Structural and Multidisciplinary Optimization (2023) 66:245
https://doi.org/10.1007/s00158-023-03684-9

RESEARCH PAPER

Yet another parameter-free shape optimization method

Kenneth E. Swartz[1] · Ketan Mittal[1] · Mathias Schmidt[1] · Jorge-Luis Barrera[1] · Seth Watts[1] · Daniel A. Tortorelli[1,2]

Received: 7 December 2022 / Revised: 5 October 2023 / Accepted: 6 October 2023 / Published online: 21 November 2023

Abstract

The use of node coordinates as design variables in shape optimization offers a larger design space than computer-aided design (CAD)-based shape parameterizations. It also allows for the optimization of legacy designs, i.e., a finite element mesh from an existing design can be readily optimized to meet new performance requirements without involving a CAD model. However, it is well known that the node coordinate parameterization method is fraught with numerical difficulties, which makes it impractical to use. This has led to several of "parameter-free" shape optimization methods that seek the advantages and avoid the pitfalls of the naïve node coordinate parameterization method. These methods come in two main varieties: sensitivity filtering (or gradient smoothing) and consistent filtering. The latter is analogous to the density filter method used in topology optimization (TO). Herein, we use the PDE filter from TO and energy-based filters to implement consistent shape optimization filtering schemes easily and efficiently. Numerical experiments demonstrate that consistent methods are more robust than sensitivity filtering methods.

Keywords Shape optimization · Node coordinate parameterization · Consistent filtering

1 Introduction

Shape optimization morphs material boundaries to achieve optimal design performance. When compared to topology optimization, shape optimization exhibits less design freedom; however, it provides an explicit boundary/interface description. Thus, when simulations involve boundary or interface-dominant physics, e.g., stress computations for optimization metrics or solid–fluid interaction, shape optimization is often advantageous. Further, shape optimization results are easily communicated to manufacturing processes due to the explicit boundary and interface representations.

Shape optimization poses two key challenges. First, a parameterization must be defined that allows design flexibility while maintaining adequate design control. Unfortunately, these two desires are often contradictory. Second, a conforming mesh of each iterate design must be generated for use in the finite element simulation. The difficulty of mesh generation depends on the complexity of the parameterization. To ease this process, we combine the parameterization with the finite element mesh. Namely, we use the mesh nodal coordinates as the optimization variables. Users only generate a mesh via their favorite preprocessor for the initial design. Thereafter, the optimizer controls the mesh updates. This "parameter-free" approach reduces design limitations imposed by restrictive parameterizations.

To motivate our work, we optimize the shape of a hole in a biaxially loaded plate to minimize the compliance subject to a maximum volume constraint Fig. 1.[1] In our so-called parameterized optimization, we require the hole to be an ellipse and optimize the lengths of the axes, Fig. 2. The compliance of the optimized design is 79% of the initial design's compliance. The analytical solution, assuming the plate is infinite, is an ellipse with 2:1 axes ratio, so our design with 1.81 axes ratio is nearly optimal. Nonetheless, to improve the design, we no longer restrict it to the 2 parameter elliptical hole optimization. Rather, an "independent node" optimization is performed in which *all* of the finite element node coordinates (except those that would affect the plate's outer

Responsible Editor: Ole Sigmund

✉ Kenneth E. Swartz
swartz10@llnl.gov

[1] Lawrence Livermore National Laboratory, 7000 East Ave, Livermore, CA 94550, USA

[2] Department of Mechanical Science and Engineering, University of Illinois Urbana-Champaign, 1206 W Green St, Urbana, IL 61801, USA

[1] The full details of this optimization problem are provided in Sect. 3.

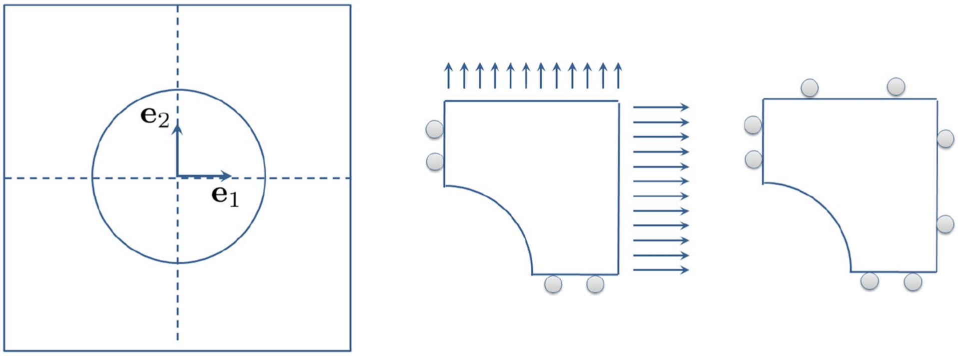
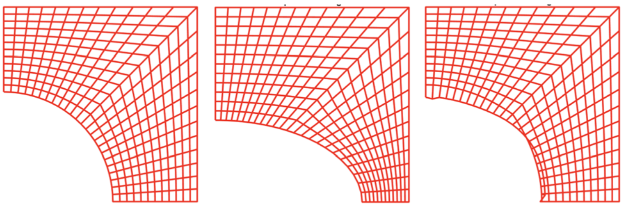


Fig. 1 Plate with hole (left), symmetry domain with biaxial loading boundary conditions (center) and design boundary conditions (right)



(a) Initial design (b) CAD parameterization, optimized (c) Independent node parameterization, after 3 iterations

Fig. 2 Finite element meshes for optimized plate with hole

boundary) serve as design parameters. However, as seen in Fig. 2, after 3 iterations, the mesh gets tangled so the optimization is terminated. This is expected since the compliance and volume are insensitive to internal node movements and hence the optimizer only moves the boundary nodes. To resolve this mesh tangling deficiency, various “parameter-free” shape optimization methods have been proposed; we propose yet another.

We aim to develop a parameter-free shape optimization method that uses readily available nonlinear programming (NLP) solvers, e.g., IPOPT (Wächter and Biegler 2006) and fmincon (Matlab 2017). In this way, we can readily accommodate an arbitrary number of constraints, verify the Karush-Kuhn-Tucker (KKT) optimality conditions, and avoid writing-specialized NLP solvers. We seek a robust method that does not require extensive parameter tuning when solving a range of optimization problems. Finally, we aim to develop an efficient method that converges rapidly and utilizes high-performance computing (HPC).

To achieve our parameter-free shape optimization goals, we mimic the implicit PDE density filter developed by Lazarov and Sigmund (2011) from topology optimization. The use of a consistent sensitivity analysis, i.e., we filter nodal coordinates and their sensitivities, provides robust convergence during the design optimization.

We aim to optimize the shape of three-dimensional continuum structures using the finite element method, and hence, we do not review literature that concerns optimization of plate and shell structures and isogeometric analysis.[2] Notwithstanding this omission, the shape optimization literature is vast, please refer to Samareh (2001); Martinelli and Jameson (2012); Upadhyay et al. (2021) for thorough reviews of the subject. Our discussion focuses on parameter-free methods. The use of node coordinates as design variables in shape optimization has the advantage of offering a large design space and eliminating the reliance on computer-aided design (CAD)-based shape parameterizations (Braibant and Fleury 1984; Chen and Tortorelli 1997) that are troublesome for industrially relevant optimization problems. This is because industrially relevant CAD models are created using numerous boolean, blending, rounding, and trimming operations. In the optimization, finite element meshes must be created from the CAD models and derivatives of the node coordinates with respect to the optimization parameters, e.g., select dimensions must be computed. While this task is easily accomplished for the simple optimization problem illustrated in Figs. 1 and 2, it is often extremely challenging for industrially relevant problems. CAD models associated with legacy designs may also be unavailable. However, if finite element models for such designs are available, they can be shape optimized via the parameter-free method to meet new design specifications. That said, the node coordinate parameterizations may lead to an ill-posed optimization problems (Chenais 1975; Bängtsson et al. 2003; Mohammadi and Pironneau 2004; Harbrecht 2008; Mohammadi and Pironneau 2008), but most definitely, it is well known that naively using the node coordinates as design variables produces designs with tangled meshes which are artifacts of

[2] The reason being that plate/shell studies and isogeometric-based formulations are less susceptible to the mesh distortion problems that plague the shape optimization of continuum bodies modeled with the finite element (or similar) method.

the finite element discretization (Braibant and Fleury 1984; Le et al. 2011), cf. the simple example above.

We compute the derivatives of the optimization cost and constraint functions using the adjoint method. The derivatives can be expressed as a combination of volumetric and surface integrals in the so-called domain method or solely with surface integrals in the so-called boundary method. Sensitivities computed via the "domain" method agree, within machine precision, to those computed via finite difference approximations (Choi and Seong 1986; Yang and Botkin 1987) because the differentiate-then-discretize and discretize-then-differentiate operations commute (Yang and Botkin 1987; Berggren 2010). However, sensitivities computed via the boundary method are error prone due to a lack of continuity in the gradient computations across the inter element boundaries (Berggren 2010). Fortunately, these errors diminish with mesh refinement and may also be reduced by using nonconventional C^1 interpolation.

Gradient smoothing is analogous to sensitivity filtering in topology optimization (Sigmund 1997) where it is well known that the filtered sensitivities are inconsistent with the NLP solver and hence the NLP convergence suffers (Matsumori et al. 2017). Two types of shape optimization gradient smoothing methods exist, explicit and implicit. The explicit approach uses a convolution filter, much like how Sigmund (1997) computes a "smooth" gradient from the "raw" gradient, e.g. (Stück and Rung 2011; Gerzen et al. 2012). In the implicit approach, a careful distinction is made between the derivative and the gradient, e.g., Pironneau (1984). We generally compute the derivative via adjoint methods as described below. But, the gradient, not the derivative, is often required by popular NLP algorithms, e.g., the method of steepest descent. The gradient belongs to an inner product space, and hence, it depends on the definition of the inner product. This is the impetus of the H^1 smoothing methods (Pironneau 1984; Jameson 2004; Gournay 2006; Nonogawa et al. 2020; Azegami 2021; Allaire et al. 2021) which roughly project the derivative onto the chosen inner product space. The projected gradient requires the solution of a PDE and hence the "implicit" filter terminology.

The traction method presented by Shimoda et al. (1997) can be viewed as a gradient smoothing method in which the H^1 inner product is replaced with the energy inner product from linear isotropic elasticity. The method applies the shape derivative as a traction to compute the smoothed gradient and hence its name. In another variant, Azegami and Takeuchi (2006) treat the shape derivative as an "ambient" displacement in a Robin-like boundary condition to obtain additional smoothness of the gradient. This latter approach is more closely related to the H^1 smoothing methods. Riehl and Steinmann (2014) extend the traction method by incorporating an adaptive mesh refinement step to ensure suitable meshes and accurate simulations.

As noted by Schwedes et al. (2017), the inconsistency introduced by only filtering the sensitivity in the implicit methods can be resolved by incorporating the chosen inner product into the NLP solver. Unfortunately, this is seldom the case as most NLP libraries work in $\mathbb{R}^n$ rather than the underlying function space in which the optimization problem is naturally formulated, e.g., H^1. Perhaps this is why most of the shape optimization methods cited above do not use popular NLP libraries; rather, they rely on augmented Lagrangian methods that incorporate a single constraint combined with inefficient steepest descent methods (Gournay 2006; Nonogawa et al. 2020; Ertl et al. 2019). Nonetheless, gradient smoothing algorithms prove effective as seen below and demonstrated by Ertl et al. (2019).

Riehl et al. (2014) and Bletzinger (2014) provide efforts to maintain consistency in the optimization formulation. Therein explicit filters are used to compute the gradient from the derivative and its transpose is used to smooth the design update. This is akin to the density filter (Bruns and Tortorelli 2001) from topology optimization where the element volume fraction vector $\boldsymbol{v}$ is optimized, but the filtered element volume fraction vector $\widetilde{\boldsymbol{v}} = \boldsymbol{F}\,\boldsymbol{v}$ obtained from the filter matrix $\boldsymbol{F}$ is used to define the design over which the response is computed. Sensitivities $\partial\theta/\partial\widetilde{\boldsymbol{v}}$ of the cost and constraint functions $\widetilde{\theta}(\widetilde{\boldsymbol{v}})$ are computed via adjoint methods and the chain-rule is used to evaluate the sensitivities that are input to the NLP algorithm $\partial\theta/\partial\boldsymbol{v} = \boldsymbol{F}^T\,\partial\widetilde{\theta}/\partial\widetilde{\boldsymbol{v}}$. Here, the smoothed and unsmoothed element volume fraction vectors $\widetilde{\boldsymbol{v}}$ and $\boldsymbol{v}$ are replaced by the smoothed and unsmoothed nodal coordinate vectors $\widetilde{\boldsymbol{X}}$ and $\boldsymbol{X}$; however, an inconsistency arises that does not appear in topology optimization. Namely, in the work by Riehl et al. (2014) and Bletzinger (2014), the mesh over which $\boldsymbol{F}$ is computed is defined by the smoothed node coordinate vector $\widetilde{\boldsymbol{X}}$, which continues to evolve during the optimization and consequently $\boldsymbol{F}$ changes during the optimization. This observation has led to the development of adaptive filter schemes, Antonau et al. (2022).[3]

In the end, it is difficult to know precisely what optimization problem is solved. Notably, the filter is recomputed at each optimization iteration using the configuration corresponding to the previous design iterate and hence the optimized design depends on the "trajectory" of the NLP algorithm. One cannot obtain KKT conditions to a predefined optimization problem, without taking this "trajectory" into account.

In the natural design variable method, Belegundu and Rajan (1988) apply "design loads" $\boldsymbol{P}_d$ to the initial design and use separate "shape morphing" analyses to deform the

[3] This publication also provides an excellent overview of the so-called vertex morphing methods.

initial design configuration Ω_0 into the optimized configuration Ω. The load $\boldsymbol{P}_d$ serves as the design variable vector in the optimization, Weeber et al. (1992); Tortorelli (1993). The initial design with node coordinate vector $\boldsymbol{X}_0$ is morphed into the optimized structure with node coordinate vector $\boldsymbol{X} = \boldsymbol{X}_0 + \boldsymbol{D}$. Here, $\boldsymbol{D}$ is the solution to the discretized shape morphing elasticity equation $\boldsymbol{K}_d \boldsymbol{D} = \boldsymbol{P}_d$. Mimicking the sensitivity filtering discussion above, we have $\boldsymbol{D} = \boldsymbol{K}_d^{-1} \boldsymbol{P}_d$ and $\partial\theta/\partial\boldsymbol{P} = \boldsymbol{K}_d^{-T}\, \partial\widetilde{\theta}/\partial\boldsymbol{X}$, i.e., $\boldsymbol{K}_d^{-1}$ plays the role of the filter matrix $\boldsymbol{F}$. This consistent method has a well-defined optimization problem that quickly converges.

The "fictitious energy method" is similar the natural design variable method; however, Scherer et al. (2010) use prescribed displacements rather than loads to morph the initial design into the optimized design. Mesh quality is ensured by using a "special" nonlinear, isotropic, hyperelastic material that is unaffected by rigid motions and dilations/contractions in the morphing analysis. In this way, rigid motions and dilations/contractions of elements are allowed whereas shear and unequal triaxial extensions/compressions of the elements are minimized. To further limit mesh distortion, an additional constraint on the total strain energy of the morphing deformation is imposed.

The explicit consistent filtering approach to shape optimization (Le et al. 2011) can be viewed as the shape optimization equivalent of the popular density-based topology optimization method (Bruns and Tortorelli 2001). The latter uses a consistent filter to formulate as well-posed optimization problem and hasten convergence of the NLP algorithm. Herein, we build upon the work of Le et al. (2011) by replacing the explicit filter with various implicit filters, another idea borrowed from topology optimization (Lazarov and Sigmund 2011) and used by Matsumori et al. (2017) in their shape optimization study. These implicit filters are more suitable to HPC and, in our opinion, easier to implement, although they require the solutions of additional PDEs. Indeed, the solvers are implemented using efficient numerical techniques leveraging scalable parallel computer architectures. On the other hand, the explicit filters require interprocessor communication to evaluate the filter convolution integrals and HPC libraries that support such computations are not generally available. A comparison of consistent explicit and implicit filters is presented by Najian and Bletzinger (2023).

Our approach is very similar to the H^1 filter approach described by Matsumori et al. (2017); however, we do not constrain a mesh quality energy or limit the extent of the filter to the boundary region. Similarities also exist between our approach and that presented by Liatsikouras et al. (2022).

Therein the coordinates of duplicate "handle nodes" on the boundary serve as the design parameters. The initial mesh is morphed into the optimized design by minimizing the morphing strain energy subject to the constraint that the "follower" surface nodes are close to the handle nodes. Because the constraint is enforced via the penalty method, the design smoothness can be controlled by varying the penalty value. To a lesser extent, our approach is similar to the fictitious energy approach (Scherer et al. 2010); however, we replace their morphing displacement boundary conditions with Robin boundary conditions so we are able to filter all of the node coordinate updates, i.e., both in the domain's interior and on its boundary and we do not constrain the fictitious energy. Tikhonov regularization is used by Bängtsson et al. (2003) to obtain a smooth boundary; the idea is motivated by Gunzburger et al. (2000). Their approach requires the solution of an ODE (for their two-dimensional study) to map the optimization's jagged boundary to a smooth boundary over which the optimization cost and constraint functions are computed. Interior nodes are relocated by solving an elasticity problem as done in the fictitious energy approach (Scherer et al. 2010); however, their problem is linear so the mesh motion is more limited. Gradients of the cost and constraint functions are computed in a consistent manner using a combination of direct and adjoint methods. Their thorough study compares many different problem formulations.

We experiment with different filters based on the vector Helmholtz PDE used by Matsumori et al. (2017), a nonlinear filter based on the fictitious energy approach of Scherer et al. (2010), and a linearized version of this nonlinear filter. To solve the optimization problems, we use popular NLP libraries IPOPT (Wächter and Biegler 2006) and fmincon (Matlab 2017) to leverage their efficient algorithms and their abilities to incorporate numerous constraints and compute KKT optimality conditions. Some of our computations utilize the MFEM library (Anderson et al. 2021), and hence, we can readily invoke parallelism and employ a wide variety of discretizations, e.g., high-order polynomials. Derivatives for the algorithms are computed consistently via adjoint methods and accurately using domain, rather than boundary, sensitivity integrations. Summarizing the salient features in our work, we highlight

1. Domain shape derivative computations,
2. Consistency with the NLP algorithm,
3. PDE and energy-based filters,
4. Multiple filter functions,
5. Use of popular NLP libraries.

We compare our consistent approaches to the fictitious energy approach of Scherer et al. (2010) and the gradient smoothing approaches of Allaire et al. (2021) and Azegami (2021). These three methods are also integrated with the popular NLP libraries and utilize the accurate domain shape derivative computations.

2 Independent node coordinate formulation

In the independent node coordinate formulation, the node coordinates contained in $\boldsymbol{X}$ that define the *physical* domain Ω serve as the design variable. A linear elastic finite element analysis is performed over Ω to compute the *physical* displacement field $\boldsymbol{u}$. The cost and constraint functions $\theta_i, i = 0, 1, 2, \cdots$ and their shape sensitivities $D\theta_i/D\boldsymbol{X}$ are then evaluated. The values of the coordinates $\boldsymbol{X}$, cost and constraint functions θ_i, and sensitivities $D\theta_i/D\boldsymbol{X}$ are input to the NLP algorithm to update the coordinates $\boldsymbol{X} \leftarrow \boldsymbol{X} + \Delta\boldsymbol{X}$. Iterations continue until convergence, or more likely until the appearance of a tangled mesh which causes the algorithm to crash.

In the finite element analysis, we find the physical displacement $\boldsymbol{u} \in \mathcal{H}(\boldsymbol{u}^p)$ that satisfies the linear elastic residual equation

$$\begin{aligned} 0 = &\ r(\boldsymbol{u}, \boldsymbol{w}) \\ = &\int_\Omega \nabla\boldsymbol{w} \cdot \mathbb{C}[\nabla\boldsymbol{u}]\, dv - \int_\Omega \boldsymbol{w} \cdot \boldsymbol{b}\, dv \\ &- \int_{A^t} \boldsymbol{w} \cdot \boldsymbol{t}^p\, da - \int_{A^p} \boldsymbol{w} \cdot p\, \boldsymbol{n}\, da \\ = &\ a(\boldsymbol{u}, \boldsymbol{w}) - \ell(\boldsymbol{w}) \end{aligned} \tag{1}$$

for all $\boldsymbol{w} \in \mathcal{H}(\boldsymbol{0})$. In the above $\mathbb{C}$ is the elasticity tensor which exhibits the usual symmetries, $\boldsymbol{b}$ is the prescribed body force per unit volume. $\partial\Omega$ is the boundary of Ω with outward unit normal vector $\boldsymbol{n}$; it is divided into three complementary regions A^u, A^t, and A^p over which the displacement $\boldsymbol{u}^p$, traction $\boldsymbol{t}^p$, and normal pressure p are prescribed. Finally, a is the bilinear form, ℓ is the load linear form, and the space of admissible functions is

$$\mathcal{H}(\boldsymbol{u}^p) = \{\boldsymbol{u} \in H^1 \ :\ \boldsymbol{u} = \boldsymbol{u}^p\ \forall\, \boldsymbol{x} \in A^u\}. \tag{2}$$

Without loss of generality, we assume $\boldsymbol{u}^p = \boldsymbol{0}$ henceforth.

Upon evaluating the displacement field, we compute the optimization cost and constraint functions θ_i

$$\begin{aligned} \theta_i(\boldsymbol{x}) = &\int_\Omega \Lambda(\nabla\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{x})\, dv + \int_{A^t} \tau(\boldsymbol{u}, \boldsymbol{x})\, da \\ &+ \int_{A^p} \boldsymbol{\tau}(\boldsymbol{u}, \boldsymbol{x}) \cdot \boldsymbol{n}\, da \end{aligned} \tag{3}$$

where $\boldsymbol{x}$ denotes the location of material particles in Ω and Λ, τ, and $\boldsymbol{\tau}$ are generic functions that can be used to compute, e.g., the volume of Ω by equating $\Lambda = 1$, $\tau = 0$, and $\boldsymbol{\tau} = \boldsymbol{0}$ or compliance by equating $\Lambda = \nabla\boldsymbol{u} \cdot \mathbb{C}[\nabla\boldsymbol{u}]$, $\tau = 0$, and $\boldsymbol{\tau} = \boldsymbol{0}$. The shape sensitivity of the above is evaluated using the adjoint method (Haug and Choi 1986). Details of the sensitivity analysis and finite element discretization are provided in Appendix.

```
Require: $\boldsymbol{X}$
Ensure: $\Theta_i$, $D\Theta_i/D\boldsymbol{X}$
  Solve primal problem $\boldsymbol{K}\boldsymbol{U} = \boldsymbol{P}$ for $\boldsymbol{U}$
  for $i \in \{0, 1, 2, \cdots\}$ do
      Compute $\Theta_i$ and the derivatives $\partial\Theta_i/\partial\boldsymbol{X}$ and $\partial\Theta_i/\partial\boldsymbol{U}$, cf. (34) and (35)
      Solve adjoint problem $\boldsymbol{K}\boldsymbol{W} = (\partial\Theta_i/\partial\boldsymbol{U})^T$ for $\boldsymbol{W}$
      Compute pseudo load $\partial R/\partial\boldsymbol{X}$, cf. (36)
      Compute sensitivity $D\Theta_i/D\boldsymbol{X} = \partial\Theta_i/\partial\boldsymbol{X} - \partial R/\partial\boldsymbol{X}$
  end for
  return $\Theta_i$, $D\Theta_i/D\boldsymbol{X}$
```

Algorithm 1 Independent node coordinate iteration,Appendix 1 for the discretization

In practice, many of the node coordinates are not moved during the optimization. To illustrate this, we optimize the shape of a hole in the center of a homogeneous isotropic 2×2 square plate that is subjected to biaxial traction, Fig. 1. To facilitate the plane stress analysis, we utilize domain symmetry whence the optimized hole geometry is necessarily symmetric with respect to reflections about the $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ axes. Since we are only optimizing the hole shape, the node coordinates are constrained such that the nodes on the vertical edges cannot move in the $\boldsymbol{e}_1$ direction and the nodes on the horizontal edges cannot move in the $\boldsymbol{e}_2$ direction.

One iteration for the independent node coordinate optimization algorithm is partially summarized in Algorithm 1. For each iteration of the optimization, we have a design Ω defined by the node coordinates $\boldsymbol{X}$ over which we compute the θ_i and their derivatives $D\theta_i/D\boldsymbol{X}$. The optimizer updates the design, i.e., $\boldsymbol{X} \leftarrow \boldsymbol{X} + \Delta\boldsymbol{X}$, whereafter another iteration commences if the convergence criterion has not been satisfied. Unfortunately, this independent node coordinate formulation invariably produces tangled meshes which cause the optimization to prematurely crash, Fig. 2.

2.1 Gradient smoothing algorithm

The gradient smoothing formulation is similar to the independent node coordinate formulation as we are given a body with material points identified by their positions $\boldsymbol{x}$ in the initial design configuration Ω_0 and $\boldsymbol{x} \leftarrow \boldsymbol{x} + \delta\boldsymbol{x}$ in the optimized design configuration Ω. The mesh tangling incurred by the independent node coordinate formulation is resolved using numerous variants of explicit (Stück and Rung 2011) or implicit (Pironneau 1984; Jameson 2004; Gournay 2006; Scherer et al. 2010; Allaire et al. 2021; Müller et al. 2021; Welker 2021) gradient smoothing algorithms. Mimicking the sensitivity filter in topology optimization (Sigmund 1997), the explicit method computes a smooth shape sensitivity by filtering the raw gradient, i.e., shape sensitivity. Implicit smoothing uses the Riesz representation theorem to relate the derivative $\delta\theta$ to the gradient $\mathrm{grad}\theta$. In $\mathbb{R}^n$, under the standard Euclidean inner product, there is no difference in the representation of the derivative and the gradient. Thus, when the design, i.e., the node coordinate vector $\boldsymbol{X}$, is viewed as an element of $\mathbb{R}^n$, the design derivative and gradient are (numerically) interchangeable. However, this is no longer the case when one views the design as an element of a function space $\mathcal{H}$, which is the case if we use a perturbation $\Delta\boldsymbol{x} \in \mathcal{H}$ to deform the current design into the next design iterate, i.e., points originally located at $\boldsymbol{x}$ in the current design are displaced to locations $\boldsymbol{x} = \boldsymbol{x} + \Delta\boldsymbol{x}$ in the next design iterate. Assuming the inner product on $\mathcal{H}$ is not the Euclidean inner product, the shape derivative is no longer (numerically) interchangeable with the gradient. Instead, assuming it exists, the derivative at Ω, $\delta\theta(\boldsymbol{x};\cdot) \in \mathrm{Lin}(\mathcal{H} \to \mathbb{R})$, is a linear operator from $\mathcal{H}$ into $\mathbb{R}$ and hence by the Riesz representation theorem there exists a unique $\mathrm{grad}\theta(\boldsymbol{x}) \in \mathcal{H}$ such that $\delta\theta(\boldsymbol{x}, \delta\boldsymbol{x}) = a(\mathrm{grad}\theta(\boldsymbol{x}), \delta\boldsymbol{x})$ for all $\delta\boldsymbol{x} \in \mathcal{H}$. In this equality $a(\cdot,\cdot)$ is the inner product for $\mathcal{H}$ and this is where things get murky as there are multiple choices for the space $\mathcal{H}$ and inner product $a(\cdot,\cdot)$. For example if we choose $\mathcal{H} = L^2$ and the standard inner product

$$a(\boldsymbol{u}, \boldsymbol{w}) = \int_\Omega \boldsymbol{u} \cdot \boldsymbol{w} \, dv \tag{4}$$

we obtain the "rough" gradient $\mathrm{grad}\theta(\boldsymbol{x})$ via

$$a(\mathrm{grad}\theta(\boldsymbol{x}), \delta\boldsymbol{x}) = \int_\Omega \mathrm{grad}\theta(\boldsymbol{x}) \cdot \delta\boldsymbol{x} \, dv = \delta\theta(\boldsymbol{x};\delta\boldsymbol{x}) \tag{5}$$

[4] However, if we choose $\mathcal{H} = H^1$ and the nonstandard inner product

$$\widetilde{a}(\boldsymbol{u}, \boldsymbol{w}) = \int_\Omega (\nabla\boldsymbol{u} \cdot \gamma \, \nabla\boldsymbol{w} + \boldsymbol{u} \cdot \boldsymbol{w}) \, dv \tag{6}$$

with $\gamma > 0$, we obtain the "smooth" gradient $\widetilde{\mathrm{grad}}\theta(\boldsymbol{x})$ via

$$\begin{aligned} \widetilde{a}(\widetilde{\mathrm{grad}}\theta(\boldsymbol{x}), \delta\boldsymbol{x}) = \int_\Omega \Big(\nabla\widetilde{\mathrm{grad}}\theta(\boldsymbol{x}) \cdot \gamma \, \nabla\delta\boldsymbol{x} + \\ \widetilde{\mathrm{grad}}\theta(\boldsymbol{x}) \cdot \delta\boldsymbol{x} \Big) \, dv \\ = \delta\theta(\boldsymbol{x};\delta\boldsymbol{x}) \end{aligned} \tag{7}$$

In the above, we emphasize that $\widetilde{\mathrm{grad}}\theta(\boldsymbol{x})$ is a vector field in H^1 and $\nabla\widetilde{\mathrm{grad}}\theta(\boldsymbol{x})$ is the spatial derivative of this field.

To see why we refer to $\mathrm{grad}\theta(\boldsymbol{x})$ and $\widetilde{\mathrm{grad}}\theta(\boldsymbol{x})$ as the rough and smooth gradients, consider the following minimization problem

$$\begin{aligned} \min_{\widetilde{\mathrm{grad}}\theta \in H^1} \quad & \int_\Omega |\widetilde{\mathrm{grad}}\theta - \mathrm{grad}\theta|^2 \, dv \\ \text{such that} \quad & \int_{\Omega_0} |\nabla\widetilde{\mathrm{grad}}\theta|^2 \, dv = 0 \end{aligned} \tag{8}$$

where $\mathrm{grad}\theta \in L^2$ is the known rough gradient and we have dropped the arguments for conciseness. Solving the above via the penalty method results in the unconstrained problem

$$\begin{aligned} \min_{\widetilde{\mathrm{grad}}\theta(\boldsymbol{x}) \in H^1} \quad & \int_\Omega |\widetilde{\mathrm{grad}}\theta(\boldsymbol{x}) - \mathrm{grad}\theta(\boldsymbol{x})|^2 \, dv + \\ \gamma & \int_{\Omega_0} |\nabla\widetilde{\mathrm{grad}}\theta(\boldsymbol{x})|^2 \, dv \end{aligned} \tag{9}$$

where $\gamma > 0$ is the penalty parameter. The stationary condition of (9) requires $\widetilde{\mathrm{grad}}\theta(\boldsymbol{x}) \in H^1$ to satisfy

$$\widetilde{a}(\widetilde{\mathrm{grad}}\theta(\boldsymbol{x}), \boldsymbol{w}) = a(\mathrm{grad}\theta(\boldsymbol{x}), \boldsymbol{w}) = \delta\theta(\boldsymbol{x};\boldsymbol{w}) \tag{10}$$

for all $\boldsymbol{w} \in H^1$. Larger values of γ increase the enforcement of the $\int_{\Omega_0} |\nabla\widetilde{\mathrm{grad}}\theta(\boldsymbol{x})|^2 \, dv = 0$ constraint, i.e., lessen the spatial variations in $\widetilde{\mathrm{grad}}\theta(\boldsymbol{x})$. An excellent tutorial on the need for using $\widetilde{\mathrm{grad}}\theta(\boldsymbol{x})$ rather than $\mathrm{grad}\theta(\boldsymbol{x})$ appears in Jameson (2004).

The H^1 smoothing algorithm uses the smooth gradient $\widetilde{\mathrm{grad}}\theta(\boldsymbol{x})$ to compute the design updates $\delta\boldsymbol{x}$. Smoother gradients result in smoother updates and hence less mesh tangling. Setting $\gamma = 1$ we see that $\widetilde{a}$ reduces to the standard inner product for H^1 and hence the H^1 terminology.

We finish the derivation with the imposition of the "boundary conditions." To fix the location of portions of the boundary $\partial\Omega$ during the shape optimization, we replace H^1 in (10) with

$$H^1_0 = \{\boldsymbol{u} \in H^1 \,|\, \boldsymbol{u} \cdot \boldsymbol{e} = 0 \text{ on } A^f\} \tag{11}$$

in which $A^f \subset \partial\Omega_0$ are the surface regions which are fixed in the $\boldsymbol{e}$ direction during the optimization. E.g. in the Fig. 1 example $\boldsymbol{u} \cdot \boldsymbol{e}_2 = 0$ on the horizontal edges and $\boldsymbol{u} \cdot \boldsymbol{e}_1 = 0$ on the vertical edges. Telling the nonlinear programming

[4] We emphasize above that both $\mathrm{grad}\theta(\boldsymbol{x}) \in \mathcal{H}$ and $\delta\boldsymbol{x} \in \mathcal{H}$ are functions of position, i.e., $a(\mathrm{grad}\theta(\boldsymbol{x}), \delta\boldsymbol{x}) = \int_\Omega \mathrm{grad}\theta(\boldsymbol{x})(\boldsymbol{y}) \cdot \delta\boldsymbol{x}(\boldsymbol{y}) \, dv_y = \delta\theta(\boldsymbol{x};\delta\boldsymbol{x})$.

```
Require: $\boldsymbol{X}$
Ensure: $\Theta_i$, $\widetilde{\mathrm{Grad}\Theta_i}$
  Solve primal problem $\boldsymbol{KU} = \boldsymbol{P}$ for $\boldsymbol{U}$
  for $i \in \{0, 1, 2, \cdots\}$ do
      Compute $\Theta_i$ and the derivatives
      $\partial\Theta_i/\partial\boldsymbol{X}$ and $\partial\Theta_i/\partial\boldsymbol{U}$, cf. (34) and (35)
      Solve adjoint problem
      $\boldsymbol{KW} = (\partial\Theta_i/\partial\boldsymbol{U})^T$ for $\boldsymbol{W}$
      Compute pseudo load
      $\partial R/\partial\boldsymbol{X}$, cf. (36)
      Compute sensitivity
      $D\Theta_i/D\boldsymbol{X} = \partial\Theta_i/\partial\boldsymbol{X} - \partial R/\partial\boldsymbol{X}$
      Solve the smoothed gradient problem
      $(\widetilde{\boldsymbol{K}} + \widetilde{\boldsymbol{M}})\,\widetilde{\mathrm{Grad}\Theta_i} = (D\Theta_i/D\boldsymbol{X})^T$ for $\widetilde{\mathrm{Grad}\Theta_i}$
  end for
  return $\Theta_i$, $\widetilde{\mathrm{Grad}\Theta_i}$
```

Algorithm 2 H^1 smoothing algorithm iteration,Appendix 2 for the discretization

algorithm that $\widetilde{\mathrm{grad}\theta}(\boldsymbol{x}) \cdot \boldsymbol{e} = 0$ on A^f ensures that these surfaces remain unchanged in these directions during the optimization.

The H^1 smoothing algorithm is analogous to the PDE filter in topology optimization (Lazarov and Sigmund 2011). However, it is not performed in a consistent manner, and hence, it is also analogous to the sensitivity filtering approach in topology optimization (Sigmund 1997).[5] As such, the optimization algorithm may converge slowly or not at all.

We note that the traction method (Shimoda et al. 1997) is a variant of the H^1 smoothing algorithm in which the elasticity bilinear form a of (1) replaces $\widetilde{a}$ of (6), i.e., we use the energy inner product. As such, "infinitesimal rigid motions" characterized by skew $\nabla\widetilde{\mathrm{grad}\theta}(\boldsymbol{x})$ are not penalized, as these do not adversely affect mesh quality. Multiple variants of the traction method are described by Azegami (2021).

The H^1 smoothing algorithm is only a slight modification to the independent node coordinate formulation. For each iteration of the optimization, we are given the node coordinates $\boldsymbol{X}$ whereupon we complete Algorithm 2 to compute the Θ_i and their H^1 smoothed gradients $\widetilde{\mathrm{Grad}\Theta_i}$. The optimizer processes these quantities and either updates the design, i.e., $\boldsymbol{X} \leftarrow \boldsymbol{X} + \Delta\boldsymbol{X}$ or declares convergence.

2.2 Fictitious energy formulation

In the fictitious energy formulation of Scherer et al. (2010), an initial design Ω_0 is deformed into the optimized design Ω such that points located at $\boldsymbol{x}_0$ in the initial design Ω_0 are displaced to the locations $\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{d}$ in the optimized design. As such, the design displacement field $\boldsymbol{d}$ serves as the control in the optimization. To be clear, $\boldsymbol{d}$ is only optimized on the boundary $\partial\Omega_0 \backslash A^f$ where A^f is defined in the discussion surrounding (11). For the Fig. 1 example, $d^p = \boldsymbol{d} \cdot \boldsymbol{e}_1$ and $d^p = \boldsymbol{d} \cdot \boldsymbol{e}_2$ are optimized over the circular arc, $d^p = \boldsymbol{d} \cdot \boldsymbol{e}_1$ is optimized over the horizontal edges and $d^p = \boldsymbol{d} \cdot \boldsymbol{e}_2$ is optimized over the vertical edges. Over the remainder of the boundary A^f, $\boldsymbol{d} \cdot \boldsymbol{e} = 0$. To obtain the displacement $\boldsymbol{d}$ on the interior Ω_0 , a "fictitious" nonlinear elasticity analysis is performed. The analysis assumes an isotropic hyperelastic material that is unaffected by rigid deformations and dilations/contractions.

The morphing analysis finds the $\boldsymbol{d}$ that solves the minimization problem

$$\min_{\boldsymbol{d} \in \mathcal{H}_f(d^p)} \int_{\Omega_0} \Psi_f(\boldsymbol{F}_d)\, dv \tag{12}$$

where Ψ_f is the fictitious energy density defined in Appendix 3, $\boldsymbol{F}_d = \boldsymbol{I} + \nabla\boldsymbol{d}$ is the fictitious deformation gradient,

$$\mathcal{H}_f(d^p) = \{\boldsymbol{d} \in H^1 \,|\, \boldsymbol{d} \cdot \boldsymbol{e} = 0 \text{ on } A^f \text{ and } \\ \boldsymbol{d} \cdot \boldsymbol{e} = d^p \text{ on } \partial\Omega_0 \setminus A^f\} \tag{13}$$

is the space of admissible fictitious displacements, and d^p is the prescribed displacement that is optimized. Stationarity of (12) requires $\boldsymbol{d} \in \mathcal{H}_f(d^p)$ to satisfy the residual equation

$$r_f(\boldsymbol{d}, \boldsymbol{w}) = \int_{\Omega_0} \nabla\boldsymbol{w} \cdot \boldsymbol{P}_f\, dv = 0 \tag{14}$$

for all $\boldsymbol{w} \in \mathcal{H}_f(0)$ where $\boldsymbol{P}_f = \partial\Psi_f/\partial\boldsymbol{F}_d$ is the fictitious First Piola-Kirchhoff stress. In the finite element method, we express $\boldsymbol{d} = \boldsymbol{d}^0 + \boldsymbol{d}^p$ where $\boldsymbol{d}^0 \in \mathcal{H}_f(0)$ and $\boldsymbol{d}^p \in \mathcal{H}_f(d^p)$ is a known extension function that satisfies the prescribed displacement boundary condition. In this way, we solve (12) by finding the $\boldsymbol{d}^0 \in \mathcal{H}_f(0)$ that satisfies

$$r_f(\boldsymbol{d}^0, \boldsymbol{w}, \boldsymbol{d}^p) = \int_{\Omega_0} \nabla\boldsymbol{w} \cdot \frac{\partial\Psi_f}{\partial\boldsymbol{F}_d}\, dv = 0 \tag{15}$$

for all $\boldsymbol{w} \in \mathcal{H}_f(0)$. We then trivially compute $\boldsymbol{d} = \boldsymbol{d}^0 + \boldsymbol{d}^p$. The solution of the above residual equation for $\boldsymbol{d}^0$ is obtained via Newton's method, Appendix 3 for details.

[5] The lack of consistency in $\mathbb{R}^n$ means $D\theta/DX \neq \widetilde{\mathrm{grad}\theta}$ which "confuses" optimization algorithms developed in $\mathbb{R}^n$.

As an added measure to prevent severe mesh distortion during the shape optimization, the fictitious energy approach places a constraint on the total fictitious energy, i.e.,

$$\theta_f(d^p) = \int_{\Omega_0} \Psi_f(\boldsymbol{F}_d)\, dv - \overline{\Psi}_f \leq 0 \tag{16}$$

where $\overline{\Psi}_f$ is the maximum permissible total fictitious energy. The above is enforced in addition to the $\theta_i(d^p) \leq 0$ and $i = 1, 2, \cdots$ constraints.

The fictitious energy approach is a consistent approach; thereby, it exhibits faster convergence verses the H^1 smoothing formulation. However, the consistency comes with a price with regard to the sensitivity analysis as two adjoint analyses are now required to evaluate the sensitivity of each cost and constraint function with respect the boundary displacement d^p, Appendix 3 for details.

In each iteration of the fictitious energy formulation, we are given the initial node coordinates $\boldsymbol{X}_0$ and the node design displacement vector $\boldsymbol{D}^p$ whereupon we complete Algorithm 3 to compute the Θ_i and their derivatives $D\Theta_i/D\boldsymbol{D}^p$, $i = f, 0, 1, 2, \cdots$. The optimizer subsequently uses these quantities to either update the design, i.e., $\boldsymbol{D}^p \leftarrow \boldsymbol{D}^p + \Delta\boldsymbol{D}^p$ or declare the convergence.

2.3 Consistent filtering formulation

The consistent filtering formulation is similar to the fictitious energy formulations as we again deform an initial design Ω_0 into the optimized design Ω and the design displacement field $\boldsymbol{d}$ controls the design. Here, however, points at locations $\boldsymbol{x}_0$ in the initial design Ω_0 are displaced to locations $\boldsymbol{x} = \boldsymbol{x}_0 + \widetilde{\boldsymbol{d}}$ in the optimized design Ω, where $\widetilde{\boldsymbol{d}}$ is the *filtered* version of $\boldsymbol{d}$. The method is analogous to PDE sensitivity filtering in topology optimization (Lazarov and Sigmund 2011). As such, it eliminates the inconsistency of the H^1 smoothing formulation and filters the boundary updates which the fictitious energy formulation does not.

To obtain the filtered design in our "PDE filter" variant, we find $\widetilde{\boldsymbol{d}} \in \mathcal{H}_c$ such that

$$\begin{aligned} 0 &= r_c(\widetilde{\boldsymbol{d}}, \boldsymbol{w}, \boldsymbol{d}) \\ &= \int_{\Omega_0} \left(\nabla \boldsymbol{w} \cdot \gamma\, \nabla \widetilde{\boldsymbol{d}} + \boldsymbol{w} \cdot \widetilde{\boldsymbol{d}} \right) dv \\ &\quad - \int_{\Omega_0} \boldsymbol{w} \cdot \boldsymbol{d}\, dv \\ &= a_c(\widetilde{\boldsymbol{d}}, \boldsymbol{w}) - \ell_c(\boldsymbol{w}, \boldsymbol{d}) \end{aligned} \tag{17}$$

for all $\boldsymbol{w} \in \mathcal{H}_c$ where

```
Require: $\boldsymbol{X}_0$, $\boldsymbol{D}^p$
Ensure: $\Theta_i$, $D\Theta_i/D\boldsymbol{D}^p$
  while $|\boldsymbol{R}_f^0| > \epsilon$ Solve mesh morphing problem do
      Solve Newton update problem $\boldsymbol{K}_f^{00}\Delta\boldsymbol{D}^0 = -\boldsymbol{R}_f^0$ for $\Delta\boldsymbol{D}^0$ on the $\Omega_0$ mesh
      Update displacement $\boldsymbol{D}^0 \leftarrow \boldsymbol{D}^0 + \Delta\boldsymbol{D}^0$
      Compute $\boldsymbol{R}_f^0$ on the $\Omega_0$ mesh
  end while
  Compute $\Theta_f$ on the $\Omega_0$ mesh
  Compute sensitivity $D\Theta_f/D\boldsymbol{D}^p = \boldsymbol{R}_f^p$
  Define the $\Omega$ mesh with node coordinates $\boldsymbol{X} = \boldsymbol{X}_0 + \boldsymbol{D}$
  Solve primal problem $\boldsymbol{K}\boldsymbol{U} = \boldsymbol{P}$ for $\boldsymbol{U}$ on the $\Omega$ mesh
  for $i \in \{0, 1, 2, \cdots\}$ do
      Compute $\Theta_i$ and the derivatives $\partial\Theta_i/\partial\boldsymbol{X}$ and $\partial\Theta_i/\partial\boldsymbol{U}$ on the $\Omega$ mesh, cf. (34) and (35)
      Solve adjoint problem $\boldsymbol{K}\,\boldsymbol{W} = (\partial\Theta_i/\partial\boldsymbol{U})^T$ for $\boldsymbol{W}$ on the $\Omega$ mesh
      Compute pseudo load $\partial R/\partial\boldsymbol{X}$ on the $\Omega$ mesh, cf. (36)
      Compute sensitivity $D\Theta_i/D\boldsymbol{X} = \partial\Theta_i/\partial\boldsymbol{X} - \partial R/\partial\boldsymbol{X}$
      Solve adjoint problem $\boldsymbol{K}_f^{00}\,\boldsymbol{W}^0 = (D\Theta_i/D\boldsymbol{X}^0)^T$ for $\boldsymbol{W}^0$
      Compute sensitivity $D\Theta_i/D\boldsymbol{D}^p = D\Theta_i/D\boldsymbol{X}^p - \boldsymbol{W}^{0^T}\,\boldsymbol{K}_f^{0p}$
  end for
  return $\Theta_i$, $D\Theta_i/D\boldsymbol{D}^p$, $i = f, 0, 1, 2, \cdots$
```

Algorithm 3 Fictitious energy formulation, Appendix 3 for the discretization

$$\mathcal{H}_c = \{\widetilde{\boldsymbol{d}} \in H^1 \,|\, \widetilde{\boldsymbol{d}} \cdot \boldsymbol{e} = \boldsymbol{0} \;\text{ on }\; A^f\} \tag{18}$$

(11).

Similar to the H^1 smoothing discussion, to obtain physical insight into the filter, we consider the minimization problem

$$\begin{aligned} &\min_{\widetilde{\boldsymbol{d}} \in \mathcal{H}_c} \int_{\Omega_0} |\widetilde{\boldsymbol{d}} - \boldsymbol{d}|^2\, dv \\ &\text{such that} \int_{\Omega_0} |\nabla \widetilde{\boldsymbol{d}}|^2\, dv = 0 \end{aligned} \tag{19}$$

and solve it via the penalty method, i.e., we solve

$$\begin{aligned}\min_{\tilde{\boldsymbol{d}}\in\mathcal{H}_c} &\int_{\Omega_0} |\tilde{\boldsymbol{d}}-\boldsymbol{d}|^2\,dv \\ &+\gamma\int_{\Omega_0}|\nabla\tilde{\boldsymbol{d}}|^2\,dv\end{aligned} \tag{20}$$

. The stationarity condition of (20) requires $\tilde{\boldsymbol{d}}\in\mathcal{H}_c$ to satisfy (17) for all $\boldsymbol{w}\in\mathcal{H}_c$. In this way, $\tilde{\boldsymbol{d}}$ is "close" to $\boldsymbol{d}$, but $|\nabla\tilde{\boldsymbol{d}}|$ will be "small," i.e., $\tilde{\boldsymbol{d}}$ is the smoothed version of $\boldsymbol{d}$. The smoothness of $\tilde{\boldsymbol{d}}$ increases with increasing γ because the enforcement of the $\int_{\Omega_0}|\nabla\tilde{\boldsymbol{d}}|^2\,dv=0$ constraint increases with increasing γ. Our filter is very similar to that used by Matsumori et al. (2017); however, they only integrate $|\tilde{\boldsymbol{d}}-\boldsymbol{d}|^2$ over regions that are close to the boundary $\partial\Omega_0$ so the interior nodes are less constrained, which is most likely fine.

A comparison of the filtering approaches is worthy of discussion. The fictitious energy residual Eq. (15) for $\boldsymbol{d}^0$ and consistent filter residual Eq. (17) for $\tilde{\boldsymbol{d}}$ are solved over the initial design configuration Ω_0 whereas the H^1 smoothing Eq. (10) for $\widehat{\mathrm{grad}\,\theta_i}$ is solved over the current design configuration Ω. One difference between the fictitious energy and consistent approaches lies in the definitions of the spaces of the admissible displacements $\mathcal{H}_f$ and $\mathcal{H}_c$, (13) and (18). In the fictitious energy method, $\boldsymbol{d}=d^p\,\boldsymbol{e}$ is prescribed by the optimizer over $\partial\Omega_0\backslash A^f$ whereas in the consistent approach, $\tilde{\boldsymbol{d}}$ is not prescribed over this surface. Because of this, no smoothing is performed on this boundary in the fictitious energy method. Another difference between these methods is that the fictitious energy method requires the solution of a nonlinear PDE; albeit in the second variant of our consistent approach described next, we must also solve a nonlinear PDE.

The $\nabla\boldsymbol{w}\cdot\gamma\,\nabla\tilde{\boldsymbol{d}}$ term in (17) penalizes $|\nabla\tilde{\boldsymbol{d}}|$. As such, rigid transformations or dilations/contractions are penalized even though they do not adversely affect mesh quality. Motivated by this observation and by Scherer et al. (2010), in our second variant "energy filter" approach, we solve a nonlinear PDE by replacing $\int_{\Omega_0}\nabla\boldsymbol{w}\cdot\gamma\,\nabla\tilde{\boldsymbol{d}}\,dv$ with $\int_{\Omega_0}\nabla\boldsymbol{w}\cdot\gamma\,\partial\Psi_f(\tilde{\boldsymbol{F}}_d)/\partial\tilde{\boldsymbol{F}}_d\,dv$ where $\tilde{\boldsymbol{F}}_d=\boldsymbol{I}+\nabla\tilde{\boldsymbol{d}}$. This has the effect of weakly enforcing the constraint $\int_{\Omega_0}\Psi_f(\tilde{\boldsymbol{F}}_d)\,dv=0$ rather than $\int_{\Omega_0}|\nabla\tilde{\boldsymbol{d}}|^2\,dv=0$. And in our third variant, the "linearized energy filter," we alternatively penalize the linearized energy by replacing $\int_{\Omega_0}|\nabla\tilde{\boldsymbol{d}}|^2\,dv$ with $1/2\int_{\Omega_0}\nabla\tilde{\boldsymbol{d}}\cdot\mathbb{A}_f[\nabla\tilde{\boldsymbol{d}}]\,dv=0$ where $\mathbb{A}_f=\partial^2\Psi_f(\boldsymbol{I})/\partial\tilde{\boldsymbol{F}}_d^2$ provides the linearized energy, Appendix 3 for details.

In these consistent approaches, we maintain a clear distinction between the filtered field $\tilde{\boldsymbol{d}}$ that is used to define Ω and the design displacement field $\boldsymbol{d}$ that we optimize. So like the fictitious energy approach, this has ramification in the sensitivity analysis, namely we again have to solve two adjoint problems, Appendix 4 for details.

```
Require: $\boldsymbol{X}_0$, $\boldsymbol{D}$
Ensure: $\Theta_i$, $D\Theta_i/D\boldsymbol{D}$
    Solve mesh morphing problem
  $(\boldsymbol{K}_c+\boldsymbol{M}_c)\,\tilde{\boldsymbol{D}}=\boldsymbol{M}_c\,\boldsymbol{D}$ for $\tilde{\boldsymbol{D}}$ on the $\Omega_0$ mesh
    Define the $\Omega$ mesh with node coordinates
  $\boldsymbol{X}=\boldsymbol{X}_0+\tilde{\boldsymbol{D}}$
    Solve primal problem
  $\boldsymbol{K}\,\boldsymbol{U}=\boldsymbol{P}$ for $\boldsymbol{U}$ on the $\Omega$ mesh
  for $i\in\{0,1,2,\cdots\}$ do
        Compute $\Theta_i$ and the derivatives $\partial\Theta_i/\partial\boldsymbol{X}$
  and $\partial\Theta_i/\partial\boldsymbol{U}$ on the $\Omega$ mesh, cf. (34) and (35)
        Solve adjoint problem
  $\boldsymbol{K}\,\boldsymbol{W}=(\partial\Theta_i/\partial\boldsymbol{U})^T$ for $\boldsymbol{W}$
        Compute pseudo load
  $\partial R/\partial\boldsymbol{X}$ on the $\Omega$ mesh, cf. (36)
        Compute sensitivity
  $D\Theta_i/D\boldsymbol{X}=\partial\Theta_i/\partial\boldsymbol{X}-\partial R/\partial\boldsymbol{X}$
        Solve adjoint problem
  $(\boldsymbol{K}_c+\boldsymbol{M}_c)\,\boldsymbol{W}=(D\Theta_i/D\boldsymbol{X})^T$ for $\boldsymbol{W}$
        Compute sensitivity
  $D\Theta_i/D\boldsymbol{D}=\boldsymbol{M}_c\,\boldsymbol{W}$
  end for
  return $\Theta_i$, $D\Theta_i/D\boldsymbol{D}$
```

Algorithm 4 Consistent filtering formulation, Appendix 4 for the discretization

In each iteration of the consistent PDE filtering formulation, we are given the initial node coordinates $\boldsymbol{X}_0$ and the design displacement $\boldsymbol{D}$ whereupon we complete Algorithm 4 to compute the Θ_i and their derivatives $D\Theta_i/D\boldsymbol{D}$. The optimizer subsequently uses these quantities to either update the design, i.e., $\boldsymbol{D}\leftarrow\boldsymbol{D}+\Delta\boldsymbol{D}$ or declare the convergence.

As expected, filtering $\boldsymbol{d}$ reduces numerical anomalies due to the discretization and hastens convergence of the optimization problem by reducing the effective design space. Indeed, the many design parameters in $\boldsymbol{d}$ are "mollified" into a fewer number of influential parameters. To see this, we use the spectral decomposition theorem to express[6] $\boldsymbol{\Phi}^T(\boldsymbol{K}_c+\boldsymbol{M}_c)\boldsymbol{\Phi}=\mathrm{diag}\big(\lambda_1,\lambda_2,\cdots\big)$ where $(\lambda_i,\boldsymbol{\Phi}_i)$ are the eigenpairs of $\boldsymbol{K}_c+\boldsymbol{M}_c$ that are arranged such that $\lambda_i\le\lambda_{i+1}$, and the eigenvectors are mass normalized such that $\boldsymbol{\Phi}_i^T\boldsymbol{M}_c\boldsymbol{\Phi}_j=\delta_{ij}$ and $\boldsymbol{\Phi}=\big[\boldsymbol{\Phi}_1\;\;\boldsymbol{\Phi}_2\;\;\cdots\big]$. In this way, with $\tilde{\boldsymbol{D}}=\boldsymbol{\Phi}\tilde{\boldsymbol{D}}^*$ and $\boldsymbol{D}=\boldsymbol{\Phi}\boldsymbol{D}^*$, we have

[6] Appendix 4 for details of this discretization.

$$\widetilde{\boldsymbol{D}}^{*} = \left\{ \begin{array}{c} \frac{1}{\lambda_1} D_1^{*} \\ \frac{1}{\lambda_2} D_2^{*} \\ \vdots \end{array} \right\} \tag{21}$$

We see that the “modes” associated with the smaller eigenvalues contribute more to $\widetilde{\boldsymbol{D}}$ than those associated with the larger eigenvalues. And the mode shapes associated with smaller eigenvalues are smoother than those associated with larger eigenvalues. Hence, $\widetilde{\boldsymbol{D}}$ is smoother than $\boldsymbol{D}$, i.e., by replacing $\boldsymbol{D}$ with $\widetilde{\boldsymbol{D}}$, we have mollified the space of possible design configurations Ω.

3 Filter comparison

To illustrate the parameter-free methods described in Sect. 2, we return to the problem discussed in Sect. 1 and optimize the shape of a hole in a plane stress plate, initially a circle with radius $r = 1/(2\sqrt{\pi})$, to minimize the compliance (or maximum stress) subject to a constraint that the area be less than $\overline{A} = 0.75\,V$ where A is the area of the plate without the hole, Fig. 1. The 2×2 plate is comprised of an isotropic linear elastic material with Young’s modulus $E = 10.0$ and Poisson ratio $\nu = 0.3$ and subjected to a biaxial tensile load with prescribed traction $\boldsymbol{t}^p = 2\,\boldsymbol{e}_1$ and $\boldsymbol{t}^p = \boldsymbol{e}_2$ over the vertical and horizontal boundaries. Quarter symmetry is utilized to reduce the computational burden. For the H^1 and traction methods with $\Delta\boldsymbol{x} = \boldsymbol{d}$, and consistent method, we solve

$$\begin{aligned}
&\min_{\boldsymbol{d}\in\mathcal{H}_H} \theta_0 \\
&\text{such that } \theta_1 = \int_{\Omega} da - \overline{A} = 0 \\
&\underline{d} \leq \boldsymbol{d}\cdot\boldsymbol{e}_i \leq \overline{d} \quad \text{for} \quad i = 1,2
\end{aligned} \tag{22}$$

. And for the fictitious energy method, we solve

$$\begin{aligned}
&\min_{d^p} \theta_0 \\
&\text{such that } \theta_1 = \int_{\Omega} da - \overline{A} = 0 \\
&\theta_f = \int_{\Omega_0} \Psi_f(\boldsymbol{F}_f)\, da - \overline{\Psi}_f \leq 0 \\
&\underline{d} \leq d^p \leq \overline{d}
\end{aligned} \tag{23}$$

In our reduced space optimization, we use the adjoint sensitivity analyses to account for the implicit dependence of the response, e.g., $\boldsymbol{u}$ with respect to the design. For the H^1, traction, and fictitious energy approaches, it is easy to place box constraints on the design because $\boldsymbol{d}$ and d^p directly affect the shape as $\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{d}$; for all of these cases $\underline{d}, \overline{d} = \mp 0.25$. On the other hand, in the consistent approaches, $\boldsymbol{x} = \boldsymbol{x}_0 + \widetilde{\boldsymbol{d}}$ where $\widetilde{\boldsymbol{d}}$ is implicitly defined by the design $\boldsymbol{d}$. So while we do provide box constraints on $\boldsymbol{d}$, their range is large in comparison with those in the other approaches. As a guide, we choose $\underline{d}, \overline{d} \approx \mp\lambda_i \underline{\widetilde{d}}$ where $\underline{\widetilde{d}}$ is the bound we desire on $\widetilde{d}$ and λ_i is the $i-th$ smallest eigenvalue of $\boldsymbol{K}_c + \boldsymbol{M}_c$ where $i \approx 4, 5, 6$. For all of these cases, $\underline{d}, \overline{d} = \mp 20 \approx \mp 0.25 \cdot \lambda_4 = 18.25$. In all cases, these box constraints are necessary to obtain well-posed problems. Indeed, removing them could result in designs with fine scale shape oscillations in the continuum setting and tangled finite element meshes in the discrete setting. All of the above functions θ_i are scaled so that the sensitivities of each function has a unit norm for the initial design, e.g., $|D\theta_i/D\boldsymbol{D}| = 1$ for $\boldsymbol{D} = \boldsymbol{0}$. An exception for this is the maximum energy constraint in the fictitious energy approach which is normalized for $d^p = 0.02$ as $D\theta_f/Dd^p = 0$ for $d^p = 0$. To keep things relatively equal, the H^1 smoothing and consistent PDE filter use $\gamma = 10$, the traction uses $E = 10$ and $\nu = 0.3$,and the fictitious energy and consistent approaches with the linearized energy function use $\gamma = 5$. On the other hand, the fictitious energy and consistent approaches with the nonlinear energy function use $\gamma = 1$. We conjecture that the lower γ value is needed due to the stiffening effect of the nonlinear energy function Ψ_f, i.e., a higher nonlinear energy is required to achieve the same level of mesh morphing as compared to the linearized energy. Loosely based on the $\overline{\Psi}_f = .006\overline{A} = 0.0045$ suggested by Scherer et al. (2010) (which does not include the γ factor), we assign $\overline{\Psi}_f = \gamma\, 0.030\overline{A} = 0.1125$ for the maximum energy for the fictitious energy method with the linearized energy. We decreased the maximum to $\overline{\Psi}_f = 4\,\gamma\,\Psi_f(0.02) = 0.0285$, i.e., $4\,\gamma$ times the “normalization” energy for the nonlinear energy case. The selection of these maximum energy bounds definitely requires some “experimentation.”

The optimization cost function θ_0 is either the compliance

$$\Theta = \int_{\Omega} \nabla\boldsymbol{u} \cdot \mathbb{C}[\nabla\boldsymbol{u}]\, da \tag{24}$$

or the p-norm aggregation of the von Mises stress σ_{VM}

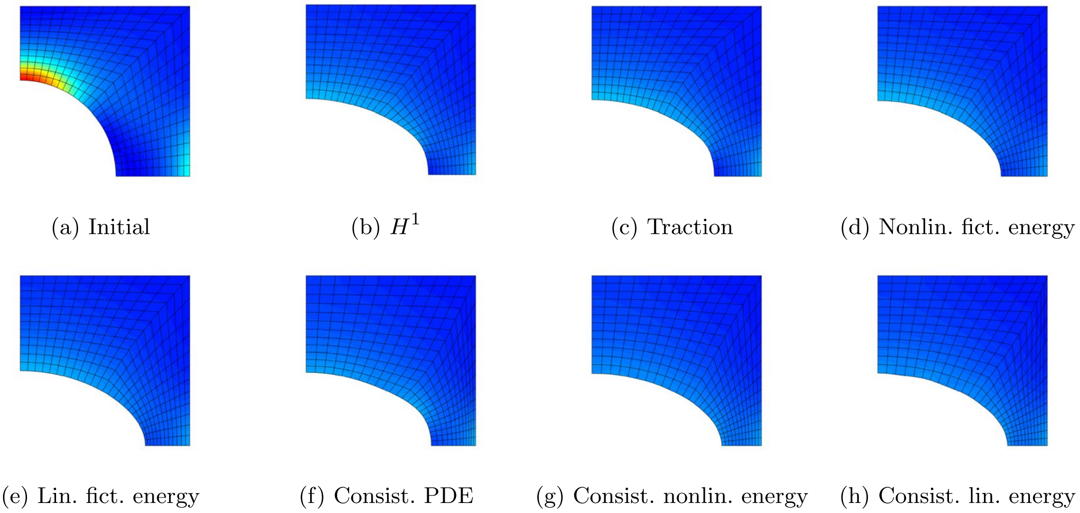
$$\Upsilon = \left(\int_{\Omega} \sigma_{VM}^{p}\, da \right)^{\frac{1}{p}} \tag{25}$$

in which $p = 6$ throughout Sect. 3. We use Θ and Υ to denote compliance and p-norm aggregation of the von Mises stress, respectively, throughout, and the superscript ^{0} to denote the metric value of the initial design; e.g., Θ^0 refers to the compliance of the initial design.

The interior-point method of Matlab’s fmincon with a limit of 100 function evaluations is used for the optimizations. The accuracies of the user supplied derivatives are verified for the fictitious energy and consistent approaches.

Table 1 Compliance minimization results

Method	Iteration	Θ/Θ^0	Area	Υ/Υ^0	KKT	Axes ratio	$D_i^{\min}$	$D_i^{\max}$
H^1	6/15	0.808	0.745	0.704	6.98×10^{-3}	1.615	– 0.117	0.158
Traction	6/15	0.814	0.743	0.713	8.99×10^{-3}	1.604	– 0.115	0.158
Fict. Energy $\gamma\,\Psi_f$	44/98	0.796	0.750	0.696	8.99×10^{-4}	1.640	– 0.120	0.163
Fict. Energy w/ lin $\gamma\,\mathbb{A}_f$	53/100	0.794	0.750	0.691	1.60×10^{-4}	1.670	– 0.122	0.173
Consistent w/ PDE	99/100	0.791	0.750	0.678	4.14×10^{-3}	1.778	– 19.8	20.0
Consistent w/ $\gamma\,\Psi_f$	59/100	0.790	0.750	0.670	2.54×10^{-4}	1.798	– 0.830	4.133
Consistent w/ lin $\gamma\,\mathbb{A}_f$	97/100	0.789	0.750	0.670	1.10×10^{-3}	1.790	– 8.98	13.4


(a) Initial (b) H^1 (c) Traction (d) Nonlin. fict. energy

(e) Lin. fict. energy (f) Consist. PDE (g) Consist. nonlin. energy (h) Consist. lin. energy

Fig. 3 Strain energy distributions of compliance optimized designs

None of the methods produce the expected elliptical hole with 2:1 axis ratio due to the finite domain size; however, they all tend in that direction.

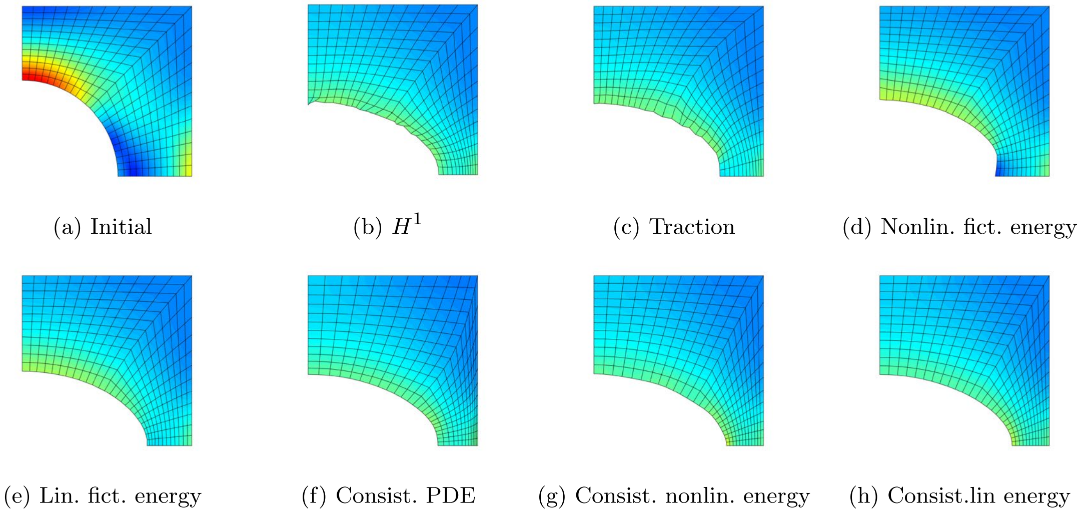
3.1 Compliance minimization

We first minimize the compliance $\theta_0 = \Theta$ of (24). This is an easy problem so all of the methods should and do perform admirably. Results from the various algorithms are presented in Table 1.

The fictitious energy values for the optimized designs obtained via the linear and nonlinear energy functions are 0.117 and 0.0271; hence, the maximum energy constraints are, respectively, violated and relatively active. The nonlinear energies for the fictitious energy and consistent approaches required some special care in that the Newton solver did not always converge. To resolve this, the load was incrementally applied over numerous load steps as needed; as such, the morphing analysis dominated the computational expense. Figure 3 illustrates the designs and their strain energy distributions. The contour plots qualitatively compare the methods with a consistent coloring scheme whereas the tables provide a quantitative comparison. The designs exhibit subtle differences which are seen through their energy distributions. The consistent methods with linearized energy filter $1/2\,\gamma\,\nabla\boldsymbol{d}\cdot\mathbb{A}_f[\nabla\boldsymbol{d}]$ and nonlinear energy filter $\gamma\,\Psi_f$ produce the best designs, although all of the designs are reasonable. The consistent method with nonlinear energy filter $\gamma\,\Psi_f$ produces the best mesh based on the "eyeball" norm. The inconsistent H^1 and traction methods converge in the fewest number of iterations because the change in the design between iterations is too small, presumably because the optimizer gets "confused" by the inconsistent sensitivity analysis, as evidenced by greater compliance values.

3.2 Stress minimization

We next minimize the p-norm of the von Mises stress, i.e., in the optimization, we replace (24) with (25) so that $\theta_0 = \Upsilon$. This is a more difficult problem as we replace the compliance which is a global quantity with an approximation of the maximum stress which is a local quantity. All of the


(a) Initial

(b) H^1

(c) Traction

(d) Nonlin. fict. energy

(e) Lin. fict. energy

(f) Consist. PDE

(g) Consist. nonlin. energy

(h) Consist.lin energy

Fig. 4 Von Mises stress distributions of stress optimized designs

Table 2 Stress minimization results

Method	Iteration	Θ/Θ^0	Area	Υ/Υ^0	KKT	Axes ratio	$D_i^{\min}$	$D_i^{\max}$
H^1	11/16	0.798	0.747	0.674	6.99×10^{-3}	1.887	– 0.156	0.206
Traction	23/99	0.792	0.749	0.678	2.13×10^{-3}	1.740	– 0.138	0.180
Fict. Energy $\gamma\,\Psi_f$	42/100	0.810	0.748	0.713	6.73×10^{-3}	1.510	– 0.116	0.138
Fict. Energy w/ lin $\gamma\,\mathbb{A}_f$	54/99	0.794	0.750	0.687	1.46×10^{-2}	1.738	– 0.125	0.173
Consistent w/ PDE	99/100	0.790	0.750	0.670	2.02×10^{-2}	1.821	– 19.6	19.9
Consistent w/ $\gamma\,\Psi_f$	34/98	0.792	0.749	0.670	3.65×10^{-3}	1.842	– 0.808	4.75
Consistent w/ lin $\gamma\,\mathbb{A}_f$	82/95	0.791	0.750	0.665	9.96×10^{-3}	1.857	– 2.90	9.40

other parameters are the same as those described above. Results from the various algorithms are denoted in Table 2 and Fig. 4. Descriptions of the table and figures are the same as those for the compliance minimization. The fictitious energy values for the optimized designs obtained via the linear and nonlinear energy functions are 0.115 and 0.0246; hence, the maximum energy constraints are again, respectively, violated and relatively active. The design obtained by the fictitious energy method with nonlinear energy is quite good; however, its nonlinear morphing analysis dominates the computational cost. All of the designs obtained by the consistent methods appear to be reasonable. And again the consistent method with nonlinear energy filter $\gamma\,\Psi_f$ produces the best mesh based on the "eyeball" norm. Anomalies exist in the designs obtained with the remaining methods. Parameter tuning could be used to improve these designs, but the point of this exercise is to find methods that are amenable to common nonlinear programming algorithms that do not require excessive parameter tuning.

4 Test problems

The advantages of consistent filtering methods were demonstrated in Sect. 3. Here, we further showcase the three consistent filtering approaches when used to design both 2D and 3D structures. Specifically, tradeoffs between ease of implementation, computational burden, and mesh quality are explored. A summary of relevant parameters used throughout this section is presented in Table 3. All results in this section used the MFEM library (Anderson et al. 2021) to solve the PDEs and the IPOPT library (Wächter and Biegler 2006) to solve the optimization problems. The Livermore Design Optimization (LiDO) code was utilized to compute the required sensitivities, and the Targeted Matrix

Table 3 Parameters used throughout Sect. 4

Description	Symbol	Value
Young's Modulus	E	10
Poisson's ratio	ν	0.3
p-norm exponent	p	10
KKT tolerance	ϵ_{kkt}	10^{-4}

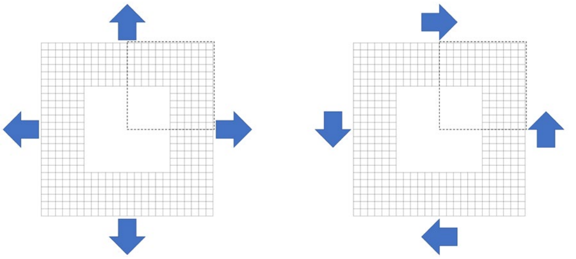
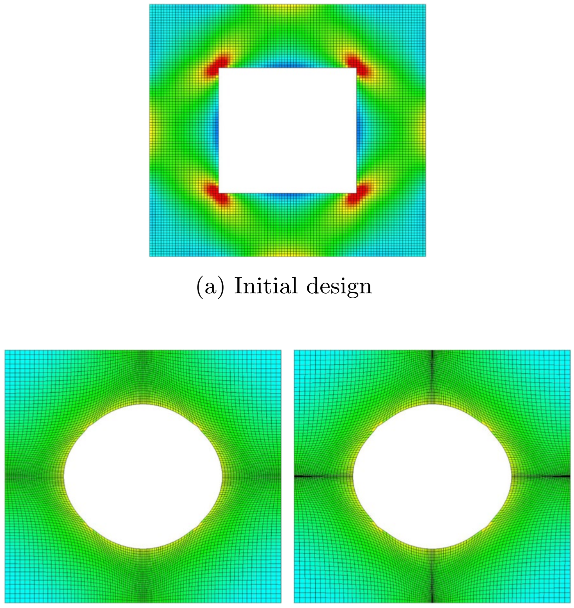

(a) Biaxial loading (b) Shear loading

Fig. 5 2D plate test problem, design/analysis symmetry cell outlined by dashed line

Optimization Paradigm (TMOP) (Knupp 2012; Dobrev et al. 2019; Barrera et al. 2022) is used to implement the consistent nonlinear energy filter.[7]

4.1 2D plate design

To compare the filtering methods, we revisit the design of the plane stress plate, but under biaxial or shear loading and with an initially square 1 × 1 hole rather than a circular hole, Fig. 5. Morphing the square into the optimal circle (biaxial loading) or diamond (shear loading) is challenging since it requires a right angle to be morphed into an arc or straight line, respectively. All optimizations presented in this section solve the stress minimization problem of Equations (22) and (25). We chose to minimize the maximum stress since it is generally more challenging than compliance, which could hide deficiencies in the methods.


(a) Initial design

(b) Optimal design for $\gamma = 10^{-1}$, $\underline{d}, \overline{d} \mp 0.4$

(c) Optimal design for $\gamma = 10^{1}$, $\underline{d}, \overline{d} \mp 25.0$

Fig. 6 Design of 2D plate under biaxial loading with the PDE filter

4.1.1 Consistent method with the PDE filter

We first consider the consistent PDE filter, which has one parameter γ that controls the tradeoff between how "smooth" the mesh is and how closely $\widetilde{\boldsymbol{d}}$ matches $\boldsymbol{d}$, as well as a set of parameter bounds $\underline{d}, \overline{d}$. A smaller γ requires smaller, i.e., more restrictive, parameter bounds to avoid mesh entanglement, while a larger γ permits more liberal parameter bounds.

Figure 6 illustrates how γ, $\underline{d}$, and $\overline{d}$ affect the optimized design. The two optimal designs boast similar performance with Υ/Υ^0 values of 0.38 and 0.39. The first takeaway is that we can use values of γ that are two orders of magnitude apart and obtain satisfactory results by tuning the $\underline{d}$ and $\overline{d}$ bounds. However, both of our optimized designs tend to "smash" elements along the symmetry planes, and further, this effect worsens for the larger value of γ. The advantage of this approach is the smooth boundary, which comes at the expense of element quality. In Fig. 7, we present the optimal design for shear loading, using the same $\gamma = 10^{-1}$ value that produced the higher quality mesh under biaxial loading. We again see a smooth boundary but notice many highly distorted elements near the symmetry planes. The stress contour plots, which use consistent color schemes for each load case, are provided to qualitatively compare designs, whereas Tables 4 and 5 quantitatively compare them.

To summarize, this PDE filter is simple to implement, especially for those familiar with the PDE filter from topology optimization. Indeed, repeated use of a

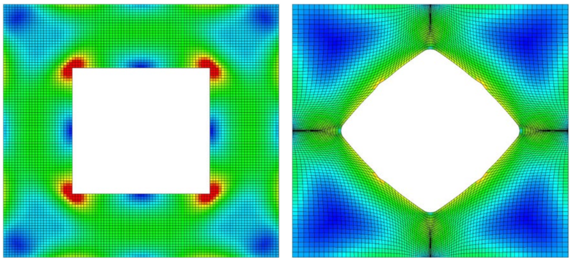
[7] In our implementation of TMOP (Barrera et al. 2022), their objective function of (8) $F(\boldsymbol{x}) = \sum_{E^t \in M} \int_{E^t} \mu(\boldsymbol{T})\, dv + w_\sigma \sum_{s \in S} \sigma^2(\boldsymbol{X}^s)$ is minimized with respect to the node coordinates of the physical mesh $\boldsymbol{X} = \boldsymbol{X}_0 + \widetilde{\boldsymbol{D}}$. In this expression, the so-called target elements E^t comprise all of the elements in the Ω_0 mesh, $\mu = \Psi_f$ is the quality metric and $\boldsymbol{T} = \widetilde{\boldsymbol{F}_d}$. TMOP constrains the nodes on A^f such that $\boldsymbol{X} \cdot \boldsymbol{e} = \boldsymbol{X}_0 \cdot \boldsymbol{e}$ and weakly enforces the $\widetilde{\boldsymbol{X}} = \boldsymbol{X}_0 + \boldsymbol{D}$ "constraint" via the penalty function $\sigma^2(\boldsymbol{X}^s) = |\widetilde{\boldsymbol{D}}^s - \boldsymbol{D}^s|^2$ for all nodes $s \in \Omega_0 \backslash A^f$ with the small penalty value $w_\sigma = 1/\gamma$. The TMOP metrics μ_2 and μ_{303} Dobrev et al. (2019) equal Ψ_f of (40) for 2 and 3 dimensions.

Table 4 Summary of 2D plate under biaxial loading optimization results

Filter	Init fig	Opt fig	Elements	Order	Dof	γ	$\underline{d}, \overline{d}\mp$	Optimal Υ/Υ^0
PDE	6a	6b	1,728	1	3650	0.1	0.4	0.38
PDE	6a	6c	1,728	1	3650	10.0	25	0.39
Lin energy	6a	8a	1,728	1	3650	0.1	0.075	0.40
Nonlin energy	9a	9b	108	1	266	0.1	0.1	0.56
Nonlin energy	9c	9d	432	1	962	0.1	0.1	0.46
Nonlin energy	9e	9f	1,728	1	3650	0.1	0.1	0.39
Nonlin energy	9g	9h	6,912	1	14,210	0.1	0.1	0.32
Nonlin energy	11a	11b	108	1	266	0.1	0.1	0.56
Nonlin energy	11c	11d	108	2	962	0.1	0.1	0.42
Nonlin energy	11e	11f	108	3	2090	0.1	0.1	0.40

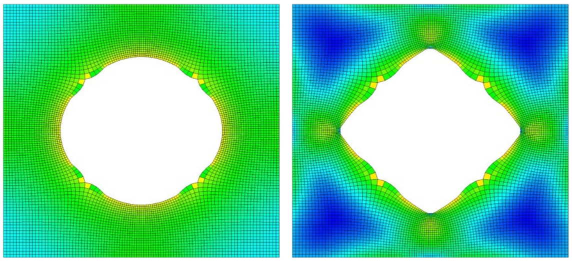
Table 5 Summary of 2D plate under shear loading optimization results

Filter	Init fig	Opt fig	Elements	Order	Dof	γ	$\underline{d}, \overline{d}\mp$	Optimal Υ/Υ^0
PDE	7a	7b	1728	1	3650	0.1	0.4	0.34
Lin energy	7a	8b	1728	1	3650	0.1	0.075	0.36
Nonlin energy	10a	10b	108	1	266	0.1	0.1	0.50
Nonlin energy	10c	10d	432	1	962	0.1	0.1	0.41
Nonlin energy	10e	10f	1728	1	3650	0.1	0.1	0.34
Nonlin energy	10g	10h	6912	1	14,210	0.1	0.1	0.28
Nonlin energy	12a	12b	108	1	266	0.1	0.1	0.50
Nonlin energy	12c	12d	108	2	962	0.1	0.1	0.37
Nonlin energy	12e	12f	108	3	2090	0.1	0.1	0.35



(a) Initial design (b) Optimal design

Fig. 7 Design of 2D plate under shear loading with the PDE filter



(a) Biaxial loading (b) Shear loading

Fig. 8 Optimal 2D plate designs obtained via the linearized energy filter

scalar-valued PDE filter could be used to compute each component of $\widetilde{\boldsymbol{d}}$, although a vector-valued implementation was used here. This method is computationally inexpensive since it only requires linear PDE solves. Optimizations using this filter generally produce smooth boundaries at the expense of poor element quality and consequently reduced simulation accuracy.

4.1.2 Consistent method with the linearized energy filter

The linearized energy filter promotes higher element quality than the PDE filter as seen in the designs for biaxial and shear loading illustrated in Fig. 8. Specifically, the elements are nearly squareshaped throughout the domain. Notably, the high mesh quality near the symmetry planes is in stark contrast to the results in Sect. 4.1.1; however, the tradeoff is that the optimizer struggles to remove the "corner" in the initial design's square hole. Thus, the maximum von Mises ratios in Fig. 8 are slightly higher than their counterparts in Sect. 4.1.1, i.e., 0.40 and 0.36 for the designs in this section, compared with 0.38 and 0.34 for the designs in Sect. 4.1.1.

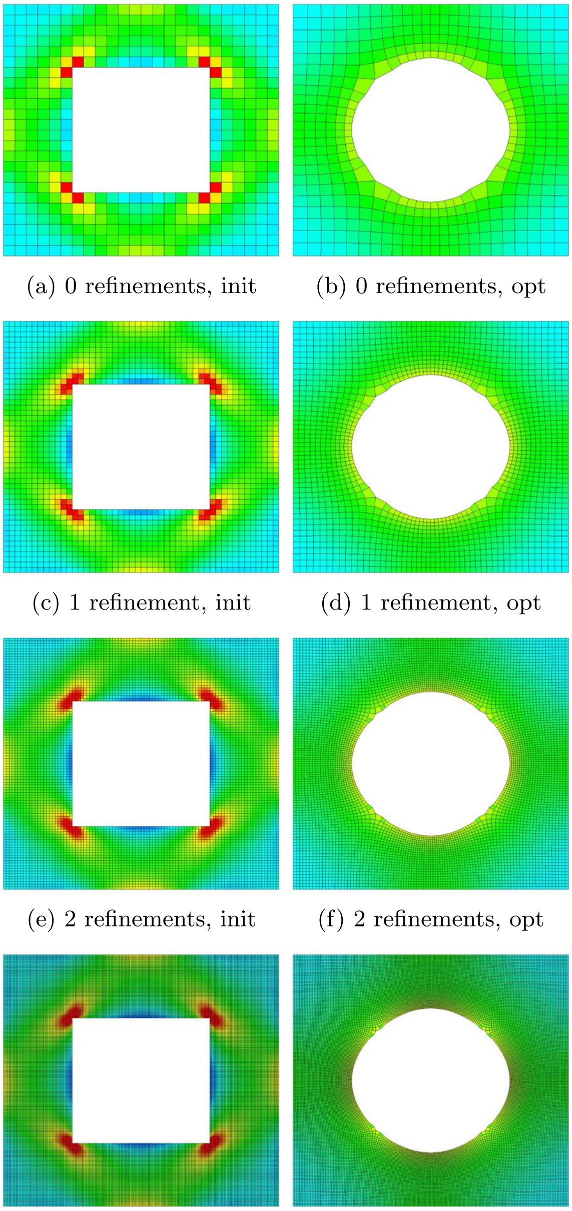
When compared to the PDE filter, we see that the linearized energy filter ensures higher mesh quality at the expense of design freedom. It is also simple to implement, especially since a linear elasticity solver can be re-purposed for this filter with proper selection of the Lamé parameters, i.e., $\lambda = -\frac{1}{3}$ and $\mu = \frac{1}{2}$ (see Appendix 3 for details), and the computational expense is therefore equivalent to a single linear elasticity PDE solve.

4.1.3 Consistent method with the nonlinear energy filter

Finally, we consider the nonlinear energy filter (with quality metric μ_2 from Dobrev et al. (2019) which is the same as the Ψ_f function of (40)). Although this filter is more computationally expensive, it proves to be more capable than the linear filters presented in Sects. 4.1.1 and 4.1.2. The robustness of the formulation is demonstrated by the mesh refinement study for the biaxial loaded design illustrated in Fig. 9. Indeed, the same parameters, $\gamma = 10^{-1}, \underline{d}, \overline{d} \mp 0.1$, were used to obtain each result, despite their differing element size. Additionally, we see very similar optimized designs, demonstrating the mesh independence of this technique. Similar observations hold for the mesh refinement study performed on the shear loaded designs, Fig. 10. Notably, the same values of γ, $\underline{d}$, and $\overline{d}$ were used for both loadings, and again, we see consistent designs for the different mesh resolutions. When compared to the consistent PDE filter in Sect. 4.1.1, we notice higher element quality throughout Figs. 9 and 10, and when compared to the linearized energy filter in Sect. 4.1.2, we notice a smoother boundary. The nonlinear energy filter produces the lowest maximum stress in the optimal design when comparing equivalent meshes across the three consistent filters, as listed in Tables 4 and 5. Thus, the nonlinear filter exhibits the advantages of both linear filters, i.e., high mesh quality and smooth boundaries, at the cost of additional computational burden.

As an alternative to increasing the number of elements, i.e., h-refinement, simulation accuracy can also be improved by increasing the polynomial order of the mesh, i.e., p-refinement. Figures 11 and 12 present optimal designs corresponding to linear, quadratic, and cubic elements. The p-refinement study used the same values of γ, $\underline{d}$, and $\overline{d}$ as the h-refinement study. The sharp corner is much improved when using quadratic elements, and nearly entirely removed when using cubic elements. Further, the element quality is high throughout the domain. The designs obtained using the cubic meshes and the nonlinear energy filter appear to be the best when considering mesh quality, design performance, and number of degrees-of-freedom.

The consistent nonlinear energy filter can be summarized as an extremely effective, but computationally expensive alternative to the consistent linear filters. Indeed morphing computations may dominate the physics computations, especially if the latter are linear. Additionally, the implementation of a nonlinear filter is more difficult than the linear filters. However, the improved optimization performance may warrant the added effort. Further, the combination of a smooth boundary representation and high element quality throughout the domain, especially when using meshes with higher-order elements, is unmatched by the other techniques. As design problems require more complex physics, the improved element quality obtained via the nonlinear energy filter may become even more important for accurate response predictions. Moreover, the computational cost will



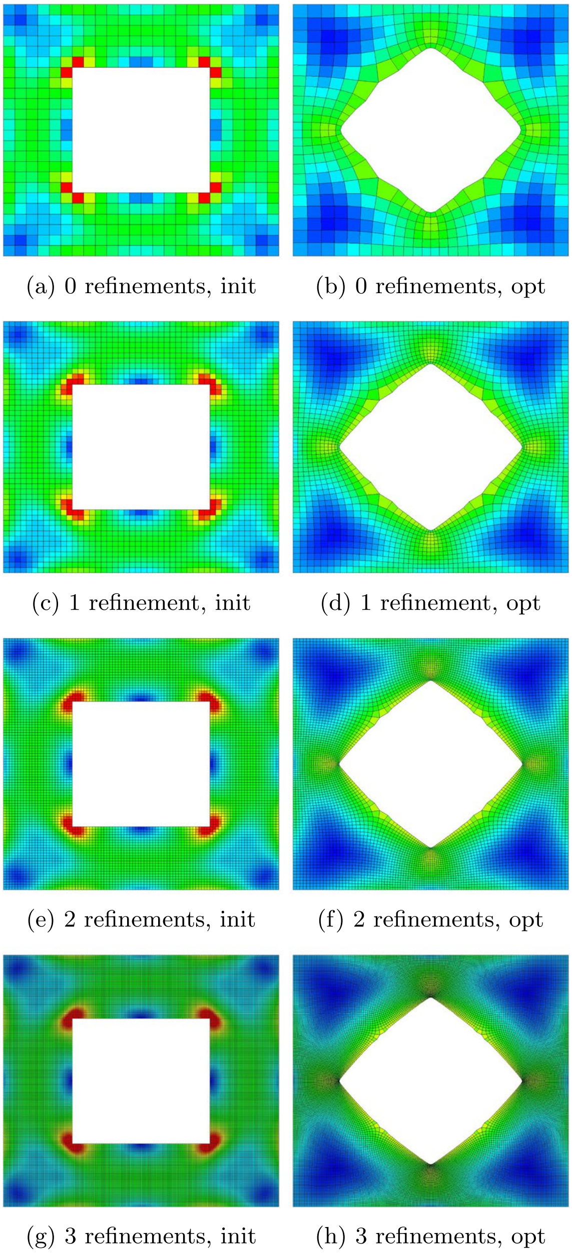
(a) 0 refinements, init (b) 0 refinements, opt

(c) 1 refinement, init (d) 1 refinement, opt

(e) 2 refinements, init (f) 2 refinements, opt

(g) 3 refinements, init (h) 3 refinements, opt

Fig. 9 Effect of h-refinement on the design of 2D plate under biaxial loading with the nonlinear energy filter. Left column is initial (init) designs, and right column is optimized (opt) designs


(a) 0 refinements, init (b) 0 refinements, opt

(c) 1 refinement, init (d) 1 refinement, opt

(e) 2 refinements, init (f) 2 refinements, opt

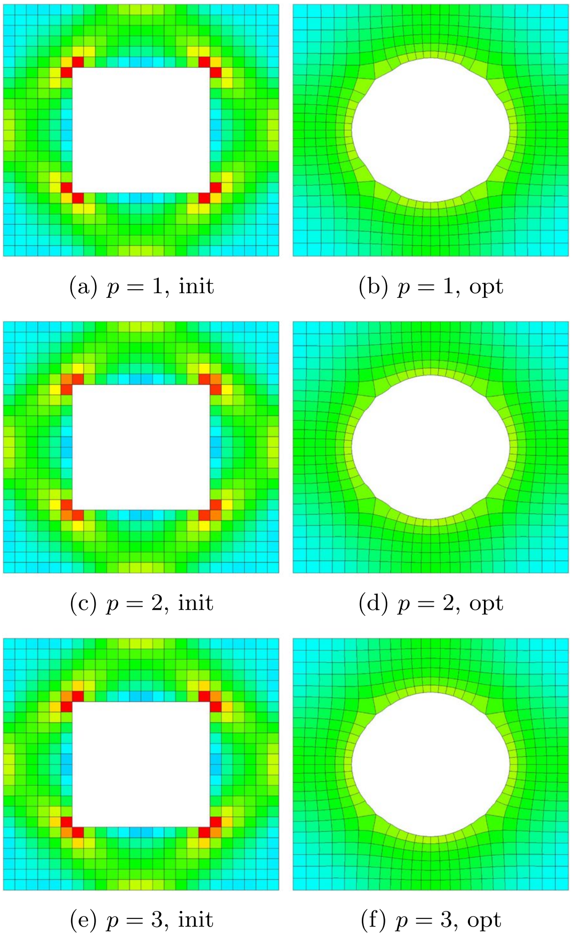
(g) 3 refinements, init (h) 3 refinements, opt

Fig. 10 Effect of h-refinement on the design of 2D plate under shear loading with the nonlinear energy filter. Left column is initial (init) designs, and right column is optimized (opt) designs

be less significant when compared to the cost of the complex nonlinear physics simulations.

4.2 3D dam design

We use the design of a 3D dam subject to a design-dependent hydrostatic pressure load to further challenge the optimization scheme. Indeed, the direction of the applied load changes as the mesh morphs. The problem is summarized in Fig. 13 by overlaying the pressure load and mechanical displacement boundary conditions on the mesh of the initial design. As pictured, we enforce homogeneous essential boundary conditions, i.e., $\boldsymbol{u} \cdot \boldsymbol{n} = 0$, on the surfaces with normals $\boldsymbol{n} = -\boldsymbol{e}_2$ and $\boldsymbol{n} = \pm\boldsymbol{e}_3$. We also enforce design constraints by fixing the normal displacement $\boldsymbol{d} \cdot \boldsymbol{n} = 0$ over the surfaces with normals $\boldsymbol{n} = \pm\boldsymbol{e}_2$ and $\boldsymbol{n} = \pm\boldsymbol{e}_3$. The initial design has dimensions [200, 700, 1200] in the $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, and $\boldsymbol{e}_3$ directions, respectively. We consider both compliance and maximum stress minimization.


(a) $p = 1$, init (b) $p = 1$, opt

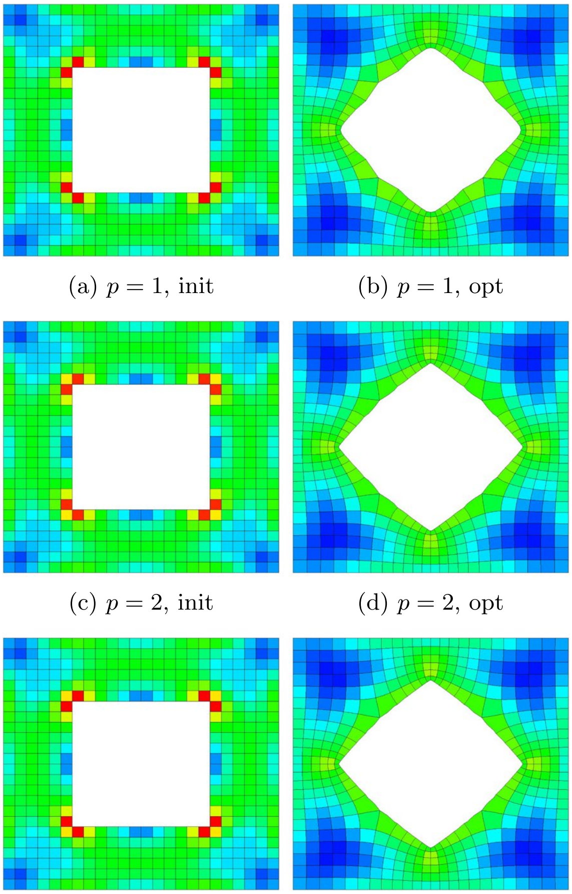
(c) $p = 2$, init (d) $p = 2$, opt

(e) $p = 3$, init (f) $p = 3$, opt

Fig. 11 Effect of p-refinement on the design of 2D plate under biaxial loading with the nonlinear energy filter. Left column is initial (init) designs, and right column is optimized (opt) designs. p denotes the order of the solution basis functions

4.2.1 Consistent method with the PDE filter

The PDE filter achieves satisfactory results for the 3D dam design problem. The minimum compliance design pictured in Fig. 14 has 50% of the initial design's compliance. Element quality is difficult to qualitatively compare for 3D structures, but we note that there are highly distorted elements near the bottom corners of the design.



(a) $p = 1$, init (b) $p = 1$, opt

(c) $p = 2$, init (d) $p = 2$, opt

(e) $p = 3$, init (f) $p = 3$, opt

Fig. 12 Effect of p-refinement on the design of 2D plate under shear loading with the nonlinear energy filter. Left column is initial (init) designs, and right column is optimized (opt) designs. p denotes the order of the solution basis functions

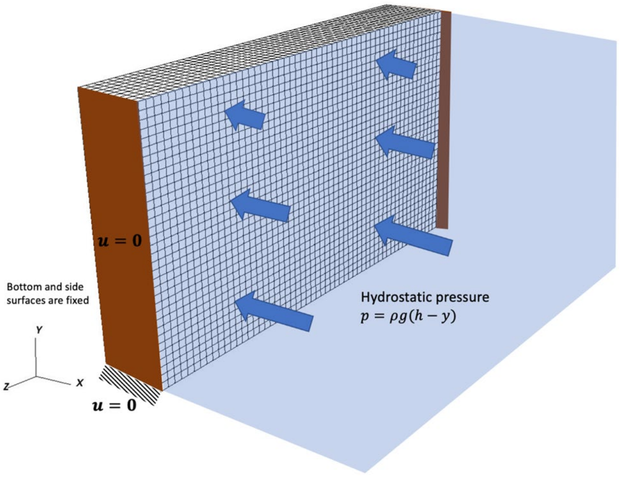
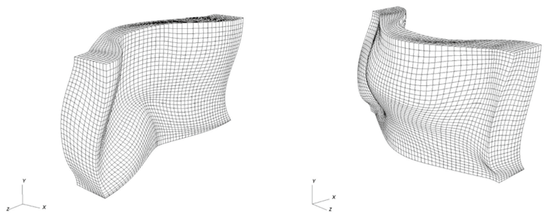
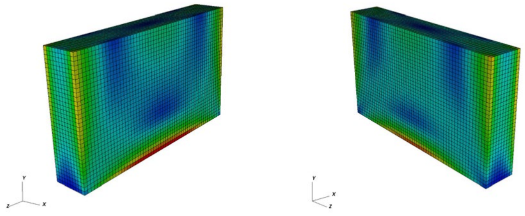


Fig. 13 Dam design problem loads and boundary conditions applied to the initial design



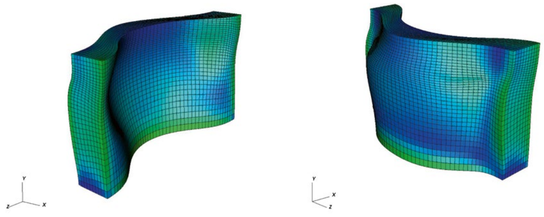
(a) Front view (b) Back view

Fig. 14 Compliance min. with the PDE filter



(a) Initial stress, front (b) Initial stress, back

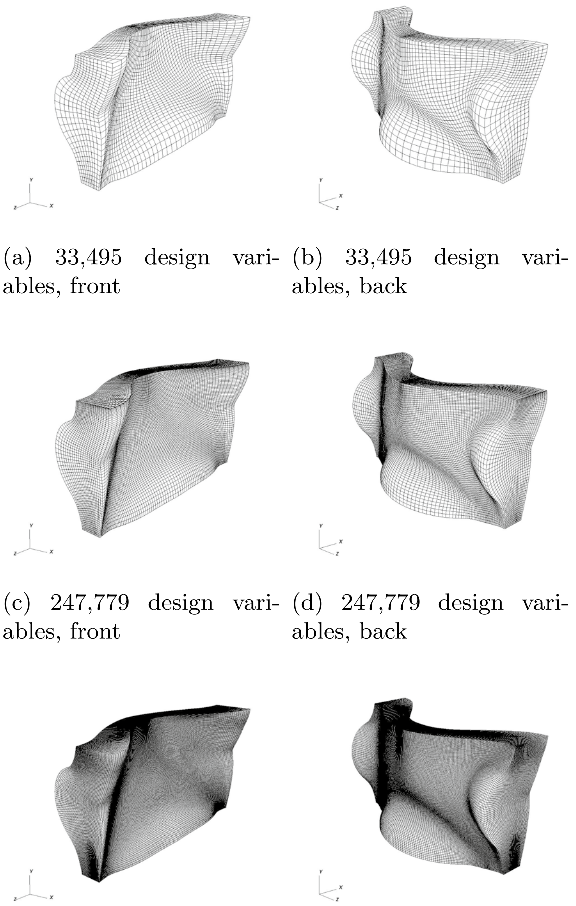
Fig. 15 Initial von Mises stress distribution



(a) Optimal stress, front (b) Optimal stress, back

Fig. 16 Maximum von Mises stress minimization with the PDE filter

A minimization of the maximum von Mises stress produces a design that resembles the compliance minimization design. The initial and optimized von Mises stress distributions are presented in Figs. 15 and 16, respectively. The optimal design reduces the peak von Mises stress by nearly half; however, this filter provides the worst performance of the three consistent filters, Table 7. Generally, the PDE filter is not able to displace the initial mesh as far as the energy-based filters before mesh entanglement occurs. Since this filter simply "smooths" the node coordinates without regard for element quality, the elements become distorted quite easily and therefore the optimal performance is limited by how far the mesh is able to distort before it tangles.

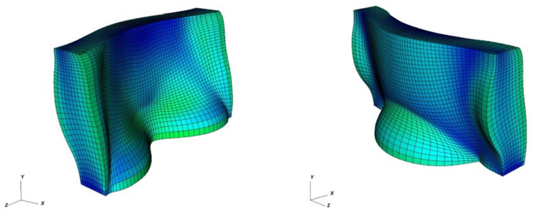


(a) 33,495 design variables, front (b) 33,495 design variables, back

(c) 247,779 design variables, front (d) 247,779 design variables, back

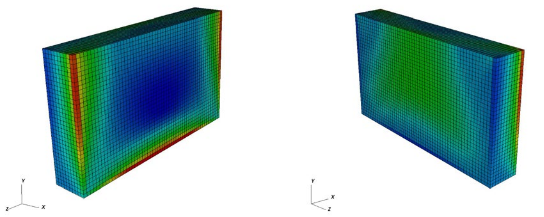
(e) 1,904,163 design variables, front (f) 1,904,163 design variables, back

Fig. 17 Effect of h-refinement on compliance minimization with the linearized energy filter



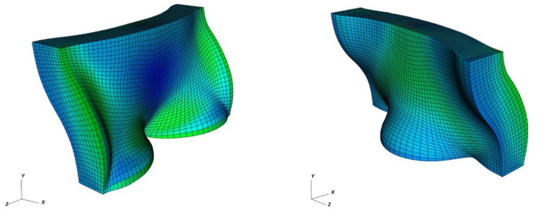
(a) Optimal stress, front (b) Optimal stress, back

Fig. 18 Maximum von Mises stress minimization with the linearized energy filter



(a) Initial stress, front (b) Initial stress, back

Fig. 19 Initial Drucker-Prager stress distribution



(a) Optimal stress, front (b) Optimal stress, back

Fig. 20 Maximum Drucker-Prager stress minimization with the linearized energy filter

4.2.2 Consistent method with linearized energy

The linearized energy filter was effective in the dam optimization. An h-refinement study of the compliance minimization problem was performed to investigate the mesh independence of the shape optimization framework, Fig. 17. Two important behaviors were observed: the same filter and bound parameters were effective for all three meshes, and the same performance, i.e., $\Theta/\Theta_0^0 = 0.42$ was obtained for all three designs. The final mesh contained 1,904,163 design variables, demonstrating the scalability of the method to large design problems. *The robustness provided by mesh-independent filter parameters allows designers to tune optimization parameters on coarser meshes before using them on refined meshes.*

The linearized energy filter produced a design with less compliance than the PDE filter, Table 6. This trend was consistent with the minimum von Mises stress designs. Inspection of Fig. 18 reveals a superior design to that in Fig. 16; the linearized energy filter allows the mesh to morph further while keeping elements from tangling and therefore allows the final design to mitigate stress more effectively.

The astute civil engineer might notice, however, that our optimal dams are to this point concave with respect to the hydrostatic pressure load, whereas real-world dams are often convex with respect to the water they are retaining. The main driver of this inconsistency is the von Mises stress metric, which assumes materials are equally strong in tension and compression. Thus, a better stress metric to design for is the Drucker-Prager criterion (Drucker and Prager 1952) since

Table 6 Summary of 3D dam compliance minimization results

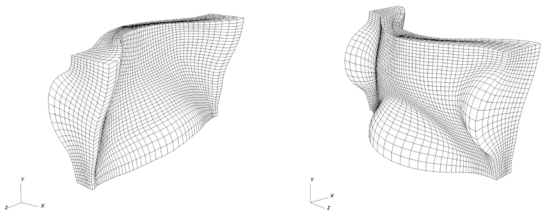
Filter	Init fig	Opt fig	Elements	Order	Dof	γ	$\underline{d}, \overline{d}\mp$	Optimal Θ/Θ^0
PDE	13	14	9520	1	33,495	10^5	700	0.50
Lin energy	13	17a, 17b	9520	1	33,495	10^5	1400	0.42
Lin energy	n/a	17c, 17d	76,160	1	247,779	10^5	1400	0.42
Lin energy	n/a	17e, 17f	609,280	1	1,904,163	10^5	1400	0.42
Nonlin energy	13	21	9520	1	33,495	10^5	14×50	0.39

Table 7 Summary of 3D dam von Mises stress optimization results

Filter	Init fig	Opt fig	Elements	Order	Dof	γ	$\underline{d}, \overline{d}\mp$	Optimal Υ/Υ^0
PDE	15	16	9520	1	33,495	10^5	500	0.57
Lin energy	15	18	9520	1	33,495	10^5	1000	0.45
Nonlin energy	15	22	9520	1	33,495	10^5	13×50	0.37

Table 8 Summary of 3D dam Drucker-Prager stress optimization results

Filter	Init fig	Opt fig	Elements	Order	Dof	γ	$\underline{d}, \overline{d}\mp$	Optimal Υ/Υ^0
Lin energy	19	20	9520	1	33,495	10^5	800	0.33
Nonlin energy	19	23	9520	1	33,495	10^5	13×50	0.19



(a) Front view (b) Back view

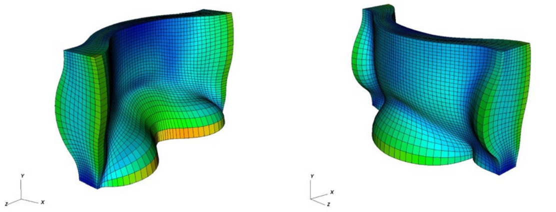
Fig. 21 Compliance minimization with nonlinear energy filter

most dams are made of concrete, which is strong in compression and weak in tension. If we assume that our constituent material is 10 times stronger in compression than in tension and use the Drucker-Prager stress criterion, we obtain the initial stress field pictured in Fig. 19. Designing for the Drucker-Prager criterion has a fascinating effect on the optimal geometry; the top surface of the dam switches from concave to convex with respect to the hydrostatic pressure load, which is a trait of arch dams, and two reinforcement columns appear on the front face, which is a trait of buttress dams, Fig. 20. In summary, when we account for the type of materials typically used to build dams, the optimized design resembles dams seen in the real world.

4.2.3 Consistent method with nonlinear energy

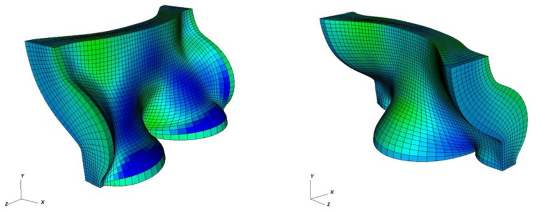
We attempted to use the nonlinear energy filter (with quality metric μ_{303} from Dobrev et al. (2019) which equals Ψ_f from (40)) in the same manner as in Sect. 4.1. A value was selected for γ and then the parameter bounds $\underline{d}, \overline{d}$ were tuned accordingly. Through trial and error, it was determined that this technique was not effective when using the nonlinear energy filter for the 3D dam optimization problem. Specifically, the optimal dam design required a much larger mesh distortion than the optimal plate design, which caused convergence issues during the filtering operation. Thus, a slightly different technique was developed for this problem. Rather than selecting a large bound range $[\underline{d}, \overline{d}]$, we follow Scherer et al. (2010) and solve a series of optimizations with smaller ranges, wherein the initial design for each subsequent optimization was the optimized design from the previous optimization. In this way, the design was allowed to distort incrementally through many, more constrained, design problems. The key difference between solving the design problem in a single pass versus breaking it into numerous steps, as outlined here, is that the reference element is incrementally updated in the latter technique which introduces inconsistencies with respect to the initial design optimization problem.

Figure 21 depicts the optimized design for compliance minimization using the nonlinear energy filter. The final design is superior to those obtained with the linear filters, Table 6. Similarly, the best designs with respect to von Mises and Drucker-Prager metrics were obtained using the nonlinear energy filter with the incremental design process, Tables 7 and 8. The final designs are displayed in Figs. 22 and 23, for von Mises and Drucker-Prager metrics, respectively. The design stress field histories for the Drucker-Prager minimization are presented in Fig. 24 to



(a) Optimal stress, front (b) Optimal stress, back

Fig. 22 Maximum von Mises stress minimization with nonlinear energy filter

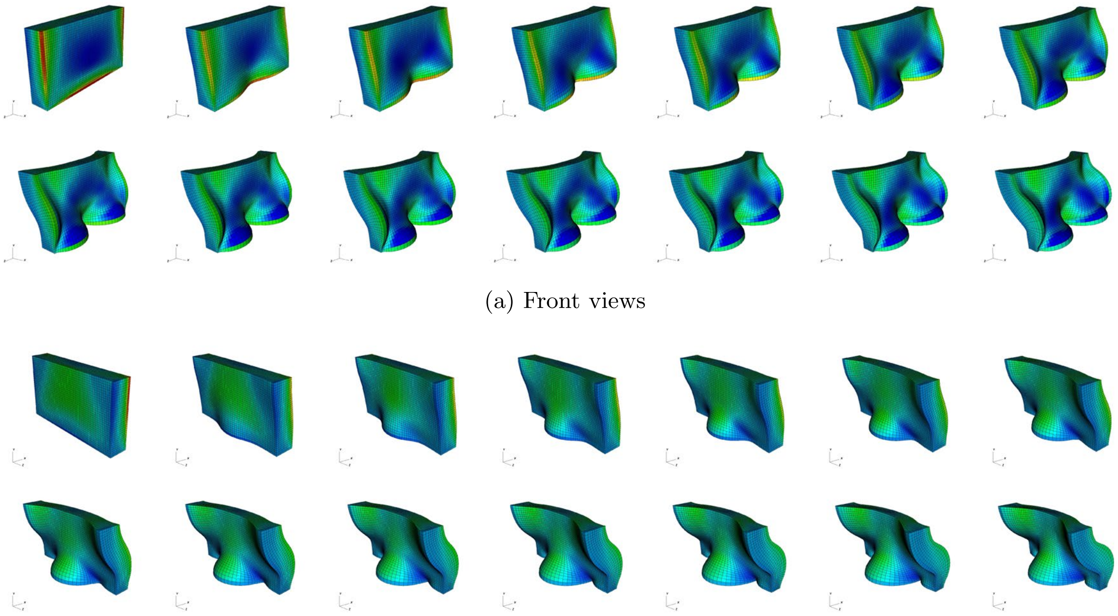


(a) Optimal stress, front (b) Optimal stress, back

Fig. 23 Maximum Drucker-Prager stress minimization with nonlinear energy filter

demonstrate the sequential optimization process described above. We emphasize that although 14 optimizations were required to obtain optimal designs, each of the optimizations converged quickly due to the smaller range $[\underline{d}, \overline{d}]$. In fact, the total number of design iterations for all of the design steps combined was similar to that required for the single-pass optimizations in Sect. 4.2.2. That said, the overall computational cost remains higher for the nonlinear filter since each design iteration requires the solution of the nonlinear filter PDE.

Again, we emphasize the tradeoff between computational cost and optimization performance must be considered. For example, the design problems using the highly refined meshes in Figs. 17e and 17f would have an increased execution time, perhaps an order of magnitude more, if using the nonlinear energy filter compared to the linearized energy filter to obtain only marginal compliance improvement, i.e., Θ/Θ^0 of 0.42 vs. 0.39. However, the design for Drucker-Prager stress using the nonlinear energy filter was significantly more effective than that obtained using the linear filter, i.e., Θ/Θ^0 of 0.33 vs. 0.19, possibly justifying the increased computational cost.



(a) Front views

(b) Back views

Fig. 24 History of Drucker-Prager stress minimization designs with $\underline{d}, \overline{d} \mp 50$

5 Conclusions

Shape optimization provides an attractive option over topology optimization, especially when the boundary phenomena dominate the design, e.g., in stress-based design. Shape optimizations typically use CAD-based parameterizations which move the positions of the finite elements through globally supported basis functions. This type of design description can be difficult to implement for industry applications and limits the design to these predefined parameterizations.

Independent node parameterization has been extensively used, but numerical difficulties, e.g., mesh entanglement, still pose a challenge. Significant progress has been made to alleviate this challenge and here we add to this body of work. Specifically, we demonstrate that consistent filtering, i.e., filtering both updated nodal positions and their sensitivities, is advantageous over filtering only sensitivities. We unify a technique for parameterizing locally supported finite element quantities, i.e., element "densities" in topology optimization and nodal coordinates in shape optimization, to minimize human intervention in the design process.

We demonstrate consistent filtering by designing a 2D plate under biaxial and shear loading and a 3D dam subject to a hydrostatic pressure load. We show that the consistent PDE, linearized energy, and nonlinear energy filters all produce satisfactory designs, although each has pros and cons relative to the others. In summary, the PDE filter maintains design regularity, is computationally inexpensive and simple to implement, but suffers from poor element quality. The linearized energy filter is computationally inexpensive and simple to implement and produces higher element quality at the expensive of less design freedom. Finally, the nonlinear energy filter exhibits superior performance by simultaneously ensuring design freedom and high element quality, but incurs a steep computational cost. The choice of an appropriate filter is therefore problem dependent; the tradeoff between design freedom, ease of implementation, and computational cost must be analyzed on a case-by-case basis.

Appendix 1: Independent node coordinate derivations

The variation of θ at $\boldsymbol{x}$ with respect to the shape perturbation $\delta\boldsymbol{x}$ is expressed as the sum of an *explicit* variation $\delta\theta^E$ and *implicit* variation $\delta\theta^I$, i.e., $\delta\theta = \delta\theta^E + \delta\theta^I$. The former is expressed

$$
\begin{aligned}
&\delta\theta^E(\boldsymbol{x};\delta\boldsymbol{x})\\
&\quad= \int_\Omega \Big((-\nabla^T\boldsymbol{u}\,\frac{\partial\Lambda}{\partial\nabla\boldsymbol{u}} + \Lambda\boldsymbol{I})\cdot\nabla\delta\boldsymbol{x} + \frac{\partial\Lambda}{\partial\boldsymbol{x}}\cdot\delta\boldsymbol{x}\Big)dv\\
&\quad+ \int_{A^t}\Big(\frac{\partial\tau}{\partial\boldsymbol{x}}\cdot\delta\boldsymbol{x} + \tau\,(\boldsymbol{I}-\boldsymbol{n}\otimes\boldsymbol{n})\cdot\nabla\delta\boldsymbol{x}\Big)\,da\\
&\quad+ \int_{A^p}\Big(\Big(\frac{\partial\boldsymbol{\tau}}{\partial\boldsymbol{x}}\Big)^T\boldsymbol{n}\cdot\delta\boldsymbol{x}\\
&\quad+((\boldsymbol{\tau}\cdot\boldsymbol{n})\boldsymbol{I}-(\boldsymbol{n}\otimes\boldsymbol{\tau}))\cdot\nabla\delta\boldsymbol{x})\,da
\end{aligned}
\tag{26}
$$

and the latter as

$$
\begin{aligned}
\delta\theta^I(\boldsymbol{x};\overset{\star}{\boldsymbol{u}}) &= \int_\Omega\Big(\frac{\partial\Lambda}{\partial\nabla\boldsymbol{u}}\cdot\nabla\overset{\star}{\boldsymbol{u}} + \frac{\partial\Lambda}{\partial\boldsymbol{u}}\cdot\overset{\star}{\boldsymbol{u}}\Big)\,dv\\
&\quad+\int_{A^t}\overset{\star}{\boldsymbol{u}}\cdot\frac{\partial\tau}{\partial\boldsymbol{u}}da + \int_{A^p}\overset{\star}{\boldsymbol{u}}\cdot\Big(\frac{\partial\boldsymbol{\tau}}{\partial\boldsymbol{u}}\Big)^T\boldsymbol{n}\,da
\end{aligned}
\tag{27}
$$

To obtain the above, we define the *material derivative* with respect the *design velocity* $\delta\boldsymbol{x}$ as $\overset{\star}{\boldsymbol{u}}(\boldsymbol{x}) = \delta\boldsymbol{u}(\boldsymbol{x};\delta\boldsymbol{x}) + \nabla\boldsymbol{u}(\boldsymbol{x})\,\delta\boldsymbol{x}$; it is implicitly defined by (1).[8] We also use the commuter identity $\overset{\star}{\overline{\nabla\boldsymbol{u}}} = -\nabla\boldsymbol{u}\nabla\delta\boldsymbol{x} + \nabla\overset{\star}{\boldsymbol{u}}$.

In the adjoint sensitivity method, we annihilate the implicit variation $\delta\theta^I$. To do this, we take the variation of the residual Eq. (1) and, as above, express it as the sum of an explicit variation δr^E and implicit variation δr^I, i.e., $\delta r = \delta r^E + \delta r^I$. The explicit and implicit variations are

$$
\begin{aligned}
&\delta r^E(\boldsymbol{u},\boldsymbol{w},\boldsymbol{x};\delta\boldsymbol{x}) =\\
&\int_\Omega\big((-\nabla^T\boldsymbol{w}\,\mathbb{C}[\nabla\boldsymbol{u}] - \nabla^T\boldsymbol{u}\,\mathbb{C}[\nabla\boldsymbol{w}]\\
&+(\nabla\boldsymbol{w}\cdot\mathbb{C}[\nabla\boldsymbol{u}] - \boldsymbol{w}\cdot\boldsymbol{b})\boldsymbol{I})\cdot\nabla\delta\boldsymbol{x}\\
&+\Big(\Big(\frac{\partial\mathbb{C}[\nabla\boldsymbol{u}]}{\partial\boldsymbol{x}}\Big)^T[\nabla\boldsymbol{w}] - \Big(\frac{\partial\boldsymbol{b}}{\partial\boldsymbol{x}}\Big)^T\boldsymbol{w}\Big)\cdot\delta\boldsymbol{x}\Big)\,dv\\
&-\int_{A^t}\Big(\Big(\frac{\partial\boldsymbol{t}^p}{\partial\boldsymbol{x}}\Big)^T\boldsymbol{w}\cdot\delta\boldsymbol{x}\\
&+(\boldsymbol{w}\cdot\boldsymbol{t}^p)\,(\boldsymbol{I}-\boldsymbol{n}\otimes\boldsymbol{n})\cdot\nabla\delta\boldsymbol{x}\,)\,da\\
&-\int_{A^p}\Big(\Big(\frac{\partial p}{\partial\boldsymbol{x}}\otimes\boldsymbol{w}\Big)\boldsymbol{n}\cdot\delta\boldsymbol{x} + p\,((\boldsymbol{w}\cdot\boldsymbol{n})\boldsymbol{I}\\
&-(\boldsymbol{n}\otimes\boldsymbol{w}))\cdot\nabla\delta\boldsymbol{x})\,da
\end{aligned}
\tag{28}
$$

and

[8] In the material derivative context, $\overset{\star}{\boldsymbol{u}}(\boldsymbol{x},t) = \partial\boldsymbol{u}(\boldsymbol{x},t)/\partial t + \nabla\boldsymbol{u}(\boldsymbol{x},t)\,\boldsymbol{v}(\boldsymbol{t})$ where $\boldsymbol{v}$ is the *design velocity field* and t is the pseudo time. The $\partial\boldsymbol{u}(\boldsymbol{x},t)/\partial t$ term accounts for the change in $\boldsymbol{u}$ due to the shape perturbation for a fixed *location* in space $\boldsymbol{x}$ whereas the $\nabla\boldsymbol{u}(\boldsymbol{x},t)\,\boldsymbol{v}(\boldsymbol{t})$ term accounts for the movement of the material point.

$$\begin{aligned}\delta r^{I}(\boldsymbol{u},\boldsymbol{w},\boldsymbol{u};\overset{\star}{\boldsymbol{u}}) &= \int_{\Omega} \nabla\boldsymbol{w}\cdot\mathbb{C}[\nabla\overset{\star}{\boldsymbol{u}}]\,dv\\ &= a(\overset{\star}{\boldsymbol{u}},\boldsymbol{w})\end{aligned} \tag{29}$$

Now we subtract the zero $\delta r = \delta r^{E} + \delta r^{I} = 0$ from $\delta\theta = \delta\theta^{E} + \delta\theta^{I}$ and rearrange to obtain

$$\begin{aligned}\delta_{i}\theta(\boldsymbol{x},\delta\boldsymbol{x}) &= \delta\theta^{E}(\boldsymbol{x},\delta\boldsymbol{x})\\ &\quad - \delta r^{E}(\boldsymbol{u},\boldsymbol{w},\boldsymbol{x};\delta\boldsymbol{x})\\ &\quad + \left(\delta\theta^{I}(\boldsymbol{x},\overset{\star}{\boldsymbol{u}}) - \delta r^{I}(\boldsymbol{u},\boldsymbol{w},\boldsymbol{u};\overset{\star}{\boldsymbol{u}})\right)\end{aligned} \tag{30}$$

To eliminate the implicit variations $\delta\theta^{I}(\boldsymbol{x},\overset{\star}{\boldsymbol{u}}) - \delta r^{I}(\boldsymbol{u},\boldsymbol{w},\boldsymbol{u};\overset{\star}{\boldsymbol{u}})$, we solve the following adjoint equation for the heretofore arbitrary $\boldsymbol{w}\in\mathcal{H}(\boldsymbol{0})$

$$\begin{aligned}0 &= a(\overset{\star}{\boldsymbol{u}},\boldsymbol{w})\\ &\quad - \int_{\Omega}\left(\nabla\overset{\star}{\boldsymbol{u}}\cdot\frac{\partial\Lambda}{\partial\nabla\boldsymbol{u}} + \overset{\star}{\boldsymbol{u}}\cdot\frac{\partial\Lambda}{\partial\boldsymbol{u}}\right)dv\\ &\quad - \int_{A^{t}}\overset{\star}{\boldsymbol{u}}\cdot\frac{\partial\tau}{\partial\boldsymbol{u}}\,da\\ &\quad - \int_{A^{p}}\overset{\star}{\boldsymbol{u}}\cdot\left(\frac{\partial\boldsymbol{\tau}}{\partial\boldsymbol{u}}\right)^{T}\boldsymbol{n}\,da\end{aligned} \tag{31}$$

for all $\overset{\star}{\boldsymbol{u}}\in\mathcal{H}(\boldsymbol{0})$. As such, the sensitivity reduces to

$$\begin{aligned}\delta\theta(\boldsymbol{x};\delta\boldsymbol{x}) &= \delta\theta^{E}(\boldsymbol{x};\delta\boldsymbol{x})\\ &\quad - \delta r^{E}(\boldsymbol{u},\boldsymbol{w},\boldsymbol{x};\delta\boldsymbol{x}).\end{aligned} \tag{32}$$

Note that $\delta\theta$ can be expressed as a boundary integral by applying the divergence method, Haug and Choi (1986), but it is well known that this introduces error in the finite element computations as mentioned in Sect. 1. Consequently, we use the "domain" expression (32) in *all* of our computations.[9]

We now discuss the finite element discretization. The Equilibrium Equation (1) becomes

$$\boldsymbol{W}^{T}(\boldsymbol{K}\,\boldsymbol{U} - \boldsymbol{P}) = 0 \tag{33}$$

where $\boldsymbol{U}$ and $\boldsymbol{W}$ are the degrees-of-freedom vectors corresponding to $\boldsymbol{u}$ and $\boldsymbol{w}$, $\boldsymbol{K}$ is the symmetric stiffness matrix defined such that $\boldsymbol{W}^{T}\boldsymbol{K}\,\boldsymbol{U} = a(\boldsymbol{u},\boldsymbol{w}) = \int_{\Omega}\nabla\boldsymbol{w}\cdot\mathbb{C}[\nabla\boldsymbol{u}]\,dv$ and $\boldsymbol{P}$ is the load vector defined such that $\boldsymbol{W}^{T}\boldsymbol{P} = \ell(\boldsymbol{w}) = \int_{\Omega}\boldsymbol{w}\cdot\boldsymbol{b}\,dv + \int_{A^{t}}\boldsymbol{w}\cdot\boldsymbol{t}^{n}\,da + \int_{A^{p}}\boldsymbol{w}\cdot p\,\boldsymbol{n}\,da$. The arbitrariness of $\boldsymbol{W}$ yields the linear equation $\boldsymbol{K}\,\boldsymbol{U} = \boldsymbol{P}$ that we solve for $\boldsymbol{U}$.

After evaluating $\boldsymbol{U}$, we can interpolate $\boldsymbol{u}$ and $\nabla\boldsymbol{u}$ within the elements as $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{N}(\boldsymbol{x})\,\boldsymbol{U}$ and $\nabla\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{x})\,\boldsymbol{U}$ where $\boldsymbol{N}$ contains the element shape functions, $\boldsymbol{B} = \partial\boldsymbol{N}/\partial\boldsymbol{x}$ and the gradient $\nabla\boldsymbol{u}$ is expressed in the usual vector representation formed by stacking its columns. These interpolations allow us to evaluate the cost and constraint functions θ of (3) by summing over the elements Ω_{e} and performing the usual quadratures. The same interpolation is used to express $\boldsymbol{x} = \boldsymbol{N}(\boldsymbol{x})\,\boldsymbol{X}$, $\delta\boldsymbol{x} = \boldsymbol{N}(\boldsymbol{x})\,\delta\boldsymbol{X}$, and $\nabla\delta\boldsymbol{x} = \boldsymbol{B}(\boldsymbol{x})\,\delta\boldsymbol{X}$. In this way, $\delta\theta^{E}(\boldsymbol{x};\delta\boldsymbol{x}) = \delta\boldsymbol{X}^{T}\,\partial\Theta/\partial\boldsymbol{X}$ where

$$\begin{aligned}\frac{\partial\Theta}{\partial\boldsymbol{X}} &= \mathbb{A}_{e=1}^{n_{\text{elem}}}[\\ &\int_{\Omega_{e}}\left(\boldsymbol{B}^{T}\left(-\nabla^{T}\boldsymbol{u}\,\frac{\partial\Lambda}{\partial\nabla\boldsymbol{u}} + \Lambda\boldsymbol{I}\right)^{T} + \boldsymbol{N}^{T}\frac{\partial\Lambda}{\partial\boldsymbol{x}}\right)dv\\ &+ \int_{A_{e}^{t}}\left(\boldsymbol{N}^{T}\,\frac{\partial\tau}{\partial\boldsymbol{x}} + \boldsymbol{B}^{T}\,\tau\,(\boldsymbol{I} - \boldsymbol{n}\otimes\boldsymbol{n})\right)da\\ &+ \int_{A_{e}^{p}}\left(\boldsymbol{N}^{T}\left(\frac{\partial\boldsymbol{\tau}}{\partial\boldsymbol{x}}\right)^{T}\boldsymbol{N}\right.\\ &\left.+\boldsymbol{B}^{T}((\boldsymbol{\tau}\cdot\boldsymbol{n})\,\boldsymbol{I} - (\boldsymbol{n}\otimes\boldsymbol{\tau}))\right)da]\end{aligned} \tag{34}$$

in which $\mathbb{A}$ is the usual finite element assembly operator and all tensors, e.g., $-\nabla^{T}\boldsymbol{u}\,\frac{\partial\Lambda}{\partial\nabla\boldsymbol{u}} + \Lambda\boldsymbol{I}$ are expressed in their vector representations. We similarly have $\delta\theta^{I}(\boldsymbol{x};\overset{\star}{\boldsymbol{u}}) = \overset{\star}{\boldsymbol{U}}{}^{T}\,\partial\Theta/\partial\boldsymbol{U}$ where

$$\begin{aligned}\frac{\partial\Theta}{\partial\boldsymbol{U}} &= \mathbb{A}_{e=1}^{n_{\text{elem}}}\left[\int_{\Omega_{e}}\left(\boldsymbol{B}^{T}\,\frac{\partial\Lambda}{\partial\nabla\boldsymbol{u}} + \boldsymbol{N}^{T}\,\frac{\partial\Lambda}{\partial\boldsymbol{u}}\right)dv\right.\\ &\qquad\left.+ \int_{A_{e}^{t}}\boldsymbol{N}^{T}\,\frac{\partial\tau}{\partial\boldsymbol{u}}\,da + \int_{A_{e}^{p}}\boldsymbol{N}^{T}\left(\frac{\partial\boldsymbol{\tau}}{\partial\boldsymbol{u}}\right)^{T}\boldsymbol{N}\,da\right]\end{aligned} \tag{35}$$

Further, $\delta r^{E}(\boldsymbol{u},\boldsymbol{w},\boldsymbol{x};\delta\boldsymbol{x}) = \delta\boldsymbol{X}^{T}\,\partial R/\partial\boldsymbol{X}$ where

$$\begin{aligned}\frac{\partial R}{\partial\boldsymbol{X}} &= \mathbb{A}_{e=1}^{n_{\text{elem}}}\left[\int_{\Omega_{e}}\left(\boldsymbol{B}^{T}\left(-\nabla^{T}\boldsymbol{w}\,\mathbb{C}[\nabla\boldsymbol{u}]\right.\right.\right.\\ &\left.-\nabla^{T}\boldsymbol{u}\,\mathbb{C}[\nabla\boldsymbol{w}] + (\nabla\boldsymbol{w}\cdot\mathbb{C}[\nabla\boldsymbol{u}] - \boldsymbol{w}\cdot\boldsymbol{b})\,\boldsymbol{I}\right)\\ &+\ \boldsymbol{N}^{T}\left(\left(\frac{\partial\mathbb{C}[\nabla\boldsymbol{u}]}{\partial\boldsymbol{x}}\right)^{T}[\nabla\boldsymbol{w}] - \left(\frac{\partial\boldsymbol{b}}{\partial\boldsymbol{x}}\right)^{T}\boldsymbol{w}\right)\Bigg)\,dv\\ &- \int_{A_{e}^{t}}\left(\boldsymbol{N}^{T}\left(\frac{\partial\boldsymbol{t}^{p}}{\partial\boldsymbol{x}}\right)^{T}\boldsymbol{w}\right.\\ &\left.+\boldsymbol{B}^{T}\,(\boldsymbol{w}\cdot\boldsymbol{t}^{p})\,(\boldsymbol{I} - \boldsymbol{n}\otimes\boldsymbol{n})\right)da\\ &- \int_{A_{e}^{p}}\left(\boldsymbol{N}^{T}\left(\frac{\partial p}{\partial\boldsymbol{x}}\otimes\boldsymbol{w}\right)\boldsymbol{n}\right.\\ &\left.\left.+\boldsymbol{B}^{T}\,p\,((\boldsymbol{w}\cdot\boldsymbol{n})\boldsymbol{I} - (\boldsymbol{n}\otimes\boldsymbol{w}))\right)da\right]\end{aligned} \tag{36}$$

in which $\boldsymbol{w}$ and $\nabla\boldsymbol{w}$ are interpolated similarly to $\boldsymbol{u}$ and $\nabla\boldsymbol{u}$. Finally

[9] The gradient smoothing and fictitious energy algorithms cited in Sect. 1 use the less accurate boundary expression. We implement those algorithms here using the more accurate domain expression.

$$\delta r^I(\boldsymbol{u},\boldsymbol{w},\boldsymbol{x};\overset{\star}{\boldsymbol{u}}) = a(\boldsymbol{w},\overset{\star}{\boldsymbol{u}}) = \overset{\star}{\boldsymbol{U}}{}^T \boldsymbol{K}\,\boldsymbol{W} \tag{37}$$

so that

$$\begin{aligned}\delta\theta^I(\boldsymbol{x},\overset{\star}{\boldsymbol{u}}) - \delta r^I(\boldsymbol{u},\boldsymbol{w},\boldsymbol{x};\overset{\star}{\boldsymbol{u}}) \\ = \overset{\star}{\boldsymbol{U}}{}^T\big((\partial\Theta/\partial\boldsymbol{U})^T - \boldsymbol{K}\,\boldsymbol{W}\big).\end{aligned} \tag{38}$$

To annihilate this implicit term, we equate $\overset{\star}{\boldsymbol{U}}{}^T\big((\partial\Theta/\partial\boldsymbol{U})^T - \boldsymbol{K}\,\boldsymbol{W}\big) = 0$ and use the arbitrariness of $\overset{\star}{\boldsymbol{U}}$ to define the adjoint problem $\boldsymbol{K}\,\boldsymbol{W} = (\partial\Theta/\partial\boldsymbol{U})^T$ that we solve for $\boldsymbol{W}$. Finally, we evaluate the sensitivity $D\Theta/DX = \partial\Theta/\partial X - \partial R/\partial X$.

Appendix 2: Gradient smoothing derivations

The finite element implementation of (10) gives the sensitivity filter equation

$$\boldsymbol{W}^T\left((\widetilde{\boldsymbol{K}}+\widetilde{\boldsymbol{M}})\,\widetilde{\text{Grad}\Theta} - \left(\frac{D\Theta}{DX}\right)^T\right) = 0 \tag{39}$$

where we define $\widetilde{\text{Grad}\Theta}$ as the degrees-of-freedom vector corresponding to $\widetilde{\text{grad}\theta}$, $\widetilde{\boldsymbol{K}}$ as the symmetric smoothing stiffness matrix that satisfies $\boldsymbol{W}^T\widetilde{\boldsymbol{K}}\,\boldsymbol{U} = \int_\Omega \nabla\boldsymbol{w}\cdot\gamma\,\nabla\boldsymbol{u}\,dv$ and $\widetilde{\boldsymbol{M}}$ as the symmetric smoothing mass matrix that satisfies $\boldsymbol{W}^T\widetilde{\boldsymbol{M}}\,\boldsymbol{U} = \int_\Omega \boldsymbol{w}\cdot\boldsymbol{u}\,dv$. The arbitrariness of $\boldsymbol{W}$ gives the linear equation $(\widetilde{\boldsymbol{K}}+\widetilde{\boldsymbol{M}})\,\widetilde{\text{Grad}\Theta} = (D\theta/DX)^T$ that we solve for $\widetilde{\text{Grad}\Theta}$.

Appendix 3: Fictitious energy derivations

Fictitious energy function

We desire a fictitious energy density functional Ψ_f that is unaffected by rigid motions and dilations/contractions as we do not want to penalize an element that is subjected to such a shape preserving transformation. To these ends, we use pose the fictitious energy function

$$\Psi_f(\boldsymbol{F}_d) = \frac{\text{tr}\boldsymbol{C}_d}{(\det\boldsymbol{C}_d)^{\frac{1}{n}}} - n \tag{40}$$

where $\boldsymbol{C}_d = \boldsymbol{F}_d^T\boldsymbol{F}_d$ is the fictitious symmetric right-Cauchy-Green strain and $n = 2(3)$ for two(three)-dimensional domains. It is readily verified that for $\boldsymbol{F}_d = \alpha\,\boldsymbol{R}_d$, i.e., a combined rotation $\boldsymbol{R}_d$ and dilation/contraction $\alpha\,\boldsymbol{I}$ that $\Psi_f(\boldsymbol{F}_d) = 0$.

To obtain the Eq. (40) energy density functional Ψ_f, we note that

$$\boldsymbol{F}_d = \boldsymbol{R}_d\,\boldsymbol{U}_d = \boldsymbol{R}_d\,(\det\boldsymbol{F}_d)^{1/n}\,\overline{\boldsymbol{U}}_d \tag{41}$$

where $\boldsymbol{R}_d$ is a rotation and $\boldsymbol{U}_d$ is symmetric positive definite. The first equality is a result of the polar decomposition theorem and the second uses the definition $\overline{\boldsymbol{U}}_d = (\det\boldsymbol{F}_d)^{-1/n}\,\boldsymbol{U}_d$ so that $\det\overline{\boldsymbol{U}}_d = 1$. As seen above, the fictitious deformation gradient consists of an isochoric distortion/contraction $\overline{\boldsymbol{U}}_d$, followed by a dilation $(\det\boldsymbol{F}_d)^{1/n}\,\boldsymbol{I}$ followed by a rotation $\boldsymbol{R}_d$. Based on this decomposition, we pose the Neo-Hookean like fictitious energy function that does not depend on the dilation/contraction $\det\boldsymbol{F}_d\,\boldsymbol{I}$ nor the rotation $\boldsymbol{R}_d$

$$\Psi_f(\boldsymbol{F}_d) = \text{tr}\overline{\boldsymbol{U}}_d^2 - n \tag{42}$$

which is equivalent to (40). Using the spectral decomposition theorem, we can express the above for two dimensions as

$$\Psi_f(\boldsymbol{F}_d) = \lambda^2 + \frac{1}{\lambda^2} - 2 \tag{43}$$

where λ and $1/\lambda$ are the positive eigenvalues of $\overline{\boldsymbol{U}}_d$ and for three dimensions as

$$\Psi_f(\boldsymbol{F}_d) = \lambda_1^2 + \lambda_2^2 + \frac{1}{\lambda_1^2\,\lambda_2^2} - 3 \tag{44}$$

where λ_1, λ_2 and $1/(\lambda_1\,\lambda_2)$ are the positive eigenvalues of $\overline{\boldsymbol{U}}_d$. A simple derivation subsequently indicates that the minimum energy is obtained for $\lambda = 1$ in two dimensions and for $\lambda_1 = \lambda_2 = 1$ in three dimensions which implies $\overline{\boldsymbol{U}}_d = \boldsymbol{I}$, i.e., there is no distortion. This further confirms that $\boldsymbol{F}_d = \alpha\,\boldsymbol{R}_d$ is an energy minimizer for which $\Psi_f(\alpha\,\boldsymbol{R}_d) = 0$.

Newton's method

To solve the residual Eq. (15), we use Newton's method and iterate by finding the $\Delta\boldsymbol{d} \in \mathcal{H}_f(0)$ such that

$$\begin{aligned}\delta_{\boldsymbol{d}^0} r_f(\boldsymbol{d}^0,\boldsymbol{w},\boldsymbol{d}^p;\Delta\boldsymbol{d}^0) &= \int_{\Omega_0} \nabla\boldsymbol{w}\cdot\frac{\partial^2\Psi_f}{\partial\boldsymbol{F}_d^2}[\nabla\Delta\boldsymbol{d}^0]\,dv \\ &= -r_f(\boldsymbol{d},\boldsymbol{w},\boldsymbol{d}^p)\end{aligned} \tag{45}$$

for all $\boldsymbol{w} \in \mathcal{H}_f(0)$. Upon evaluating $\Delta\boldsymbol{d}^0$ we update $\boldsymbol{d}^0 \leftarrow \boldsymbol{d}^0 + \Delta\boldsymbol{d}^0$ and continue iterating until $r(\boldsymbol{d},\boldsymbol{w},\boldsymbol{d}^p) \le \epsilon$ for some suitable convergence tolerance ϵ. Note that the fictitious energy residual Eq. (14) for $\boldsymbol{d}^0$ is solved over the initial configuration Ω_0 whereas the H^1 smoothing Eq. (10) for $\widetilde{\text{grad}\theta}$ is solved over the current design configuration Ω.

To evaluate the residual and tangent, we need

$$
\begin{aligned}
\frac{\partial \Psi_f}{\partial \boldsymbol{F}_d} &= 2\,\boldsymbol{F}_d\,\frac{\partial \Psi_f}{\partial \boldsymbol{C}_d}\\
\frac{\partial^2 \Psi_f}{\partial \boldsymbol{F}_d^2} &= 2\,\boldsymbol{I} \boxtimes \frac{\partial \Psi_f}{\partial \boldsymbol{C}_d}\\
&\quad + 4(\boldsymbol{F} \boxtimes \boldsymbol{I})\,\frac{\partial^2 \Psi_f}{\partial \boldsymbol{C}_d^2}\,(\boldsymbol{F} \boxtimes \boldsymbol{I})^T
\end{aligned}
\tag{46}
$$

where, for any 2-tensors $\boldsymbol{A}$ and $\boldsymbol{B}$, the conjugate product $\boldsymbol{A} \boxtimes \boldsymbol{B}$ is the 4-tensor that satisfies $(\boldsymbol{A} \boxtimes \boldsymbol{B})[\boldsymbol{C}] = \boldsymbol{A}\,\boldsymbol{C}\,\boldsymbol{B}^T$ for any 2-tensor $\boldsymbol{C}$. For the energy function of (40), we have

$$
\begin{aligned}
\frac{\partial \Psi_f}{\partial \boldsymbol{C}_d} &= (\det \boldsymbol{C}_d)^{-\frac{1}{n}}\left(\boldsymbol{I} - \frac{1}{n}\,\mathrm{tr}\boldsymbol{C}_d\,\boldsymbol{C}_d^{-1}\right)\\
\frac{\partial^2 \Psi_f}{\partial \boldsymbol{C}_d^2} &= \frac{1}{n}(\det \boldsymbol{C}_d)^{-\frac{1}{n}}\left(-\boldsymbol{I} \otimes \boldsymbol{C}_d^{-1} - \boldsymbol{C}_d^{-1} \otimes \boldsymbol{I}\right.\\
&\quad \left. + \frac{1}{n}\mathrm{tr}\boldsymbol{C}_d\,\boldsymbol{C}_d^{-1} \otimes \boldsymbol{C}_d^{-1} + \mathrm{tr}\boldsymbol{C}_d\,\boldsymbol{C}_d^{-1} \boxtimes \boldsymbol{C}_d^{-1}\right)
\end{aligned}
\tag{47}
$$

where the dyadic product $\boldsymbol{A} \otimes \boldsymbol{B}$ is the 4-tensor that satisfies $(\boldsymbol{A} \otimes \boldsymbol{B})[\boldsymbol{C}] = (\boldsymbol{B} \cdot \boldsymbol{C})\,\boldsymbol{A}$ for any 2-tensor $\boldsymbol{C}$.

A linearization about $\boldsymbol{F}_d = \boldsymbol{I}$, which is accurate when $\nabla \boldsymbol{d}$ is "small," provides

$$
\begin{aligned}
\Psi_f &= \frac{1}{2}\,\nabla \boldsymbol{d} \cdot \mathbb{A}_f[\nabla \boldsymbol{d}]\\
\frac{\partial \Psi_f}{\partial \boldsymbol{F}_d} &= \mathbb{A}_f[\nabla \boldsymbol{d}]\\
\frac{\partial^2 \Psi_f}{\partial \boldsymbol{F}_d^2} &= \mathbb{A}_f
\end{aligned}
\tag{48}
$$

where

$$
\mathbb{A}_f = \left.\frac{\partial^2 \Psi_f}{\partial \boldsymbol{F}_d^2}\right|_{\boldsymbol{F}_d = \boldsymbol{I}} = 4\left(\mathbb{I}_{\mathrm{sym}} - \frac{1}{n}\boldsymbol{I} \otimes \boldsymbol{I}\right)
\tag{49}
$$

wherein the symmetrizer $\mathbb{I}_{\mathrm{sym}}$ is the 4-tensor that satisfies $\mathbb{I}_{\mathrm{sym}}[\boldsymbol{C}] = \frac{1}{2}(\boldsymbol{C} + \boldsymbol{C}^T)$ for any 2-tensor $\boldsymbol{C}$. Note that $\mathbb{A}_f[\boldsymbol{C}] = 4\,(\mathbb{I}_{\mathrm{sym}}[\boldsymbol{C}] - \mathrm{tr}\boldsymbol{C}\,\boldsymbol{I})$ is deviatoric and symmetric and hence as expected $\mathbb{A}_f[\nabla \boldsymbol{d}] = \boldsymbol{0}$ for $\nabla \boldsymbol{d} = \alpha\,\boldsymbol{I}$ an infinitesimal dilation/contraction, for $\nabla \boldsymbol{d} = \boldsymbol{W}$ where $\boldsymbol{W}$ is skew, i.e., an infinitesimal rotation and for their combination $\nabla \boldsymbol{d} = \alpha\,\boldsymbol{W}$. The stiffness tensor $\mathbb{C}$ for a linear elastic isotropic material can be expressed as $\mathbb{C} = 2\,\mu\,\mathbb{I}_{\text{sym-dev}} + 3\,\kappa\,\mathbb{I}_{\mathrm{sph}}$ where $\mathbb{I}_{\text{sym-dev}} = \mathbb{I}_{\mathrm{sym}} - \frac{1}{n}\boldsymbol{I} \otimes \boldsymbol{I}$ and $\mathbb{I}_{\mathrm{sph}} = \frac{1}{n}\boldsymbol{I} \otimes \boldsymbol{I}$ are the projections onto the spaces of symmetric deviatoric tensors and spherical tensors, respectively. And hence, the "fictitious" material can be thought of as an elastic material with shear stiffness $\mu = 1/2$ and bulk modulus $\kappa = 0$ which also implies it has first Lamé parameter $\lambda = -1/3$, Poisson ratio $\nu = -1$, and Young's modulus $E = 0$.[10][11]

Sensitivity analysis

Noting that $\delta \boldsymbol{x} = \delta \boldsymbol{d}$ allows us to write

$$
\begin{aligned}
\delta\theta(\boldsymbol{d};\delta\boldsymbol{d}) &= \delta\theta^E(\boldsymbol{x};\delta\boldsymbol{x})\\
&\quad - \delta r^E(\boldsymbol{u},\boldsymbol{w},\boldsymbol{x};\delta\boldsymbol{x})
\end{aligned}
\tag{50}
$$

from (32). The problem arises because $\boldsymbol{d}$ is an implicit function of d^p as seen through (15), i.e., we have $\boldsymbol{d} = \boldsymbol{d}^0(d^p) + \boldsymbol{d}^p(d^p)$ and hence, with a slight abuse of notation, we have

$$
\theta = \theta(d^p) = \theta(\boldsymbol{d}(d^p)) = \theta(\boldsymbol{d}^0(d^p), \boldsymbol{d}^p(d^p)).
\tag{51}
$$

We annihilate the implicit variation $\delta\boldsymbol{d}^0(d^p;\delta d^p)$ in (50) by again applying the adjoint method. To do this, the total variation of (15) with respect to d^p in the direction δd^p, which equals zero, is subtracted from (50) to give

$$
\begin{aligned}
\delta\theta(d^p;\delta d^p) &= \delta_{d^0}\theta(\boldsymbol{d}^0(d^p), \boldsymbol{d}^p(d^p);\delta\boldsymbol{d}^0(d^p;\delta d^p))\\
&+ \delta_{d^p}\theta(\boldsymbol{d}^0(d^p), \boldsymbol{d}^p(d^p);\delta\boldsymbol{d}^p(d^p;\delta d^p)\\
&- \left\{\delta_{d^0} r_f(\boldsymbol{d}^0(d^p), \boldsymbol{w}, \boldsymbol{d}^p(d^p);\delta\boldsymbol{d}^0(d^p;\delta d^p))\right.\\
&\left. + \delta_{d^p} r_f(\boldsymbol{d}^0(d^p), \boldsymbol{w}, \boldsymbol{d}^p(d^p);\delta\boldsymbol{d}^p(d^p;\delta d^p))\right\}
\end{aligned}
\tag{52}
$$

where $\delta_{d^0} r_f$ is the tangent operator of (45) and the term in braces equals zero. Upon requiring the heretofore arbitrary $\boldsymbol{w} \in \mathcal{H}_f(0)$ to satisfy

$$
\begin{aligned}
&\delta_{d^0} r_f(\boldsymbol{d}^0(d^p), \boldsymbol{w},\boldsymbol{d}^p(d^p);\delta\boldsymbol{d}^0(d^p;\delta d^p))\\
&\qquad = \delta\theta(\boldsymbol{d}(d^p);\delta\boldsymbol{d}^0(d^p;\delta d^p))
\end{aligned}
\tag{53}
$$

for all $\delta\boldsymbol{d}^0 \in \mathcal{H}_f(0)$, (62) reduces to

$$
\begin{aligned}
\delta\theta(d^p;\delta d^p) &= \delta_{d^p}\theta(\boldsymbol{d}^0(d^p), \boldsymbol{d}^p(d^p);\delta\boldsymbol{d}^p(d^p;\delta d^p))\\
&\quad - \delta_{d^p} r_f(\boldsymbol{d}^0(d^p), \boldsymbol{w}, \boldsymbol{d}^p(d^p);\delta\boldsymbol{d}^p(d^p;\delta d^p))
\end{aligned}
\tag{54}
$$

which is the desired explicit sensitivity expression. The fact that r_f and θ are defined over different domains Ω_0 and Ω is not a concern as the change-of-variable theorem can always be used to express θ over Ω_0. Indeed, the change-of-variable theorem is often used to obtain the shape sensitivity expressions $\delta\theta_i^E$ and δr^E of (26) and (28).

A simplification arises for the sensitivity analysis of the Equation (16) constraint on the total fictitious energy. Indeed, using the adjoint sensitivity analysis, one can show that

[10] For the $n = 2$ case, $\mathbb{C}$ is obtained by using the plane stress assumption.

[11] A constraint on the linearized energy, i.e., mesh quality metric, $\nabla \boldsymbol{d} \cdot \mathbb{I}_{\text{sym-dev}}[\nabla \boldsymbol{d}]$ is used by Matsumori et al. (2017) to limit mesh distortion.

$$\begin{aligned}\delta\theta_f(d^p;\delta d^p) = &\int_{\Omega_0} \frac{\partial\Psi_f(\boldsymbol{F}_d)}{\partial \boldsymbol{F}_d}\cdot\nabla\delta\boldsymbol{d}_p(d^p;\delta d^p)\,dv\\ &- \delta_{d^p} r_f(\boldsymbol{d}^0(d^p),\boldsymbol{w},\boldsymbol{d}^p(d^p);\delta\boldsymbol{d}^p(d^p;\delta d^p))\\ = &\, r_f(\boldsymbol{d}^0(d^p),\delta\boldsymbol{d}^p(d^p;\delta d^p),\boldsymbol{d}^p(d^p))\end{aligned} \tag{55}$$

where we use (15) with $\boldsymbol{w} = \delta\boldsymbol{d}^p(d^p;\delta d^p)$[12] and the fact that the adjoint response $\boldsymbol{w} = \boldsymbol{0}$. The latter claim is true since, for all $\delta\boldsymbol{d}^0 \in \mathcal{H}_f(0)$, $\boldsymbol{w} \in \mathcal{H}_f(0)$ must satisfy

$$\begin{aligned}&\delta_{\boldsymbol{d}^0} r_f(\boldsymbol{d}^0(d^p),\boldsymbol{w},\boldsymbol{d}^p(d^p);\delta\boldsymbol{d}^0(d^p;\delta d^p))\\ &= \int_{\Omega_0} \frac{\partial\Psi_f(\boldsymbol{F}_d)}{\partial \boldsymbol{F}_d}\cdot\nabla\delta\boldsymbol{d}^0(d^p;\delta d^p)\,dv\\ &= r_f(\boldsymbol{d}^0(d^p),\delta\boldsymbol{d}^0(d^p;\delta d^p),\boldsymbol{d}^p(d^p))\end{aligned} \tag{56}$$

where we used (15) with $\boldsymbol{w} = \delta\boldsymbol{d}^0(d^p;\delta d^p) \in \mathcal{H}_f(0)$ to obtain the last equality. We trivially obtain $\boldsymbol{w} = \boldsymbol{0}$ since the load linear term on right-hand side equals zero.

Discretization

In the finite element discretization, we partition $\boldsymbol{D}$ the degrees-of-freedom vector corresponding to $\boldsymbol{d} = \boldsymbol{d}^0 + \boldsymbol{d}^p$, into the degrees-of-freedom vectors $\boldsymbol{D}^0$ and $\boldsymbol{D}^p$ that correspond to $\boldsymbol{d}^0$ and $\boldsymbol{d}^p$; the discretization and partition of $\boldsymbol{w}$ are similarly performed. The residual Equation (15) is then expressed as

$$\boldsymbol{W}^{0^T}\boldsymbol{R}_f^0(\boldsymbol{D}^0,\boldsymbol{D}^p) = 0 \tag{57}$$

where $\boldsymbol{R}_f$ is the residual vector defined such that $\boldsymbol{W}^T\boldsymbol{R}_f = \int_{\Omega_0}\nabla\boldsymbol{w}\cdot\frac{\partial\Psi_f}{\partial\boldsymbol{F}_d}\,dv$ and $\boldsymbol{R}_f^0$ is the partition of $\boldsymbol{R}_f$ that corresponds to $\boldsymbol{W}^0$. Notably, we only solve the residual equation $r_f = 0$ for the degrees-of-freedom that corresponds to the unknown $\boldsymbol{D}^0$. The discretization of the Newton update equation (45) yields

$$\boldsymbol{W}^{0^T}\left(\boldsymbol{K}_f^{00}\,\Delta\boldsymbol{D}^0 + \boldsymbol{R}_f^0\right) = 0 \tag{58}$$

where $\boldsymbol{K}_f$ is the symmetric tangent stiffness matrix defined such that $\boldsymbol{W}^T\boldsymbol{K}_f\,\Delta\boldsymbol{D} = \int_{\Omega_0}\nabla\boldsymbol{w}\cdot\partial^2\Psi_f/\partial\boldsymbol{F}_d^2[\nabla\Delta\boldsymbol{d}]\,dv$ and $\boldsymbol{K}_f^{00}$ is the partition of $\boldsymbol{K}_f$ that corresponds to $\boldsymbol{W}^0$ and $\Delta\boldsymbol{D}^0$. The arbitrariness of $\boldsymbol{W}^0$ renders the linear update equation $\boldsymbol{K}_f^{00}\,\Delta\boldsymbol{D}^0 = -\boldsymbol{R}_f^0$ that we solve for $\Delta\boldsymbol{D}^0$. Updates $\boldsymbol{D}^0 \leftarrow \boldsymbol{D}^0 + \Delta\boldsymbol{D}^0$ continue until $|\boldsymbol{R}_f^0| \le \epsilon$ for suitably small tolerance ϵ.

The adjoint Eq. (53) is discretized as

$$\delta\boldsymbol{D}^{0^T}\left(\boldsymbol{K}_f^{00}\,\boldsymbol{W}^0 - \left(\frac{D\theta}{D\boldsymbol{X}^0}\right)^T\right) = 0 \tag{59}$$

where $D\theta/D\boldsymbol{X}^0$ is the partition of $D\theta/D\boldsymbol{X}$ that is associated with the $\boldsymbol{D}^0$ degrees-of-freedom. The arbitrariness of $\delta\boldsymbol{D}^0$ renders the linear adjoint equation $\boldsymbol{K}_f^{00}\,\boldsymbol{W}^0 = \left(D\theta/D\boldsymbol{X}^0\right)^T$ that we solve for $\boldsymbol{W}^0$. Finally, the sensitivity Eq. (54) is discretized such that

$$\frac{D\theta}{D\boldsymbol{D}^p} = \frac{D\theta}{D\boldsymbol{X}^p} - \boldsymbol{W}^{0^T}\boldsymbol{K}_f^{0p} \tag{60}$$

where $D\theta/D\boldsymbol{X}^p$ is the partition of $D\theta/D\boldsymbol{X}$ that is associated with $\boldsymbol{D}^p$ and $\boldsymbol{K}_f^{0p}$ is the partition of $\boldsymbol{K}_f$ that is associated with $\boldsymbol{W}^0$ and $\Delta\boldsymbol{D}^p$. The sensitivity of the energy constraint (16) is obtained directly by evaluating $D\Theta_f/D\boldsymbol{D}^p = \partial\Theta_f/\partial\boldsymbol{D}^p$. The computation of $\partial\Theta_f/\partial\boldsymbol{D}^p = \boldsymbol{R}_f^p$ follows from (15) and (55) where $\boldsymbol{R}_f^p$ is the partition of $\boldsymbol{R}_f$ that corresponds to the $\boldsymbol{D}^p$ degrees-of-freedom.

Appendix 4: Consistent filtering derivations

Noting that $\delta\boldsymbol{x} = \delta\tilde{\boldsymbol{d}}$ allows us to trivially obtain

$$\begin{aligned}\delta\theta(\tilde{\boldsymbol{d}};\delta\tilde{\boldsymbol{d}}) = &\,\delta\theta^E(\boldsymbol{x};\delta\boldsymbol{x})\\ &- \delta r^E(\boldsymbol{u},\boldsymbol{w},\boldsymbol{x};\delta\boldsymbol{x})\end{aligned} \tag{61}$$

from (32). The problem arises because $\tilde{\boldsymbol{d}}$ is an implicit function of $\boldsymbol{d}$ as seen through (17), i.e., we have $\tilde{\boldsymbol{d}} = \tilde{\boldsymbol{d}}(\boldsymbol{d})$ and hence, using our abusive notation, we have $\theta = \theta(\boldsymbol{d}) = \theta(\tilde{\boldsymbol{d}}(\boldsymbol{d}))$. We annihilate the implicit variation $\delta\tilde{\boldsymbol{d}}(\boldsymbol{d};\delta\boldsymbol{d})$ in (61) by again applying the adjoint method. The variation of (17) with respect to $\boldsymbol{d}$ in the direction $\delta\boldsymbol{d}$, which equals zero, is subtracted from (61) to give[13]

$$\begin{aligned}\delta\theta(\boldsymbol{d};\delta\boldsymbol{d}) = &\,\delta\theta(\tilde{\boldsymbol{d}};\delta\tilde{\boldsymbol{d}})\\ &\overbrace{-a_c(\delta\tilde{\boldsymbol{d}},\boldsymbol{w}) + \ell_c(\boldsymbol{w},\delta\boldsymbol{d})}^{=0}\end{aligned} \tag{62}$$

Upon requiring the heretofore arbitrary $\boldsymbol{w} \in \mathcal{H}_c$ to satisfy

$$a_c(\delta\tilde{\boldsymbol{d}},\boldsymbol{w}) = \delta\theta(\tilde{\boldsymbol{d}};\delta\tilde{\boldsymbol{d}}) \tag{63}$$

[12] The fact that $\delta\boldsymbol{d}^p(d^p;\delta d^p) \notin \mathcal{H}_f(0)$ is not an issue here.

[13] The bilinear form a_c in (17) uses $\int_{\Omega_0}\nabla\boldsymbol{w}\cdot\gamma\,\nabla\delta\tilde{\boldsymbol{d}}\,dv$ for the PDE filter. For the nonlinear energy filter replace this integral with $\int_{\Omega_0}\nabla\boldsymbol{w}\cdot\partial\Psi_f/\partial\tilde{\boldsymbol{F}}_d[\nabla\delta\tilde{\boldsymbol{d}}]\,dv$ and for the linearized energy filter replace it with $\int_{\Omega_0}\nabla\boldsymbol{w}\cdot\mathbb{A}_f[\nabla\delta\tilde{\boldsymbol{d}}]\,dv$.

for all $\delta\widetilde{\boldsymbol{d}} \in \mathcal{H}_c$, (62) reduces to the desired explicit sensitivity expression

$$\delta\theta(\boldsymbol{d};\delta\boldsymbol{d}) = \ell_c(\boldsymbol{w}, \delta\boldsymbol{d}) \tag{64}$$

As discussed above, the fact that a_c, ℓ_c and $\delta\theta$ are defined over different domains in (62) - (64) is not a concern.

The finite element discretization of (17) yields the equation

$$\boldsymbol{W}^T((\boldsymbol{K}_c + \boldsymbol{M}_c)\widetilde{\boldsymbol{D}} - \boldsymbol{M}_c\boldsymbol{D}) = 0 \tag{65}$$

where $\boldsymbol{D}$, $\widetilde{\boldsymbol{D}}$, and $\boldsymbol{W}$ are the degrees-of-freedom vectors corresponding to $\boldsymbol{d}$, $\widetilde{\boldsymbol{d}}$, and $\boldsymbol{w}$, $\boldsymbol{K}_c$ is the symmetric filter stiffness matrix defined such that[14] $\boldsymbol{W}^T\boldsymbol{K}_c\widetilde{\boldsymbol{D}} = \int_{\Omega_0} \nabla\boldsymbol{w} \cdot \gamma\, \nabla\widetilde{\boldsymbol{d}}\, dv$ and $\boldsymbol{M}_c$ is the symmetric filter mass matrix defined such that $\boldsymbol{W}^T\boldsymbol{M}_c\boldsymbol{D} = \int_{\Omega_0} \boldsymbol{w} \cdot \boldsymbol{d}\, dv$. Using the symmetries of $\boldsymbol{K}_c$ and $\boldsymbol{M}_c$, the adjoint problem (63) is discretized as

$$\delta\widetilde{\boldsymbol{D}}^T(\boldsymbol{K}_c + \boldsymbol{M}_c)\,\boldsymbol{W} = \delta\widetilde{\boldsymbol{D}}^T\left(\frac{D\theta}{D\boldsymbol{X}}\right)^T \tag{66}$$

. The arbitrariness of $\delta\widetilde{\boldsymbol{D}}$ renders the linear adjoint equation $(\boldsymbol{K}_c + \boldsymbol{M}_c)\,\boldsymbol{W} = (D\Theta/D\boldsymbol{X})^T$ that we solve for $\boldsymbol{W}$. Finally, the derivative of (64) is computed from $D\Theta/D\boldsymbol{D} = \boldsymbol{M}_c\,\boldsymbol{W}$.

Acknowledgements This work was performed under the auspices of the US Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and was supported by the LLNL-LDRD Program under Project No. 22-ERD-023. LLNL-JRNL-841196. The authors would like to thank Professor Mathias Wallin and Dr. Anna Dalklint of Lund University and Dr. Miguel Salazar de Troya of Corintis Inc. for invaluable suggestions throughout this work.

Declarations

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

Replication of results The source code used to generate the presented results cannot at this time be released as it is proprietary to Lawrence Livermore National Laboratory. However, the authors believe that sufficient information has been provided to allow readers to replicate the presented results.

References

Allaire G, Dapogny C, Jouve F (2021) In: Bonito, A., Nochetto, R.H. (eds) Geometric partial differential equations-Part II. Elsevier, Amsterdam, 22:1–132. https://www.sciencedirect.com/handbook/handbook-of-numerical-analysis/vol/22/suppl/C

Anderson R, Andrej J, Barker A, Bramwell J, Camier J-S, Cerveny J, Dobrev V, Dudouit Y, Fisher A, Kolev T, Pazner W, Stowell M, Tomov V, Akkerman I, Dahm J, Medina D, Zampini S (2021) Mfem: a modular finite element methods library. Comput Math Appl 81:42–74. https://doi.org/10.1016/j.camwa.2020.06.009

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. https://doi.org/10.1007/s00158-022-03279-w

Azegami H (2021). Pardalos P, Thai M (eds) Shape Optimization Problems. Singapore, Springer Optimization and Its Applications. Springer, Singaporehttps://doi.org/10.1007/978-981-15-7618-8

Azegami H, Takeuchi K (2006) A smoothing method for shape optimization: traction method using the robin condition. Int J Comput Methods 3(1):21–33. https://doi.org/10.1142/S0219876206000709

Bängtsson E, Noreland D, Berggren M (2003) Shape optimization of an acoustic horn. Comput Methods Appl Mech Eng 192(11–12):1533–1571. https://doi.org/10.1016/S0045-7825(02)00656-4

Barrera JL, Kolev T, Mittal K, Tomov V (2022) Implicit high-order meshing using boundary and interface fitting. arXiv preprint arXiv:2208.05062 10.48550/arXiv.2208.05062

Belegundu AD, Rajan SD (1988) A shape optimization approach based on natural design variables and shape functions. Comput Methods Appl Mech Eng 66(1):87–106. https://doi.org/10.1016/0045-7825(88)90061-8

Berggren M (2010) A unified discrete-continuous sensitivity analysis method for shape optimization. Comput Methods Appl Sci 15:25–39. https://doi.org/10.1007/978-90-481-3239-3_4

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. https://doi.org/10.1007/s00158-013-1031-5

Braibant V, Fleury C (1984) Shape optimal design using b-splines. Comput Methods Appl Mech Eng 44(3):247–267. https://doi.org/10.1016/0045-7825(84)90132-4

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. https://doi.org/10.1016/S0045-7825(00)00278-4

Chen S, Tortorelli DA (1997) Three-dimensional shape optimization with variational geometry. Struct Optim 13(2–3):81–94. https://doi.org/10.1007/BF01199226

Chenais D (1975) On the existence of a solution in a domain identification problem. J Math Anal Appl 52(2):189–219. https://doi.org/10.1016/0022-247X(75)90091-8

Choi KK, Seong HG (1986) A domain method for shape design sensitivity analysis of built-up structures. Comput Methods Appl Mech Eng 57(1):1–15. https://doi.org/10.1016/0045-7825(86)90066-6

Dobrev V, Knupp P, Kolev T, Mittal K, Tomov V (2019) The target-matrix optimization paradigm for high-order meshes. SIAM J Sci Comput 41(1):50–68. https://doi.org/10.1137/18M1167206

Drucker DC, Prager W (1952) Soil mechanics and plastic analysis or limit design. Quarterly of Applied Mathematics 10(2):157–165 http://www.jstor.org/stable/43633942

Ertl FJ, Dhondt G, Bletzinger KU (2019) Vertex assigned morphing for parameter free shape optimization of 3-dimensional solid structures. Comput Methods Appl Mech Engi 353:86–106. https://doi.org/10.1016/j.cma.2019.05.004

Gerzen N, Materna D, Barthold F-J (2012) The inner structure of sensitivities in nodal based shape optimisation. Comput Mech 49(3):379–396. https://doi.org/10.1007/s00466-011-0648-8

Gournay F (2006) Velocity extension for the level-set method and multiple eigenvalues in shape optimization. SIAM J Control Optim 45(1):343–367. https://doi.org/10.1137/050624108

Gunzburger MD, Hongchul K, Manservisi S (2000) On a shape control problem for the stationary Navier-stokes equations. Math

[14] Footnote 13 for the necessary modifications to $\boldsymbol{K}_c$ required when using the energy filter and linearized energy filter.

Model Numer Anal 34(6):1233–1258. https://doi.org/10.1051/m2an:2000125

Harbrecht H (2008) Analytical and numerical methods in shape optimization. Math Methods Appl Sci 31(18):2095–2114. https://doi.org/10.1002/mma.1008

Haug EJ, Choi VKKK.: Design sensitivity analysis of structural systems. Mathematics in science and engineering. v. 177, Academic Press, Orlando

Jameson A (2004) Efficient aerodynamic shape optimization 2:813–833. https://doi.org/10.2514/1.J059254

Knupp P (2012) Introducing the target-matrix paradigm for mesh optimization via node-movement. Eng Comput 28(4):419–429. https://doi.org/10.1007/978-3-642-15414-0_5

Lazarov BS, Sigmund O (2011) Filters in topology optimization based on helmholtz-type differential equations. Int J Numer Methods Eng 86(6):765–781. https://doi.org/10.1002/nme.3072

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. https://doi.org/10.1016/j.cma.2010.10.004

Liatsikouras AG, Pierrot G, Megahed M (2022) A coupled cad-free parameterization-morphing method for adjoint-based shape optimization. Int J Numer Methods Fluids 94(11):1745–1763. https://doi.org/10.1002/fld.5123

Martinelli L, Jameson A (2012) Computational aerodynamics: solvers and shape optimization. J Heat Trans 10(1115/1):4007649

Matlab (2017) MATLAB optimization toolbox. The MathWorks, Natick

Matsumori T, Kawamoto A, Kondoh T, Nomura T, Saomoto H (2017) Boundary shape design by using pde filtered design variables. Struct Multidisc Optim 56(3):619–629. https://doi.org/10.1007/s00158-017-1678-4

Mohammadi B, Pironneau O (2004) Shape optimization in fluid mechanics. Ann Rev Fluid Mech 36:255–279. https://doi.org/10.1146/annurev.fluid.36.050802.121926

Mohammadi B, Pironneau O (2008) Theory and practice of optimal shape design. Eur J Comput Mech 17(1–2):13–30. https://doi.org/10.3166/remn.17.13-30

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. https://doi.org/10.1007/s00158-021-03030-x

Najian Asl R, Bletzinger K-U (2023) The implicit bulk-surface filtering method for node-based shape optimization and a comparison of explicit and implicit filtering techniques. Struct Multidisc Optim. https://doi.org/10.1007/s00158-023-03548-2

Nonogawa M, Takeuchi K, Azegami H (2020) Shape optimization of running shoes with desired deformation properties. Struct Multidisc Optim 62:1535–1546. https://doi.org/10.1007/s00158-020-02560-0

Pironneau O (1984) Optimal shape design for elliptic systems. Springer, Heidelberg. https://doi.org/10.1007/978-3-642-87722-3

Riehl S, Steinmann P (2014) An integrated approach to shape optimization and mesh adaptivity based on material residual forces. Comput Methods Appl Mech Eng 278:640–663. https://doi.org/10.1016/j.cma.2014.06.010

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. https://doi.org/10.1016/j.cma.2014.05.009

Samareh JA (2001) Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. AIAA J 39(5):877–884. https://doi.org/10.2514/2.1391

Scherer M, Denzer R, Steinmann P (2010) A fictitious energy approach for shape optimization. Int J Numer Methods Eng 82(3):269–302. https://doi.org/10.1002/nme.2764

Schwedes T, Ham DA, Funke SW, Piggott MD (2017) Mesh dependence in PDE-constrained optimisation: an application in tidal turbine array layouts. Springer, Cham. https://doi.org/10.1007/978-3-319-59483-5

Shimoda M, Azegami H, Sakurai T (1997) Traction method approach to optimal shape design problems. SAE Technical Papers https://www.jstor.org/stable/44720004

Sigmund O (1997) On the design of compliant mechanisms using topology optimization. Mech Struct Mach 25(4):493–524. https://doi.org/10.1080/08905459708945415

Stück A, Rung T (2011) Adjoint rans with filtered shape derivatives for hydrodynamic optimisation. Comput Fluids 47(1):22–32. https://doi.org/10.1016/j.compfluid.2011.01.041

Tortorelli DA (1993) A geometric representation scheme suitable for shape optimization. Mech Struct Mach 21(1):95–121. https://doi.org/10.1080/08905459308905182

Upadhyay BD, Sonigra SS, Daxini SD (2021) Numerical analysis perspective in structural shape optimization: a review post 2000. Adv Eng Softw. https://doi.org/10.1016/j.advengsoft.2021.102992

Wächter A, Biegler LT (2006) On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Mathe Progr 106(1):25–57. https://doi.org/10.1007/s10107-004-0559-y

Weeber K, Ratnajeevan S, Hoole H (1992) A structural mapping technique for geometric parametrization in the optimization of magnetic devices. Int J Numer Methods Eng 33(10):2145–2179. https://doi.org/10.1002/nme.1620331010

Welker K (2021) Suitable spaces for shape optimization. Appl Math Optim 84:869–902. https://doi.org/10.1007/s00245-021-09788-2

Yang RJ, Botkin ME (1987) Accuracy of the domain material derivative approach to shape design sensitivities. AIAA J 25(12):1606–1610. https://doi.org/10.2514/3.9831

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.