivities, viz.

$$\frac{dj}{d\boldsymbol{s}_{\Gamma}} = \boldsymbol{M}_{\Gamma}^{-1} \cdot \frac{dJ}{d\boldsymbol{s}_{\Gamma}} = (\boldsymbol{K}_{\Gamma} + \boldsymbol{M}_{\Gamma})^{-1} \cdot \frac{dJ}{d\boldsymbol{x}_{\Gamma}} \tag{24a}$$

$$\frac{dj}{d\boldsymbol{s}} = \boldsymbol{M}_{\Omega}^{-1} \cdot \frac{dJ}{d\boldsymbol{s}} = (\boldsymbol{K}_{\Gamma} + \boldsymbol{K}_{\Omega} + \boldsymbol{M}_{\Omega})^{-1} \cdot \frac{dJ}{d\boldsymbol{x}} \tag{24b}$$

The scaled gradients do not include disturbing discretization effects and ensure computations of efficient search directions without the necessity of second-order

information. Therefore, the discrete control field and subsequently the geometry are updated as follows

$$\begin{aligned} \Delta \boldsymbol{s}_\Gamma &= -\alpha \cdot \frac{dj}{d\boldsymbol{s}_\Gamma} \rightarrow \Delta \boldsymbol{x}_\Gamma = \boldsymbol{A}^I_\Gamma \cdot \Delta \boldsymbol{s}_\Gamma \\ \Delta \boldsymbol{s} &= -\alpha \cdot \frac{dj}{d\boldsymbol{s}} \rightarrow \Delta \boldsymbol{x} = \boldsymbol{A}^I \cdot \Delta \boldsymbol{s} \end{aligned} \tag{25}$$

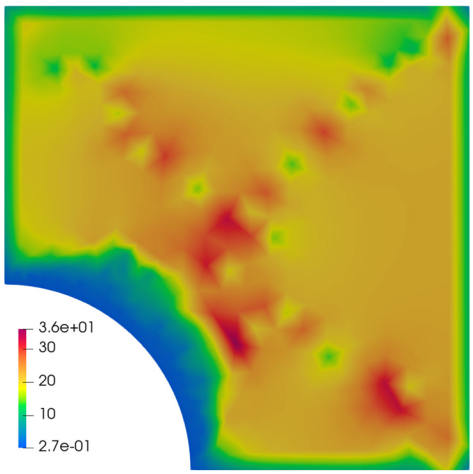
As mentioned earlier, one can interpret the proposed control update rule using the scaled gradients as a quasi-Newton step with the diagonal approximation of the Hessian matrix by the mass matrix.

3.5 Consistency check

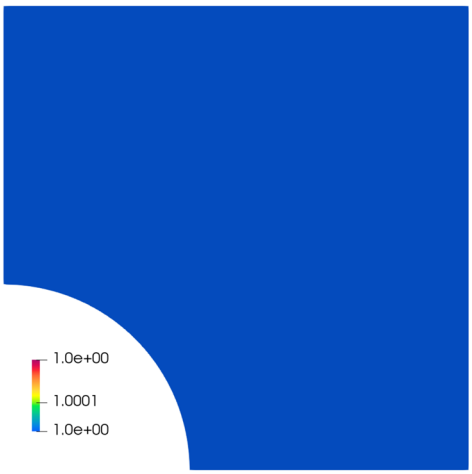
The discrete implicit filtering was derived and discussed using conventional finite elements. As is well known in finite element theory, a proper finite element formulation includes the rigidbody motion capability. However, for the sake of completeness, we numerically study the consistency aspects of implicit filtering. Perforated plate used in Sect. 2.2 is reconsidered, and the synthetic consistent sensitivity field in Fig. 2 is used for the test. Control sensitivities obtained after implicit (Sobolev/Helmholtz) filtering are demonstrated in Fig. 4. Similar to explicit filtering, the consistent/discrete control sensitivities are highly mesh-dependent, whereas the scaled ones are fully uniform and mesh-independent. The obtained results indicate that the consistency (rigid-body-movement production) can be perfectly achieved throughout the implicit shape filtering process if control sensitivities are scaled by the inverse mesh mass matrix, independent of the discretization.

4 Comparison of the explicit and implicit filters

The literature has discussed the relation between the convolution-based (explicit) and the Sobolev-/Helmholtz-based (implicit) filters. Stück and Rung (2011) state that explicit filtering using a Gaussian kernel function is a first-order approximation to the Sobolev-based filtering. On the other hand, Lazarov and Sigmund (2011) have shown in 1D the correlation between the explicit filtering using the linear hat function and the Helmholtz-based filtering. The fundamental solution of the non-homogeneous modified Helmholtz equation in Eq. 14 for an infinite domain reads (Polyanin and Nazaikinskii 2016)



(a) Discrete control sensitivities $dJ/d\boldsymbol{s}_\Gamma$.



(b) Scaled control sensitivities $dj/d\boldsymbol{s}_\Gamma = \boldsymbol{M}_\Gamma^{-1} \cdot dJ/d\boldsymbol{s}_\Gamma$.

Fig. 4 Control/filtered sensitivities computed with implicit filtering for the consistent/discrete sensitivities in Fig. 2

$$\begin{aligned} \boldsymbol{x}_0 &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F^H(\boldsymbol{x},\boldsymbol{x}_0)\, \boldsymbol{s}(\boldsymbol{x})\, dx^1\, dx^2\, dx^3 \\ F^H(\boldsymbol{x},\boldsymbol{x}_0) &= \frac{1}{4\pi \|\boldsymbol{x}-\boldsymbol{x}_0\|/(r^H)^2} e^{-\|\boldsymbol{x}-\boldsymbol{x}_0\|/r^H} \end{aligned} \tag{26}$$

which is basically a convolutional filtering using Green's function F^H as the kernel. It should be mentioned that the integral of Green's function is always one; therefore, the consistency property is naturally included. We also note that the Green's function in Eq. 26 is singular at the evaluation point $\boldsymbol{x}_0$, therefore we propose a regularized form of it as

$$F^{RH}(\boldsymbol{x},\boldsymbol{x}_0) = \frac{1}{1+(4\pi \|\boldsymbol{x}-\boldsymbol{x}_0\|/(r^H)^2)} e^{-\|\boldsymbol{x}-\boldsymbol{x}_0\|/r^H} \tag{27}$$

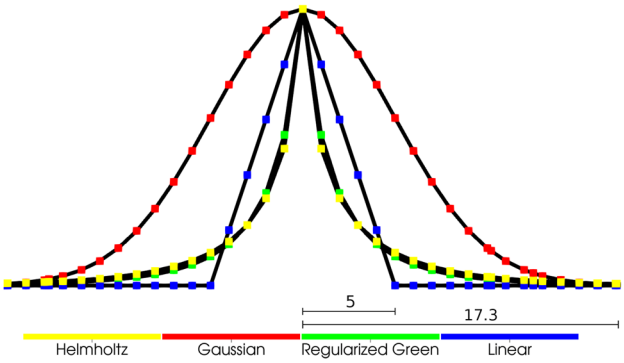


Fig. 5 Discretized profiles of the considered filter functions

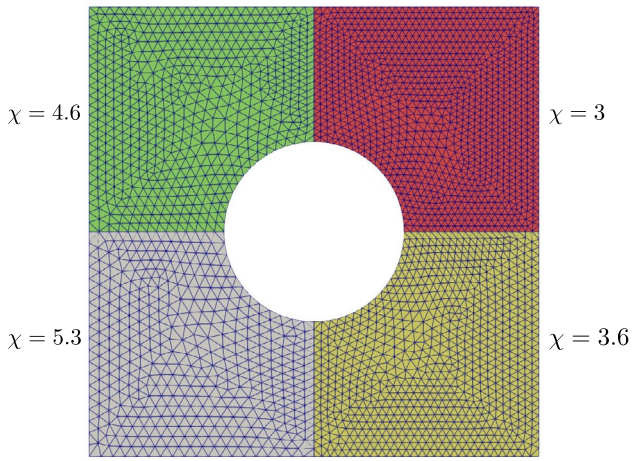


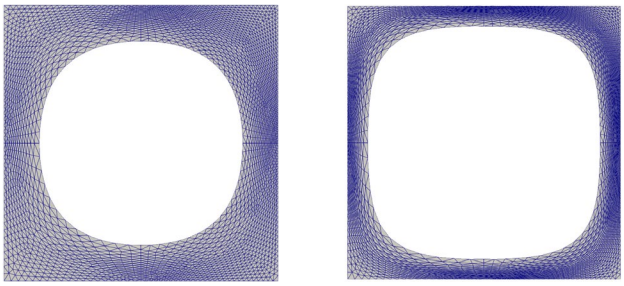
Fig. 6 Mirrored perforated plate example. The average element size χ changes at each quarter

In order to compare qualitatively explicit and implicit filters, a square flat plate of 100×100 meshed by triangles with the average element size of 1 is used. Figure 5 illustrates schematically the considered filter functions for a given radius of 5. It should be mentioned that the Helmholtz's kernel at every mesh point is calculated numerically from the corresponding row in the inverted Helmholtz's stiffness matrix, i.e., $(\boldsymbol{K}_\Gamma + \boldsymbol{M}_\Gamma)^{-1}$. The first observation indicates that the proposed regularized Green's function almost matches the numerical kernel of the Helmholtz filter. We also notice that the linear hat function is approximately spanning over $1/2\sqrt{3}$ of the other's support span, which confirms the relation developed by Lazarov and Sigmund (2011). Furthermore, during the upcoming sections, the performance of the presented filters is critically discussed for shape optimization of shells and solids receptively.

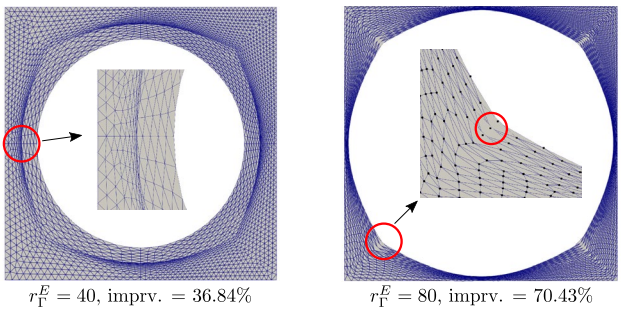
4.1 Mesh dependency in filtering

Discrete mesh is only a means for describing the geometry, and ideally, it should be independent of parameters like the element size and type. Subsequently, mesh-based calculations like FEM analysis and filtering are desired to be faultlessly mesh-independent. Let us reconsider the perforated plate example for studying the dependency of optimal shape on the mesh and filtering. For this study, the geometry is mirrored about the X and Y planes, and each quarter is meshed with a different average element size χ, as shown in Fig. 6.

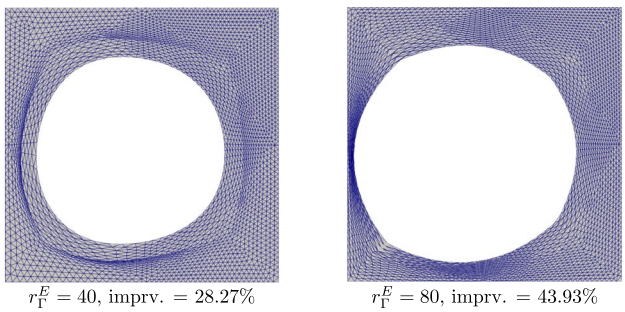
In order to ensure the convexity of the problem and avoid simulation errors that may influence the accuracy of shape gradients, the total surface area of the plate is chosen as the objective function. The outer boundaries of the plate are fully fixed during optimization, and two filter radii (40,80) are used for filtering. The results of this study for different cases are shown in Fig. 7. This optimization considers three shape filtering configurations. These include implicit filtering with the newly introduced mesh-independency treatment (a), explicit filtering with damping and mesh-independency treatment (b), and finally, explicit filtering with damping but without mesh-independency treatment (c). In all three cases, the optimization process terminates when the mesh becomes distorted because it interferes with the validity of the numerical simulation. Additionally, for the implicit filtering case, we provide shape evolutions at different objective improvement levels (imprv.) to demonstrate the robustness of this technique in preserving mesh quality. Based on the provided results, several crucial observations can be made, including: (I) the implicit filtering outperforms the explicit filtering in preserving surface mesh quality and geometrical properties (such as plane symmetry) regardless of the filter and mesh size. This observation is particularly significant since, in practice, meshes tend to be highly non-uniform in complex geometries and physics. Therefore, it is crucial that the filtering and mesh size are independent of each other. (II) By comparing Fig. 7b and c, we can evaluate the impact of the introduced mesh-independency treatment (i.e., scaling discrete sensitivities with the inverse of the mesh mass matrix) on the performance of the explicit filter. This observation serves to confirm the significance of the rigid-body-movement production property for filtering, as discussed in detail in Sect. 2.2. (III) As observed from the circled and zoomed regions in Fig 7b, the damping technique used in explicit filtering to prevent fixed domains from moving results in unsuitable interference between filtering and damping near these regions, leading to mesh distortion. In contrast, enforcing fixed boundaries is straightforward in implicit filtering because it is naturally achieved through the Dirichlet boundary condition of the governing PDE of the filter (see Eq. 13).



(a) Design evolution using implicit filtering. Results for $r_r^H = 40, 80$ are overlaying.



(b) Optimization using explicit filtering. Discrete control sensitivities are scaled by the inverse of mesh mass matrix.



(c) Optimization using explicit filtering and without scaling the discrete control sensitivities.

Fig. 7 Area minimization of the perforated plate in Fig. 6. Distorted or collapsed mesh is used as the optimization stopping criterion

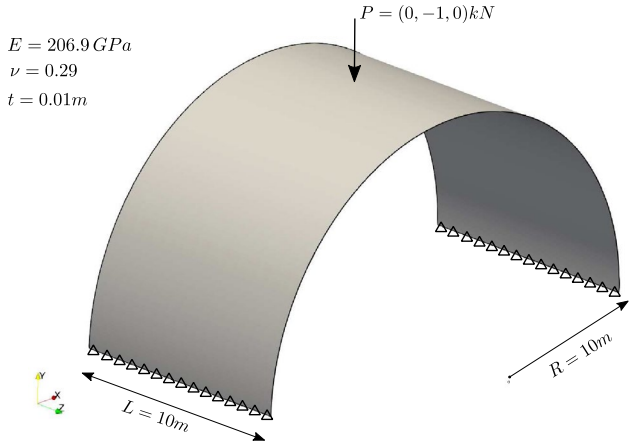


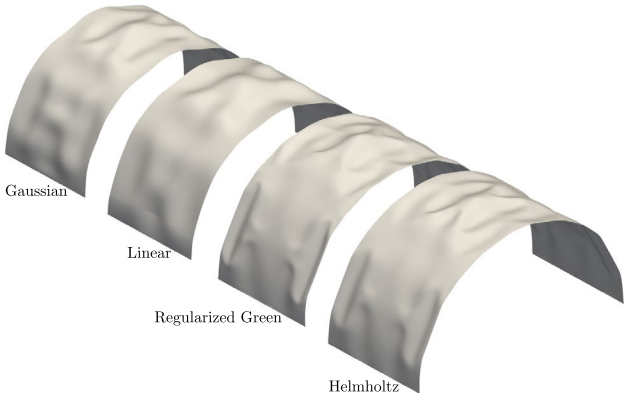
Fig. 8 Geometry, support and loading of half-cylinder shell

4.2 Numerical properties

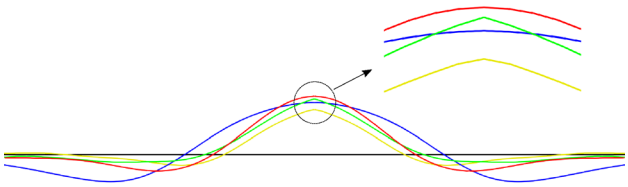
This section intends to compare the presented filter kernels mainly concerning numerical properties. Figure 8 shows a half-cylinder shell fully clamped at the edges and loaded by a nodal force at the center. The specified geometry is described by 73522 unstructured triangles with an average element size of $0.1m$. As a linear system solver, we use AMGCL - an efficient parallel iterative linear solver Demidov (2020).

Helmholtz, regularized Green, Gaussian, and linear hat kernel functions are used for the following investigations. The projected steepest descent algorithm with constant step size (Najian Asl 2019) is employed to minimize the linear strain energy of the structure while constraining its mass. Optimal geometries obtained for the considered kernel function are presented in Fig.9a. Filter radii are set so that the kernel functions have the same support span $p = 2m$. It is observed that the Helmholtz and regularized Green kernels resulted in similar optimal geometries, which are sharper than the others due to the resolution of local features. On the other hand, it can be seen from Fig.9b that the Helmholtz and regularized Green kernels allow the creation of a kink at the center; therefore, not only do they obtain smooth shapes but could develop a kink as well (Müller et al. 2021). The improved objective function is depicted in Fig.9c. It shows that all filter functions reduced the structural strain energy significantly; however, optimization with the Helmholtz filtering seems to converge faster.

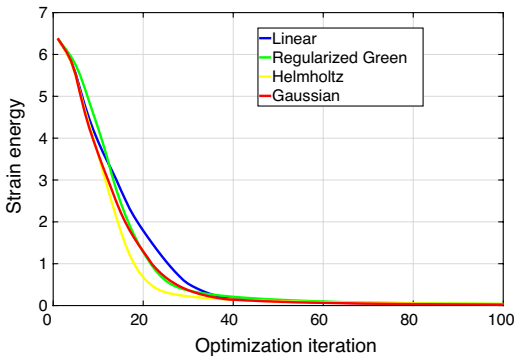
A comparison of the time necessary for applying the explicit and implicit filters for different support span to element size ratios p/a is shown in Fig. 10a. For reference, this study is performed on a 6 core XEON W-2133 3.60GHz processor using hyper-threads-based parallelism. The neighbor search and distance evaluation needed for the explicit filtering is done by use of octrees for the sake of efficiency. The



(a) Optimal geometries obtained with different kernels of the same support span $p = 2m$.



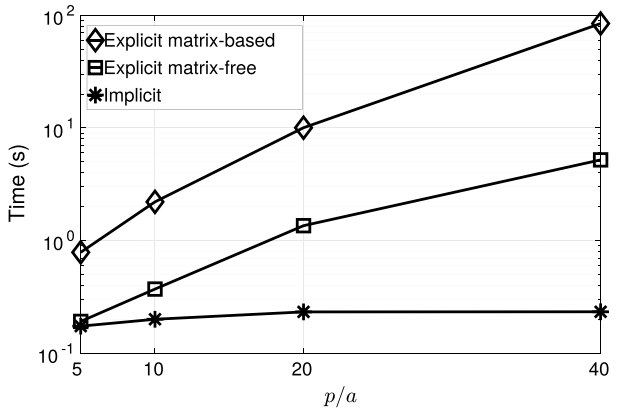
(b) Cross sections of optimal geometries through the YZ plane at the center.



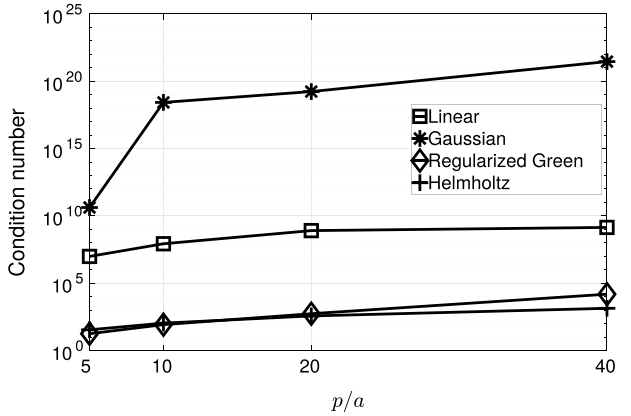
(c) Convergence of objective.

Fig. 9 Optimal design of the half-cylinder shell

results indicate that the explicit filter, regardless of whether it was implemented with the matrix-based method or without storing the filter matrix, is slower than the implicit PDE filter. Additionally, as the ratio of *p*/*a* increases, the cost of the explicit filter rises excessively, whereas the cost of the implicit filter shows minimal growth and remains nearly constant even for high ratios. In conclusion, the cost of constructing a search tree and conducting a neighbor search to explicitly filter a surface field is higher than the cost of solving a linear system that is associated with implicit surface filtering. This aligns with the findings and conclusions made by Lazarov and Sigmund (2011).



(a) Time necessary for computing filtered field using implicit (Helmholtz) filter and the explicit filter with (matrix-based) and without (matrix-free) storing the filter matrix for different support span to element size ratios p/a.



(b) Condition number of the discrete filter operator matrix of the considered kernels for different support span to element size ratios p/a.

Fig. 10 Numerical properties of different kernel functions

To investigate the sensitivity of filtering and subsequently the optimal shape to the filter radius, the condition number of the discrete filter operator is used in this work. This is particularly important since the conditioning of the filter matrix directly influences the conditioning of the optimization problem, see Najian Asl et al. (2017) for discussions and studies on this subject. Conditioning of the discrete filter operator matrix is shown in Fig. 10b as a function of the support span. It was predictable that the Helmholtz and regularized Green kernel functions would behave very similarly. They show very good-conditioning with much less sensitivity to the *p*/*a* ratio. The linear and Gaussian kernels generally have ill-conditioning behavior, which worsens by increasing the support span. So we can conclude that the regularized Green kernel function may be preferred over the others for explicit shape filtering.

4.3 Challenge of fixing domains

In node-based optimization, interpolating or fixing the non-design domains during the optimization iterations can present a challenge, depending on the problem setup. For instance, in Sect. 4.1, we noticed artifacts around the outer boundaries (non-design/fixed edges) due to the conflict between explicit filtering and damping. However, no such artifacts were observed in Sect. 4.2. To better understand the challenge, let us consider the optimization problem of the shell roof shown in Fig. 11. The roof is secured at its corners and experiences a uniform load. The load distribution will not alter even if the roof undergoes shape modifications. Additionally, the linear static analysis is conducted using Kratos-Multiphysics and the surface is divided into 5416 nodes and 10498 triangular shell elements with an average size of $0.3m$. In the structural analysis and optimization, the four corners are completely fixed and during optimization, one-quarter of the inner holes (as indicated) are also fixed. Examining the results, the shape obtained through implicit filtering appears highly smooth, natural, and satisfies the geometrical constraint. On the other hand, the shape obtained through explicit filtering has undesirable and meaningless artifacts near the fixed edges. Geometry interpolation represents a strong and point-wise Dirichlet boundary condition for the shape optimization problem. Explicit shape filtering, however, uses damping to limit modifications of fixed boundaries in a practical manner. In contrast, implicit filtering consistently satisfies geometric constraints through Dirichlet boundary conditions, regardless of the filter radius selected. This example clearly demonstrates the superiority of implicit filtering over the other method for node-based shape optimization.

4.4 Solid optimization

This section is devoted to the analysis and comparison of the presented methods for the shape optimization of solid structures. It is significant because attention must be paid not only to smoothing or regularizing the surface mesh but also to preserving internal mesh quality by properly handling the volume mesh. In the upcoming sections, we will evaluate the implicit bulk-surface filtering approach outlined in Sect. 3.2 against the explicit handling of surface and volume mesh for two complex geometry problems.

4.4.1 Tensegrity-tower node: mass minimization

In Ghantasala et al. (2021a), the Vertex-Morphing technique as an explicit surface filtering was used in shape optimization of the connecting nodes of a five-meter tensegrity ("tensional integrity") tower, see Fig. 12a. The tower is float in the air and built out of tensile and compressive elements. The nodes are highly complex connections between the rods and cables, and they are additively manufactured using the laser powder bed fusion (LPBF) process.

Here, an aluminum node of the tower is reconsidered for shape optimization. Figure 12b illustrates the problem setup, including the boundary conditions for structural analysis and design optimization. Volume mesh consists of 212689 tetrahedrons and the outer surface of the node colored in green is subject to optimization. In the first step, the mass of the node is optimized without any constraint. This is especially challenging due to the large compression that mesh undergoes to reach the global optimal shape, which is basically the projection of the design surface onto the non-design surfaces. Figure 13 depicts optimal shapes obtained with the

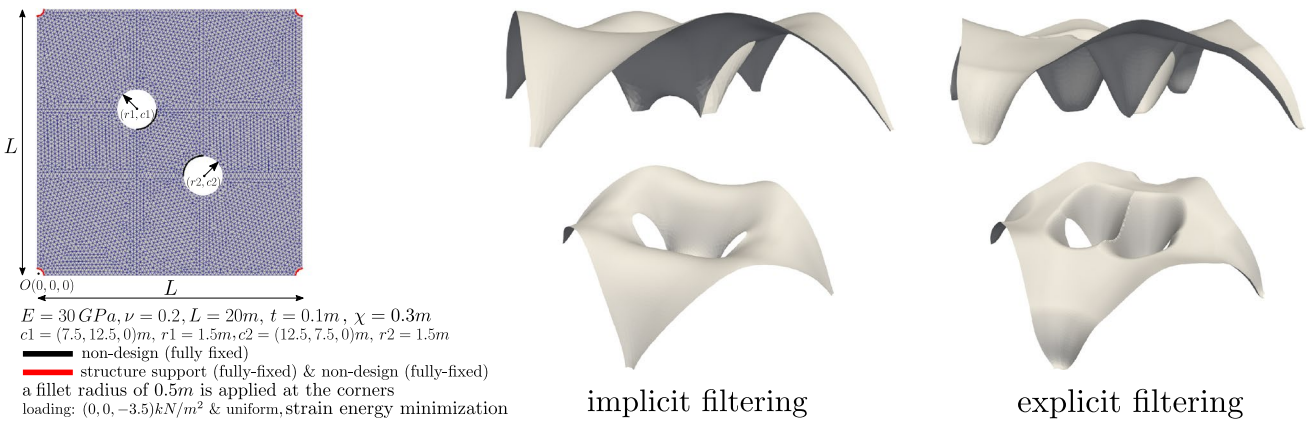
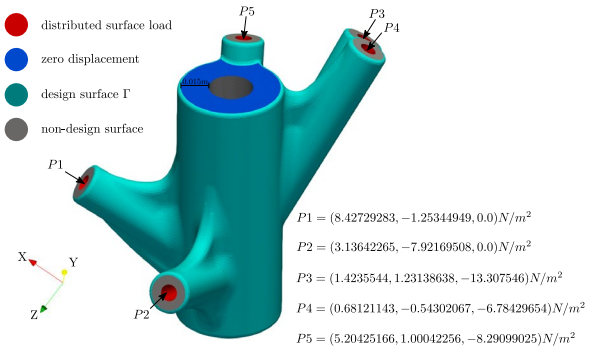


Fig. 11 Settings and optimization results for the square roof with holes

(a) The five-meter "tensegrity tower" exhibited in the Deutsches Museum of Science and Technology in Munich.



(b) Problem setup of an aluminum connecting node.

Fig. 12 Shape optimization of the nodes of a tensegrity-tower

bulk-surface Helmholtz filtering using $\beta = 1$ and the Vertex-Morphing technique. While the former handles boundary and internal nodes implicitly and simultaneously, the latter treats them sequentially by first applying the explicit filtering on the surface and then using a pseudo-structural model with

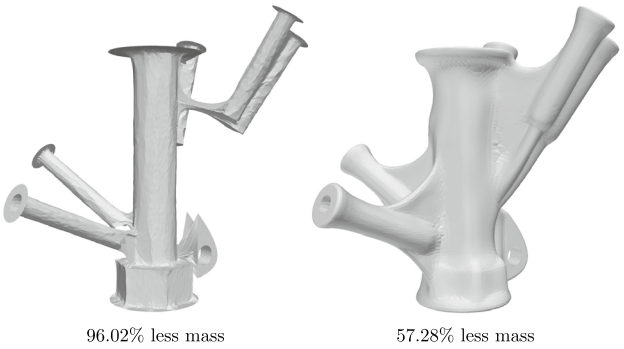
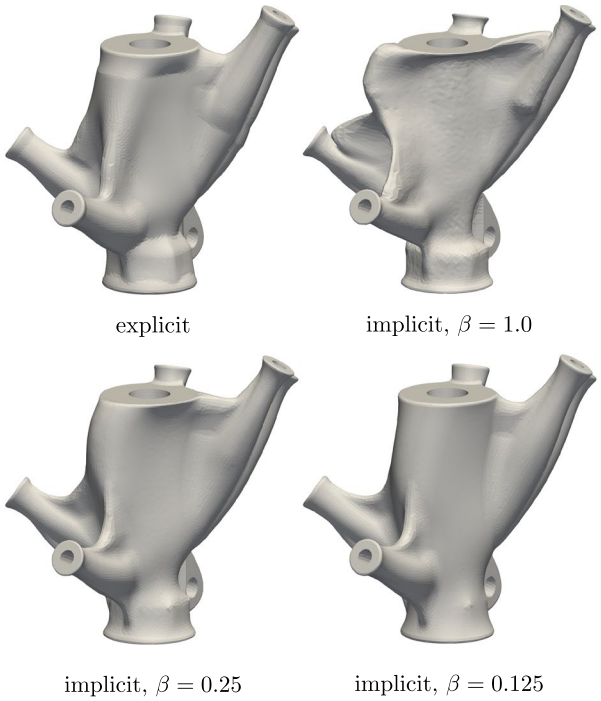


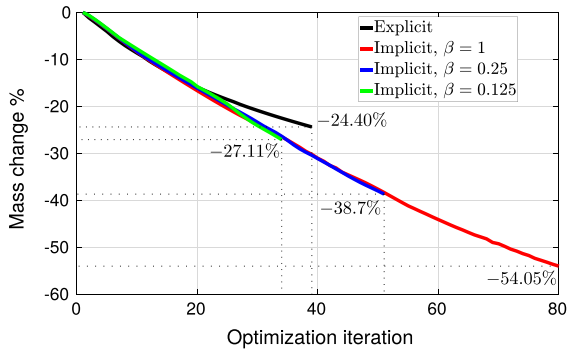
Fig. 13 Unconstrained mass minimization of the considered tensegrity node using the bulk-surface Helmholtz shape filtering with $\beta = 1$ (left) and the explicit filtering (right)

the Jacobian-based stiffening to deform the internal mesh. Furthermore, the distorted or collapsed mesh is used as the optimization stopping criterion. Although both methods leverage the Jacobian-based stiffening technique to improve the deformed internal mesh quality, with the bulk-surface Helmholtz filtering much bigger surface deformation and objective improvement (96.02% vs. 57.28%) could be achieved.

To study the influence of the weighting factor β on the performance of the proposed bulk-surface filtering, strain energy minimization of the node is performed under mass constraint, and the result is compared against that of the Vertex-Morphing technique. It should be noticed that the lower the factor is, the less attention or weight is given to the preservation of the internal mesh quality, which results in an earlier mesh distortion. This is clearly seen in the results presented in Fig. 14. However, we observe that even for a small value $\beta = 0.125$, the proposed implicit filtering outperforms the explicit one. On the other hand, we notice that the proposed bulk-surface filtering requires a linear system solve (Eq. 23b) for calculating each response function derivative with respect to solid control points. A comparison of the time necessary for filtering a surface sensitivity field using the explicit surface filter and the bulk-surface filter for different support span to element size ratios p/a is shown in Fig. 15. Similar observations and conclusions as in the shell optimization case (Sect. 4.2) can be made. For moderate ratios, e.g., $p/a = 10$, the cost of the bulk-surface filter is comparable to that of the matrix-free explicit filtering, whereas for higher ratios cost of the matrix-free explicit filtering hinders the performance.



(a) Geometries obtained with the explicit filter and the implicit filter for various weighting factor β .



(b) History of objective function until volume mesh distortion (elements with negative Jacobian) stops FEM calculation.

Fig. 14 Strain energy constrained mass minimization of the tensegrity node using the bulk-surface Helmholtz shape filtering (implicit) and the Vertex-Morphing technique (explicit)

4.4.2 Hook: stiffness maximization

The second solid shape optimization example is the stiffness maximization (strain energy minimization) of a hook under the initial volume constraint. Figure 16 provides settings and optimization results using the explicit and bulk-surface implicit techniques. The geometry is discretized by 39,292 nodes (× 3 = number of design variables) and 197,541 small displacement elements. A relatively large filter radius of $0.06m$ was intentionally used to ensure global (low-frequency) shape changes during

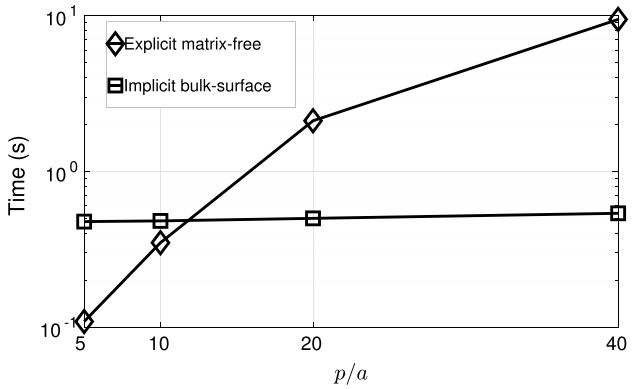


Fig. 15 Time necessary for filtering a surface sensitivity field using the bulk-surface filter and the explicit surface filter without (matrix-free) storing the filter matrix for various *p*/*a* ratios

optimization. The β parameter of the bulk-surface regularization (implicit) is set to one, which means that the volumetric and surface criteria are treated equally. The distorted or collapsed mesh was also used as the optimization stopping criterion. From the results, it is evident that explicit filtering is not as effective as the implicit bulk-surface filtering in maintaining mesh quality and improving the design. Upon close examination of the explicit filtering result at the cross-section (b), it is evident that damping was not successful in ensuring low shape mode during the transition from the non-design to design domains. Based on the observations made and the fact that the explicit technique becomes highly computationally demanding for large filter radii, the implicit technique is the better choice for robust and efficient node-based optimization of solids.

5 Conclusions

In this work, we visited two primary filtering techniques for node-based shape optimization: convolution-based (explicit) and PDE-based (implicit). Consistency and mesh independency, as two strict requirements for any kind of filtering, were carefully discussed and demonstrated for both techniques. Mesh-independent filtering was achieved by scaling discrete control sensitivities with the inverse of the mesh mass matrix. Supported by numerical experiments, a regularized Green's function was introduced as an equivalent explicit form of the so-called Helmholtz-/Sobolev-based (implicit) filter. It was observed that the Helmholtz and regularized Green kernels not only do they obtain smooth shapes but can allow the generation of features like kink or a corner if the optimum requires. With the aim of controlling the boundary smoothness and preserving the internal mesh quality simultaneously, this work introduced the implicit bulk-surface filtering technique for the shape optimization

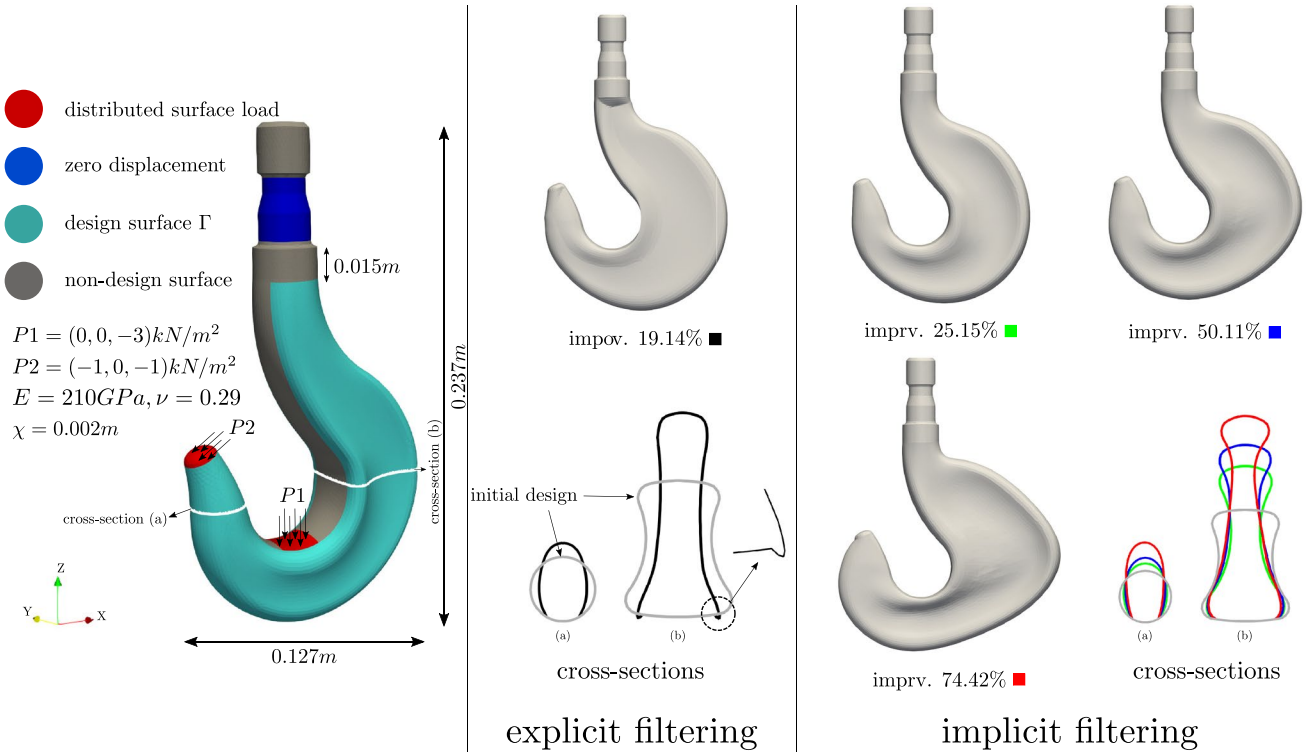


Fig. 16 Stiffness maximization of a hook under initial volume constraint

of volumetric domains. Its superior efficiency and robustness from explicit filtering were demonstrated for shape optimization of complex solid structures. Overall, whether the surface or bulk domain is filtered, the implicit approach was numerically more efficient and unconditionally consistent than the explicit one.

Acknowledgements This research is funded by Deutsche Forschungsgemeinschaft (DFG) through the project 414265976 TRR 277 C-02. The authors gratefully acknowledge the support.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Replication of results The software package used in this work, *Kratos-Multiphysics*, is open-source and available for download at https://github.com/KratosMultiphysics/Kratos. The used datasets are available from the corresponding author upon reasonable request.

References

Antonau I, Warnakulasuriya S, Bletzinger K-U, Bluhm FM, Hojjat M, Wüchner R (2022) Latest developments in node-based shape optimization using vertex morphing parameterization. Struct Multidisc Optim 65(7):1–19

Azegami H (1994) Solution to domain optimization problems. Trans Jpn Soc Mech Eng 60(574):1479

Azegami H, Takeuchi K (2006) A smoothing method for shape optimization: traction method using the robin condition. Int J Comput Methods 3(01):21–33

Azegami H, Wu ZC (1996) Domain optimization analysis in linear elastic problems: approach using traction method. JSME Int J Ser A Mech Mater Eng 39(2):272–278

Baumgärtner D (2020) On the grid-based shape optimization of structures with internal flow and the feedback of shape changes into a CAD model. PhD thesis, Technische Universität München

Bletzinger K-U (2014) A consistent frame for sensitivity filtering and the vertex assigned morphing of optimal shape. Struct Multidisc Optim 49(6):873–895

Bletzinger K-U, Firl M, Linhard J, Wüchner R (2010) Optimal shapes of mechanically motivated surfaces. Comput Methods Appl Mech Eng 199(5–8):324–333

Bletzinger K-U, Wüchner R, Daoud F, Camprubí N (2005) Computational methods for form finding and optimization of shells and membranes. Comput Methods Appl Mech Eng 194(30–33):3438–3452

Bourdin B (2001) Filters in topology optimization. Int J Numer Methods Eng 50(9):2143–2158

Bruns TE, Tortorelli DA (2001) Topology optimization of non-linear elastic structures and compliant mechanisms. Comput Methods Appl Mech Eng 190(26–27):3443–3459

Demidov D (2020) Amgcl—a c++ library for efficient solution of large sparse linear systems. Softw Impacts 6:100037

Dick T, Schmidt S, Gauger NR (2021) Combining Sobolev smoothing with parameterized shape optimization. arXiv preprint. arXiv:2109.15279

Dick T, Gauger NR, Schmidt S (2022) Combining Sobolev smoothing with parameterized shape optimization. Comput Fluids 244:105568

Dziuk G, Elliott CM (2013) Finite element methods for surface PDEs. Acta Numer 22:289–396

Edelmann D (2021) Isoparametric finite element analysis of a generalized robin boundary value problem on curved domains. SMAI J Comput Math 7:57–73

Eppler K, Schmidt S, Schulz V, Ilic C (2009) Preconditioning the pressure tracking in fluid dynamics by shape Hessian information. J Optim Theory Appl 141(3):513–531

Firl M, Bletzinger K-U (2012) Shape optimization of thin walled structures governed by geometrically nonlinear mechanics. Comput Methods Appl Mech Eng 237:107–117

Ghantasala A, Diller J, Geiser A, Wenzler D, Siebert D, Radlbeck C, Wüchner R, Mensinger M, Bletzinger K-U (2021a) Node-based shape optimization and mechanical test validation of complex metal components and support structures, manufactured by laser powder bed fusion. In: International conference on applied human factors and ergonomics, pp 10–17. Springer, Cham

Ghantasala A, Najian Asl R, Geiser A, Brodie A, Papoutsis E, Bletzinger K-U (2021b) Realization of a framework for simulation-based large-scale shape optimization using vertex morphing. J Optim Theory Appl 189(1):164–189

Hojjat M, Stavropoulou E, Bletzinger K-U (2014) The vertex morphing method for node-based shape optimization. Comput Methods Appl Mech Eng 268:494–513

Jameson A, Vassberg JC (2000) Studies of alternative numerical optimization methods applied to the brachistochrone problem. Comput Fluid Dyn J 9(3):281–296

Kawamoto A, Matsumori T, Yamasaki S, Nomura T, Kondoh T, Nishiwaki S (2011) Heaviside projection based topology optimization by a PDE-filtered scalar function. Struct Multidisc Optim 44(1):19–24

Kröger J, Rung T (2015) Cad-free hydrodynamic optimisation using consistent kernel-based sensitivity filtering. Ship Technol Res 62(3):111–130

Lazarov BS, Sigmund O (2011) Filters in topology optimization based on helmholtz-type differential equations. Int J Numer Methods Eng 86(6):765–781

Le C, Bruns T, Tortorelli D (2011) A gradient-based, parameter-free approach to shape optimization. Comput Methods Appl Mech Eng 200(9–12):985–996

Mohammadi B, Pironneau O (2009) Applied shape optimization for fluids. OUP, Oxford

Müller PM, Kühl N, Siebenborn M, Deckelnick K, Hinze M, Rung T (2021) A novel p-harmonic descent approach applied to fluid dynamic shape optimization. Struct Multidisc Optim 64(6):3489–3503

Najian Asl R (2019) Shape optimization and sensitivity analysis of fluids, structures, and their interaction using vertex morphing parametrization. PhD thesis, Technische Universität München

Najian Asl R, Shayegan S, Geiser A, Hojjat M, Bletzinger K-U (2017) A consistent formulation for imposing packaging constraints in shape optimization using vertex morphing parametrization. Struct Multidisc Optim 56(6):1507–1519

Oden J, Reddy J (1972) Mixed conjugate finite-element approximations of linear operators. J Struct Mech 1(1):113–131

Polyanin AD, Nazaikinskii VE (2016) Handbook of linear partial differential equations for engineers and scientists. Chapman and Hall/CRC, Boca Raton

Riehl S, Friederich J, Scherer M, Meske R, Steinmann P (2014) On the discrete variant of the traction method in parameter-free shape optimization. Comput Methods Appl Mech Eng 278:119–144

Schmidt S, Ilic C, Gauger N, Schulz V (2008) Shape gradients and their smoothness for practical aerodynamic design optimization. Optim Eng. Preprint No. SPP1253-10-03

Sigmund O (1994) Design of material structures using topology optimization. PhD thesis, Technical University of Denmark Lyngby

Stein K, Tezduyar T, Benney R (2003) Mesh moving techniques for fluid-structure interactions with large displacements. Journal of Appl Mech 70(1):58–63

Stück A, Rung T (2011) Adjoint rans with filtered shape derivatives for hydrodynamic optimisation. Comput Fluids 47(1):22–32

Tezduyar T, Behr M, Mittal S, Johnson A (1992) Computation of unsteady incompressible flows with the stabilized finite element methods: space–time formulations, iterative strategies and massively parallel implementations. In: New methods in transient analysis, American Society of Mechanical Engineers, Pressure Vessels and Piping Division (Publication) PVP, pp 7–24. ASME. Winter Annual Meeting of the American Society of Mechanical Engineers, 08–11 December 1992

Tonon P, Sanches RAK, Takizawa K, Tezduyar TE (2021) A linear-elasticity-based mesh moving method with no cycle-to-cycle accumulated distortion. Comput Mech 67(2):413–434