Examples and Counterexamples 9 (2026) 100225

Contents lists available at ScienceDirect

Examples and Counterexamples

journal homepage: www.elsevier.com/locate/exco

An adaptive algorithm for shape optimization☆

Oğuz Han Altıntaş [a,b], Hamdullah Yücel [a,*]

[a] Institute of Applied Mathematics, Middle East Technical University, Üniversiteler Mahallesi, Dumlupınar BulvarıNo:1, Ankara, 06800, Turkiye
[b] ROKETSAN, Elmadağ, Ankara, 06780, Turkiye

ARTICLE INFO

Communicated by F. Bertrand

Keywords:
Shape optimization
Adaptivity
Level-set method

ABSTRACT

The numerical investigation of shape optimization problems is both computationally and theoretically more complex than solving direct analysis problems using the finite element method. The limitations are mainly due to the involvement of both structural analysis and optimization processes. To overcome these numerical challenges, we present an adaptive approach to solve the shape optimization problem associated with the compliance objective. This approach takes into account not only the errors due to the discretization of the constraining PDE, in particular the linear elasticity system, but also the errors due to the discretization of the deformation bilinear form that provides a descent direction. Moreover, the step length in gradient-based optimization is adaptively adjusted by estimating the Lipschitz constant. Finally, in numerical simulations, we present the superiority of the adaptive procedure compared to the fixed mesh refinement, in addition to some cases of failure.

1. Introduction

Shape optimization, a class of optimization problems where the objective functional depends on the shape of the domain in which the partial differential equation (PDE) is defined and the solution of the PDE itself, is formulated as follows:

$$\min_{\Omega\in\mathcal{U}^{\text{ad}}\subset\mathbb{R}^2} J(\Omega) := j(\Omega, u(\Omega)) \quad \text{subject to} \quad \mathcal{G}u(\Omega) = 0, \tag{1}$$

with the cost functional J, the partial differential operator $\mathcal{G}$, and the set of admissible domains $\mathcal{U}^{\text{ad}}$. Within this framework, the problem (1) can be viewed as a PDE-constrained optimization problem where the functional depends on the shape, with the domain serving as the optimization variable and the PDE acting as the constraint.

In the realm of shape optimization using finite element techniques, one encounters more complex challenges compared to simply solving direct analysis problems with the finite element method. Shape-optimal structural design involves combining structural analysis with optimization processes. Errors in this framework stem from approximating the domain and the solution of the PDE constraint, along with their derivatives within the finite elements, as well as approximating design variables and optimality conditions. To reduce the computational complexity while maintaining high accuracy in the geometric description of the optimal design and the approximation of the objective functional, it is crucial to keep the degrees of freedom (DoF) as low as possible and to ensure that the shape gradient is strictly negative along the specified descent direction. To address these concerns, *a posteriori* error estimators can offer valuable insights during the resolution of the shape optimization problem (1).

Several works in the literature have highlighted the significant potential of a posteriori error estimators in achieving optimal designs when addressing shape/topology optimization problems. In the work by [1], the authors introduced the Zienkiewicz-Zhu error estimator to improve the error arising from the approximation of the differential problem, while the norm of the design gradient of the Lagrange function was used as an a posteriori error estimator for optimality conditions. This research was further developed by [2], where the dual-weighted residual method was employed to manage the PDE error, and the Laplace–Beltrami error indicator was used to address the geometric error related to domain deformations. We also refer to [3–5] for the application of adaptive mesh refinement in the shape/topology optimization problems.

In this work, in order to enhance the quality of finite element analysis and improve the numerical solution beyond what fixed mesh refinement offers, we propose an adaptive approach using a posteriori error estimators based on the dual-weighted residual (DWR) method [6]. This approach considers not only errors from the discretization of the constraint PDE, specifically the linear elasticity system, but also accounts for errors from the discretization of the deformation bilinear form, which provides a descent direction. To evolve the mesh Ω according

☆ This article is part of a Special issue entitled: 'EXCO_Numerical Methods 2025' published in Examples and Counterexamples.

* Corresponding author.
E-mail addresses: oguz.altintas@metu.edu.tr (O.H. Altıntaş), yucelh@metu.edu.tr (H. Yücel).

https://doi.org/10.1016/j.exco.2026.100225
Received 17 June 2025; Received in revised form 8 December 2025; Accepted 7 May 2026
Available online 14 May 2026
2666-657X/© 2026 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

to the descent direction, we employ the level-set method [7], which uses an implicit representation of the geometry. The step length in the gradient-based optimization required to update the level-set function is adaptively adjusted by estimating the Lipschitz constant.

The rest of this paper is organized as follows. In Section 2, we introduce the problem statement under the appropriate assumptions and notation. The shape derivative of our shape optimization problem regarding the compliance minimization objective with penalized volume fraction is given in Section 3. The deformation bilinear form that provides a descent direction and the level-set method are also provided in this section. Section 4 focuses on a posteriori error estimators for adjusting the step length and the mesh size and presents an overall adaptive procedure for solving the underlying shape optimization problem. Finally, some numerical results are provided in Section 5 to test the performance of the proposed adaptive procedure.

2. The problem

Throughout this paper, we consider a bounded, Lipschitz domain $\Omega \subset D \subset \mathbb{R}^2$, where D is a fixed working domain. The boundary of Ω is composed of three disjoint subregions Γ_d, Γ_n, and Γ_f such that $\partial\Omega = \overline{\Gamma}_d \cup \overline{\Gamma}_n \cup \overline{\Gamma}_f$. The regions Γ_d and Γ_n are fixed parts of the boundary and correspond to Dirichlet and Neumann boundary conditions, respectively, whereas Γ_f represents the free boundary, which is the only part of the boundary that can be optimized. Under the appropriate regularity conditions, the displacement $u : \Omega \to \mathbb{R}^2$ is solution of the linearized elasticity system

$$
\begin{aligned}
-\mathrm{div}(A\,\nabla^s u) &= 0 && \text{in } \Omega, && (2a)\\
u &= 0 && \text{on } \Gamma_d, && (2b)\\
(A\nabla^s u)\,n &= g && \text{on } \Gamma_n, && (2c)\\
(A\nabla^s u)\,n &= 0 && \text{on } \Gamma_f, && (2d)
\end{aligned}
$$

where $g : \Gamma_n \to \mathbb{R}^2$ is the loading force, $\nabla^s u = \frac{1}{2}(\nabla u^\top + \nabla u) : \Omega \to \mathbb{R}^{2\times 2}$ is the symmetric strain tensor field, A is the Hooke elasticity tensor, $A\,\nabla^s u : \Omega \to \mathbb{R}^{2\times 2}$ is the stress tensor field, and n is the unit outward normal vector to $\partial\Omega$.

Instead of working with the variable domain Ω, we transfer the computational effort to a fixed domain D by employing the method known as the *ersatz material*; see, e.g., [8]. In the ersatz material method, the strong phase material is discriminated from void (or weak phase) material by stating the Hooke tensor as

$$
A_\Omega = A\,\chi_\Omega + \epsilon A\,\chi_{D\setminus\Omega}, \tag{3}
$$

where ϵ is a given small parameter, i.e., $0 < \epsilon \ll 1$, χ_S denotes the indicator function of a region S. Here, A lays in the strong phase $\Omega \subset D$, whereas ϵA stays in the weak phase $D\setminus\Omega$. For any symmetric matrix ζ, the Hooke's law A of a linear isotropic elastic material is characterized by $A\zeta = 2\mu\zeta + \lambda\mathrm{tr}(\zeta)\mathtt{Id}$, where $\mu, \lambda \in \mathbb{R}$ are the Lamé moduli of the material, $\mathrm{tr}(\cdot)$ denotes the trace operator, and $\mathtt{Id}$ is the 2×2 identity matrix. In the ersatz material approach, the optimization process is still carried out concerning the variable set Ω, but Ω is incorporated within the fixed, larger set D. Hence, the elasticity system is rewritten as follows

$$
\begin{aligned}
-\mathrm{div}(A_\Omega\,\nabla^s u) &= 0 && \text{in } D, && (4a)\\
u &= 0 && \text{on } \Gamma_d, && (4b)\\
(A_\Omega\nabla^s u)\,n &= g && \text{on } \Gamma_n, && (4c)\\
(A_\Omega\nabla^s u)\,n &= 0 && \text{on } \Gamma_f, && (4d)
\end{aligned}
$$

where $\partial D = \overline{\Gamma}_d \cup \overline{\Gamma}_n \cup \overline{\Gamma}_f$. In this formulation the interface between the weak and strong phases, denoted by $\partial\Omega$, corresponds to the sole free interface. For small values of ϵ, the solution of (4) corresponds to an approximation of the solution to (2); see [9] for possible stability issues.

Introducing the Hilbert space $H^1_d(D)^2 := \{v \in H^1(D)^2 : v = 0 \text{ on } \Gamma_d\}$ equipped with $H^1(D)^2$-norm, we state the weak form of (4) as follows: Find $u \in H^1_d(D)^2$ such that

$$
a(u,v) := \int_D A_\Omega\,\nabla^s u : \nabla^s v\,dx = \int_{\Gamma_n} g \cdot v\,ds \quad \forall\, v \in H^1_d(D)^2, \tag{5}
$$

where the notation : represents the Frobenious inner product of tensors. Under the appropriate regularity conditions, the Lax–Milgram theorem allows us to assert that the problem described by (5) is well-posed; see, e.g., [10].

In this paper, our goal is the minimization of compliance functional under a volume constraint

$$
\min_{\Omega\in\mathcal{U}^{\mathrm{ad}}} J(\Omega) = \int_{\Gamma_n} g \cdot u\,ds \quad \text{subject to} \quad (4), \tag{6}
$$

with the set of admissible shapes $\mathcal{U}^{\mathrm{ad}}$, given by

$$
\mathcal{U}^{\mathrm{ad}} = \left\{\Omega \in \Omega_\varepsilon : \Gamma_d \cup \Gamma_n \subset \partial\Omega,\ 0 < |\Omega| = \varsigma\,|D|\right\},
$$

where $\Omega_\varepsilon := \{\Omega \subset D : \Omega \text{ is open and satisfies the } \varepsilon\text{-cone property}\}$ (see, e.g., [11]) and $\varsigma \in (0,1)$ is the given fraction of perturbed domain volume $|\Omega| = \int_\Omega dx$ to the fixed working domain volume $|D|$. To satisfy the volume constraint in practice, we penalize the constraint as follows

$$
\min_{\Omega\in D} J^P(\Omega) = J(\Omega) + \frac{\Lambda}{2}\left(\varsigma - \frac{|\Omega|}{|D|}\right)^2 \tag{7}
$$

with the penalty parameter $\Lambda > 0$.

3. Shape derivative with level-set method

To compute the derivatives with respect to shape Ω, we recall the notion of Hadamard's boundary variation method; see, e.g., [12]. In this framework, variations of a given shape Ω is defined by $\Omega_t = (\mathtt{Id}+\theta)(\Omega)$, where $\mathtt{Id}$ is the identity map and the deformation vector $\theta \in W^{1,\infty}(\mathbb{R}^2)$ is given by $\dot{x}(t) = \theta(x(t)), x(0) = x_0$ with the associated flow $\Phi_t : \mathbb{R}^2 \to \mathbb{R}^2$ and $t \in [0,\tau]$ defined for each $x_0 \in \mathbb{R}^2$ as $\Phi_t(x_0) = x(t)$, where $x : [0,\tau] \to \mathbb{R}^2$.

The functional $J(\Omega)$ is then said to be shape differentiable at Ω if it has the Eulerian semiderivative at Ω defined by $\mathrm{d}J(\Omega)(\theta) = \lim_{t\to 0+}(J(\Phi_t(\Omega)) - J(\Omega))/t$ for all directions $\theta \in C^\infty_0(D,\mathbb{R}^2)$ and the mapping $\theta \mapsto \mathrm{d}J(\Omega)(\theta) : C^\infty_0(D,\mathbb{R}^2) \to \mathbb{R}$ is linear and continuous. Under this setting, the objective function in (7), that is, $J^P(\Omega)$, is shape differentiable at Ω and the domain expression of the shape derivative is given by

$$
\mathrm{d}J^P(\Omega)(\theta) = \int_D S_\Omega : \nabla\theta\,dx - \widetilde{\Lambda}\int_\Omega \mathrm{div}\theta\,dx, \tag{8}
$$

where $S_\Omega = 2\nabla u^\top A_\Omega \nabla^s u - A_\Omega \nabla^s u : \nabla^s u\mathtt{Id}$, $\widetilde{\Lambda} = \frac{\Lambda}{|D|}\left(\varsigma - \frac{|\Omega|}{|D|}\right)$. For the details of the derivation, we refer to [13,14] and references therein. We also note that the volumetric shape derivative is preferred over the surface expression by considering the performance of the volumetric expression with finite element approximation [15].

To find a descent direction $\theta \in H^1(D)^2$ such that $\mathrm{d}J^P(\Omega;\theta) < 0$ in the gradient-based optimization process, we solve a deformation equation [16] in the form

$$
b(\theta,\varphi) = -\mathrm{d}J^P(\Omega;\varphi) \quad \forall\varphi \in H^1(D)^2, \tag{9}
$$

where a positive definite bilinear form $b : H^1(D)^2 \times H^1(D)^2 \to \mathbb{R}$ is given by

$$
b(\theta,\varphi) = \int_D \left(\tau_1\nabla\theta : \nabla\varphi + \tau_2\theta\cdot\varphi\right)dx
$$

with $\tau_1, \tau_2 > 0$ and $\theta\cdot n = 0$ on ∂D. For $H^1(D)^2 \ni \theta \neq 0$, we directly have $\mathrm{d}J^P(\Omega;\theta) = -b(\theta,\theta) < 0$, which implies that θ is a descent direction.

With the computation of θ, it is sufficient to deform the mesh Ω following the descent direction θ. However, capturing the shapes of domains that evolve or are perturbed in the direction of θ is not

straightforward. To address this, we adopt the level-set method [7], which utilizes an implicit representation of the geometry. In the level-set framework, the evolving domain $\Omega_t \subset D$ for $t \in (0,T)$ is described by a Lipschitz-continuous function $\psi : D \times (0,T) \to \mathbb{R}$ such that

$$\Omega_t := \{x \in D : \psi(x,t) < 0\} \quad \text{with} \quad \partial\Omega_t = \{x \in D : \psi(x,t) = 0\}.$$

Let the flow $x(t) = \Phi_t(x_0)$ be the position of points on $\partial\Omega_t$ from an initial Ω_0, with velocity vector $\dot{x}(t) = \theta(x(t))$, $x(0) = x_0$. Differentiating the relation $\psi(x(t),t) = 0$ with respect to t, we obtain a Hamilton–Jacobi type equation (HJE), which characterizes the evolution of the shape during the optimization process

$$\frac{\partial \psi}{\partial t} + \theta \cdot \nabla \psi = 0 \quad \text{in } D \times (0,T). \tag{10}$$

Thus, we can state that solving (10) with an appropriate initial condition $\psi(x,0)$ is equivalent to deform the boundary of each Ω_t along the descent direction $-\mathrm{d}J^P(\Omega_t)$. To solve the HJE in (10), we apply the Lax–Friedrichs flux scheme with a forward Euler discretization in fictitious time steps Δt; see [17] for a detailed discussion. The time step size is determined by the Courant–Friedrichs–Lewy (CFL): $\Delta t = \alpha_k \min(\Delta x, \Delta y)/\|\theta\|_\infty$, where α_k is the step length in the optimization process, Δx and Δy are the step sizes of the finite difference discretization in the x and y directions, respectively.

4. Adaptive process

In this section, we first adaptively determine the step length in the gradient-based optimization process and then propose a posteriori error estimates by following the dual-weighted residuals (DWR) method. Finally, the overall adaptive process will be presented.

4.1. Adaptive choice of step length

Taking into account the level-set method, we iterate the shapes $\Omega_{k+1} = (\mathrm{Id} + \alpha_k \theta_k)(\Omega_k)$ for $\Omega_k \subset D \ \forall k \in \mathbb{N}$, where $\alpha_k > 0$ is the step length of how much vector field θ_k is scaled to move the points of Ω_k. To obtain a sufficient decrease in the smooth objective $J^P(\Omega)$, it is required that $\alpha_k := \alpha = 1/L$, where L represents the Lipschitz constant of the shape derivative $\mathrm{d}J^P(\Omega)$. To efficiently estimate the Lipschitz constant, we suggest an adaptive approach inspired by the Lipschitz continuity of the shape derivative

$$\|\mathrm{d}J^P(\Omega_k) - \mathrm{d}J^P(\Omega_{k-1})\| \le L_k \, d(\Omega_k, \Omega_{k-1}), \tag{11}$$

where the function $d(\Omega_k, \Omega_{k-1})$ measures the distance between the points of the current and the previous shapes. Then, we approximate it as done in [18] via

$$\widehat{d}(\Omega_k, \Omega_{k-1}) = \int_{\Omega_k} \max_{y \in \Omega_{k-1}} \|x - y\| \, dx.$$

Recall that the points are evolved with respect to initial value problem $\dot{x}(t) = \theta(x(t))$, $x(0) = x_0$. For each fictitious time duration t, we have

$$x(t) = x(0) + \int_0^t \theta(x(\tau))\mathrm{d}\tau \approx x(0) + \theta(x(0))t, \tag{12}$$

where $t > 0$ corresponds to the final time of the discrete HJE problem. Then, by (12), we get an approximation for the distance between points for each iteration $k \in \mathbb{N}$ as

$$\begin{aligned} \widehat{d}(\Omega_k, \Omega_{k-1}) &= \int_{\Omega_k} \max_{x_{k-1} \in \Omega_{k-1}} \|x - x_{k-1}\| \, dx \\ &\approx \int_{\Omega_k} \max_{x_{k-1} \in \Omega_{k-1}} \|\theta(x_{k-1})t\| \, dx = t\|\theta(x_{k-1})\|_\infty |\Omega_k|. \end{aligned} \tag{13}$$

Inserting (13) into (11), we obtain an estimate for L_k

$$L_k = \frac{\|\mathrm{d}J^P(\Omega_k) - \mathrm{d}J^P(\Omega_{k-1})\|}{t\|\theta(x_{k-1})\|_\infty |\Omega_k|}. \tag{14}$$

4.2. Goal-oriented mesh refinement

Let $\{\mathcal{T}_h\}_h$ be a quasi-uniform family of triangulations of D with $\overline{D} = \bigcup_{K \in \mathcal{T}_h} \overline{K}$ and $h = \max_{K \in \mathcal{T}_h} h_K$, where h_K is the diameter of an element K. Moreover, $\mathcal{E}_h$ denotes the partition of the boundary ∂D induced by $\mathcal{T}_h$, meaning that each $E \in \mathcal{E}_h$ with $E \subset \partial D$ is an edge of some triangle $K \in \mathcal{T}_h$. Denoting the set of all linear polynomials on K by $\mathbb{P}(K)$, we introduce the following finite element space

$$V_h = \{v \in C(D)^2 : v_h|_K \in \mathbb{P}(K) \ \forall K \in \mathcal{T}_h \text{ and } v_h|_{\Gamma_d} = 0\} \subset H^1_d(D)^2.$$

Then, we seek to find $u_h \in V_h$ such that

$$a(u_h, v_h) = \int_D A_\Omega \nabla^s u_h : \nabla^s v_h \, dx = \int_{\Gamma_n} g \cdot v_h \, ds \ \ \forall v_h \in V_h, \tag{15}$$

see, e.g., [10, Chapter 11] for the existence and uniqueness of the Galerkin solution.

To construct a sequence of meshes $\{\mathcal{T}_h^k\}_{k \ge 0}$ in an adaptive manner, we use the tools of the DWR method [6] for both compliance and deformation errors. In the following we report only the main results, and their proofs can be easily derived by following the steps in [6].

Proposition 1. *Let u and u_h be solutions of (5) and (15), and let further z be the exact dual solution. For the linear functional $\mathcal{Q}^c := J(\Omega)$, it holds that*

$$|\mathcal{Q}^c(u) - \mathcal{Q}^c(u_h)| \le \sum_{K \in \mathcal{T}_h} \rho_K^u \omega_K^z := \eta^c, \tag{16}$$

where the element residual ρ_K^u and the local weight ω_K^z are given by

$$\begin{aligned} \rho_K^u &= \|\mathrm{div}(A_\Omega \nabla^s u_h)\|_K + h_K^{-1/2} \|[\![(A_\Omega \nabla^s u_h)n]\!]\|_{\partial K}, \\ \omega_K^z &= \|z - I_h z\|_K + h_K^{1/2} \|z - I_h z\|_{\partial K}, \end{aligned}$$

respectively. Here, $I_h z \in V_h$, $h_K = |K|^{1/2}$, and $[\![\cdot]\!]$ denotes jump across interelement sides, defined by

$$[\![(A_\Omega \nabla^s u_h)n]\!]|_E = \frac{1}{2}\left((A_\Omega \nabla^s u_h)n|_{K_1} - (A_\Omega \nabla^s u_h)n|_{K_2}\right)$$

for $E \in K_1 \cap K_2$ with $K_1, K_2 \in \mathcal{T}_h$, and n is unit normal vector to ∂K_1. On the boundary Γ_n, we set $[\![(A_\Omega \nabla^s u_h)n]\!] = g - (A_\Omega \nabla^s u_h)n$.

Motivated by the role of the shape derivative $\mathrm{d}J^P(\Omega)$ in the update of the geometry of the current domain Ω, we determine the deformation error by setting $\mathcal{Q}^d := -\mathrm{d}J^P(\Omega)(\theta)$ as an additional quantity of interest.

Proposition 2. *Let θ be the continuous solution of (9) and θ_h be the corresponding discrete solution. Let further ζ denote the dual solution of the auxiliary variational form (9). For the linear functional $\mathcal{Q}^d$, we have*

$$|\mathcal{Q}^d(\theta) - \mathcal{Q}^d(\theta_h)| \le \sum_{K \in \mathcal{T}_h} \rho_K^\theta \omega_K^\zeta := \eta^d, \tag{17}$$

where the element residual ρ_K^θ and the local weight ω_K^ζ are

$$\begin{aligned} \rho_K^\theta &= \|\mathrm{div}(\tau_1 \nabla \theta_h + S_\Omega) - \tau_2 \theta_h\|_K \\ &\quad + h_K^{-1/2} \|[\![-(\tau_1 \nabla \theta_h + S_\Omega - \widetilde{A}\,\chi_\Omega \, \mathrm{Id})n]\!]\|_{\partial K}, \\ \omega_K^\zeta &= \|\zeta - I_h \zeta\|_K + h_K^{1/2} \|\zeta - I_h \zeta\|_{\partial K}. \end{aligned}$$

Here, it is noted that S_Ω depends on u_h, that is, $S_\Omega = S_\Omega(u_h)$.

In (16) and (17), one still needs to compute the exact dual solutions z and ζ in order to obtain a fully a posteriori error estimator. As a remedy, we define an extrapolation procedure to the Galerkin approximations $z_h, \zeta_h \in V_h$. Let $\widetilde{V}_h \supset V_h$ be a higher order finite element (or enriched) space on the same triangulation $\mathcal{T}_h$. Then, we define an extrapolation operator $E : V_h \to \widetilde{V}_h$ to compute the higher order approximations $Ez_h, E\zeta_h \in \widetilde{V}_h$ with increased polynomial order.

Finally, in order to balance the sum of different goals as discussed in [19], we scale the errors as follows:

$$\mathcal{Q} = \frac{|\mathcal{Q}^c(u) - \mathcal{Q}^c(u_h)|}{|\mathcal{Q}^c(u_h)|} + \frac{|\mathcal{Q}^d(\theta) - \mathcal{Q}^d(\theta_h)|}{|\mathcal{Q}^d(\theta_h)|}. \tag{18}$$

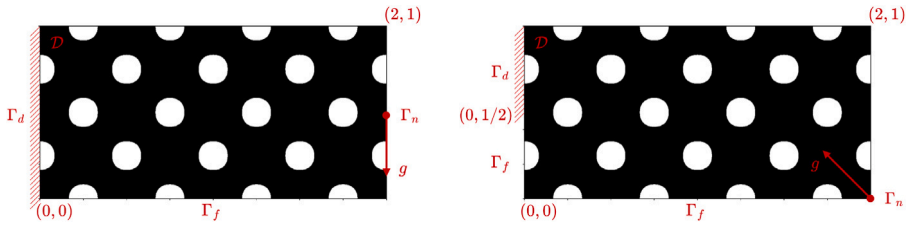

Fig. 1. Initial level-set function defined on the computational domain $\mathcal{D}$, including prescribed boundary conditions and load force $g(x)$ for the **Case I** (left) and **Case II** (right) setups.

Algorithm 1 An adaptive procedure for shape optimization.

Require: Initialize the working domain $\mathcal{D}$, the level-set function ψ^0, the parameters $\alpha_0, \Lambda, \varsigma, \epsilon, \tau_1, \tau_2$, the initial triangulation $\mathcal{T}_h^0$ with h^0, the reference mesh size h^*, tolerance values $\varepsilon_1, \varepsilon_2$.

$\Omega_0 \leftarrow \{x \in \mathcal{D} : \psi^0(x) < 0\}$ and $\theta_0 \leftarrow 0, k \leftarrow 0$
while $\max |J^P_{k:k-5} - J^P_k| > \varepsilon_1 |J^P_k|$ **do**
  $u_h, z_h, \theta_h, \zeta_h \leftarrow$ `solve`$(\mathcal{T}_h^k, \Omega_k)$
  $\eta_K^c, \eta_K^d \leftarrow$ `estimate`$(\mathcal{T}_h^k, \Omega_k, u_h, z_h, \theta_h, \zeta_h)$
  $J_k^P \leftarrow$ `compute_cost`$(\mathcal{T}_h^k, \Omega_k, u_h)$
  $\mathrm{d}J_k^P \leftarrow$ `compute_shape_derivative`$(\mathcal{T}_h^k, \Omega_k, u_h, z_h, \theta_h)$
  if $\eta(\eta_K^c, \eta_K^d) > \varepsilon_2$ **and** $h_{min}(\mathcal{T}_h^k) > h^*$ **then**
    $\mathcal{T}_h^{k+1} \leftarrow$ `mark_and_refine`$(\mathcal{T}_h^k, \eta^c, \eta^d)$, and **go to** Step 3
  end if
  if $k > 1$ **then**
    $\alpha_k \leftarrow$ `estimate_step_length`$(\mathrm{d}J_k, \mathrm{d}J_{k-1}, \Omega_k)$
  end if
  $\mathcal{T}_h^{k+1} \leftarrow$ `reset_mesh`$(\mathcal{T}_h^k)$
  $\psi^{k+1} \leftarrow$ `solve_hje`$(\psi^k, \theta_h, \alpha_k)$
  $\Omega_{k+1} \leftarrow \{x \in \mathcal{D} : \psi^{k+1}(x) < 0\}$
  $\theta_{k+1} \leftarrow \theta_k$ and $k \leftarrow k+1$
end while

4.3. Adaptive algorithm

Now we give an adaptive algorithm to solve the shape optimization problem (6) in Algorithm 1 and detail the subroutines in the algorithm. Recall that $k \geq 1$ stands for the optimization iteration and Ω_k is the current shape with deformable boundary $\partial\Omega_k$. Let $V_h^k = V_h(\mathcal{T}_h^k)$ be the finite element space with the current shape Ω_k.

In the subroutine `solve`, we first solve the primal problems in (5) and (9), and the corresponding dual problems by utilizing the finite element space $V_h^k = V_h(\mathcal{T}_h^k)$ with the current shape Ω_k. Then, the dual-weighted residual error estimators defined in (16) and (17) are computed in the subroutine `estimate`. In the computation of error indicators, we also solve the following dual problems

$$a^*(z_h, v) = \left(\delta_u \mathcal{Q}^c(u_h), v\right) \quad \text{and} \quad b^*(\zeta_h, \varphi) = \left(\delta_\theta \mathcal{Q}^d(\theta_h), \varphi\right)$$

for all $v, \varphi \in \widetilde{V}_h^k$. Here, $\widetilde{V}_h^k$ is the enriched space via quadratic polynomial order. Next, the subroutines `compute_cost` and `compute_shape_derivative` simply compute the penalized compliance (7) and the corresponding shape derivative in (8), respectively. The subroutine `mark_and_refine` marks some elements of $\mathcal{T}_h^k$ according to the bulk criteria and then refine the marked elements via bisection while keeping the conformity of $\mathcal{T}_h^k$, if the scaled estimator is higher than the given tolerance value or the current mesh size is not reached the minimum mesh size h^*. Otherwise, we reset the current mesh and return to the initial original mesh. Before the update current shape Ω_k, the step length α_k is computed according to the approximated Lipschitz constant L_k in the subroutine `estimate_step_length` via (14). After solving the HJE in (10) by using a finite difference scheme in the subroutine `solve_hje`, we update the current shape Ω_k.

5. Numerical results

In this section, we carry out some numerical experiments to illustrate the validity and efficiency of the proposed adaptive algorithm in Section 4.3 to solve the well-known structural optimization problem, namely, the *cantilever beam*; see, e.g., [4,8,13]. To solve the bilinear forms (5) and (9), we use a conventional finite element approach with linear Lagrange elements on a *crossed* triangulation $\mathcal{T}_h$. Isotropic materials are considered in the form of $\mu = E/(2(1+\nu)), \lambda = E\nu/((1+\nu)(1-2\nu))$, where Young's module $E = 1$ and Poisson's ratio $\nu = 0.3$. The parameters used in the process of Algorithm 1 are taken as follows:

$$\epsilon = 10^{-3}, \quad \alpha_0 = 10^{-2}, \quad \tau_2 = 1, \quad \varsigma = 0.3,$$
$$\varepsilon_1 = 10^{-3}, \quad \varepsilon_2 = 10^{-1}, \quad \Lambda = 100, \quad h^* = 1/180.$$

We consider two test setup that differ in the location of the boundary conditions and in the direction of the applied point load, defined as follows:

- **Case I:** $\Gamma_d = \{0\} \times [0, 1]$ and $\Gamma_n = \{(2, \frac{1}{2})\}$, with $g(x) = (0, -10)$.
- **Case II:** $\Gamma_d = \{0\} \times \left[\frac{1}{2}, 1\right]$ and $\Gamma_n = \{(2, 0)\}$, with

$$g(x) = \left(-10\cos(\tfrac{\pi}{4}),\ 10\sin(\tfrac{\pi}{4})\right).$$

The remaining portion of the boundary is treated as the free boundary Γ_f. Furthermore, the initial domain Ω_0 is defined by the level-set function $\psi^0(x, y) = -\cos(4\pi x)\cos(4\pi y) - 0.5$, as illustrated in Fig. 1 for both test cases. All computations are performed using the FEniCS/DOLFIN library on an Intel(R) Core(TM) i5-6500 processor (3.20 GHz) with 8 GB of RAM, based on the implementation of Laurain [13].

We start by analyzing the convergence of the adaptive process in Algorithm 1 in Fig. 2. Fig. 3 shows that the final designs obtained from both the fixed mesh and the adaptively refined mesh are almost identical. To set up a comparison in terms of computational cost between fixed mesh and adaptive mesh processes, we define a computation index in the form of

$$\mathtt{CI} \propto \sum_{i=1}^{N} \sum_{j=1}^{P} \varrho(d_{ij}), \tag{19}$$

where d_{ij} is the number of degrees of freedom (DoF) at ith iteration and jth level of mesh refinement with N number of iterations and P number of levels. Here, ϱ represents the time complexity of linear solver in terms of DoF. In the FEM solutions, we use MUMPS linear solver implemented in FEniCS/DOLFIN libraries with the time complexity of $O(d^{3/2})$; see, e.g., [20]. Table 1 demonstrates that, in both cases, the adaptive process significantly reduces the total runtime and the CI

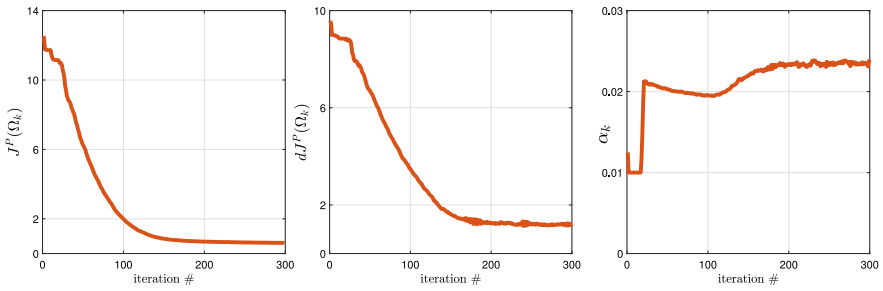

Fig. 2. Convergence behavior of the penalized compliance $J^P(\Omega_k)$ (left), the shape derivative $\mathrm{d}J^P(\Omega_k)$ (middle), and the step length α_k (right) on adaptively refined meshes for the **Case I** configuration with an initial mesh size of $h^0 = 1/60$.

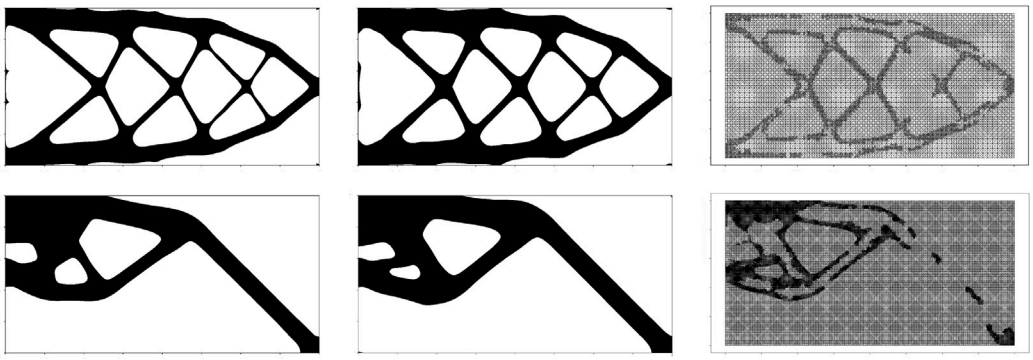

Fig. 3. Final designs obtained on fixed and adaptively generated meshes for **Case I** (top) with initial mesh size $h^0 = 1/60$ and **Case II** (bottom) with initial mesh size $h^0 = 1/90$. For each case, the fixed-mesh design (left), the adaptive-mesh design (middle), and the corresponding adaptive mesh (right) are shown.

Table 1
Comparison of the computational effort on fixed and adaptively generated meshes in terms of degrees of freedom (DoF), total runtime T_{total}, and the computational index (CI) defined in (19), for Case I and Case II.

	DoF	T_{total} (hours)	CI
Case I			
Fixed mesh	260,282	3.68	2.25×10^{10}
Adaptive mesh	29,162 → 45,004	1.46	4.35×10^{9}
Case II			
Fixed mesh	260,282	10.26	6.87×10^{10}
Adaptive mesh	65,342 → 94,046	4.28	1.69×10^{10}

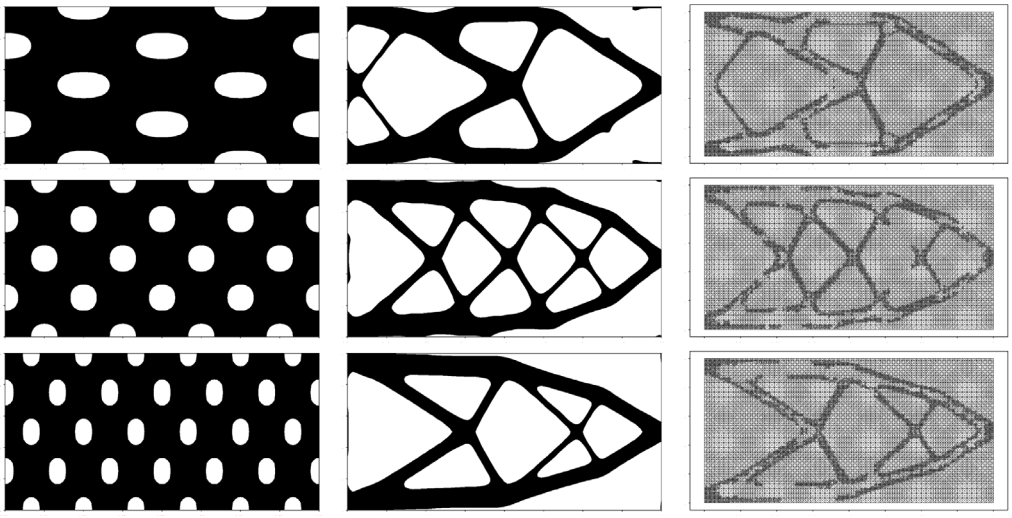

Fig. 4. Effect of different initial level-set functions (left) on the resulting final designs (middle) and their associated adapted meshes (right) for the **Case I** configuration with an initial mesh size of $h^0 = 1/60$.

index compared to the fixed-mesh approach, even though both methods yield comparable final designs.

One of drawbacks of the level-set approach is that the optimal solution usually depends on the initial guess, as shown in Fig. 4. To mitigate this issue, future studies can use a reaction–diffusion equation-based level-set method [21]. Another challenge in this study concerns mesh coarsening. Coarsening enables the mesh to follow the evolving level-set interface by removing unnecessary fine elements, thereby improving efficiency while maintaining accuracy. This is particularly important in level-set-based methods, where topological changes occur naturally and over-refinement away from the interface may introduce numerical noise and affect optimization stability. Since the FEniCS/DOLFIN framework does not support mesh coarsening, Algorithm 1 employs an alternative approach in which the mesh is reset to the initial coarse discretization. While this strategy is adequate for the benchmark problems studied, more sophisticated applications will require a fully integrated mesh coarsening capability. Furthermore, adaptive mesh generation influences the step length computation, which depends on differences between successive mesh configurations; abrupt or overly small mesh changes may negatively affect the stability and convergence of the gradient-based optimization process.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Hamdullah Yücel reports financial support was provided by Scientific and Technological Research Council of Türkiye. Hamdullah Yücel reports a relationship with Turkish Academy of Sciences that includes: funding grants. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

HY gratefully acknowledges the research support provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under the program TÜBİTAK 2219-International Postdoctoral Research Fellowship (Project No. : 1059B192302207) and would like to thank the Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg for its excellent hospitality.

Data availability

No data was used for the research described in the article.

References

[1] N. Banichuk, F.-J. Barthold, A. Falk, E. Stein, Mesh refinement for shape optimization, Struct. Optim. 9 (1995) 46–51.
[2] P. Morin, R.H. Nochetto, M.S. Pauletti, M. Verani, Adaptive finite element method for shape optimization, ESAIM: COCV 18 (2012) 1122–1149.
[3] M. Giacomini, O. Pantz, K. Trabelsi, Certified descent algorithm for shape optimization driven by fully-computable a posteriori error estimators, ESAIM: COCV 23 (2017) 977–1001.
[4] M. Bruggi, M. Verani, A fully adaptive topology optimization algorithm with goal-oriented error control, Comput. Struct. 89 (2011) 1481–1493.
[5] A. Schleupen, K. Maute, E. Ramm, Adaptive FE-procedures in shape optimization, Struct. Multidiscip. Optim. 19 (2000) 282–302.
[6] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, Acta Numer. 10 (2001) 1–102.
[7] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[8] G. Allaire, F. Jouve, A.-M. Toader, Structural optimization using sensitivity analysis and a level-set method, J. Comput. Phys. 194 (2004) 363–393.
[9] M. Dambrine, D. Kateb, On the ersatz material approximation in level-set methods, ESAIM Control Optim. Calc. Var. 16 (2010) 618–634.
[10] S.C. Brenner, L.R. Scott, The Mathematical Theory of Finite Element Methods, third ed., Springer, Berlin, 2008.
[11] D. Chenais, On the existence of a solution in a domain identification problem, J. Math. Anal. Appl. 52 (1975) 189–219.
[12] J. Sokołowski, J.-P. Zolésio, Introduction to shape optimization, in: Springer Series in Computational Mathematics, vol. 16, Springer-Verlag, Berlin, 1992.
[13] A. Laurain, A level set-based structural optimization code using FEniCS, Struct. Multidiscip. Optim. 58 (2018) 1311–1334.
[14] K. Sturm, Minimax Lagrangian approach to the differentiability of nonlinear PDE constrained shape functions without saddle point assumption, SIAM J. Control Optim. 53 (2015) 2017–2039.
[15] R. Hiptmair, A. Paganini, S. Sargheini, Comparison of approximate shape gradient, BIT Numer. Math. 55 (2014) 459–485.
[16] V. Schulz, M. Siebenborn, Computational comparison of surface metrics for PDE constrained shape optimization, Comput. Methods Appl. Math. 16 (2016) 485–496.
[17] S. Osher, C.-W. Shu, High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations, SIAM J. Numer. Anal. 28 (1991) 907–922.
[18] D. Luft, K. Welker, Computational investigations of an obstacle-type shape optimization problem in the space of smooth shapes, in: Geometric Science of Information: 4th International Conference, GSI 2019, Toulouse, France, August 27–29, 2019, Proceedings 4, Springer, 2019, pp. 579–588.
[19] R. Hartmann, Multitarget error estimation and adaptivity in aerodynamic flow simulations, SIAM J. Sci. Comput. 31 (2008) 708–731.
[20] P.R. Amestoy, I.S. Duff, J.-Y. L'Excellent, J. Koster, A fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM J. Matrix Anal. Appl. 23 (2001) 15–41.
[21] T. Yamada, K. Izui, S. Nishiwaki, A. Takezawa, A topology optimization method based on the level set method incorporating a fictitious interface energy, Comput. Methods Appl. Mech. Engrg. 199 (2010) 2876–2891.