Nesterov’s acceleration for level set-based topology optimization using reaction-diffusion equations

Tomoyuki Oka\textsuperscript{a}, Ryota Misawa\textsuperscript{b}, Takayuki Yamada\textsuperscript{a,*}

\textsuperscript{a}Graduate School of Engineering, The University of Tokyo, 
Yayoi 2-11-16, Bunkyo-ku, Tokyo 113-8656, Japan 
\textsuperscript{b}Graduate School of Science and Engineering, Saitama University, 
255 Shimo-Okubo, Sakura-ku, Saitama City, Saitama 338-8570, Japan

Abstract

This paper discusses level set-based structural optimization. Level set-based structural optimization is a method used to determine an optimal configuration for minimizing objective functionals by updating level set functions characterized as solutions to partial differential equations (PDEs) (e.g., Hamilton-Jacobi and reaction-diffusion equations). In this study, based on Nesterov’s accelerated gradient method, a nonlinear (damped) wave equation will be derived as a PDE satisfied by level set functions and applied to minimum mean compliance problems. Numerically, the method developed in this study will yield convergence to an optimal configuration faster than methods using only a reaction-diffusion equation, and moreover, its FreeFEM++ code will also be described.

Keywords: topology optimization, level set method, reaction-diffusion equation, nonlinear (damped) wave equation, Nesterov’s accelerated gradient method

2010 MSC: Primary: 80M50; Secondary: 35Q93, 47J35

\textsuperscript{*}This study is partially supported by a project JPNP20004 subsidized by the New Energy and Industrial Technology Development Organization (NEDO) and JSPS KAKENHI Grant Number JP22K20331.

\textsuperscript{*}Corresponding author

Email addresses: tomoyuki-oka@ecc.u-tokyo.ac.jp (Tomoyuki Oka),
t.yamada@mech.t.u-tokyo.ac.jp (Takayuki Yamada)
1. Introduction

Topology optimization is a structural optimization with the highest degree of design freedom. It is a method for determining an optimal material configuration (denoted by a set $\Omega_{\text{opt}} \subset \mathbb{R}^d$ ($d = 2$ or $3$) below) to minimize objective functionals and is currently being developed in various fields. Generally, the following objective functional $F_{\Omega} : H^1(\Omega)^d \to \mathbb{R}$ is treated in usual variational methods:

$$F_{\Omega}(v) = \int_{\Omega} f_d(x, v, \nabla v) \, dx + \int_{\partial\Omega} f_b(x, v, \nabla v) \, d\sigma,$$

where $f_d$ and $f_b$ are functions defined on $\Omega \times \mathbb{R}^d$ and $\partial\Omega \times \mathbb{R}^d \times \mathbb{R}^{d \times d}$, respectively. On the other hand, the following set function $F : \mathcal{U}_{\text{ad}} \to \mathbb{R}$ is treated in topology optimization:

$$F(\Omega) = \int_{\Omega} f_d(x, u_\Omega, \nabla u_\Omega) \, dx + \int_{\partial\Omega} f_b(x, u_\Omega, \nabla u_\Omega) \, d\sigma.$$

Here, $\mathcal{U}_{\text{ad}} := \{\Omega \subset D: \text{constraint conditions}\}$ and $D \subset \mathbb{R}^d$ denote a family of sets for admissible domains and a fixed design domain such that $\partial\Omega \cap \partial D \neq \emptyset$, respectively, and the state variable $u_\Omega \in H^1(\Omega)^d$ is a (vector-valued) function determined by fixing $\Omega \subset D$; for instance, a solution to the Euler-Lagrange equation (i.e., its Fréchet derivative $F'_{\Omega}$ is zero). Then consider the following minimization problem:

$$\inf_{\Omega \in \mathcal{U}_{\text{ad}}} F(\Omega). \quad (1.1)$$

Therefore, topology optimization can be regarded as a minimization problem with a set $\Omega \in \mathcal{U}_{\text{ad}}$ as a variable, and (1.1) can be replaced by the following distribution problem of materials:

$$\inf_{\chi_{\Omega} \in L^\infty(D;\{0,1\})} F(\chi_{\Omega}) \quad \text{subject to constraint conditions}, \quad (1.2)$$

where $F : L^\infty(D;\{0,1\}) \to \mathbb{R}$ is a functional given by

$$F(\chi_{\Omega}) = \int_D f_d(x, u_\Omega, \nabla u_\Omega)\chi_{\Omega}(x) \, dx + \int_{\partial D} f_b(x, u_\Omega, \nabla u_\Omega) \, d\sigma.$$
and $\chi_{\Omega} \in L^\infty(D)$ is a characteristic function defined by

$$
\chi_{\Omega}(x) = \begin{cases} 
1, & x \in \overline{\Omega} := \Omega \cup \partial \Omega, \\
0, & x \in D \setminus \overline{\Omega}.
\end{cases}
$$

Hence, topology optimization is the minimization problem for $\chi_{\Omega} \in L^\infty(D)$. Moreover, $\Omega_{\text{opt}} \in \mathcal{U}_{\text{ad}}$ may be characterized as the domain of the minimizer $\chi_{\Omega} \in L^\infty(D)$ in (1.2) such that $\chi_{\Omega} = 1$. Thus, topology optimization implies that changes in the shape of $\partial \Omega$ and the topology of $\Omega \in \mathcal{U}_{\text{ad}}$, such as an increase or decrease in the number of holes, can be allowed in the optimization procedure. However, various issues remain on how to determine $\Omega_{\text{opt}} \in \mathcal{U}_{\text{ad}}$ even if such configurations exist.

### 1.1. Homogenization-based topology optimization.

The existence of $\Omega_{\text{opt}} \in \mathcal{U}_{\text{ad}}$ for generalized problems is obtained by the homogenization theory based on $H$ (or $G$)-convergence (see, e.g., [1, Theorem 3.2.1] and [2]), and therefore, it forms the basis for numerical analysis. Homogenization is a method for replacing heterogeneous materials, which possess many microstructures, with an equivalent homogeneous material. As for the numerical analysis, the so-called homogenization design method was first developed in [3]. Moreover, its simplified version, the so-called *Solid Isotropic Material with Penalization* (SIMP) method [4], is frequently used by replacing the characteristic functions with density functions, but there are certain issues: (i) An optimized configuration $\Omega_{\text{opt}} \in \mathcal{U}_{\text{ad}}$ typically includes grayscale domains since the density function takes a value in $[0,1]$. In other words, $\partial \Omega_{\text{opt}}$ is not clearly expressed unless such domains are removed. (ii) True material properties of composite materials with microstructures are not devised due to oversimplification; indeed, it holds only material density (see, e.g., [5] for the resurrection of the homogenization method [6, 7] for filtering and [8, 9] for meshless).

### 1.2. Level set-based structural optimization

To overcome the aforementioned issues, a level set method, which was first proposed in a previous study [10] to implicitly represent the evolution of in-
terfaces, was introduced, and the following level set function \( \phi \in H^1(D) \) was employed:

\[
\begin{align*}
\phi(x) = \begin{cases} 
> 0, & x \in \Omega, \\
0, & x \in \partial \Omega, \\
< 0, & x \in D \setminus \overline{\Omega}.
\end{cases}
\end{align*}
\] (1.3)

Thus, \([ \phi > 0 ] := \{ x \in D : \phi(x) > 0 \}, [ \phi < 0 ] \) and \([ \phi = 0 ] \) represent material domains, void domains and structural boundaries, respectively, and \( \chi_\Omega \in L^\infty(D;\{0,1\}) \) in (1.2) can be replaced by

\[
\chi_\phi(x) = \begin{cases} 
1 & \text{if } \phi(x) \geq 0, \\
0 & \text{if } \phi(x) < 0.
\end{cases}
\]

Moreover, \( \phi \in H^1(D) \) is determined by combining Lagrange’s method of undetermined multipliers with the Karush-Kuhn-Tucker (KKT) conditions since \( F(\chi_\Omega) \) can also be replaced with \( F(\phi) \). However, the direct derivation of \( \phi \in H^1(D) \) is nearly impossible in general.

Alternatively, some methods that can be used to update (1.3) by introducing a fictitious time variable \( t \in (0, +\infty) \) have been devised. As a typical example for (1.3), the following signed distance function is known:

\[
\phi(x) = \begin{cases} 
d(x), & x \in \Omega, \\
0, & x \in \partial \Omega, \\
-d(x), & x \in D \setminus \overline{\Omega},
\end{cases}
\] (1.4)

where \( d(x) = \inf_{y \in \partial \Omega} |x - y| \). By noting that (1.4) solves an eikonal equation, the following Hamilton-Jacobi equation is derived by differentiating for (1.4) regarding \( t \in (0, \infty) \):

\[
\partial_t \phi(x,t) + v(x,t)|\nabla \phi(x,t)| = 0, \quad (x,t) \in D \times (0, +\infty),
\] (1.5)

where \( \partial_t = \partial / \partial t \) and \( v \) is a given function. In previous studies [11, 12, 13], the equivalence of transporting the solution to (1.5) and moving \( \partial \Omega \) along the
The descent gradient direction of functionals was used, and the shape derivative (based on the Fréchet derivative) was applied; therefore, optimized shape \( \partial \Omega_{opt} \) can be expressed by employing the solution to (1.5) and the topology can also be changed by reducing the number of holes, but changing the topologies by generating new holes is not; in other words, \( \Omega_{opt} \in \mathcal{U}_{ad} \) deeply depends on initial configurations (see also for other level set methods [14, 15, 16, 17, 18]).

With the aid of the bubble method [19], a concept of topological derivative was introduced in [20], and the issue of the dependence on initial configurations might have been resolved. The topological derivative [21, 22] represents an influence when a sufficiently small ball \( B_{\varepsilon}(x) \subset \Omega \) is created and is defined as follows:

**Definition 1.1 (Topological derivative).** Let \( A = \{ \Omega \subset D : \Omega \text{ is open in } D \} \). A function \( J \) defined in \( A \) is said to be topologically differentiable at \( \Omega_0 \) and at point \( x \in \Omega_0 \) if the following limit exists:

\[
d_TJ(\Omega_0, x) := \lim_{\varepsilon \to 0^+} \frac{J(\Omega_0 \setminus B_{\varepsilon}(x)) - J(\Omega_0)}{|B_{\varepsilon}(x)|}.
\]

Here, \( B_{\varepsilon}(x) := \{ y \in \Omega_0 : |x - y| < \varepsilon \} \) and \( |B_{\varepsilon}(x)| \) denotes the Lebesgue measure of \( B_{\varepsilon}(x) \).

**Example 1.2.** Let \( J(\Omega) = |\Omega| \). Then \( d_TJ(\Omega) = -1 \) and \( J(\Omega) \) can be regarded as the functional for \( \chi_\phi \in L^\infty(D) \) by noting that

\[
J(\Omega) = \int_D \chi_\phi(x) \, dx =: J(\chi_\phi),
\]

which implies that \( d_TJ(\Omega) \) can be identified with \(-J'(\chi_\phi)\).

Generally, the solution to (1.5) does not coincide with (1.4), and note that the bubble method is heuristic. Thus the issue of the dependence on initial configurations still remains (e.g., [23] for the reinitialization of level set functions).
1.3. Level set-based topology optimization using a reaction-diffusion equation

To avoid the dependence on initial configurations, in a previous study [24], (1.3) was modified as follows:

\[
\phi(x) = \begin{cases} 
  \in (0, 1], & x \in \Omega, \\
  = 0, & x \in \partial \Omega, \\
  \in [-1, 0), & x \in D \setminus \Omega.
\end{cases}
\]

(1.6)

Notably, (1.4) can no longer be taken as (1.6). Furthermore, based on the concept of the gradient descent method, the following time evolution equation was employed:

\[
\partial_t \phi = \rho d_T F,
\]

(1.7)

where \( \rho > 0 \). In terms of the regularity, the following (reaction) diffusion equation was applied:

\[
\partial_t \phi = \rho d_T F + \tau \Delta \phi \quad \text{in } D \times (0, +\infty),
\]

(1.8)

where \( \tau > 0 \). Thus \( \tau \Delta \phi \) plays a role in the regularization term since (1.8) is expected to have the smoothing effect, and the solution to (1.7) may be approximated by the solution to (1.8) for \( \tau > 0 \) small enough. Actually, \( \Omega_{opt} \in U_{ad} \) exhibits complex configurations if \( \tau > 0 \) is small enough and vice versa. Hence, an appropriate value \( \tau > 0 \) prevents the generation of excessively geometrically complex configurations, and \( \Omega_{opt} \in U_{ad} \) with the geometric simplicity is obtained. On the other hand, by employing \( d_T F \), the dependence on initial configurations is entirely removed since generating holes in \( D \) can be allowed. Therefore, in terms of practicality, level set-based topology optimization with (1.8) may be valid for various problems since the issues (grayscale domains, reinitialization for level set functions and dependence on initial configurations) are all removed. However, the mathematical justification of replacing the topological derivative for the Fréchet derivative and optimality remain future issues.
1.4. Setting of the optimization problem

This paper concerns an optimal design that minimizes the following volume-constrained mean compliance:

$$\inf_{\Omega \in \mathcal{U}_{\text{ad}}} \left\{ F(\Omega) := \langle \mathbf{t}, \mathbf{u}\Omega \rangle_{H^{1/2}(\Gamma_t)} \right\},$$

(1.9)

where

$$\mathcal{U}_{\text{ad}} = \{ \Omega \subset D : |\Omega| \leq G_{\text{max}} |D| \},$$

$G_{\text{max}} > 0$, the vector-valued function $\mathbf{u}\Omega \in C^2(\Omega)^d \cap C(\overline{\Omega})^d$ is a unique (classical) solution to the following linearized elasticity system:

$$\begin{cases}
-\text{div}[\mathbf{D}\varepsilon(\mathbf{u}\Omega)] = 0 & \text{in } \Omega, \\
\mathbf{u}\Omega = 0 & \text{on } \Gamma_D, \\
-\mathbf{D}\varepsilon(\mathbf{u}\Omega) \cdot \mathbf{n} = \mathbf{t} & \text{on } \Gamma_t, \\
-\mathbf{D}\varepsilon(\mathbf{u}\Omega) \cdot \mathbf{n} = 0 & \text{on } \Gamma_N := \partial\Omega \setminus (\Gamma_D \cup \Gamma_t).
\end{cases}$$

Here and henceforth, $\Gamma_D \cap \Gamma_t = \emptyset$, $|\Omega|$ represents the Lebesgue measure of $\Omega \subset \mathbb{R}^d$. The forth-order elastic tensor $\mathbf{D} = \mathbf{D}_{ijkl} e_i \otimes e_j \otimes e_k \otimes e_l$ and the strain tensor $\varepsilon(\mathbf{u}\Omega)$ are given by

$$\mathbf{D}_{ijkl} = \frac{E \nu}{(1 + \nu)(1 - 2\nu)} \delta_{ij} \delta_{kl} + \frac{E}{2(1 + \nu)} (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk})$$

for some $E, \nu > 0$ and

$$\varepsilon(\mathbf{u}\Omega) = \frac{1}{2} (\nabla \mathbf{u}\Omega + (\nabla \mathbf{u}\Omega)^T), \quad \nabla \mathbf{u}\Omega = \partial_{x_i} (\mathbf{u}\Omega)_j e_i \otimes e_j,$$

respectively. The traction $\mathbf{t} \in \mathbb{R}^d$ is a constant vector. In particular, $\mathbf{n}$, $\delta_{ij}$ and $e_k$ stand for the outer unit normal vector, Kronecker delta and $k$-th vector of the canonical basis of $\mathbb{R}^d$, respectively.

The optimization problem (1.9) can be replaced by the following minimization problem with the level set function $\phi \in H^1(D)$ given by (1.6):

$$\inf_{\phi \in H^1(D;[-1,1])} \left\{ F(\phi) := \int_{\Gamma_t} \mathbf{t} \cdot \mathbf{u}_\phi(x) \, d\sigma \right\}$$

(1.10)
subject to

\[ G(\phi) := \int_D (\chi_\phi(x) - G_{\text{max}}) \, dx \leq 0, \]  

(1.11)

where \( D \) is a bounded domain in \( \mathbb{R}^d \) such that \( \Omega \subset D \) and \( \partial D \) is the boundary of \( D \) with \( \Gamma_D \) and \( \Gamma_t \) such that \( \Gamma_D \cap \Gamma_t = \emptyset \), the state variable \( u_\phi \in V \) satisfies

\[ \int_D A_\phi(x) \varepsilon(u_\phi)(x) : \varepsilon(v)(x) \, dx = \int_{\Gamma_t} t \cdot v(x) \, d\sigma \quad \text{for all} \quad v \in V, \]  

(1.12)

\[ A_\phi = \mathbb{D}\chi_\phi + \varepsilon(1-\chi_\phi) \] and \( V = \{ v \in H^1(D)^d : v = 0 \text{ on } \Gamma_D \} \). Furthermore, by Lagrange’s method of undetermined multipliers, the objective functional \( F(\phi) \) in (1.10) is replaced with

\[ F(\phi, u_\phi, \tilde{u}, \lambda) = \int_{\Gamma_t} t \cdot (u_\phi(x) + \tilde{u}(x)) \, d\sigma \]

\[ - \int_D A_\phi(x) \varepsilon(u_\phi)(x) : \varepsilon(\tilde{u})(x) \, dx + \lambda G(\phi), \]  

(1.13)

which implies that the optimization problem (1.10)–(1.11) may also be replaced with the unconstrained minimization problem for (1.13). Here and henceforth, \( F \) denotes the Lagrangian of \( F \) and \( \lambda \geq 0 \) and \( \tilde{u} \in V \) stand for the Lagrange multiplier.

1.5. Aims and plan of this paper

In this paper, we shall provide a method for level set-based topology optimization that converges to optimized configurations faster than the method based on the (reaction) diffusion equation (1.8). To this end, instead of the usual gradient descent method, Nesterov’s accelerated gradient method [25] will be introduced, and a nonlinear (damped) wave equation will be applied as a partial differential equation (PDE) to update the level set function (see the next section below for a derivation).

This paper is organized as follows. In the next section, we shall set up time evolution equations to update the level set functions. In particular, we shall show that the level set function satisfies (1.8) and a nonlinear damped wave equation, according to the usual gradient descent method and Nesterov’s accelerated gradient method [25], respectively. Thus, Section 2 offers a new idea
and is the most contributing. Section 3 describes the numerical algorithm for the minimization problem of (1.13) and Section 4 deals with the main results of this paper. Furthermore, we shall emphasize that convergence to optimal configurations for the minimization problem of (1.13) is improved through typical numerical examples. The FreeFEM++ code will be described in Appendix A (see also [27]). The final section will conclude this paper.

2. Formulation of nonlinear hyperbolic-parabolic equations

To find \( \Omega_{\text{opt}} \in \mathcal{U}_{\text{ad}} \) for the minimization problem of (1.13), we shall formulate the equations satisfied by the level set functions. To this end, we shall first derive a reaction-diffusion equation. Noting that, for any functional \( F : H^1(D) \to \mathbb{R} \), it holds that

\[
F(\phi) \leq \mathcal{F}(\phi) + \frac{\tau}{2} \int_D |\nabla \phi(x)|^2 \, dx \quad \text{for all } \tau > 0. \tag{2.1}
\]

Let \( \mathcal{F} : H^1(D) \to \mathbb{R} \) be the right-hand side in (2.1). Then, by replacing \( F \) with \( \mathcal{F} \), the gradient descent method such as (1.7) yields

\[
\partial_t \phi = -\rho \mathcal{F}'(\phi) + \tau \Delta \phi, \tag{2.2}
\]

which implies that, for \( \tau > 0 \) small enough, one can choose (2.2) as the equation which the level set function in (1.13) satisfies by noting that

\[
\mathcal{F}(\phi) - \mathcal{F}(\phi) \to 0 \quad \text{as } \tau \to 0_+.
\]

**Remark 2.1.** We note that (2.2) does not coincide with (1.8). However, it is well-known that the replacement of \( F' \) with \( -d_T \mathcal{F} \) is adequate in various optimization problems; indeed, if \( \varphi \in H^1(D) \) is a critical point for \( \mathcal{F} \), then \( \partial_{x^a} \mathcal{F} |_{x^a = 0} = 0 \) follows since \( \mathcal{F}' = \partial_{x^a} \mathcal{F} \partial_{x^a} \varphi \) formally. As in Example 1.2 identifying \( \partial_{x^a} \mathcal{F} \) with \( -d_T \mathcal{F} \), we observe that, for any \( \psi \in H^1(D) \), \( \mathcal{F}(\varphi + k\psi) = \mathcal{F}(\varphi) + k\langle \mathcal{F}'(\varphi), \psi \rangle_{H^1(D)} + o(k) = \mathcal{F}(\varphi) - k\langle d_T \mathcal{F}, \psi \rangle_{H^1(D)} + o(k) \) as \( k \to 0_+ \). Hence, setting \( \psi = d_T \mathcal{F} \), we obtain (1.7) for a suitable initial level set function \( \phi_0 \in L^\infty(D) \), which implies that (2.2) can be replaced with (1.8) under this setting, and therefore, the replacement of \( \mathcal{F}' \) with \( -d_T \mathcal{F} \) and a perturbation of the Dirichlet energy yield (1.8).
2.1. Nonlinear damped wave equation

In this subsection, we shall establish another equation satisfied by (1.6) to converge faster to an optimized configuration. Similar to the gradient descent method, we recall that the following improved gradient descent method, so-called Nesterov’s accelerated gradient method, was developed in [25]:

\[
\psi_{n+1}(x) = \phi_n(x) - k\mathcal{F}'(\phi_n),
\]

(2.3)

\[
\phi_n(x) = \psi_n(x) + \frac{n-1}{n+2}(\psi_n - \psi_{n-1})
\]

(2.4)

for \( n \in \mathbb{N} \). Here, \( k > 0 \), \( \phi_0 \) stands for the initial level set function and \( \psi_0 = \phi_0 \). Therefore, the gradient descent method and (2.3)-(2.4) are equivalent until \( n = 1 \) (i.e., \( \phi_m = \psi_m \) for \( m = 0, 1 \)). Conversely, as for \( n \geq 2 \), the second term of the right-hand side in (2.4) plays a role in the inertia term.

Now, we consider another equation satisfied by (1.6). Let \( n \in \mathbb{N} \) be large enough and identified with \( n + 2 \) (i.e., \( n \simeq n + 2 \)). Then, setting \( k = (\Delta t)^2 \rho \) and noting that

\[
\psi_{n+1} = \psi_n + (1 - 3/n)(\psi_n - \psi_{n-1}) - k\mathcal{F}'(\phi_n),
\]

one can derive that

\[
\frac{\psi_{n+1} - 2\psi_n + \psi_{n-1}}{(\Delta t)^2} + \frac{3}{n\Delta t} \frac{\psi_n - \psi_{n-1}}{\Delta t} = -\rho\mathcal{F}'(\phi_n).
\]

Hence, by setting \( \psi_{n+i} = \psi(x, (n \pm i)\Delta t) \) and \( i = 0, 1 \), we have

\[
\partial_t^2 \psi + (3/t)\partial_t \psi = -\rho\mathcal{F}'(\psi)
\]

(2.5)

formally (see [28] for justification). Thus, combining (2.1) with (2.5), we obtain

\[
\partial_t^2 \psi + (3/t)\partial_t \psi = -\rho\mathcal{F}'(\psi) + \tau \Delta \psi.
\]

(2.6)

Remark 2.2. Notably, two initial conditions are required to solve uniquely (2.6), which implies that it is necessary to have an initial data and another data updated by employing it. In this study, (2.2) will be applied the first few times to construct the initial data and get the same regularity in (2.2) (see

10
also Remarks 2.3 and 2.4 below). Actually, hyperbolic equations do not have smoothing effects in general. Hence, we note that the scheme (2.3)-(2.4) does not mean a regularization scheme.

2.2. Update of level set functions

Based on (2.2) and (2.6), we shall set up equations which (1.6) for the minimization problem of (1.13) satisfies. By [29, Appendix B], it holds that

$$d_T F(\phi) = A \chi_\phi \varepsilon(u) : \varepsilon(u) - \lambda,$$

where

$$A_{ijkl} = \frac{-3(1 - \nu)}{2(1 + \nu)(7 - 5\nu)} \left( \frac{1 - 14\nu + 15\nu^2}{(1 - 2\nu)^2} E(\delta_{ik}\delta_{j\ell} + \delta_{i\ell}\delta_{jk}) \right).$$

As in Remark 2.1, we replace $F'$ with $-d_T F$, which implies that (2.2) coincides with (1.8). Here we put $F = F$. Then we set $\phi \in L^\infty(0, +\infty; H^1_0(D))$ as a unique weak solution to

$$\begin{cases}
\partial_t^2 \phi + (3/t) \partial_t \phi - \tau \Delta \phi = \rho d_T F(\phi) \text{ in } D \times (0, +\infty), \\
\phi|_{\partial D} = 0, \quad \phi|_{t=0} = \phi_0, \quad \partial_t \phi|_{t=0} = \phi_1,
\end{cases} \quad (2.7)$$

where $\phi_0 \in H^2(D) \cap H^1_0(D)$ and $\phi_1 \in H^1_0(D)$.

**Remark 2.3.** (Well-posedness and boundary conditions). The characteristic function $\chi_{\phi} \in L^\infty(D; \{0, 1\})$ is replaced by an approximated Lipchitz continuous function in terms of numerical analysis (see Appendix A below). Thus standard general theories ensure well-posedness for (2.7) (see, e.g., [30] for details). Here, the homogeneous Dirichlet boundary condition for (2.7) is imposed, but only for the uniqueness of solutions. Thus, other boundary conditions can also be allowed.

**Remark 2.4.** Since $(3/t) \to 0$ as $t \to +\infty$, the damping term $(3/t) \partial_t \phi$ may be ignored in (2.7) for simplicity. Indeed, in order to construct $\phi_0 \in H^2(D) \cap H^1_0(D)$ and $\phi_1 \in H^1_0(D)$, let $\psi \in L^\infty(s, +\infty; H^1_0(D))$ be a unique weak solution to

$$\begin{cases}
\partial_t^2 \psi + (3/t) \partial_t \psi - \tau \Delta \psi = \rho d_T F(\psi) \text{ in } D \times (s, +\infty), \\
\psi|_{\partial D} = 0, \quad \psi|_{t=s} = \varphi|_{t=s}, \quad \partial_t \psi|_{t=s} = \partial_t \varphi|_{t=s}
\end{cases}$$

11
for some \( s \in (0, +\infty) \) and \( \Delta t > 0 \). Here, \( \varphi \in L^2((0, s]; H^1_0(D)) \) is a unique weak solution to
\[
\begin{cases}
\partial_t \varphi - \tau \Delta \varphi = \rho d_T F(\varphi) & \text{in } D \times (0, s], \\
\varphi|_{\partial D} = 0, \quad \varphi|_{t=0} = \varphi_0 \in L^\infty(D).
\end{cases}
\]
Since the reaction term is numerically treated as a Lipchitz continuous function, there exists \( T_{\text{max}} > 0 \) such that \( \varphi \in C(0, T_{\text{max}}; H^2(D) \cap H^1_0(D)) \). In particular, by (1.6), one can choose \( T_{\text{max}} \in (0, +\infty) \) as a large number, and so is \( s \in (0, +\infty) \); in other words, \((3/t)\partial_t \psi\) is small enough to be negligible.

Furthermore, in terms of the regularity of solutions, the control of geometric complexity will be expected, and hence, the optimized configuration \( \Omega_{\text{opt}} \in \mathcal{U}_{\text{ad}} \) will also be as smooth as that for the reported method [24].

3. Numerical algorithm

In this section, we shall describe a numerical algorithm to solve the optimization problem for (1.13) by updating the level set function (see Appendix A below for technical details).

**Step 1.** Set the fixed design domain \( D \subset \mathbb{R}^d \), boundary conditions for (1.12) and the initial level set function (\( \phi := \phi_0 \) in the code).

**Step 2.** Determine the state value \( u_\phi \in V \). To this end, discretizing \( D \subset \mathbb{R}^d \) with finite elements, we solve (1.12) using the finite element method.

**Step 3.** Compute the functionals \( F(\phi) \) and \( G(\phi) \) (\( \text{obj} \) and \( Gv \) in the code, respectively). Here, we note that \( G(\phi) \) is normalized in the code.

**Step 4.** Check for convergence. In the code, based on the gradient descent method, we define convergence conditions as follows:
\[
\|Lsfdiff\|_{L^\infty(D)} < \text{eps0pt}, \quad Gv \leq 0. \tag{3.1}
\]
Here, \( \text{eps0pt} \in \mathbb{R} \) is the criterion for convergence (see Step 7 below for \( Lsfdiff \)). If the conditions in (3.1) are all satisfied, then we terminate the optimization. Otherwise, we proceed to the next step.
Step 5. Compute the topological derivative $d_T F$ and the Lagrange multiplier $\lambda \geq 0$ ($Td((\cdot,\cdot)$ and $LagGV$ in the code, respectively). In particular, we set the following normalizer for dimensionless in the code:

$$AbsTd1 := \frac{\int_D |d_T F| \, dx}{|D|}.$$ 

On the other hand, as for the Lagrange multiplier $LagGV$, we employ augmented Lagrangian’s method as follows:

$$LagGV = LagGVp + LagGVD,$$

where $LagGVp$ is the previous version of $LagGV$ and $LagGVD$ is some normalized volume functional (see Appendix A below for details).

Step 6. Solve PDEs. In the code, let $Iter \in \mathbb{Z}$ be an iteration number. Choosing $StatIt \in \mathbb{Z}$ to which (2.7) applies, we solve the following either (i) or (ii) using the finite difference method discretized in the time direction:

(i) In case $Iter \leq StatIt$, for all $\tilde{\Phi} \in \tilde{V} := \{ \tilde{\Phi} \in H^1(D) : \tilde{\Phi} = 0$ on $\Gamma_D \}$,

$$0 = \int_D \left( \frac{\phi}{dt} \right) \tilde{\Phi}(x) \, dx + \int_D \tau \nabla (\phi) \cdot \nabla \tilde{\Phi}(x) \, dx - \int_D CdF \left( \frac{Td1(u, o\phi)}{AbsTd1} - LagGv \right) \tilde{\Phi}(x) \, dx - \int_D \left( \frac{o\phi}{dt} \right) \tilde{\Phi}(x) \, dx.$$

Here $dt := \Delta t > 0$, $\tau := \tau > 0$ and $CdF := \rho > 0$ are given parameters.

(ii) In case $Iter > StatIt$, for all $\tilde{\Phi} \in \tilde{V}$,

$$0 = \int_D \left( \frac{\phi}{dt} \right) \tilde{\Phi}(x) \, dx + \int_D \tau \nabla (\phi) \cdot \nabla \tilde{\Phi}(x) \, dx$$

$$- \int_D CdF \left( \frac{Td1(u, o\phi)}{AbsTd1} - LagGv \right) \tilde{\Phi}(x) \, dx - \int_D \left( \frac{2o\phi - o o\phi}{dt} \right) \tilde{\Phi}(x) \, dx.$$

Here we used the fact in Remark 2.4 for simplicity since the damping term eventually becomes negligibly small.

Step 7. Normalize $\phi$ as follows:

$$\phi = \begin{cases} 
\text{sgn}(\phi) & \text{if } |\phi| > 1, \\
\phi & \text{otherwise}
\end{cases}$$
and set \( LsfDiff := \text{phi} - \text{ophi} \). Return to Step 2 after setting the next initial level
set functions as \( ophi = \text{phi} \) and \( oophi = \text{ophi} \).

4. Main results

In this section, we shall describe numerical examples for the two-dimensional
case mainly and numerically show that the method based on (2.7) and (1.8)
converges to an optimized configuration faster than the method based on only
(1.8).

Let \( D \subset \mathbb{R}^2 \) be a rectangle and set Young’s modulus \( E \) > 0, Poisson’s ratio
\( \nu \) > 0, and the Lame coefficients \( \hat{\lambda} \) > 0 and \( \mu \) > 0 as follows:

\[
E = 2.1 \times 10^{11}, \quad \nu = 0.3, \quad \hat{\lambda} = \frac{E \nu}{(1 + \nu)(1 - 2\nu)}, \quad \mu = \frac{E}{2(1 + \nu)}.
\]

In order to solve (1.12) in Step 2, the elasticity tensor and traction vector replaced by \( D \) and \( g \) according to the code, respectively, are set as follows:

\[
D = \begin{pmatrix}
\hat{\lambda} + 2\mu & \hat{\lambda} & 0 \\
\hat{\lambda} & \hat{\lambda} + 2\mu & 0 \\
0 & 0 & \mu
\end{pmatrix}, \quad g = (0, -1.0 \times 10^3).
\]

Here, we note that the elasticity tensor is rewritten as the matrix in terms of
the finite element method. Moreover, we set the topological derivative \( d_T F \) as
follows:

\[
d_T F = A \chi_\phi \epsilon(u_\phi) \cdot \epsilon(u_\phi) - \lambda,
\]

where, \( \epsilon(u_\phi) = (\partial_{x_1} u_1, \partial_{x_2} u_2, \partial_{x_2} u_1 + \partial_{x_1} u_2), \ u_\phi = (u_1, u_2) \in V \) and

\[
A = \begin{pmatrix}
A_1 + 2A_2 & A_1 & 0 \\
A_1 & A_1 + 2A_2 & 0 \\
0 & 0 & A_2
\end{pmatrix}.
\]

Here, \( A_1 \) and \( A_2 \) are given by

\[
A_1 = -\frac{3(1 - \nu)(1 - 14\nu + 15\nu^2)}{2(1 + \nu)(7 - 5\nu)(1 - 2\nu)^2} E \quad \text{and} \quad A_2 = \frac{15(1 - \nu)}{2(1 + \nu)(7 - 5\nu)} E,
\]
respectively. We consider the three models (see Figure 1 below). For simplicity, the method with (1.8) and the method with (2.7) (i.e., nonlinear hyperbolic-parabolic equations) are described as (RD) and (NLHP), respectively.

In this paper, we define (NLHP) converging faster than (RD), if (NLHP) has fewer iteration numbers that satisfy all convergence conditions than (RD) (see Step 4 in §3); indeed, the convergence condition for the level set functions mentioned in the previous section is standard in the gradient descent method, and moreover, if \( \|\phi_{n+1} - \phi_n\|_{L^\infty(D)} < \varepsilon \) for \( \varepsilon > 0 \) small enough, then the configurations \( \Omega_{\phi_n} = \{x \in D: \chi_{\phi_n}(x) = 1\} \) and \( \Omega_{\phi_{n+1}} \) can be (almost) identified.

![Figure 1: Fixed design domain D and boundary conditions.](image)

**4.1. Cantilever model**

Based on Figure 1a, we consider the so-called cantilever model. The given parameters are the same as in the code (see Appendix A). In particular, we set \((\tau, G_{\text{max}}) = (5.0 \times 10^{-4}, 0.45)\), and the number of triangles \(n_t\) and maximum edge size \(h_{\text{max}}\) are set to \((n_t, h_{\text{max}}) = (38400, 0.0144)\). As for the convergence criterion, we choose \(\text{eps0pt} = 1.0 \times 10^{-3}\) in the code.

**Case (i) (Periodically perforated domain).** We first consider the case where the initial configuration is a periodically perforated domain. Then Figures 2 and 3 are obtained, and one can confirm that (NLHP) satisfies the convergence condition in Figure 3b faster than (RD). In particular, it is noteworthy that (NLHP) optimizes the topology in only 15 steps (see Figure 2h).

**Case (ii) (Whole domain).** We next consider the case where the initial configuration is the whole domain \(D \subset \mathbb{R}^2\). Then Figures 4 and 5 are obtained.
Figure 2: Configuration $\Omega_{\phi_n} \subset D$ for the case where the initial configuration is the periodically perforated domain. Figures (a)–(e) and (f)–(j) represent the results of (RD) and (NLHP), respectively. The symbol $#$ implies the final step.

Figure 3: Objective functional and convergence condition for §4.1-(i).
In this case, the difference in methods obviously arises; indeed, there is no considerable difference up to Step 50, but their topologies do not coincide at Step 150. Furthermore, at Step 250, (NLHP) is as close as possible to the optimal configuration. However, in (RD), even the topology is different from the optimal configuration.

![Figure 4: Configuration $\Omega_{\phi_n} \subset D$ for the case where the initial configuration is the whole domain. Figures (a)–(e) and (f)–(j) represent the results of (RD) and (NLHP), respectively. The symbol $\#$ implies the final step.](image)

![Figure 5: Objective functional and convergence condition for §4.1(ii).](image)

**Case (iii) (Upper domain).** We consider the case where the initial configuration is an upper domain. Then Figures 6 and 7 are obtained. In this case, we first note that the topology and the shape must be significantly modified. Large differences exist in convergence among the methods (see Figure 7). In particular, at Step 300, Figure 6i is considerably closer to the optimal configuration.
than Figure 6d.

Figure 6: Configuration $\Omega_{\phi_n} \subset D$ for the case where the initial configuration is the upper domain. Figures (a)–(e) and (f)–(j) represent the results of (RD) and (NLHP), respectively. The symbol # implies the final step.

Figure 7: Objective functional and convergence condition for §4.1-(iii).

Case (iv) (Three-dimensional domain). Let us finally consider the corresponding three-dimensional case. Here we set $(n_t, h_{\text{max}}) = (187280, 0.0422)$ and $(\tau, G_{\text{max}}) = (5.0 \times 10^{-4}, 0.3)$. Then Figures 8 and 9 are obtained. Obviously, Figure 9b shows that (NLHP) converges faster than (RD). In particular, at Step 300, (NLHP) is almost identical to the final configuration, and therefore, we see that the boundary structure in (NLHP) is moving faster than that in (RD).
Figure 8: Configuration $\Omega_{\phi_n} \subset D \subset \mathbb{R}^3$ for the case where the initial configuration is the whole domain. Figures (a)–(d) and (e)–(h) represent the results of (RD) and (NLHP), respectively. The symbol $\#$ implies the final step. Here the depth of $D$ is set to 0.2.

Figure 9: Objective functional and convergence condition for §4.1-(iv).
4.2. Bridge model

As another boundary condition, we next consider the so-called bridge model (see Figure 1b) and show numerically that the same assertion in the previous subsection is obtained. In this subsection, we set \((n_t, h_{\text{max}}) = (35000, 0.0141)\) and \((\tau, G_{\text{max}}) = (8.0 \times 10^{-5}, 0.35)\). Here and henceforth, the convergence criterion is set to \(\epsilon^{\text{opt}} = 1.0 \times 10^{-2}\) in terms of practicality.

**Case (i) (Periodically perforated domain).** As in § 4.1-(i), one takes the initial configuration as the periodically perforated domain. Then Figures 10 and 11 ensure the assertion in this study; indeed, at Step 180 (see Figures 10b and 10g), the topology of \(\Omega_{\phi_n} \subset D\) in (NLHP) can be optimized, and moreover, (NLHP) satisfies the convergence condition in at least half the number of iterations for (RD).

![Figure 10: Configuration \(\Omega_{\phi_n} \subset D\) for the case where the initial configuration is the periodically perforated domain. Figures (a)–(e) and (f)–(j) represent the results of (RD) and (NLHP), respectively. The symbol # implies the final step.](image)

**Case (ii) (Whole domain).** We next choose the whole domain as the initial configuration. By comparing Figure 12b with 12g, the topology is optimized at the same speed, and there seems to be no difference in the speed for convergence; however, from Figure 13, one can see that the boundary structure moves and converges faster in (NLHP) than in (RD). As in the previous case, it is noteworthy that more than twice the number of iterations can be improved.

**Case (iii) (Upper domain).** As for the case where the initial configuration
Figure 11: Objective functional and convergence condition for §4.2-(i).

Figure 12: Configuration $\Omega_{\phi_n} \subset D$ for the case where the initial configuration is the whole domain. Figures (a)–(e) and (f)–(j) represent the results of (RD) and (NLHP), respectively. The symbol # implies the final step.

Figure 13: Objective functional and convergence condition for §4.2-(ii).
is the upper domain, Figures 14 and 15 yield the assertion in this study; indeed, it is confirmed from Figures 14b and 14g that (NLHP) optimizes the topology faster than (RD). Moreover, we can also see that the boundary structure in (NLHP) is moving faster than that of (RD) in terms of satisfying the convergence condition more quickly.

Figure 14: Configuration $\Omega_{\phi_n} \subset D$ for the case where the initial configuration is the upper domain. Figures (a)–(e) and (f)–(j) represent the results of (RD) and (NLHP), respectively. The symbol # implies the final step.

Figure 15: Objective functional and convergence condition for §4.2-(iii).

Case (iv) (Three-dimensional domain). The same results as in the two-dimensional cases can also be obtained for the three-dimensional case. Actually, we readily deduce from Figure 16 that the boundary structure in (NLHP) is moving faster than that of (RD). Moreover, Figure 17 ensures the validity of the proposed method. Here we set $(n_t, h_{\text{max}}) = (176629, 0.0433)$ and $(\tau, G_{\text{max}}) = (1.0 \times 10^{-3}, 0.2)$. 

22
Figure 16: Configuration $\Omega_{\phi_n} \subset D \subset \mathbb{R}^3$ for the case where the initial configuration is the whole domain. Figures (a)–(d) and (e)–(h) represent the results of (RD) and (NLHP), respectively. The symbol "#" implies the final step. Here the depth of $D$ is set to 0.3.

Figure 17: Objective functional and convergence condition for §4.2(iv).
4.3. Radiator model

We finally consider the so-called \textit{radiator model} as a more complex geometry model (see, e.g., [5, Example 5.9] and [31]). Here we set $(n_t, h_{\text{max}}) = (25600, 0.0088)$ and $(\tau, G_{\text{max}}) = (2.0 \times 10^{-5}, 0.5)$. The setting for the fixed domain $D \subset \mathbb{R}^2$ and boundary conditions are as in Figure 1c.

\textbf{Case (i) (Periodically perforated domain).} Let us consider the case where the initial configuration is the periodically perforated domain. From Figures 18c and 18h, we infer that the topology of $\Omega_{\phi_n} \subset D$ in (NLHP) can be optimized faster than that of (RD), and then Figure 19 yields the assertion in this study. Here we switched to the reaction-diffusion in (NLHP) after 15 steps to avoid oscillation.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure18}
\caption{Configuration $\Omega_{\phi_n} \subset D$ for the case where the initial configuration is the periodically perforated domain. Figures (a)–(e) and (f)–(j) represent the results of (RD) and (NLHP), respectively. The symbol # implies the final step.}
\end{figure}

\textbf{Case (ii) (Whole domain).} As for the case where the initial configuration is the whole domain, we first note that optimal configurations in Figure 20 are not the same as that in Figure 20e; however, from Figure 21a, we see that the convergence values are the same, and (NLHP) is optimized to the same degree as (RD). On the other hand, by noting that objective functional $F(\phi_n)$ is oscillating in Figure 20a and switching to the reaction-diffusion in 50 steps, the convergence condition can be satisfied quickly, which implies that the use of a slightly nonlinear damped wave equation improves convergence.
Figure 19: Objective functional and convergence condition for 4.3(i).

Figure 20: Configuration $\Omega_{\phi_n} \subset D$ for the case where the initial configuration is the whole domain. Figures (a)–(e) and (f)–(j) represent the results of (RD) and (NLHP), respectively. The symbol # implies the final step.

Figure 21: Objective functional and convergence condition 4.3(ii).
Case (iii) (Upper domain). We take the initial configuration as the upper domain. Then Figures 22 and 23 are obtained. As in the previous case, since the objective functional \( F(\phi_n) \) in (NLHP) oscillates, we also switched to the reaction-diffusion at 50 steps. By comparing Figures 22c with 22h one can see that the topology in (NLHP) is optimized faster than that in (RD), and moreover, it is noteworthy that convergence is greatly improved.

![Figure 22](image)

**Figure 22:** Configuration \( \Omega_{\phi_n} \subset D \) for the case where the initial configuration is the upper domain. Figures (a)–(e) and (f)–(j) represent the results of (RD) and (NLHP), respectively. The symbol \# implies the final step.

![Figure 23](image)

**Figure 23:** Objective functional and convergence condition §4.3(iii).

Case (iv) (Three-dimensional domain). We finally show numerically that the assertion in this study can also be obtained for the corresponding three-dimensional case. Here we set \((n_H, h_{max}) = (227955, 0.0192) \) and \((\tau, G_{max}) = (1.0 \times 10^{-4}, 0.2) \). Then Figures 24 and 25 are obtained, and then we see that the
boundary structure in (NLHP) is moving faster than that in (RD). In particular, Figure 25b ensures the validity of the proposed method. This completes the confirmation for the assertion.

Figure 24: Configuration $\Omega_{\phi_n} \subset D \subset \mathbb{R}^3$ for the case where the initial configuration is the whole domain. Figures (a)–(d) and (e)–(h) represent the results of (RD) and (NLHP), respectively. The symbol $#$ implies the final step. Here the depth of $D$ is set to 0.2.

Figure 25: Objective functional and convergence condition for §4.3-(iv).

5. Conclusion

Based on Nesterov’s accelerated gradient descent method, this study devised a level set-based topology optimization, which employed reaction-diffusion and
nonlinear (damped) wave equations. Furthermore, it was applied to minimum mean compliance problems and yielded numerically faster convergence to an optimized configuration than a reported result [23]. Besides, we presented our FreeFEM++ code (for the cantilever model) in the appendix.

References

[1] G. Allaire, *Shape optimization by the homogenization method*, Vol. 146 of Applied Mathematical Sciences, Springer-Verlag, New York, 2002.

[2] F. Murat, L. Tartar, H-convergence, In: Topics in the mathematical modeling of composite materials, Progr. Nonlinear Differential Equations Appl., vol. 31, Birkhäuser Boston, Boston, MA, pp.21–43, 1997.

[3] M.P. Bendsøe, N. Kikuchi, Generating optimal topologies in structural design using a homogenization method, Comput. Methods. Appl. Mech. Eng. 71 (1988), 197–224.

[4] M.P. Bendsøe, Optimal shape design as a material distribution problem, Struct. Multidisc. Optim. 1 (1988), 193–202.

[5] G. Allaire, L. Cavallina, N. Miyake, T. Oka, T. Yachimura, The homogenization method for topology optimization of structures: old and new, Interdiscip. Inform. Sci. 25 (2019), 75–146.

[6] O. Sigmund, Morphology-based black and white filters for topology optimization, Struct. Multidisc. Optim. 22 (2007), 401–424.

[7] F. Wang, B.S. Lazarov, O. Sigmund, On projection methods, convergence and robust formulations in topology optimization, Struct. Multidisc. Optim. 43 (2011), 767–784.

[8] Z. Ullah, B. Ullah, W. Khan, S.Ul-Islam, Proportional topology optimisation with maximum entropy-based meshless method for minimum compliance and stress constrained problems, Eng. Comput. 38 (2022), 1–21.
[9] W. Khan, S. Ul-Islam, B. Ullah, Analysis of meshless weak and strong formulations for boundary value problems, Eng. Anal. Bound. Elem. 80 (2017), 1–17.

[10] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. 79 (1988), 12–49.

[11] G. Allaire, F. Jouve, A.M. Toader, A level-set method for shape optimization, C. R. Acad. Sci. Paris 334 (2002), 125–1130.

[12] G. Allaire, F. Jouve, A.M. Toader, Structural optimization using sensitivity analysis and a level-set method, J. Comp. Phys. 194 (2004), 363–393.

[13] M.Y. Wang, X. Wang, D. Guo, A level set method for structural topology optimization, Comput. Methods. Appl. Mech. Eng. 192 (2003), 227–246.

[14] B. Ullah, W. Khan, S. Ul-Islam, Z. Ullah, A coupled meshless element-free Galerkin and radial basis functions method for level set-based topology optimization, J. Braz. Soc. Mech. Sci. 44 (2022), 1–19.

[15] B. Ullah, S. Ul-Islam, Z. Ullah, W. Khan, A parametrized level set based topology optimization method for analysing thermal problems, Comput. Math. Appl. 99 (2022), 99–112.

[16] S. Ul-Islam, W. Khan, B. Ullah, Z. Ullah, The localized radial basis functions for parameterized level set based structural optimization, Eng. Anal. Bound. Elem. 113 (2020), 296–305.

[17] W. Khan, S. Ul-Islam, B. Ullah, Structural optimization based on meshless element free Galerkin and level set methods, Comput. Methods. Appl. Mech. Eng. 344 (2019), 144–163.

[18] B. Ullah, J. Trevelyan, S. Ul-Islam, A boundary element and level set based bi-directional evolutionary structural optimization with a volume constraint, Eng. Anal. Bound. Elem. 80 (2017), 152–161.
[19] H.A. Eschenauer, V.V Kobelev, A. Schumacher, Bubble method for topology and shape optimization of structures, Struct. Multidisc. Optim. 8 (1994), 42–51.

[20] G. Allaire, F. De Gournay, F. Jouve, A.M. Toader, Structural optimization using topological and shape sensitivity via a level set method, Control and Cybernetics 34 (2005), 59–80.

[21] S. Amstutz, H. Andrä, A new algorithm for topology optimization using a level-set method, J. Comput. Phys. 216(2) (2006), 573–588.

[22] A. Novotny, J. Sokolowski, Topological derivatives in shape optimization. Springer, Heidelberg, 2013.

[23] J.A. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science, Cambridge University Press (1999).

[24] 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. Eng. 199 (2010), 2876–2891.

[25] Y. Nesterov, A method of solving a convex programming problem with convergence rate $O(1/k^2)$, Sov. math. Dokl. 27 (1983), 372–376.

[26] F. Hecht, New development in FreeFem++, J. Numer. Math. 20 (2012), 251–266.

[27] G. Allaire, O, Pantz, Structural optimization with FreeFem++, Struct. Multidisc. Optim. 32 (2006),173–181.

[28] W. Su, S. Boyd, E.J. Candés, A differential equation for modeling Nesterov’s accelerated gradient method: theory and insights, J. Mach. Learn. Res. 17 (2016), 1–43.
[29] M. Otomori, T. Yamada, K. Izui, S. Nishiwaki, Matlab code for a level set-based topology optimization method using a reaction diffusion equation, Struct. Multidisc. Optim. 51 (2015), 1159–1172.

[30] T. Cazenave, A. Haraux, An introduction to semilinear evolution equations, Oxford Lecture Series in Mathematics and its Applications, Oxford University Press, New York, 1998.

[31] T. Yamada, Y. Noguchi, Topology optimization with a closed cavity exclusion constraint for additive manufacturing based on the fictitious physical model approach Addit. Manuf. 52 (2022), 102630.

Appendix A. Free FEM++ code

The code for the optimization problem of the cantilever model is as follows:

```cpp
// (c) 2021 Takayuki YAMADA
// All right reserved
// THE UNIVERSITY OF TOKYO
verbosity = 0; // Set message level

// parameters

// Solid Mechanics
real E; E = 210e9;
// Young's modulus
real nu; nu = 0.3;
// Poisson's ratio
real lambda; lambda = E*nu*(1+nu)/(1-2.*nu);
// Lame coefficient
real mu; mu = E/2./(1+nu);
// Lame coefficient
real A1; A1 = E*(-3.*(1-nu)/(2.*(1+nu)*(7-5.*nu)))*
((1-14.*nu+15.*nu^2)*1./(1-2.*nu)^2);
real A2; A2 = E*(15.*(1-nu)*1./(2.*(1+nu)*(7-5.*nu)));
real matd; matd = 1e-3;
real matw; matw = 0.8;

// Optimization
int MaxLoop; MaxLoop = 50000;
```
// Maximal iteration number
real epsOpt; epsOpt = 1.e-3;
// Criterion for convergence
int FlagOptMax; FlagOptMax = 10;
// period for convergence check

// Volume Constraint
real GvMax; GvMax = 0.45;
// upper limit of the volume constraint
int GvLoop; GvLoop = 15;
// extended
real LagGvA; LagGvA = 2.0;
// coefficient of the linear factor
real LagGvinit; LagGvinit = 1.0;
// initial value of lagrange multiplier
real LagGvD; real LagGvC; LagGvC = 1.0;
// coefficient for modification of the Lagrange multiplier
real LagGvMax; LagGvMax = 5.0;
// upper limit of the Lagrange multiplier
real LagGvMin; LagGvMin = 0.1;
// lower limit of the Lagrange multiplier

// Level Set Function
bool UseAcceration = true;
int StatIt; StatIt = 5;
// iteration number of starting the acceration
real L; L = 1.0;
// characteristic length
real tau; tau = 5.e-4;
// regularization parameter
real CdF; CdF = 1.2;
// normalization parameter for sensitivity
real dt; dt = 0.7;
// initial time increment

// geometry and Mesh

int [int] labs(4);
// labels on boundaries of square: (bottom, right, top, left)
int [int] MeshNum(2);
// number of mesh: (x coordinate, y coordinate)
real [int] Pos0(2), Pos1(2);
// Positions of diagonal points of square: (x coordinate, y coordinate)
int resMesh; resMesh = 4;
// mesh resolution

mesh Shc1;
// Mesh of the square 1
labs = [1,2,3,4];
MeshNum = [50,11]*resMesh;
Pos0 = [0.0,0.0];
Pos1 = [2.0,0.5-0.04];
Shc1 = square(MeshNum(0),MeshNum(1),
[Pos0(0)+(Pos1(0)-Pos0(0))*x,Pos0(1)+(Pos1(1)-Pos0(1))*y],
label=labs,flags=1,region=1);

mesh Shc2;
// Mesh of the square 2
labs = [5,6,7,8];
MeshNum = [50,2]*resMesh;
Pos0 = [0.0,0.5-0.04];
Pos1 = [2.0,0.5+0.04];
Shc2 = square(MeshNum(0),MeshNum(1),
[Pos0(0)+(Pos1(0)-Pos0(0))*x,Pos0(1)+(Pos1(1)-Pos0(1))*y],
label=labs,flags=1,region=1);

mesh Shc3;
// Mesh of the square 3
labs = [9,10,11,12];
MeshNum = [50,11]*resMesh;
Pos0 = [0.0,0.5+0.04];
Pos1 = [2.0,1.0];
Shc3 = square(MeshNum(0),MeshNum(1),
[Pos0(0)+(Pos1(0)-Pos0(0))*x,Pos0(1)+(Pos1(1)-Pos0(1))*y],
label=labs,flags=1,region=1);

// combine the meshes
mesh Sh=Shc1+Shc2+Shc3;

// define Boundary Label
int [int] wall(3); wall = [4,8,12];
// fixed displacement
int [int] traction(1); traction = [6];
// traction
int [int] phiOne(1); phiOne = [6];
// phi=1.0

// Define functional spaces
// functional space for governing and joint equation
fespace VhS2(Sh,P2), VhS1(Sh,P1), VhS0(Sh,P0);
fespace VhV2(Sh,[P2,P2]), VhV1(Sh,[P1,P1]);

// Definition of variables
VhS1 ophi,oophi,LsfDiff;
// ophi: previous level set function

// Define Characteristic function

macro chiX (phi) ((0.5+phi/matw*(15./16−phi^2/matw^2*(5./8−3./16*phi^2/matw^2)))*
( phi>==matw)*(phi<matw)+1.*(phi>matw))//
macro chiP (phi) ( max((1.−matd)*chiX(phi)+matd,matd) ) //
macro chiV (phi) ( (0.5+phi*(15./16−phi^2*(5./8−3./16*phi^2)))*
( phi>==1.0)*(phi<=1.0)+1.*(phi>1.0))//

// Define Governing Equation

macro u [u1,u2]
// displacement vector
macro tu [tu1,tu2]
// test function for displacement
macro e(u) [dx(u[0]),dy(u[1]),(dx(u[1])+dy(u[0]))] // strain tensor
macro D [[[2.*mu+lambda,lambda,0],[lambda,2.*mu+lambda,0],[0,0,mu]] // elastic tensor
macro A [[[2.*A2+A1,1,0],[A1,2.*A2+A1,0],[0,0,A2]] // for topoligical derivative
macro g [0.0,−1.0e3]
// traction vector
macro Td1 (u,ophi) ((A*e(u))^*e(u)*chiP(ophi))
// Topological derivative 1

// define displacement
VhV1 u,tu;

// governing equation
problem gov(u,tu) 
  =int1d(Sh,traction)(g'*tu) 
  −int2d(Sh)((D*e(u))^*e(tu)*chiP(ophi)) 
  + on(wall,u1=0,u2=0);
// Define PDEs and Initialize the LSF

VhS1 phi, Vphi;
// level set function and its test function
real TotTime;
// time
real AbsTd1;
// absolute value of Td1
real LagGv;
// Lagrange multiplier for volume constraint
real LagGvp;
// Previous Lagrange multiplier

// Accelerated Reaction Diffusion Equation
problem arde(phi, Vphi)
= int2d(Sh) ( (phi*Vphi/dt)+(dx(phi)*dx(Vphi)+dy(phi)*dy(Vphi))*tau*L^2)
- int2d(Sh) ( (CdF*(Td1(u, ophi)/AbsTd1-LagGv)+((2.*ophi-oophi)/dt))*Vphi)
+ on(phiOne, phi = 1);

// Reaction Diffusion Equation
problem rde( phi, Vphi)
= int2d(Sh) ( (phi*Vphi/dt)+(dx(phi)*dx(Vphi)+dy(phi)*dy(Vphi))*tau*L^2)
- int2d(Sh) ( (CdF*(Td1(u, ophi)/AbsTd1-LagGv)+ophi/dt)*Vphi)
+ on(phiOne, phi = 1);

// Initialize Level Sets
phi = 4. * ( max( cos(10*x*pi), cos(10*y*pi)) - 0.5);
// case (i) of initialize level set function
//phi = 1.0/2.0*(((y-0.3)>0) - 0.5);
// case (ii) of initialize level set function
//phi = 2.0*(((y-0.4)>0) - 0.5);
// case (iii) of initialize level set function
ophi = phi*(phi<1.0)*(phi>=1.0)+1.*(phi>1.0)-1.*(phi<1.0);
// ophi is mapped to satisfy the upper and lower constraints.

// Define objective functions and constraint functions

real obj;
// objective function
real objPre;
// Previous objective function
real volFDD;
volFDD = int2d(Sh)(1.);
// volume of the fixed design domain
real vol;
// normalized volume
real volInit;
// initial normalized volume
real Gv;
// constraint functional for volume constraint
real GvEx;
// extended constraint functional for volume constraint
real GvMaxEx;
// extended maximal constraint functional for volume constraint

// compute initial volume
volInit = int2d(Sh)(chiV(ophi));

---
// Start Optimization Loop
---

int FlagOpt; FlagOpt = 0;
// flag for check convergence
for (int iter=0; iter<MaxLoop; iter++) {

// Solve Governing Equations (Step 2)

// Compute Objective and Constraint functions (Step 3)

// compute objective function
objPre = obj;
obj = int1d(Sh, traction)(g’*u);

// compute volume
vol = int2d(Sh)(chiV(ophi));

// compute extended upper limit
if (iter<GvLoop) {
    GvMaxEx = volInit/volFDD-((volInit/volFDD-GvMax)/GvLoop)*iter;
} else {
    GvMaxEx = GvMax;
}

GvEx = vol/volFDD-GvMaxEx;
// extended volume constraint functional
Gv = vol/volFDD-GvMax;
// volume constraint functional
// Plot

{// start plot
    plot(ophi,cmm="ophi", fill=1,
        wait=false, boundary=false, WindowIndex=0);
}// end plot

// Check Convergence (Step 4)

if (LsDiff[].max<=epsOpt && iter>GLoop && Gv<epsOpt) {
    FlagOpt++;
    if(FlagOpt>FlagOptMax) { break; }
} else{ FlagOpt = 0; }

// Compute Lagrange Multipliers and several normalizers (Step 5)

// Compute Normalizers
AbsTd1 = int2d(Sh)(abs(Td1(u, ophi)))/volFDD;

// Compute Lagrange Multiplier for volume constraint
if(iter==0){
    LagGv = LagGvinit;
    LagGvp = LagGv;
}

// compute update value for Lagrange multiplier
if(GvEx<0){
    LagGvD = -LagGvC*abs((vol/volFDD-GvMaxEx)/GvMaxEx);
} else{
    LagGvD = LagGvC*abs((vol/volFDD-GvMaxEx)/GvMaxEx);
}

// update Lagrange multiplier
LagGv = LagGvp+LagGvD;

// set upper and lower limmit of the Lagrange multiplier
LagGv = max(min(LagGvMax,LagGv),LagGvMin);

// save update Lagrange multiplier
LagGvp=LagGv;

// set extended Lagrange multiplier using the linear modification
LagGv=LagGvp*(1.0+LagGvA*((vol/volFDD-GvMaxEx)/GvMaxEx));
/ Solve Level set function (Step 6)
if (iter <= StatIt) {rde;}
else {
  if (UseAcceration){
    arde;
  } else {
    rde;
  }
}

// map LSF to "ophi" (Step 7)
ophi = ophi;
ophi = phi*(phi<=1.0)*(phi>=-1.0)+1.0*(phi>1.0)-1.0*(phi<-1.0);
LsfDiff = abs(ophi-oophi);
} // End optimization loop