INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2017; **00**:1–22
Published online in Wiley Online Library (www.onlinelibrary.wiley.com). DOI: 10.1002/nme.5621

Acoustic shape optimization using cut finite elements

Anders Bernland*, Eddie Wadbro, and Martin Berggren

Department of Computing Science, Umeå University, SE-901 87 Umeå, Sweden.

SUMMARY

Fictitious-domain methods are attractive for shape optimization applications, since they do not require deformed or re-generated meshes. A recently developed such method is the CutFEM approach, which allows crisp boundary representations and for which uniformly well-conditioned system matrices can be guaranteed. Here we investigate the use of the CutFEM approach for acoustic shape optimization, using as test problem the design of an acoustic horn for favorable impedance-matching properties. The CutFEM approach is used to solve the Helmholtz equation, and the geometry of the horn is implicitly described by a level-set function. To promote smooth algorithmic updates of the geometry, we propose to use the nodal values of the Laplacian of the level-set function as design variables. This strategy also improves the algorithm's convergence rate, counteracts mesh dependence, and, in combination with Tikhonov regularization, controls small details in the optimized designs. An advantage with the proposed method is that the exact derivatives of the discrete objective function can be expressed as boundary integrals, as opposed to when using a traditional method that employs mesh deformations. The resulting horns possess excellent impedance-matching properties and exhibit surprising sub-wavelength structures, not previously seen, which are possible to capture due to the fixed-mesh approach.

Received . . .

KEY WORDS: shape optimization; level set; CutFEM; sensitivity analysis; acoustic horn; Helmholtz equation

*Correspondence to: Anders Bernland, Department of Computing Science, Umeå University, SE-901 87 Umeå, Sweden. E-mail: anders.bernland@cs.umu.se.

Contract/grant sponsor: Swedish Foundation for Strategic Research; contract/grant number: AM13-0029; Swedish Research Council; contract/grant number: 621-2013-3706; eSSENCE, a strategic collaborative eScience program funded by the Swedish Research Council

1. INTRODUCTION

Numerical shape optimization algorithms are increasingly applied to aid in the design of devices in engineering fields such as elasticity, fluid mechanics, electromagnetics, and acoustics. Such algorithms operate by iteratively updating the geometry of the device, typically by modifying the shape of its boundary. At each iteration, the performance of the resulting device needs to be determined. To do so, the finite element method is a common choice when partial differential equations are involved. However, a traditional body-fitted finite element mesh needs to be updated as the geometry changes, and, irregardless if this is done with a deformation of the mesh or by a complete re-meshing, there is a computational cost, and the process is not always robust. Re-meshing can also influence the optimization process negatively. An alternative is to use a fixed mesh that does not need to conform to the geometry. Methods employing this strategy are usually labeled fictitious domain, domain embedding, or immersed boundary methods. In this paper, we consider shape optimization of an acoustic horn on a fixed finite element mesh. The horn has been used as a model problem in a series of previous studies [1–7].

Finite element methods on fixed meshes are commonly combined with a level-set description of the geometry, where the boundary is implicitly defined as the zero-level contour of a function, often referred to as the level-set function. Initially used for handling evolving boundaries by Osher & Sethian [8], the level-set method was also adopted for structural optimization [9], as detailed in several reviews [10–12]. In early studies of level-set-based optimization methods with finite elements applied to cantilever beam model problems, the geometrical boundaries were blurred by giving intersected elements average material properties, or by using approximate Heaviside functions [13–15]. In contrast, Van Miegroet & Duysinx [16] used exact Heaviside functions to achieve a crisp description of the boundary, effectively not integrating over the part of the finite elements that were void. Similar strategies have been proposed and refined by Wei et al. [17], Li et al. [18], Makhija & Maute [19], and Villanueva & Maute [20]. More recent studies have concerned multiple materials and inclusions [21–23]. There are also publications covering other fields of engineering, such as fluid flow [24, 25], electrostatics [26], and magnetic actuators [27]. However, to the authors' best knowledge, there does not seem to be any previous study on level-set-based optimization using finite element methods on fixed meshes for wave propagation problems.

Finite element approaches allowing crisp boundary descriptions on meshes that do not conform to the geometry are, for example, XFEM [28] and CutFEM [29]. An undesired consequence of the fixed mesh is that the condition numbers of the system matrices can become large when a boundary cuts the mesh in a way that includes only a fraction of an element inside a domain of interest. Countering this issue is essential, and the solution used here is to add a stabilization term ("ghost penalty") involving the jump in the fluxes over the edges (or faces in 3D) of intersected elements, as proposed by Burman [30].

The choice of parametrization of the geometry to be optimized is crucial, since the optimization algorithm can only explore the chosen parameter space. Therefore, instead of parametrizing the design by a set of geometric primitives [16, 17, 23], or an explicit boundary representation [1], and thus bias the optimization, we choose to discretize the level-set function and use its nodal values as parameters [11, 19, 20]. Smoothing, or filtering, together with regularization is often applied to shape and topology optimization problems to promote smooth designs, control the feature size, counteract

mesh dependence, and improve the convergence rate [11]. We choose to employ a smoothing algorithm that promotes smooth level-set functions. The approach has previously been employed to acoustic horn optimization by Bängtsson et al. [1], but here we adapt it for level-set parametrization. Furthermore, we find that Tikhonov regularization [31] works well to control the feature size in the final design.

Since a large number of design variables are used, a gradient based optimization scheme is suitable. Consequently, information about the change in performance due to changes in the design variables are needed. A common approach is "differentiate then discretize", where the derivatives in the continuous optimization problem are discretized. However, to obtain accurate descent directions and better convergence in the discrete optimization, it is often advantageous to "discretize then differentiate", that is, to compute the derivatives in the discrete setting. The latter approach is employed here.

Sensitivity analysis for optimization using body-fitted finite elements is well studied, but there are essential differences when using a fixed mesh. Noël et al. [32] derive expressions for the derivatives of the system matrices in a bimaterial linear elasticity optimization problem. Sharma et al. [33] derive similar expressions in a general setting, and propose, due to their complexity, a semi-analytical approach, partly using finite differences, or reverting to the differentiate-then-discretize approach. Najafi et al. [34] perform similar shape sensitivity analysis for IGFEM, a finite element method on fixed meshes where additional nodes are introduced at the interface. Finally, Barthold & Materna [35] introduce additional, boundary-conforming basis functions inside the intersected elements in the YFEM method to make the sensitivity analysis more similar to the well-studied case with a body-fitted mesh. We adopt a different approach for our optimization problem; we derive an exact expression for the discrete derivative in terms of an integral over the design boundary. That this is possible is an unexpected benefit of the fixed mesh, as opposed to a body-fitted mesh. In the latter case, the boundary integral needs to be exchanged for a domain integral involving the mesh deformation to avoid consistency errors [36].

2. PROBLEM STATEMENT

A planar symmetric acoustic horn is considered, see Figure 1. This geometry corresponds to a horn extending infinitely in the direction perpendicular to the plane and models a wide planar horn. A model of a rotationally symmetric horn can be obtained with a change to polar coordinates. The waveguide is assumed infinite to the left. The purpose of the horn is to convert incoming plane waves from the waveguide to outgoing cylindrical waves, and the optimization problem, described in more detail later in this section, is to minimize the amount of power being reflected back to the waveguide. The horn has been used as a model problem for acoustic design optimization in a series of papers [1–7].

2.1. The acoustical problem

The horn is assumed symmetric, and thus only half the horn needs to be modelled, see Figure 2, and the walls of the horn are assumed to be sound hard. A perfectly matched layer (PML) [37] absorbs the outgoing waves where the computational domain is truncated, and therefore the boundary

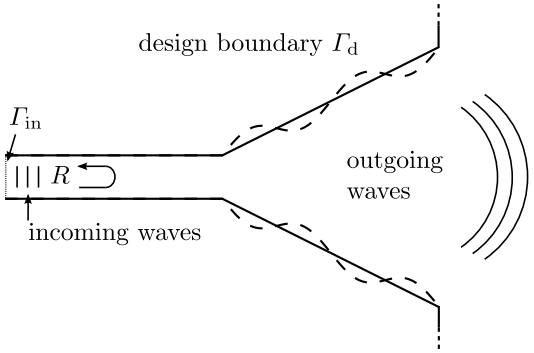

Figure 1. The considered planar symmetric horn.

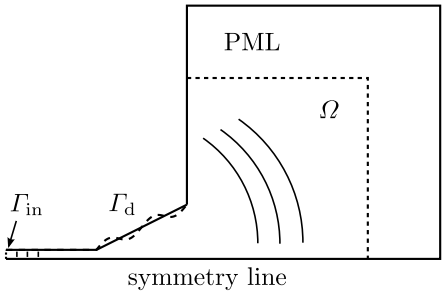

Figure 2. The computational domain Ω.

conditions at the outer boundaries have negligible influence on the solution and are set to be sound hard as well. An incoming plane wave of unit amplitude is imposed by the boundary condition on Γ_{in}.

We assume that the acoustic pressure $P(\boldsymbol{x}, t)$ satisfies the linear wave equation. Here $\boldsymbol{x}$ denotes the spatial coordinate and t denotes time. For time harmonic solutions $P(\boldsymbol{x}, t) = \mathrm{Re}(p(\boldsymbol{x})\mathrm{e}^{\mathrm{i}2\pi f t})$, the complex amplitude function p satisfies the system of equations

$$\nabla \cdot (\mathbf{G}\nabla p) + k^2 \gamma p = 0 \quad \text{in } \Omega, \tag{2.1a}$$

$$\mathrm{i}kp + \frac{\partial p}{\partial n} = 2\mathrm{i}k \quad \text{on } \Gamma_{\mathrm{in}}, \tag{2.1b}$$

$$\frac{\partial p}{\partial n} = 0 \quad \text{on } \Gamma \setminus \Gamma_{\mathrm{in}}, \tag{2.1c}$$

where $k = 2\pi f/c$ is the wavenumber, f is the frequency, c is the speed of sound, i is the imaginary unit, $\partial/\partial n = \hat{\boldsymbol{n}} \cdot \nabla$ is the normal derivative, and $\hat{\boldsymbol{n}}$ is the outwards directed unit normal. The complex diagonal matrix $\mathbf{G}$ and complex constant γ perform the damping of the outgoing waves in the PML, and reduce to the identity matrix $\mathbf{I}$ and 1, respectively, outside the PML [37]. For more details on the acoustic modelling, see Bängtsson et al. [1].

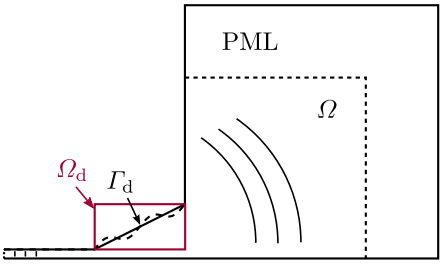

Figure 3. The design domain Ω_d covers the area where the design boundary Γ_d is allowed to take its path. Note that the domain Ω is the same as in Figure 2.

2.2. *Level-set parametrization*

To specify the geometry of the horn, we choose a level-set parametrization, where the design boundary Γ_d is defined as the points where a level-set function $\phi : \Omega_\mathrm{d} \to \mathbb{R}$ vanishes, that is,

$$\Gamma_\mathrm{d} = \{\boldsymbol{x} : \phi(\boldsymbol{x}) = 0\},$$

where $\boldsymbol{x} = (x_1, x_2)$, and $\phi(\boldsymbol{x}) < 0$ implies that $\boldsymbol{x}$ is in the interior of Ω. We assume that the level-set function ϕ is defined in a neighborhood Ω_d around the design boundary, see Figure 3. The initial level-set function gives the initial design, which here is chosen as the straight, solid line Γ_d in Figure 3. The level-set function ϕ is in the set U_ad of admissible functions if ϕ is sufficiently regular and if

$$\begin{cases} \phi > 0 & \text{on the upper and left side of } \partial\Omega_\mathrm{d}, \\ \phi < 0 & \text{on the lower and right side of } \partial\Omega_\mathrm{d}, \text{ and} \\ \phi = 0 & \text{on the lower left and upper right corners of } \partial\Omega_\mathrm{d}. \end{cases} \tag{2.2}$$

These conditions force the design boundary to start at the lower left corner of Ω_d and end at the upper right corner, but allows it to take any path strictly inside Ω_d between those two points.

2.3. *Optimization problem*

Our goal is to optimize the impedance-matching properties of the horn by minimizing the reflection coefficient R, as defined below. The boundary condition on the input boundary Γ_in in problem (2.1) excites a plane wave of unit amplitude and zero phase, and at the same time absorbs reflected plane waves [1]. Thus, the acoustic pressure at Γ_in is simply the sum of the unit input pressure and the reflected pressure. Non-planar modes have negligible influence at frequencies below the cutoff frequency for the first non-planar mode, and are neglected. Consequently, the complex reflection coefficient is defined as

$$R = \frac{1}{|\Gamma_\mathrm{in}|} \int_{\Gamma_\mathrm{in}} (p - 1)\, \mathrm{d}\Gamma, \tag{2.3}$$

and it depends on the wavenumber k as well as on the shape of the design boundary Γ_d, given implicitly by the level-set function ϕ, through the acoustic pressure p at Γ_in. In order to evaluate the

reflection properties in a wavenumber interval $[k_1, k_{N_k}]$, we introduce the objective function

$$J_R(\phi) = \frac{1}{2N_k} \sum_{n=1}^{N_k} |R(\phi, k_n)|^2 , \tag{2.4}$$

which will be minimized for a discrete set of wavenumbers $k_n \in [k_1, k_{N_k}]$.

2.4. *Smoothing and regularization*

Smoothing, or filtering, combined with regularization, is routinely applied in shape optimization to promote smooth designs and counteract mesh dependence, see for example Bängtson et al. [1, § 3] and van Dijk et al. [11, § 6]. Smoothing and regularization may also improve the convergence rate, and can be used to control the feature size of the design [11, 19]. There are several ways to apply smoothing and regularization.

Here, we use an elliptic PDE to promote smooth level-set functions. The approach is similar to the one used in Bängtson et al. [1], but here it is adopted for level-set parametrization. An auxiliary function $\hat{\phi}$ is introduced to indirectly specify the level-set function ϕ. More precisely, ϕ is defined as the solution to the equation

$$\begin{cases} -\nu\Delta\phi + \mu\phi = \hat{\phi} & \text{in } \Omega_\mathrm{d}, \\ \phi = \phi_0 & \text{on } \partial\Omega_\mathrm{d}, \end{cases} \tag{2.5}$$

with $\nu \geq 0$ and $\mu \geq 0$. The set $\hat{U}_\mathrm{ad}$ of admissible functions $\hat{\phi}$ is chosen as the space of square-integrable functions. The set of admissible level-set functions U_ad is thus the solutions ϕ to system (2.5) with the boundary values ϕ_0 satisfying requirement (2.2). For non-zero μ, the quotient ν/μ determines the amount of smoothing. The choice $\nu = 0$ leads to no smoothing, and the choice $\nu > 0$ implies that ϕ by elliptic regularity is in $H^2(\Omega_\mathrm{d}) \subset C^0(\overline{\Omega}_\mathrm{d})$ for smooth or convex Ω_d.

To further control the feature size of the design boundary, we combine the smoothing algorithm with Tikhonov regularization by adding a penalty term to promote solutions with smaller norms. More precisely,

$$J_\epsilon(\hat{\phi}) = \frac{1}{2} \int_{\Omega_\mathrm{d}} \hat{\phi}^2 \, \mathrm{d}\Omega \tag{2.6}$$

is added to objective function (2.4). This gives the total objective function

$$J(\hat{\phi}) = J_R(\phi(\hat{\phi})) + \epsilon_\mathrm{r} J_\epsilon(\hat{\phi}), \tag{2.7}$$

where $\phi(\hat{\phi})$ is the solution of system (2.5). The parameter $\epsilon_\mathrm{r} \geq 0$ determines the amount of regularization. Note that, with $\hat{\phi}$ of the same dimension as one over length, the regularization term J_ϵ is dimensionless.

The optimization problem reads:

$$\begin{aligned} &\text{Find } \hat{\phi}_* \in \hat{U}_\mathrm{ad} \text{ such that} \\ &J(\hat{\phi}_*) \leq J(\hat{\phi}) \quad \forall \hat{\phi} \in \hat{U}_\mathrm{ad}, \end{aligned}$$

with the objective function J given by expression (2.7).

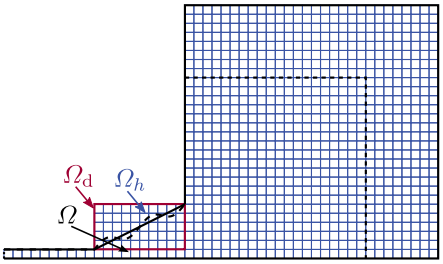

Figure 4. An illustration of the mesh, which is structured, fixed, and independent of the design boundary Γ_d, allowed to cut through the elements. The domain Ω_h equals $\Omega \cup \Omega_\mathrm{d}$, so that it covers the computational domain Ω for all admissible designs. Note that the domains Ω and Ω_d are the same as in Figure 3.

3. DISCRETIZATION

3.1. Finite element discretization with cut elements

The finite element method is used to solve problem (2.1) for the acoustic pressure p, which is needed to determine reflection coefficient (2.3). The geometry is partitioned into a set of square elements, such that the union of elements Ω_h equals the union of Ω and Ω_d; in other words, the domain Ω_h covers the computational domain Ω for all admissible designs, see Figure 4. The boundary is allowed to cut through the elements, so that the mesh is fixed and independent of the design. Thus there is no need for re-meshing or mesh deformation.

Lagrangian elements of first and second order l are used in this paper. The standard nodal basis functions are denoted $w_i(\boldsymbol{x})$, and the space of discrete solutions is $V_h = \mathrm{span}\{w_i\}_{i=1}^N$, where N is the number of nodes. The discretized variational formulation of the governing equations (2.1) is:

$$\begin{aligned} &\text{Find } p_h \in V_h \text{ such that} \\ &A(p_h, q_h) + \epsilon_\mathrm{s} S_h(p_h, q_h) = L(q_h) \quad \forall q_h \in V_h, \end{aligned} \tag{3.1}$$

where

$$A(p_h, q_h) = \int_\Omega \nabla q_h \cdot (\mathbf{G} \nabla p_h) \,\mathrm{d}\Omega - k^2 \int_\Omega q_h \gamma p_h \,\mathrm{d}\Omega + \mathrm{i}k \int_{\Gamma_\mathrm{in}} q_h p_h \,\mathrm{d}\Gamma,$$

$$L(q_h) = 2\mathrm{i}k \int_{\Gamma_\mathrm{in}} q_h \,\mathrm{d}\Gamma,$$

and S_h is a stabilization term defined and explained in the next section, and the parameter $\epsilon_\mathrm{s} \geq 0$ determines the amount of stabilization. The solution p_h can be expanded in the basis functions w_i,

$$p_h(\boldsymbol{x}) = \sum_{i=1}^{N} p_i w_i(\boldsymbol{x}), \tag{3.2}$$

where $p_i = p_h(\boldsymbol{x}_i)$ and $\boldsymbol{x}_i$ is the location of node i. Equation (3.1) uniquely defines the restriction of p_h to Ω, and sets the nodal values p_i for nodes with any neighboring element at least partly inside Ω. For nodes i with all neighboring elements completely outside Ω, the corresponding equation in (3.1) is changed to $p_i = 0$. In this way, the size of the system of equations is independent of the design.

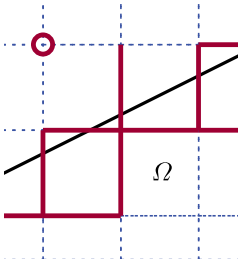

Figure 5. The solution p_i at the node marked with a circle contributes only marginally to the solution inside Ω, since only a fraction of neighboring elements are inside Ω. The stabilization term (3.4) has contributions from the fat, solid edges.

The matrix form of the discretized variational formulation (3.1) is

$$\begin{aligned} \left(\mathbf{K} - k^2\mathbf{M} + \mathrm{i}k\mathbf{M}^{\Gamma_{\mathrm{in}}} + \epsilon_{\mathrm{s}}\mathbf{S}\right)\mathbf{p} &= \mathrm{i}k\mathbf{M}^{\Gamma_{\mathrm{in}}}\mathbf{e}, \\ \mathbf{g}^{\mathrm{T}}\mathbf{p} &= 0 \end{aligned} \tag{3.3}$$

where the matrices have elements

$$\begin{aligned} K_{i,j} &= \int_\Omega \nabla w_j \cdot (\mathbf{G}\nabla w_i)\,\mathrm{d}\Omega, & M_{i,j} &= \int_\Omega w_j \gamma w_i\,\mathrm{d}\Omega, \\ M^{\Gamma_{\mathrm{in}}}_{i,j} &= \int_{\Gamma_{\mathrm{in}}} w_j w_i\,\mathrm{d}\Gamma, & S_{i,j} &= S_h(w_j, w_i), \end{aligned}$$

and the vectors of length N are $\mathbf{p} = [p_1, p_2, \ldots, p_N]$, $\mathbf{e} = [1, 1, \ldots, 1]^{\mathrm{T}}$, and $\mathbf{g} = [g_1, g_2, \ldots, g_N]$. Here $g_i = 1$ if all elements neighboring node i are completely outside Ω, and $g_i = 0$ otherwise.

3.2. *Stabilization*

It is evident that all nodes i belonging to elements that are at least partly inside Ω affect the solution p_h. However, the nodes outside Ω affect the solution very little when only a fraction of neighboring elements are inside Ω, see Figure 5. This causes the system matrix to become ill-conditioned.

The remedy for the ill-conditioning chosen in this paper is the addition of the "ghost penalty" stabilization term

$$S_h(p_h, q_h) = \sum_{F\in\mathcal{F}_S}\sum_{i=1}^{l} h_F^{2i-1}\int_F \left[\!\left[\frac{\partial^i p_h}{\partial n^i}\right]\!\right]_F \left[\!\left[\frac{\partial^i q_h}{\partial n^i}\right]\!\right]_F \mathrm{d}\Gamma \tag{3.4}$$

suggested by Burman [30]. Here, the set $\mathcal{F}_S$ contains all edges $F = \overline{T_1} \cap \overline{T_2}$, shared by two elements T_1 and T_2, such that both neighboring elements T_1 and T_2 are at least partly inside Ω, and at least one of the elements is cut by the boundary, see the illustration in Figure 5. Furthermore, $\partial/\partial n = \hat{\boldsymbol{n}} \cdot \nabla$ is the normal derivative, $\hat{\boldsymbol{n}}$ is the unit normal to the edge (arbitrary but fixed direction), and h_F is the length of the edge. The jump of function values across the edge F is denoted $[\![\cdot]\!]_F$, that is,

$$\left[\!\left[\frac{\partial^i p_h}{\partial n^i}\right]\!\right]_F = \frac{\partial^i p_{T_1}|_F}{\partial n^i} - \frac{\partial^i p_{T_2}|_F}{\partial n^i},$$

where p_{T_1} and p_{T_2} denotes the restriction of p_h to the neighboring elements T_1 and T_2, respectively. Note that the stabilized formulation (3.1) is consistent with problem (2.1) since the jump term vanishes for the exact solution.

3.3. Discrete level set, smoothing, and regularization

The level-set function is discretized using bilinear elements on the mesh that is also used for the finite element analysis. The discretized level-set function is denoted ϕ_h, and likewise the discretization of the auxiliary function $\hat{\phi}$ is denoted $\hat{\phi}_h$. Furthermore, the discretized design boundary Γ_{d}^h is defined as a linear interpolant, whose interpolation nodes are located at the points where the zero contour of the level-set function ϕ_h intersects the mesh edges. This approximation is convenient in the implementation of the cut finite element method [29].

The smoothing and regularization introduced previously for the continuous problem are straightforward to transfer to the discrete setting. Discretization of equations (2.5) for the smoothing yields the linear equations

$$\left(\nu \mathbf{K}_{II}^{\mathrm{d}} + \mu \mathbf{M}_{II}^{\mathrm{d}}\right) \boldsymbol{\phi}_I = \mathbf{M}_{II}^{\mathrm{d}} \hat{\boldsymbol{\phi}}_I - \left(\nu \mathbf{K}_{IB}^{\mathrm{d}} + \mu \mathbf{M}_{IB}^{\mathrm{d}}\right) \boldsymbol{\phi}_B, \tag{3.5}$$

where $\boldsymbol{\phi} = [\phi_1, \phi_2, \ldots, \phi_{N^{\mathrm{d}}}]$ is the vector of level-set nodal values in Ω_{d} and N^{d} is the number of nodes in Ω_{d}. The similar notation applies for $\hat{\boldsymbol{\phi}}$. The matrices $\mathbf{K}^{\mathrm{d}}$ and $\mathbf{M}^{\mathrm{d}}$ are defined as $\mathbf{K}$ and $\mathbf{M}$ in system (3.3), with the exception that the integrals are taken over Ω_{d} instead of Ω (and thus there are no cut elements here). The index I denotes nodes in the interior of Ω_{d}, while B denotes nodes on $\partial\Omega_{\mathrm{d}}$. The term with $\boldsymbol{\phi}_B$ in the right hand side is due to the Dirichlet boundary conditions in system (2.5). The discrete Tikhonov regularization follows directly from equation (2.6) and is

$$J_\epsilon^h(\hat{\boldsymbol{\phi}}_I) = \frac{1}{2} \hat{\boldsymbol{\phi}}_I^{\mathrm{T}} \mathbf{M}_{II}^{\mathrm{d}} \hat{\boldsymbol{\phi}}_I. \tag{3.6}$$

3.4. Discrete optimization problem

Objective function (2.4) and reflection coefficient (2.3) are conveniently determined by evaluating the solution p_h at the input boundary Γ_{in}, which gives

$$R_h = \frac{1}{|\Gamma_{\mathrm{in}}|} \int_{\Gamma_{\mathrm{in}}} (p_h - 1)\, \mathrm{d}\Gamma = \frac{1}{|\Gamma_{\mathrm{in}}|} \mathbf{e}^{\mathrm{T}} \mathbf{M}^{\Gamma_{\mathrm{in}}} \mathbf{p} - 1. \tag{3.7}$$

The discrete optimization problem can then be formulated as follows:

$$\begin{aligned} &\text{Find } \hat{\boldsymbol{\phi}}_I^* \in \mathbb{R}^{N_I^{\mathrm{d}}} \text{ such that} \\ &J^h(\boldsymbol{\phi}_I(\hat{\boldsymbol{\phi}}_I^*)) \leq J^h(\hat{\boldsymbol{\phi}}_I) \quad \forall \hat{\boldsymbol{\phi}}_I \in \mathbb{R}^{N_I^{\mathrm{d}}}, \end{aligned} \tag{3.8}$$

where

$$J^h(\hat{\boldsymbol{\phi}}_I) = J_R^h(\boldsymbol{\phi}_I(\hat{\boldsymbol{\phi}}_I)) + \epsilon_{\mathrm{r}} J_\epsilon^h(\hat{\boldsymbol{\phi}}_I). \tag{3.9}$$

Here N_I^{d} is the number of design nodes, that is, nodes in the interior of the design domain Ω_{d}, see Figure 6; the values $\hat{\phi}_i$ at those nodes are the design variables. Furthermore, J_ϵ^h is given by

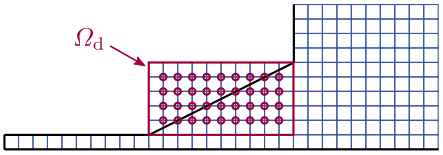

Figure 6. The inner nodes of the design domain Ω_d, marked by circles, are design nodes: the values $\hat{\phi}_i$ at these nodes are the design variables in optimization problem (3.8).

expression (3.6), and the discrete objective function J_R^h corresponding to J_R is obtained by simply exchanging R for R_h in definition (2.4).

4. SENSITIVITY ANALYSIS

We employ a gradient-based optimization method to solve problem (3.8) and will thus need to calculate derivatives of the function $\hat{\boldsymbol{\phi}}_I \mapsto J^h$. This function is a composite mapping: the design variables $\hat{\boldsymbol{\phi}}_I$ are mapped to the vector $\boldsymbol{\phi}_I$ of level-set nodal values through the solution of equation (3.5), which implicitly defines the design boundary Γ_d^h. The boundary, on the other hand, affects the solution to state equation (3.1), which in turn determines the value of the objective function. The conceptually central part of this mapping is the mapping of the level-set function ϕ_h to the reflection coefficient R_h. The sensitivity analysis of this mapping can be worked out using standard tools of shape sensitivity analysis, as shown in Appendix A. However, there are two aspects of this analysis that are not entirely standard and therefore worth pointing out. First, by exploiting the structure of the problem, the shape derivative can be expressed, as shown in Appendix A, solely as a function of the solution to state equation (3.1), without the need of a separate solution of an adjoint problem.

Second, the fact that the finite element solution p_h is defined on a mesh that is fixed implies that the shape derivative of the reflection coefficient R_h can be exactly expressed, up to round-off, as a boundary integral over the design boundary. This is in contrast to the case when a mesh deformation algorithm is employed. Then the exact shape derivative of R_h can only be expressed as a domain integral that involves the mesh deformation mapping, and the use of the boundary-integral expression will introduce a consistency error [36].

To see why this difference occur, consider a given level-set function ϕ_h and perturb its mth nodal value to obtain the family

$$\phi_{h,t} = \phi_h + t w_m \tag{4.1}$$

of level-set functions, parametrized by $t \geq 0$[†], which induces a family of perturbed domains Ω_t. Let $p_{h,t}$ denote the solution to state equation (3.1) where Ω has been replaced by Ω_t. The *shape derivative* of $p_h (= p_{h,0})$ is defined for each $\boldsymbol{x} \in \Omega_h$ as

$$\delta p_h(\boldsymbol{x}) = \lim_{t \to 0^+} \frac{p_{h,t}(\boldsymbol{x}) - p_h(\boldsymbol{x})}{t}. \tag{4.2}$$

[†]The dependency of $\phi_{h,t}$ on m has been suppressed for simplicity of notation.

Now, note that $p_{h,t}$, for each t, can be written as in expression (3.2) as an expansion in a *fixed* set of basis function that do not depend on the position of the design boundary. Consequently, the shape derivative can likewise be written in the form $\delta p_h(\boldsymbol{x}) = \sum_{i=1}^{N} \delta p_i w_i(\boldsymbol{x})$. That is, the shape derivative belongs to the same space V_h as p_h.

Let us compare with the standard case of mesh deformation. Since the mesh moves with the deformation, also the basis functions will depend on deformation parameter t. However, the shape derivative (4.2) is defined as a limit in t for each *fixed* $\boldsymbol{x}$, which means that $p_{h,t}(\boldsymbol{x})$ will sweep over different parts of the basis functions as t is changed. As a consequence, the shape derivative does *not* belong to V_h. The calculations carried out in Appendix A requires that the shape derivative belongs to V_h, and the sensitivity analysis needs to be carried out in a different way in the case of mesh deformations [36].

The expression for the derivative of the reflection coefficient R_h with respect to the level set ϕ_h, derived in Appendix A, is

$$\frac{\partial R_h}{\partial \phi_m} = \frac{1}{2\mathrm{i}k|\Gamma_{\mathrm{in}}|}\left[k^2 \int_{\Gamma_{\mathrm{d}}^h} \hat{\boldsymbol{n}} \cdot \boldsymbol{V}_m p_h^2 \,\mathrm{d}\Gamma - \int_{\Gamma_{\mathrm{d}}^h} \hat{\boldsymbol{n}} \cdot \boldsymbol{V}_m \nabla p_h \cdot \nabla p_h \,\mathrm{d}\Gamma\right], \tag{4.3}$$

where $\hat{\boldsymbol{n}}$ is the outwards directed unit normal to Γ_{d}^h and $\boldsymbol{V}_m$ is the velocity field associated with the domain deformation due to perturbation (4.1) of the level-set function ϕ_h. The velocity field $\boldsymbol{V}_m$ is given by expression (A.6) and extended as described. In matrix form, the derivative is

$$\frac{\partial R_h}{\partial \phi_m} = \frac{1}{2\mathrm{i}k|\Gamma_{\mathrm{in}}|}\mathbf{p}^{\mathrm{T}}\left(k^2\frac{\partial \mathbf{M}^{\Gamma_{\mathrm{d}}}}{\partial \phi_m} - \frac{\partial \mathbf{K}^{\Gamma_{\mathrm{d}}}}{\partial \phi_m}\right)\mathbf{p},$$

where the matrices have elements

$$\frac{\partial K_{i,j}^{\Gamma_{\mathrm{d}}}}{\partial \phi_m} = \int_{\Gamma_{\mathrm{d}}} \hat{\boldsymbol{n}} \cdot \boldsymbol{V}_m \nabla w_j \cdot \nabla w_i \,\mathrm{d}\Gamma, \quad \frac{\partial M_{i,j}^{\Gamma_{\mathrm{d}}}}{\partial \phi_m} = \int_{\Gamma_{\mathrm{d}}} \hat{\boldsymbol{n}} \cdot \boldsymbol{V}_m w_j w_i \,\mathrm{d}\Gamma.$$

The derivative of the objective function J^h with respect to the design variables $\hat{\boldsymbol{\phi}}_I = [\hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_{N_I^{\mathrm{d}}}]$ is given by

$$\frac{\mathrm{d}J^h}{\mathrm{d}\hat{\boldsymbol{\phi}}_I} = \frac{\mathrm{d}\boldsymbol{\phi}_I}{\mathrm{d}\hat{\boldsymbol{\phi}}_I}\frac{\mathrm{d}J_R^h}{\mathrm{d}\boldsymbol{\phi}_I} + \epsilon_{\mathrm{r}}\frac{\mathrm{d}J_\epsilon^h}{\mathrm{d}\hat{\boldsymbol{\phi}}_I}, \tag{4.4}$$

where

$$\frac{\mathrm{d}J_R^h}{\mathrm{d}\boldsymbol{\phi}_I} = \frac{1}{N_k}\sum_{n=1}^{N_k} \mathrm{Re}\left[R_h(\boldsymbol{\phi}_I, k_n)\frac{\mathrm{d}R_h(\boldsymbol{\phi}_I, k_n)}{\mathrm{d}\boldsymbol{\phi}_I}\right],$$

and $\mathrm{d}R_h/\,\mathrm{d}\boldsymbol{\phi}_I$ is the vector with elements $\partial R_h/\partial\phi_m$. The matrix $\mathrm{d}\boldsymbol{\phi}_I/\,\mathrm{d}\hat{\boldsymbol{\phi}}_I$ and the vector $\mathrm{d}J_\epsilon^h/\,\mathrm{d}\hat{\boldsymbol{\phi}}_I$ follow directly from formula (3.5) and formula (3.6), respectively.

The derivative expressions (4.3) and (4.4) have been compared to finite difference approximations with excellent agreement. First we consider the case without smoothing or regularization. The mth nodal value of the discrete level set function ϕ_h is perturbed by t according to equation (4.1), and the acoustic problem is solved on the perturbed domain Ω_t to compute the perturbed reflection coefficient $R_{h,t}$ and objective function $J^{h,t}$, respectively. Such perturbations are then

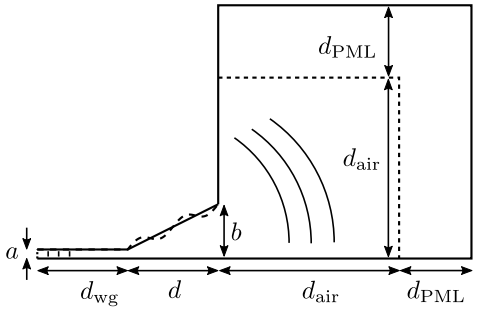

Figure 7. Definitions of the horn dimensions.

used to compute the central finite difference approximations

$$\frac{\partial J^h}{\partial \phi_m} \approx \frac{J^{h,t} - J^{h,-t}}{2t}, \quad \frac{\partial R_h}{\partial \phi_m} \approx \frac{R_{h,t} - R_{h,-t}}{2t}.$$

The same procedure is repeated for all nodal values ϕ_m inside the design domain Ω_{d}. With a perturbation $t \approx 10^{-6}$, the maximum absolute relative difference between the finite difference approximations and the derivatives computed from expressions (4.3) and (4.4) is approximately $5 \cdot 10^{-8}$. The maximum is taken over all level set nodal values ϕ_m, and also over all wavenumbers k_n for $\partial R_h / \partial \phi_m$. This accuracy is as good as can be expected in the context of floating-point numbers, since the errors are only slightly higher than the square root of the machine precision $\sqrt{\texttt{eps}} \approx 1.5 \cdot 10^{-8}$. The same accuracy test is also carried out when including the smoothing (2.5) and the regularization term (2.6), with no negative effect on the accuracy. Note that in the discrete case, the smoothing operation is given by the solution of the linear equation (3.5), and that the regularization term (3.6) is a quadratic form. Corresponding exact derivative expressions are therefore straightforward to compute.

5. RESULTS

We consider an acoustic horn with dimensions as in Bängtsson et al. [1] and Wadbro and Berggren [2], so that the results can be compared, see Figure 7 and Table I. The lower limit of the frequency interval is chosen to be 200 Hz, since optimization below this frequency does not seem feasible. This is not surprising, since the horn dimensions then are small compared to the wavelength. With the speed of sound $c = 340\,\mathrm{m/s}$, the wavelength at 200 Hz is $\lambda = 1.7\,\mathrm{m}$, which gives horn length over wavelength $d/\lambda \approx 0.29$ and horn width over wavelength $2b/\lambda \approx 0.35$. The upper limit is chosen to be 1600 Hz, which is three octaves above the lower limit, and at the same time well below the cutoff frequency for non-planar modes in the waveguide. With the width of the waveguide $2a = 0.1\,\mathrm{m}$, the cutoff frequency for the first non-planar mode is $c/(4a) = 1700\,\mathrm{Hz}$; at 1600 Hz, this mode decays with a factor $\approx 8 \cdot 10^{-3}$ over the length of the waveguide $d_{\mathrm{wg}} = 0.5\,\mathrm{m}$ [1].

The mesh size is chosen fine enough to allow a detailed design boundary; recall that the mesh used for the solution of discretized variational form (3.1) is also used for the level-set function, which implicitly describes the design boundary Γ_{d}^h. Setting the side length of the square mesh cells

a	b	d	d_{wg}	d_{air}	d_{PML}
0.05 m	0.3 m	0.5 m	0.5 m	1.0 m	0.4 m

Table I. Values for the horn dimensions, defined in Figure 7, used in the numerical examples.

to $h = a/3 \approx 0.0167\,\mathrm{m}$ means that the design domain Ω_{d} is discretized by 30×15 elements, and consequently that there are $29 \times 14 = 406$ design variables (the auxiliary function nodal values $\hat{\phi}_i$ at the inner nodes of the design domain Ω_{d}). This mesh resolution should be more than enough to accurately solve the acoustic problem, since the number of elements per wavelength is 12.75 at 1600 Hz. Biquadratic elements are used for the solution of acoustic problem (3.1). The stabilization parameter ϵ_{s} is set to 0.0025.

The parameters of the perfectly matched layer (PML) are tuned to create a low reflection. We use a matrix $\mathbf{G}$ and a constant γ, which perform the damping of the outgoing waves in the PML, according to formulas (6)–(8) in Heikkola et al. [37]. The choice of parameters $\sigma_0 = 20$ and $p = 2$ in those formulas are found to work well for the frequency interval $[200, 1600]$ Hz, when the depth of the PML is $d_{\mathrm{PML}} = 0.4\,\mathrm{m}$. To estimate the effect of the waves reflected back due to the truncated computational domain, the reflection coefficient R_h of the original horn design is calculated with varying distances to the PML (given by the side length d_{air} of the surrounding air box, see Figure 7). The reflections affect the complex reflection coefficient R_h differently depending on the distance, but it is found that the magnitude of the maximum difference is below 10^{-5}.

The implementation is carried out in Matlab. The system matrices are calculated using tensor product Gaussian quadrature over the elements (or edges for $\mathbf{M}^{\Gamma_{\mathrm{in}}}$), as is the line integral needed for derivatives (4.3), and the integrals over the edges to compute stabilization term (3.4). The Levenberg–Marquardt algorithm, as implemented in the Matlab function lsqnonlin, is employed to solve the optimization problem. This algorithm utilizes the fact that problem (3.8) can be formulated as a least squares problem with objective function

$$J^h(\hat{\boldsymbol{\phi}}_I) = \frac{1}{2N_k} \sum_{n=1}^{N_k} \left[(\mathrm{Re}\, R_h(\hat{\boldsymbol{\phi}}_I, k_n))^2 + (\mathrm{Im}\, R_h(\hat{\boldsymbol{\phi}}_I, k_n))^2 \right] + \left[\sqrt{\epsilon_{\mathrm{r}} J^h_\epsilon(\hat{\boldsymbol{\phi}}_I)} \right]^2,$$

where N_k is the number of frequencies $f_n = ck_n/2\pi$ sampled in the interval $[200, 1600]$ Hz.

5.1. Optimization results

The results presented here were obtained using $N_k = 37$ frequency points, logarithmically placed in the frequency interval $[200, 1600]$ Hz, for objective function (3.9). Figure 8 shows the original and optimized horn designs, for optimization without smoothing, with smoothing, and with both smoothing and regularization, and Figure 9 shows the magnitude of the reflection coefficient R_h in the frequency interval $[200, 1600]$ Hz for the horns.

The optimization without smoothing is successful in the sense that the magnitude of the reflection coefficient R_h is small in the whole frequency interval. However, the resulting design is jagged and not smooth (see the upper mid horn in Figure 8). Furthermore, as will be shown below, the optimization result is mesh-dependent: changing the resolution of the mesh alters the design significantly.

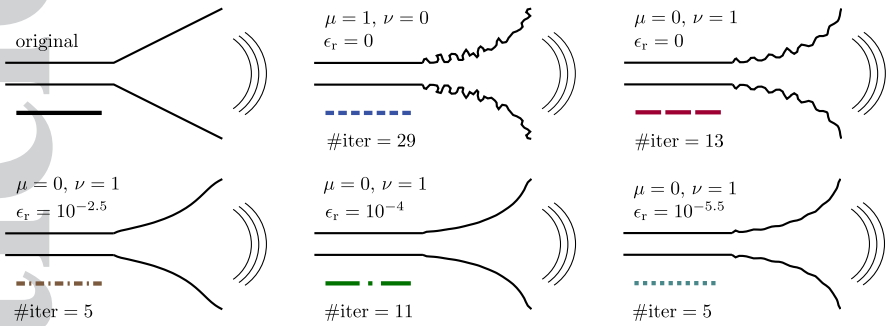

Figure 8. The original and optimized horn designs, and the number of iterations needed. The parameters μ and ν determines smoothing (2.5); recall that the choice $\nu = 0$ leads to no smoothing. The parameter ϵ_r determines the amount of regularization in objective function (3.9).

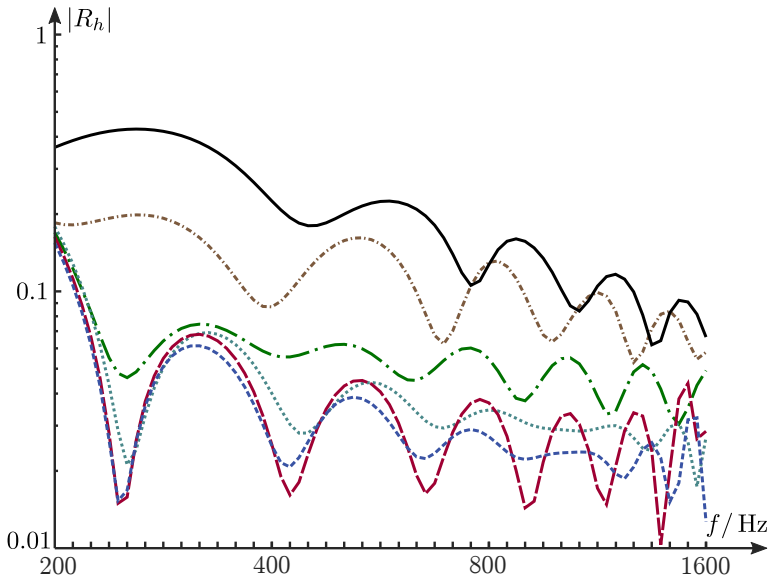

Figure 9. The magnitude of the reflection coefficient R_h for the original and optimized horn designs in Figure 8 (where the line styles can also be found). The 37 target frequencies in the optimization are marked by ticks on the frequency axis.

The optimization using smoothing (2.5), with parameters $\nu = 1$ and $\mu = 0$, results in a horn that looks similar to the horn optimized without smoothing, but which is less jagged and more rounded (see the upper right horn in Figure 8). As shown later, it is also less affected by changing the mesh resolution. Note that the magnitude of the reflection coefficient is comparable to that of the horn optimized without smoothing. The optimization results for smoothing with parameters ν and μ both different from zero are somewhere between the case $\nu = 1, \mu = 0$ and the other extreme case $\nu = 0, \mu = 1$ (no smoothing), and are not shown here.

With Tikhonov regularization, the shape of the optimized horn depends strongly on the amount of regularization, given by the parameter ϵ_r in objective function (3.9). Not surprisingly, using a small value ($\epsilon_r = 10^{-5.5}$) gives a horn design similar in shape and reflection coefficient values to the one obtained without regularization (see the lower right horn in Figure 8). Using a higher value ($\epsilon_r = 10^{-2.5}$), on the other hand, gives a design similar to the original design (see the lower left horn in Figure 8). Since the smoothing parameters $\nu = 1$ and $\mu = 0$ are used, the regularization

penalizes designs with $\hat{\phi} = \Delta\phi \neq 0$ in the design domain Ω_d. The original design, where the design boundary Γ_d^h is a straight line, is given by a harmonic level-set function, with $\Delta\phi = 0$, and consequently the regularization penalizes deviations from the starting design in this case. Finally, using an intermediate value ($\epsilon_\mathrm{r} = 10^{-4}$), the resulting horn design is radically different from the design obtained without regularization, with slightly larger, albeit small, magnitude of the reflection coefficient R_h (see the lower mid horn in Figure 8).

The results are somewhat surprising: optimization without or with little regularization leads to horns with quasi-periodic sub-wavelength features that have not been seen in previous analyses [1]. It seems like the implicit level-set parametrization used here captures these features more easily than the explicit boundary representation used before. The horn obtained with little regularization ($\epsilon_\mathrm{r} = 10^{-5.5}$) has less distinct sub-wavelength features, but almost identical reflection coefficient. However, it seems disadvantageous to completely remove the features, since the horn obtained with intermediate regularization ($\epsilon_\mathrm{r} = 10^{-4}$), which lacks the features, has slightly higher reflection.

The number of iterations needed for the optimizations can also be found in Figure 8. We see that only a few iterations are needed when using smoothing, and that more iterations are needed when not using smoothing. The Levenberg–Marquardt algorithm is set to terminate when the infinity norm of the discrete gradient, $|\,\mathrm{d}J^h/\,\mathrm{d}\hat{\boldsymbol{\phi}}_I|_\infty$, is less than 10^{-8} times its initial value $|\,\mathrm{d}J_0^h/\,\mathrm{d}\hat{\boldsymbol{\phi}}_I|_\infty$. The algorithm terminates also if the absolute relative change in the objective function between two iterations is less than 10^{-4}.

To examine the effect of mesh resolution on the optimization results, the optimization procedure has been carried out also on finer meshes. The results when the side lengths of the square mesh cells are $h = a/3 \approx 0.0167\,\mathrm{m}$ (first row, also included in Figure 8), $h = a/4 = 0.0125\,\mathrm{m}$ (second row), and $h = a/5 = 0.01\,\mathrm{m}$ (third row), respectively are presented in Figure 10. The corresponding number of design variables are $29 \times 14 = 406$, $39 \times 19 = 741$, and $49 \times 24 = 1176$, respectively. For the horns optimized without smoothing (first column), it is seen that finer details emerge as the mesh is refined. The horn optimized with smoothing but without regularization (second column) are similar on the two fine meshes, but the first horn is missing some details. The designs obtained with smoothing and regularization using $\epsilon_\mathrm{r} = 10^{-4}$ are very similar. In conclusion, it seems like the optimization results using smoothing are similar if the mesh is fine enough to resolve the details in the design, and adding regularization removes many small details.

For verification purposes, we import the optimized geometries into Comsol Multiphysics, create a body-fitted, unstructured, triangular mesh, and calculate the reflection coefficient for the design optimized using smoothing and no regularization (top right horn in Figure 8) and using smoothing and intermediate regularization (bottom mid horn in Figure 8). In Comsol Multiphysics, the element size setting is “Extremely fine”, and biquadratic elements are used. The absolute difference in the magnitude of the refection coefficient R_h computed with the cut element code and Comsol can be found in Figure 11. The difference for the horn optimized with regularization is below 10^{-4} in the entire frequency band. For the horn optimized without regularization, however, the difference is a bit higher, especially at higher frequencies; the maximum value is $4.6 \cdot 10^{-3}$. The larger difference can be explained by the small features in the design optimized without regularization, which are better resolved by the finer mesh used in Comsol. These small features are not present in the design optimized with regularization.

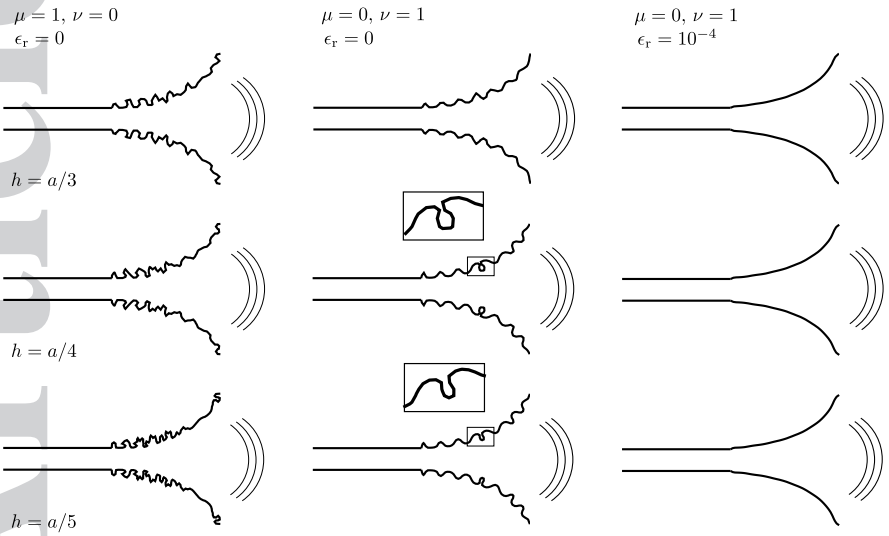

Figure 10. The horn optimized using different mesh resolutions (h denotes the side length of the square elements, and $2a$ is the width of the waveguide). The parameters μ and ν determines smoothing (2.5); recall that the choice $\nu = 0$ leads to no smoothing. The parameter ϵ_r determines the amount of regularization in objective function (3.9).

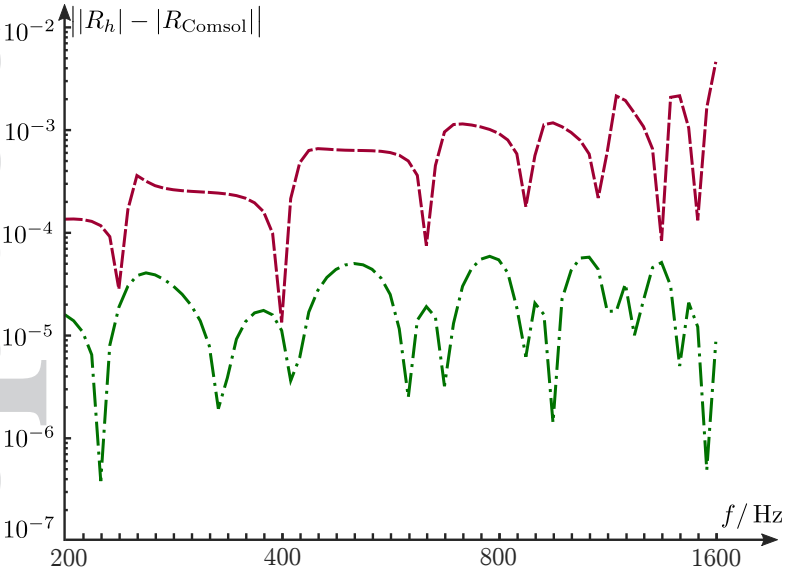

Figure 11. The absolute difference in the magnitude of the refection coefficient R_h for the solution with Comsol Multiphysics and the cut finite element code for the top right and bottom mid horns in Figure 8 (where the line styles can be found).

6. CONCLUSIONS

We presented an approach, based on cut finite elements and a level-set description of the geometry, to optimize the design of acoustic horns with regard to reflection. The cut finite element method neither needs mesh deformation nor re-meshing. Moreover, the method is robust and computationally inexpensive. The implicit level-set geometry description provides a vast design freedom and allows for solutions with high curvature of the design boundary. The optimized horns

contain surprising sub-wavelength features not seen in previous studies [1, 2], and possess excellent impedance-matching properties in the entire frequency band considered.

We found that the discrete shape sensitivity can be expressed exactly as a boundary integral over the design boundary. This is possible thanks to the fact that, due to the fixed mesh, the shape derivatives of the finite element basis functions belong to the same function space as the basis functions themselves. This is in contrast to methods using body-fitted finite elements and mesh deformation, where the shape derivatives of the basis functions are less regular, and the discrete shape sensitivity must be expressed as a domain integral involving the mesh deformation. With a fixed mesh, however, no domain integral or mesh deformation is needed for the discrete shape sensitivity.

We propose smoothing the level-set function by solving Poisson's equation on the design domain, since this treatment was found to counteract mesh-dependent solutions and to improve the convergence rate of the optimization. Furthermore, we propose combining this smoothing with Tikhonov regularization, to penalize small features in the optimized designs.

A benefit of the method we propose is that there is no need to force the level-set function to be a signed distance function. Many level-set methods suffer from slow convergence for highly varying gradients of the level-set function around the design boundary, and one way commonly employed to circumvent this problem is to use costly re-initialization procedures [11]. We observe fast convergence without re-initialization, which is partly thanks to the smoothing of the level-set function, especially when combined with Tikhonov regularization. The fast convergence is also thanks to the Levenberg–Marquardt algorithm employed to update the design, since the algorithm uses approximate second order derivatives and therefore is less sensitive to the scaling of the problem.

However, the amount of regularization added to the objective function must be determined with caution, since too much may result in performance degradation of the optimal solution. For the horn, we noticed that adding enough regularization to completely remove the sub-wavelength features decreased the performance of the optimized design. This observation agrees with previous studies [1, 2] and the fact that the impedance-matching properties of acoustic horns are remarkably sensitive to the exact shape. Even so, there are visually very different solutions with similar performance, which suggests that the optimization problem is underdetermined. Hence, additional constraints, such as on the far field characteristics, could be imposed without affecting the reflection coefficient too much. Finally, it should be mentioned that the benefits of the method, such as the absence of mesh deformation, and the straightforward and computationally inexpensive regularization and gradient computation, can be expected to be even greater in three than in two dimensions.

A. DETAILS ON THE SENSITIVITY ANALYSIS

Here, we show that the derivative of $\phi_h \mapsto R_h$, the mapping from the level-set function to the reflection coefficient, can be expressed as a boundary-integral over the design boundary Γ_{d}^h. As noted in Section 4, this expression will be exact for the present method, thanks to the fact that the shape derivative conforms with the finite element space. That is, the shape derivatives of the

finite element basis functions belong to the same function space as the basis functions themselves. When a mesh deformation is used, however, the lack of regularity of the basis functions [38, Ch. 10, Remark 2.3] implies that the exact derivative of the mapping $\phi_h \mapsto R_h$ can only be given in terms of a domain integral, whose integrand will depend on the mesh deformation [36].

Given a domain Ω, assume that Ω_t, where $\Omega_0 = \Omega$, is a family of deformed domains parametrized by $t \geq 0$. The standard tools of shape sensitivity analysis require that each point $\boldsymbol{x} \in \Omega$ can be mapped into Ω_t by a smooth invertible function $t \mapsto \boldsymbol{x}(t)$, where $\boldsymbol{x}(0) = \boldsymbol{x}$. For each such family of domain deformations, we may associate the "velocity" field

$$\boldsymbol{V}(\boldsymbol{x}) = \lim_{t \to 0^+} \frac{\boldsymbol{x}(t) - \boldsymbol{x}}{t} \stackrel{\text{def}}{=} \frac{\mathrm{d}^+}{\mathrm{d}t} \boldsymbol{x}(t)\big|_{t=0},$$

which is a quantity representing the domain deformation that will appear in the formulas below. Let ψ_t be a family of functions defined on Ω_t and let the shape derivative of $\psi = \psi_0$ be defined by

$$\delta\psi(\boldsymbol{x}) = \frac{\mathrm{d}^+}{\mathrm{d}t} \psi_t(\boldsymbol{x})\big|_{t=0} \tag{A.1}$$

for each $\boldsymbol{x} \in \Omega$. The shape derivative of the family of integrals $I_t = \int_{\Omega_t} \psi_t \, \mathrm{d}\Omega$ is likewise defined by

$$\delta I = \frac{\mathrm{d}^+}{\mathrm{d}t} I_t\big|_{t=0}. \tag{A.2}$$

The classic formula for the shape derivative (A.2) is

$$\delta I(\boldsymbol{V}) = \int_\Omega \big(\delta\psi + \nabla \cdot (\boldsymbol{V}\psi)\big) \, \mathrm{d}\Omega. \tag{A.3}$$

Delfour & Zolesio [38, Ch. 9, § 4.1] prove this formula under the assumption that ψ, $\delta\psi$, and $\nabla\psi$ are in $L^1(\Omega)$, and that $\boldsymbol{V}$ is in $C^1(\Omega)$.

Let us now see how formula (A.3) can be used in the present framework. We start by considering how a vector field $\boldsymbol{V}$ can be constructed from perturbations of the level-set function ϕ_h. Recall that the discretized design boundary Γ_{d}^h is defined as a linear interpolant, whose interpolation nodes are located at the points where the zero contour of the level-set function ϕ_h intersects the mesh edges. The location of such an interpolation node depends solely on the nodal values of the level-set function on that edge.

Consider a point $\boldsymbol{x}_0$ where the zero contour of the level-set function ϕ_h intersects the mesh. Let $F_{i,j}$ denote an intersected edge such that $\boldsymbol{x}_0 \in F_{i,j}$, and where the values of ϕ_h at the end points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ of the edge satisfy $\phi_i \geq 0$ and $\phi_j < 0$. Note that, in the case $\phi_i = 0$, when the level-set function crosses the mesh vertex $\boldsymbol{x}_i$, we have chosen to consider the edge associated with $\boldsymbol{x}_i$ where the level-set function is negative. Since the level-set function varies linearly along the edges of the mesh, the point $\boldsymbol{x}_0$ is given by

$$\boldsymbol{x}_0(\phi_i, \phi_j) = \frac{\phi_i \boldsymbol{x}_j - \phi_j \boldsymbol{x}_i}{\phi_i - \phi_j}, \quad \text{for } \phi_i \geq 0 \text{ and } \phi_j < 0,$$

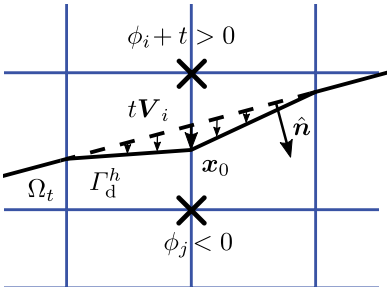

Figure 12. Perturbation (A.5) of the level-set function ϕ_h (for $m = i$), in the limit as $t \to 0^+$ moves the design boundary Γ_{d}^h, given by the zero contour $\phi_h = 0$, by $t\boldsymbol{V}_i$.

which means that

$$\begin{aligned} \frac{\partial}{\partial \phi_j} \boldsymbol{x}_0(\phi_i, \phi_j) &= -\frac{\phi_i}{(\phi_i - \phi_j)^2} (\boldsymbol{x}_i - \boldsymbol{x}_j), \quad \text{for } \phi_i \geq 0 \text{ and } \phi_j < 0, \\ \frac{\partial}{\partial \phi_i} \boldsymbol{x}_0(\phi_i, \phi_j) &= \frac{\phi_j}{(\phi_i - \phi_j)^2} (\boldsymbol{x}_i - \boldsymbol{x}_j), \quad \text{for } \phi_i > 0 \text{ and } \phi_j < 0. \end{aligned} \tag{A.4}$$

Note that these are vectors directed along the edge $F_{i,j}$. In the case $\phi_i = 0$, the function $\phi_i \mapsto \boldsymbol{x}_0$ is only semi differentiable; the formula above gives the correct value for its right derivative, but the left derivative will depend on the nodal value at the end of the other edge associated with the mesh vertex $\boldsymbol{x}_i$.

Now let w_m be a basis function for the level-set functions. Consider a family

$$\phi_{h,t} = \phi_h + t w_m \tag{A.5}$$

of perturbed level-set functions parametrized by $t \geq 0$. Letting $t \mapsto \boldsymbol{x}(t)$ denote the corresponding displacement of an arbitrary point $\boldsymbol{x} \in \Gamma_{\mathrm{d}}^h$, we may define the design boundary velocity function $\hat{\boldsymbol{V}}_m : \Gamma_{\mathrm{d}}^h \to \mathbb{R}^2$ by

$$\hat{\boldsymbol{V}}_m(\boldsymbol{x}) = \frac{\mathrm{d}^+}{\mathrm{d}t} \boldsymbol{x}(t)\big|_{t=0}$$

for each $\boldsymbol{x} \in \Gamma_{\mathrm{d}}^h$. At the intersection with the mesh edges, it holds that

$$\hat{\boldsymbol{V}}_m\big(\boldsymbol{x}_0(\phi_i, \phi_j)\big) = \frac{\partial^+}{\partial \phi_m} \boldsymbol{x}_0(\phi_i, \phi_j), \tag{A.6}$$

where the right side vanishes except when $m = i$ or $m = j$, in which case formulas (A.4) hold. Since the discretized design boundary Γ_{d}^h is given by linear interpolation between the points where the zero contour of the level-set function ϕ_h intersects the mesh, the values of $\hat{\boldsymbol{V}}_m$ are also given by linear interpoolation between the values at the points of intersection, given by expression (A.6), see Figure 12. To utilize formula (A.3), we need to extend the function $\hat{\boldsymbol{V}}_m$ into a function $\boldsymbol{V}_m$ on Ω_h. Since $\hat{\boldsymbol{V}}_m$ is continuous, piecewise linear on Γ_{d}^h, and always directed along the mesh edges, this extension can be defined by linear interpolation between $\hat{\boldsymbol{V}}_m$ and the function that vanishes at all vertices not on Γ_{d}^h.

Consider solutions to variational problem (3.1) defined on the perturbed domains Ω_t, associated with the perturbed level-set functions $\phi_{h,t}$, and define the shape derivative δp_h according to

definition (A.1). A crucial observation is that $p_{h,t} \in V_h$ can be expanded, as in expression (3.2), in a space of basis functions w_i *that do not depend on the perturbation*. Thus, $\delta p_h \in V_h$, which is a property that does not hold when mesh deformations are used, since the basis functions then will be perturbed by the mesh deformation.

Since Γ_{in} is unaffected by the perturbation, the derivative of reflection coefficient (3.7) satisfies

$$\delta R_h = \frac{1}{|\Gamma_{\text{in}}|} \int_{\Gamma_{\text{in}}} \delta p_h \, \mathrm{d}\Gamma = \frac{1}{2\mathrm{i}k|\Gamma_{\text{in}}|} L(\delta p_h), \tag{A.7}$$

where L is the right-hand side linear form in state equation (3.1). To determine how $L(\delta p_h)$ depends on the level-set perturbation, we differentiate state equation (3.1) and choose $q_h = p_h$. First, consider $\epsilon_{\text{s}} = 0$; the case $\epsilon_{\text{s}} > 0$ will be covered later. We write all integrals in the state equation as a sum over all mesh elements intersected with the domain. Differentiating this expression and using formula (A.3), noting that the restriction of both p_h and $\boldsymbol{V}_m$ on each such intersection are polynomials, we obtain

$$\begin{aligned} 0 &= \delta A(p_h, p_h) - \delta L(p_h) \\ &= 2 \sum_{T \in \mathcal{T}_h} \left[\int_{T \cap \Omega} \nabla p_h \cdot \mathbf{G} \nabla \delta p_h \, \mathrm{d}\Omega - k^2 \int_{T \cap \Omega} p_h \gamma \, \delta p_h \, \mathrm{d}\Omega \right] - \delta L(p_h) \\ &\quad + \sum_{T \in \mathcal{T}_h} \left[\int_{T \cap \Omega} \nabla \cdot \left(\boldsymbol{V}_m (\nabla p_h \cdot \mathbf{G} \nabla p_h) \right) \mathrm{d}\Omega - k^2 \int_{T \cap \Omega} \nabla \cdot \left(\boldsymbol{V}_m p_h \gamma p_h \right) \mathrm{d}\Omega \right], \end{aligned} \tag{A.8}$$

where in the second equality, we have used the product rule and that shape differentiation commutes with the gradient operator. Since $\delta p_h \in V_h$ and p_h satisfies variational problem (3.1), we find that the first terms on the right side of expression (A.8) can be written

$$\begin{aligned} &2 \sum_{T \in \mathcal{T}_h} \left[\int_{T \cap \Omega} \nabla p_h \cdot \mathbf{G} \nabla \delta p_h \, \mathrm{d}\Omega - k^2 \int_{T \cap \Omega} p_h \gamma \, \delta p_h \, \mathrm{d}\Omega \right] - \delta L(p_h) \\ &\qquad = 2A(p_h, \delta p_h) - L(\delta p_h) = L(\delta p_h). \end{aligned} \tag{A.9}$$

Note that $T \cap \Omega$ is a polygon, whose boundary segments either belong to an edge of the mesh or to Γ_{d}^h. Thus, the divergence theorem implies that the last two terms in expression (A.8) can be written

$$\begin{aligned} &\sum_{T \in \mathcal{T}_h} \left[\int_{T \cap \Omega} \nabla \cdot \left(\boldsymbol{V}_m (\nabla p_h \cdot \mathbf{G} \nabla p_h) \right) \mathrm{d}\Omega - k^2 \int_{T \cap \Omega} \nabla \cdot \left(\boldsymbol{V}_m p_h \gamma p_h \right) \mathrm{d}\Omega \right] \\ &\qquad = \int_{\Gamma_{\text{d}}^h} \hat{\boldsymbol{n}} \cdot \boldsymbol{V}_m \nabla p_h \cdot \nabla p_h \, \mathrm{d}\Gamma - k^2 \int_{\Gamma_{\text{d}}^h} \hat{\boldsymbol{n}} \cdot \boldsymbol{V}_m p_h^2 \, \mathrm{d}\Gamma, \end{aligned} \tag{A.10}$$

where the contributions from the edges of the mesh that do not belong to Γ_{d}^h vanish, since the velocity field is tangent to or vanish on each such mesh edge. In expression (A.10), we have also used that $\gamma = 1$ and $\mathbf{G} = \mathbf{I}$ on Γ_{d}^h.

By substituting expressions (A.9) and (A.10) into the differentiated state equation (A.8), we find that

$$L(\delta p_h) + \int_{\Gamma_{\text{d}}^h} \hat{\boldsymbol{n}} \cdot \boldsymbol{V}_m \nabla p_h \cdot \nabla p_h \, \mathrm{d}\Gamma - k^2 \int_{\Gamma_{\text{d}}^h} \hat{\boldsymbol{n}} \cdot \boldsymbol{V}_m p_h^2 \, \mathrm{d}\Gamma = 0,$$

which by expression (A.7) implies that

$$\delta R_h = \frac{1}{2\mathrm{i}k|\Gamma_{\mathrm{in}}|}\left[k^2\int_{\Gamma_{\mathrm{d}}^h}\hat{\boldsymbol{n}}\cdot\boldsymbol{V}_m p_h^2\,\mathrm{d}\Gamma - \int_{\Gamma_{\mathrm{d}}^h}\hat{\boldsymbol{n}}\cdot\boldsymbol{V}_m\nabla p_h\cdot\nabla p_h\,\mathrm{d}\Gamma\right]. \tag{A.11}$$

As long as Γ_{d}^h does not cross any mesh node, the inclusion of a nonzero stabilization term $\epsilon_{\mathrm{s}}S_h(p_h,q_h)$ in state equation (3.1) does not change expression (A.11). The reason is that the set of edges $\mathcal{F}_S$ in stabilization (3.4) in this case is unchanged by perturbation (A.5). Consequently, the shape derivative of stabilization term (3.4) is $\delta S_h(p_h,p_h) = 2S_h(p_h,\delta p_h)$, and since $\delta p_h \in V_h$ it follows that $2A(p_h,\delta p_h) + 2\epsilon_{\mathrm{s}}S_h(p_h,\delta p_h) - L(\delta p_h) = L(\delta p_h)$ and (A.11) follows. If, however, Γ_{d}^h crosses a mesh node $\boldsymbol{x}_i$, the function $\phi_i \mapsto S_h(p_h,p_h)$ will not be continuous at zero, since mesh edges will be added to the set of stabilization edges $\mathcal{F}_S$ as ϕ_i is perturbed from zero. We expect the effect of the nondifferentiability of S_h to be insignificant, and use equation (A.11) also in this special case.

ACKNOWLEDGEMENT

The computations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at the High Performance Computing Center North (HPC2N).

REFERENCES

1. Bängtsson E, Noreland D, Berggren M. Shape optimization of an acoustic horn. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**(11–12):1533–1571, .
2. Wadbro E, Berggren M. Topology optimization of an acoustic horn. *Computer Methods in Applied Mechanics and Engineering* 2006; **196**(1–3):420–436, .
3. Wadbro E, Udawalpola R, Berggren M. Shape and topology optimization of an acoustic horn–lens combination. *Journal of Computational and Applied Mathematics* 2010; **234**(6):1781–1787, .
4. Noreland D, Udawalpola R, Seoane P, Wadbro E, Berggren M. An efficient loudspeaker horn designed by numerical optimization: an experimental study. *Technical Report UMINF 10.1*, Umeå University, Department of Computing Science, SE-901 87 Umeå, Sweden 2010. `http://www.cs.umu.se`.
5. Farhadinia B. Structural optimization of an acoustic horn. *Applied Mathematical Modelling* 2012; **36**(5):2017–2030, .
6. Barbieri R, Barbieri N. Acoustic horns optimization using finite elements and genetic algorithm. *Applied Acoustics* 2013; **74**(3):356–363, .
7. Barbieri R, Barbieri N, de Lima KF. Some applications of the PSO for optimization of acoustic filters. *Applied Acoustics* 2015; **89**:62–70, .
8. Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics* 1988; **79**(1):12–49, .
9. Sethian J, Wiegmann A. Structural boundary design via level set and immersed interface methods. *Journal of Computational Physics* 2000; **163**(2):489–528, .
10. Burger M, Osher SJ. A survey on level set methods for inverse problems and optimal design. *European journal of applied mathematics* 2005; **16**(02):263–301.
11. van Dijk NP, Maute K, Langelaar M, van Keulen F. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization* 2013; **48**(3):437–472, .
12. Gain AL, Paulino GH. A critical comparative assessment of differential equation-driven methods for structural topology optimization. *Structural and Multidisciplinary Optimization* 2013; **48**(4):685–710, .
13. Wang MY, Wang X, Guo D. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**(12):227–246, .

14. Belytschko T, Xiao SP, Parimi C. Topology optimization with implicit functions and regularization. *International Journal for Numerical Methods in Engineering* 2003; **57**(8):1177–1196, .
15. Allaire G, Jouve F, Toader AM. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics* 2004; **194**(1):363–393, .
16. Van Miegroet L, Duysinx P. Stress concentration minimization of 2D filets using X-FEM and level set description. *Structural and Multidisciplinary Optimization* 2007; **33**(4):425–438, .
17. Wei P, Wang MY, Xing X. A study on X-FEM in continuum structural optimization using a level set model. *Computer-Aided Design* 2010; **42**(8):708–719, .
18. Li L, Wang MY, Wei P. XFEM schemes for level set based structural optimization. *Frontiers of Mechanical Engineering* 2012; **7**(4):335–356, .
19. Makhija D, Maute K. Numerical instabilities in level set topology optimization with the extended finite element method. *Structural and Multidisciplinary Optimization* 2014; **49**(2):185–197, .
20. Villanueva CH, Maute K. Density and level set-XFEM schemes for topology optimization of 3-D structures. *Computational Mechanics* 2014; **54**(1):133–150, .
21. Wang Y, Luo Z, Kang Z, Zhang N. A multi-material level set-based topology and shape optimization method. *Computer Methods in Applied Mechanics and Engineering* 2015; **283**:1570–1586, .
22. Liu P, Luo Y, Kang Z. Multi-material topology optimization considering interface behavior via XFEM and level set method. *Computer Methods in Applied Mechanics and Engineering* 2016; **308**:113–133, .
23. Noël L, Duysinx P. Shape optimization of microstructural designs subject to local stress constraints within an XFEM-level set framework. *Structural and Multidisciplinary Optimization* 2016; **Online**:1–16, .
24. Kreissl S, Maute K. Levelset based fluid topology optimization using the extended finite element method. *Structural and Multidisciplinary Optimization* 2012; **46**(3):311–326, .
25. Villanueva H, Maute K. CutFEM topology optimization of 3D laminar incompressible flow problems 2017. ArXiv:1702.02473v1.
26. Avram A, Topa V, Purcar M, Munteanu C. Numerical optimization of an electrostatic device based on the 3D XFEM and genetic algorithm. *49th International Universities Power Engineering Conference (UPEC)*, 2014; 1–5, .
27. Jeong S, Lim S, Min S. Level-set-based topology optimization using remeshing techniques for magnetic actuator design. *IEEE Transactions on Magnetics* Mar 2016; **52**(3):1–4, .
28. Fries TP, Belytschko T. The extended/generalized finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering* 2010; **84**(3):253–304, .
29. Burman E, Claus S, Hansbo P, Larson MG, Massing A. CutFEM: Discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering* 2015; **104**(7):472–501, .
30. Burman E. Ghost penalty. *Comptes Rendus Mathematique* 2010; **348**(21–22):1217–1220, .
31. Tikhonov AN, Arsenin VY. *Solutions of ill-posed problems*. Winston-Wiley, 1977.
32. Noël L, Van Miegroet L, Duysinx P. Analytical sensitivity analysis using the extended finite element method in shape optimization of bimaterial structures. *International Journal for Numerical Methods in Engineering* 2016; **107**(8):669–695, .
33. Sharma A, Villanueva H, Maute K. On shape sensitivities with heaviside-enriched XFEM. *Structural and Multidisciplinary Optimization* 2017; **55**(2):385–408, .
34. Najafi AR, Safdari M, Tortorelli DA, Geubelle PH. A gradient-based shape optimization scheme using an interface-enriched generalized FEM. *Computer Methods in Applied Mechanics and Engineering* 2015; **296**:1–17, .
35. Barthold FJ, Materna D. A modified extended finite element method approach for design sensitivity analysis. *International Journal for Numerical Methods in Engineering* 2015; **104**(3):209–234, .
36. Berggren M. A unified discrete–continuous sensitivity analysis method for shape optimization. *Applied and Numerical Partial Differential Equations: Scientific Computing in Simulation, Optimization and Control in a Multidisciplinary Context*. Springer Netherlands: Dordrecht, 2010; 25–39, .
37. Heikkola E, Rossi T, Toivanen J. Fast direct solution of the Helmholtz equation with a perfectly matched layer or an absorbing boundary condition. *International Journal for Numerical Methods in Engineering* 2003; **57**(14):2007–2025, .
38. Delfour MC, Zolézio JP. *Shapes and Geometries. Metrics, Analysis, Differential Calculus, and Optimization*. Second edn., SIAM: Philadelphia, 2011.