Shape and topology optimization for elliptic boundary value problems using a piecewise constant level set method

Shengfeng Zhu\textsuperscript{a,}\textsuperscript{*}, Qingbiao Wu\textsuperscript{a}, Chunxiao Liu\textsuperscript{b}

\textsuperscript{a}Department of Mathematics, Zhejiang University, Hangzhou 310027, Zhejiang, PR China
\textsuperscript{b}Center of Mathematical Sciences, Zhejiang University, Hangzhou 310027, Zhejiang, PR China

Abstract

The aim of this paper is to propose a variational piecewise constant level set method for solving elliptic shape and topology optimization problems. The original model is approximated by a two-phase optimal shape design problem by the ersatz material approach. Under the piecewise constant level set framework, we first reformulate the two-phase design problem to be a new constrained optimization problem with respect to the piecewise constant level set function. Then we solve it by the projection Lagrangian method. A gradient-type iterative algorithm is presented. Comparisons between our numerical results and those obtained by level set approaches show the effectiveness, accuracy and efficiency of our algorithm.

Key words: Shape optimization; Topology optimization; Piecewise constant level set method; Projection Lagrangian method

1 Introduction

With wide applications in engineering, such as structure optimization [6] and control of fluid flows [19], shape and topology optimization is a very important and challenging research area which has received considerable attention by scientists and engineers. The

\* Corresponding author.
Email addresses: shengfengzhu@zju.edu.cn (Shengfeng Zhu), qbwu@zju.edu.cn (Qingbiao Wu), xxliu198431@126.com (Chunxiao Liu).

Preprint submitted to Applied Numerical Mathematics 13 December 2010
typical problem is to find the shape which is optimal in that it minimizes or maximizes a given objective function under certain PDE constraints. The essential difficulty for solving such problems is that the topology of the optimal design is unknown a priori.

In classical shape sensitivity analysis for shape optimization [11,13,27], the shape derivative measuring the sensitivity of boundary perturbations is derived to obtain the shape gradient of the objective functional. Then gradient-type iterative algorithms are usually developed. But such methods have two main drawbacks: the tendency to fall into local minimizers and too much computational effort due to remeshing. The homogenization method [1,5] can overcome the two drawbacks, but it is mainly restricted to linear structural optimization and gives optimal shapes that are composite. Penalization methods, such as the solid isotropic material with penalization (SIMP) method [4], are needed to project the composite shape to a classical two-phase (black and white) design.

Recently, the combination of level set methods with shape sensitivity analysis has become a popular tool for solving a variety of shape optimization and reconstruction problems [3,7,23,30]. The level set method was first devised by Osher and Sethian in [24] for interface tracing and shape evolution. In this method, the interface is represented implicitly by the zero level set of a Lipschitz continuous level set function. For more about level set methods, we refer to see the monographs [22,25] and surveys [7,9,28]. When applied to shape reconstruction and optimal design, level set approaches can handle certain types of shape and topological changes in an automatic way. However, as pointed out in [3,8,12,14,31], the conventional shape gradient based level set algorithm cannot create small holes during the evolution, which may lead to solutions with fewer holes than the optimal geometry. Therefore, the initial guess generally has to contain many holes. To remedy this flaw of level set algorithms, the so-called topology derivatives [26], which measure the influence of creating small holes centered at a point in the domain, were suggested to combine with shape derivatives for inverse obstacle problems [8], structural optimization [2] and shape design [12,14].

More recently, as variants of the traditional level set method, piecewise constant level set methods (PCLSM) were proposed by Tai et al. for image segmentation [15,17] and elliptic coefficient inverse problems [29]. One has to employ \( N \) level set functions to represent up to \( 2^N \) subregions in the multiple level set method [28], while PCLSM can identify an arbitrary number of subregions using one discontinuous piecewise constant level set function (PCLSF). One advantage of the PCLSM is that it can create small holes without topological derivatives during the evolution. Iterative algorithms having little dependence on the initial guess can be designed. Another merit is that the periodic re-initialization process in level set approaches is eliminated. Generally, using PCLSM requires to add a piecewise constant constraint to the original problem. The augmented Lagrangian method (cf. [21]) was employed to deal with the piecewise constant constraint and the volume constraint in using the PCLSM to solve structural topology optimization problems [18,32]. The two-phase and multi-phase cases of PCLSM are respectively used in [32] and [18]. In [34], the augmented Lagrangian method and Lagrange multiplier approach were combined for the two constraints to devise the effective algorithms for
optimal shape design of a vibrating membrane.

In this paper, we devise a projection Lagrangian algorithm of Uzawa type for solving shape optimization problems under the PCLSM framework. Our approach can be used to solve different models in shape and topology optimization. The original PDE was approximated by a two-phase problem defined on the fixed domain. Then the new two-phase optimal design model was formulated into an optimization problem with respect to one level set function under a piecewise constant constraint. We use a similar projection Lagrangian method as in [16] to construct an iterative scheme for the Lagrange multiplier. A gradient-type algorithm is presented. Furthermore, we use an acceleration technique to speed up convergence. With no need to perform the complicated shape and topological analysis and implement re-initialization, our algorithm is shown to be more efficient than level set algorithms in [12].

The paper is structured as follows. In Section 2, we present the model problem with an elliptic state constraint. Using the ersatz material method, we approximate the original model by a two-phase optimal shape design problem. In Section 3, we first present piecewise constant level set formulations and then apply the PCLSM to the two-phase problem. The resulting variational problem is solved by the projection Lagrangian method. A gradient-type algorithm is given. Then in the next section, we discuss some numerical implementation issues and devise an acceleration technique. Numerical results are presented in Section 5. We draw conclusions in the last section.

2 Elliptic shape optimization problem and its two-phase approximation

We consider the model problem taken from [12]. Let $U$ and $V$ be two bounded open subsets of $\mathbb{R}^2$ satisfying $V \subset\subset U$. For any open set $\omega \subset \mathbb{R}^2$, we denote by $\#\bar{\omega}$ the number of connected components of $\bar{\omega}$ and consider the set of admissible domains

$$O_k = \{\Omega = U \setminus \bar{\omega} ; \ \omega \subset V, \ \#\bar{\omega} \leq k\}.$$  

For any $\Omega \in O_k$ with some positive integer $k$, the boundary of $\Omega$ is split into $\partial\Omega = \Gamma_D \cup \Gamma_N$ with $\Gamma_D = \partial U$ and $\Gamma_N = \partial \omega$. The boundaries $\Gamma_D$ and $\Gamma_N$ receive homogeneous Dirichlet boundary conditions and Neumann conditions, respectively. We illustrate the details in Fig. 1, where $\omega$ may not be connected like that.

Consider the following elliptic boundary value problem

$$\left\{ \begin{array}{ll}
-\Delta u + u &= f \quad \text{in } \Omega, \\
uu &= 0 \quad \text{on } \Gamma_D, \\
\partial\nu u &= 0 \quad \text{on } \Gamma_N,
\end{array} \right. \quad (1)$$

where $f \in C^\infty(\bar{U})$ and $\nu$ is the outward unit normal to $\Gamma_N$. Define the following energy
The characteristic functional:

\[ E(\Omega, u) := \frac{1}{2} \int_\Omega (|\nabla u|^2 + u^2) \, dx - \int_\Omega f u \, dx \]

\[ = -\frac{1}{2} \int_\Omega (|\nabla u|^2 + u^2) \, dx = -\frac{1}{2} \int_\Omega f u \, dx. \]  

The objective function \( E(\Omega, u) \) is something like the compliance over the domain \( \Omega \). In mechanics the compliance is associated with the total work of the body force \( f \). We introduce the regularized functional

\[ J(\Omega) = E(\Omega, u) + \lambda A(\Omega) - \mu P_c(\Omega)^2. \]  

The regularization terms \( A(\Omega) \) and \( P_c(\Omega) \) are defined as

\[ A(\Omega) = |\Omega|, \]  

\[ P_c(\Omega) = \max(0, |\partial \Omega| - c), \]  

where the constants \( \lambda, \mu > 0 \) are weights of the two regularization terms. Here, \(|\Omega|\) and \(|\partial \Omega|\) denote the area of \( \Omega \) and the length of \( \partial \Omega \), respectively. The constant \( c > |\partial U| \) corresponds to a shifted perimeter functional.

We shall solve the following maximization problem

\[ \max_{\Omega \in \mathcal{O}_h} J(\Omega). \]  

The existence of an optimum domain for (6) has already been given by Fulmanski et al. [12]. This original model in [12] was used as an academic shape optimization problem to demonstrate that the shape-topological gradient based level set method enjoying a hole nucleation mechanism is more likely to escape local extrema than the traditional shape gradient based level set method for solving this model. Here we show that the variational PCLSM is also effective for solving this type of problems.

Before applying the PCLSM to the shape optimization problem (6), we first reformulate the original model to be a two-phase problem on the fixed domain \( U \). By the ersatz...
material method \cite{3,2}, which amounts to fill the holes by a weak phase, we introduce a two-density function

\[ \rho(x) = \begin{cases} \epsilon, & \text{if } x \in \omega, \\ 1, & \text{if } x \in \Omega, \end{cases} \]  

(7)

where \( \epsilon > 0 \) is a small constant. Then the original state equation (1) can now be approximated by the following two-phase elliptic problem

\[ \begin{cases} -\nabla \cdot (\rho(x) \nabla u_\epsilon) + \rho(x) u_\epsilon = \chi_\Omega f & \text{in } U, \\ u_\epsilon = 0 & \text{on } \Gamma_D, \end{cases} \]  

(8)

where \( \chi_\Omega \) denotes the characteristic function of \( \Omega \). Correspondingly, (6) turns to

\[ \max_{\Omega \in \Omega_k} J_{\epsilon}(\Omega), \quad J_{\epsilon}(\Omega) = E(\Omega, u_\epsilon) + \lambda A(\Omega) - \mu P_c(\Omega)^2. \]  

(9)

The original shape optimization problem (6) can be seen as the limit of the two-phase problem (9), where the weak phase degenerates \( (\epsilon \to 0) \) so as to become holes.

3 Variational PCLSM for two-phase shape optimization

In this section, we first present the basic formulations of the PCLSM. Then we apply the variational PCLSM to the two-phase model and obtain a smooth constrained optimization problem with respect to the PCLSF. Finally, based on the Lagrange multiplier approach, we develop a projection Lagrangian method of Uzawa type to solve the obtained problem.

3.1 Piecewise constant level set formulations

Let \( \tilde{\Omega} \) be an open bounded domain in \( \mathbb{R}^2 \). Assume that it is partitioned into \( N \) subregions \( \{\Omega_i\}_{i=1}^N \) such that

\[ \tilde{\Omega} = \bigcup_{i=1}^N (\Omega_i \cup \partial \Omega_i), \]  

(10)

where \( N \) is known a priori and \( \partial \Omega_i \) denotes the boundary of \( \Omega_i \). In order to identify the subregions, we define a \( N \)-phase function \( \phi : \tilde{\Omega} \mapsto \mathbb{R} \) satisfying

\[ \phi = i \quad \text{in } \Omega_i, \quad i = 1, 2, \cdots, N, \]  

(11)

which implies that

\[ H(\phi) \equiv \prod_{i=1}^N (\phi - i) = 0 \quad \text{in } \Omega. \]  

(12)
The constraint (12) is used to guarantee that there is no vacuum and overlap between different subregions. Then the characteristic functions of the subdomains are represented as

\[ \chi_i = \frac{1}{\alpha_i} \prod_{j=1, j \neq i}^{N} (\phi - j) \quad \text{with} \quad \alpha_i = \prod_{k=1, k \neq i}^{N} (i - k). \quad (13) \]

By the properties of the characteristic functions, we can calculate the area inside \( \Omega_i \) and the length of the boundary of \( \Omega_i \) as

\[ |\Omega_i| = \int_{\Omega} \chi_i \, dx \quad \text{and} \quad |\partial \Omega_i| = \int_{\Omega} |\nabla \chi_i| \, dx. \quad (14) \]

In numerical implementations, we use the approximation

\[ \int_{\Omega} |\nabla \chi_i| \, dx \approx \int_{\Omega} \sqrt{|\nabla \chi_i|^2 + \varepsilon_1} \, dx \quad (15) \]

for a small positive number \( \varepsilon_1 \) and approximate the derivatives by finite differences.

Any function \( \rho(x) \) with piecewise constant values \( \rho = \rho_i \) in \( \Omega_i \) can be expressed as

\[ \rho(x) = \sum_{i=1}^{N} \rho_i \chi_i(\phi). \quad (16) \]

### 3.2 Variational PCLSM for the model problem

Now we can apply the PCLSM formulation corresponding to the special case \( N = 2 \) to our two-phase model problem. Then we have \( H(\phi) = (\phi - 1)(\phi - 2) \). Let \( \Omega = U \), \( \Omega_1 = \omega \) and \( \Omega_2 = \Omega \). The density \( \rho \) is represented as

\[ \rho(\phi) = \rho_1 \chi_1 + \rho_2 \chi_2 = (1 - \epsilon)\phi + 2\epsilon - 1. \quad (17) \]

Then the state problem (8) and the cost functional of (9) can be rewritten as

\[
\begin{cases}
- \nabla \cdot (\rho(\phi) \nabla u_\epsilon) + \rho(\phi) u_\epsilon = (\phi - 1) f & \text{in } U \\
u_\epsilon = 0 & \text{on } \partial U
\end{cases}
\]  

and

\[ J_\epsilon(\phi) = E(\phi) + \lambda A(\phi) - \mu P_\epsilon(\phi)^2, \quad (19) \]
where

\[ E(\phi) = -\frac{1}{2} \int_U f u_\epsilon \chi_{\Omega} \, dx = -\frac{1}{2} \int_U f u_\epsilon (\phi - 1) \, dx, \]  
\[ A(\phi) = \int_U \chi_{\Omega} \, dx = \int_U (\phi - 1) \, dx, \]  
\[ P_c(\phi) = \max(0, |\partial \Omega| - c) = \max \left(0, |\partial U| + \int_U |\nabla \phi| \, dx - c\right). \]  

We have embedded the PCLSF into the cost functional in a variational way similarly as the variational level set method [33]. Therefore, we refer this approach as the variational PCLSM. The sophisticated computation of shape or topological derivatives can be avoided in this way. Now the optimization problem (9) can be reformulated as

\[ \max_{\phi} J_\epsilon(\phi) \text{ subject to } H(\phi) = 0, \]  

We use the Lagrange multiplier method to solve the constrained problem (23). The Lagrangian functional of (23) is defined by

\[ L(\phi, \tilde{\lambda}) = J_\epsilon(\phi) + \int_U \tilde{\lambda} H(\phi) \, dx, \]  

where the Lagrange multiplier \( \tilde{\lambda} \in L^\infty(U) \). We actually maximize \( L \) with respect to \( \phi \) and minimize \( L \) with respect to \( \tilde{\lambda} \). At a saddle point of the smooth functional (24), we have the first-order necessary optimality conditions

\[ 0 = \frac{\partial L}{\partial \phi} = J_\epsilon'(\phi) + \tilde{\lambda} H'(\phi) = E'(\phi) + \lambda A'(\phi) - 2\mu P_c'(\phi) + \tilde{\lambda}(2\phi - 3), \]  
\[ 0 = \frac{\partial L}{\partial \tilde{\lambda}} = H(\phi), \]  

where \( E'(\phi), A'(\phi), P_c'(\phi) \) and \( H'(\phi) \) are the Fréchet derivatives of \( E(\phi), A(\phi), P_c(\phi) \) and \( H(\phi) \), respectively.

Let us first derive \( E'(\phi) \) in the following. The weak formulation of (18) reads: Find \( u_\epsilon \in H^1_0(U) \) such that

\[ \int_U \rho(\phi) \nabla u_\epsilon \cdot \nabla v \, dx + \int_U \rho(\phi) u_\epsilon v \, dx = \int_U (\phi - 1) f v \, dx \quad \forall v \in H^1_0(U). \]  

Differentiating both sides of (27) with respect to \( \phi \) in a direction \( \psi \in C_0^\infty(U) \) gives

\[ \int_U \rho'(\phi) \psi(\nabla u_\epsilon \cdot \nabla v + u_\epsilon v) \, dx + \int_U \rho \left[ \nabla (u_\epsilon'(\phi) \psi) \cdot \nabla v + u_\epsilon'(\phi) \psi v \right] \, dx = \int_U \psi f v \, dx. \]  

Setting \( v = u_\epsilon \) yields

\[ \int_U \rho'(\phi) \psi(|\nabla u_\epsilon|^2 + u_\epsilon^2) \, dx + \int_U \rho \left[ \nabla (u_\epsilon'(\phi) \psi) \cdot \nabla u_\epsilon + u_\epsilon'(\phi) \psi u_\epsilon \right] \, dx = \int_U \psi f u_\epsilon \, dx. \]
On the other hand, the Gâteaux derivative of $E(\phi)$ in the direction $\psi$ reads

$$E'(\phi)\psi = -\frac{1}{2} \int_U fu_\epsilon \psi \ dx - \frac{1}{2} \int_U f(\phi - 1)u'_\epsilon(\phi)\psi \ dx$$

$$= -\frac{1}{2} \int_U fu_\epsilon \psi \ dx - \frac{1}{2} \int_U [\rho \nabla u_\epsilon \cdot \nabla (u'_\epsilon(\phi)\psi) + \rho u_\epsilon u'_\epsilon(\phi)\psi] \ dx,$$

where we have used (27) with $v = u'_\epsilon(\phi)\psi$. Then using (28), we obtain further

$$E'(\phi)\psi = -\frac{1}{2} \int_U fu_\epsilon \psi \ dx - \frac{1}{2} \int_U \left[ 2fu_\epsilon \psi - \rho'(\phi)\psi(|\nabla u_\epsilon|^2 + u^2_\epsilon) \right] \ dx,$$

which implies that

$$E'(\phi) = -fu_\epsilon + \frac{1}{2}\rho'(\phi)(|\nabla u_\epsilon|^2 + u^2_\epsilon) = -fu_\epsilon + \frac{1}{2}(1 - \epsilon)(|\nabla u_\epsilon|^2 + u^2_\epsilon).$$

Besides, we can obtain easily the following derivatives:

$$A'(\phi) = 1,$$

$$P'_c(\phi) = \chi_{\{|\partial\Omega|=c\}} \max \left\{ 0, -\nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right) \right\} - \chi_{\{|\partial\Omega|>c\}} \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right),$$

$$\frac{\partial P_c(\phi)}{\partial \phi}^2 = 2P_c(\phi)P'_c(\phi) = -2P_c(\phi)\nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right).$$

Then, substituting (30), (31), (33) and (22) into (25), we get

$$\frac{\partial L}{\partial \phi} = -fu_\epsilon + \frac{1}{2}(1 - \epsilon)(|\nabla u_\epsilon|^2 + u^2_\epsilon) + \lambda$$

$$+ 2\mu \max \left( 0, |\partial U| + \int_\Omega |\nabla \phi| \ dx - c \right) \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right) + \tilde{\lambda}(2\phi - 3),$$

In the following subsection, we will present an algorithm for solving (25)-(26).

### 3.3 Projection Lagrangian algorithm

The projection Lagrangian algorithm was applied to image segmentation under a framework of the binary level set method in [16]. Based on the Lagrange multiplier approach for constrained optimization, the algorithm is essentially a projection Lagrangian algorithm of Uzawa type [10]. Here we employ it to solve the problem (23) under the piecewise constant level set frame.

To find a saddle point of $L(\phi, \tilde{\lambda})$, we apply an iterative algorithm. Starting from initial guesses $\phi^0$ and $\tilde{\lambda}^0$ respectively for $\phi$ and $\tilde{\lambda}$, we get iteratively better approximations $\phi^n$. 
and $\tilde{\lambda}^n$ with $n = 1, 2, 3, \cdots$. The iterations will stop when the changes of the two variables approach to zero. The maximization of $L(\phi, \tilde{\lambda})$ with respect to $\phi$ is realized by solving the time dependent PDE

$$
\begin{cases}
\frac{\partial \phi}{\partial t} = \frac{\partial L}{\partial \phi} & \text{in } U \times (0, \infty) \\
\phi(x, 0) = \phi^0(x) & \text{in } U
\end{cases}
$$

(35)

to the steady state, where $\phi^0(x) \in L^\infty(U)$ is the initial guess. When $\partial \phi/\partial t = 0$, we get $\partial L/\partial \phi = 0$. To solve (35) numerically, we use the explicit Euler scheme

$$
\phi^{n+1} = \phi^n + \Delta t^n \frac{\partial L}{\partial \phi} (\phi^n, \tilde{\lambda}^n),
$$

(36)

where $\Delta t^n$ is the $n$-th time step and $\partial L/\partial \phi$ is given in (34). Considering the Courant-Friedrichs-Lewy stability condition for an explicit scheme, we choose

$$
\Delta t^n = \theta h \max_{x \in U} \left| \frac{\partial L}{\partial \phi} (\phi^n(x), \tilde{\lambda}^n) \right|,
$$

where $h$ is the uniform mesh size and $\theta > 0$ is a constant to be chosen by experience.

The updating of $\tilde{\lambda}$ is done by a projection Lagrangian method [10,16]. Multiplying by $2\phi - 3$ on both sides of (25) yields

$$
(2\phi - 3)J'_\epsilon(\phi) + \tilde{\lambda}(2\phi - 3)^2 = 0.
$$

(37)

Then using (26), we obtain

$$
\tilde{\lambda} = (3 - 2\phi)J'_\epsilon(\phi).
$$

(38)

The explicit expression (38) can be used to update $\tilde{\lambda}$:

$$
\tilde{\lambda}^{n+1} = (3 - 2\phi^{n+1})J'_\epsilon(\phi^{n+1}).
$$

(39)

Now the updating formulations (36) and (39) can be combined to give an iterative algorithm.

**Algorithm.** Projection Lagrangian algorithm

Choose $\epsilon$ and initialize $\phi^0$, $\tilde{\lambda}^0$. For $n = 0, 1, 2, \cdots$,

- Use (17) to compute $\rho^n = (1 - \epsilon)\phi^n + 2\epsilon - 1$.
- Solve (18) using linear finite elements with $\phi$ replaced by $\phi^n$.
- Update the level set function $\phi$ by (36).
- Update the Lagrange multiplier $\tilde{\lambda}$ by (39).
- If not converged, set $n = n + 1$ and repeat.
4 Implementation issues

For numerical implementation, we first give the finite difference discretization of some derivatives and then present an acceleration technique for speeding up our algorithm.

Denote by $\phi_{i,j}$ the approximation of $\phi$ at the grid point $(x_i, y_j)$. Let us introduce some notations concerning the finite differences. We define

$$
\delta^+_x \phi_{i,j} = \frac{\phi_{i+1,j} - \phi_{i,j}}{h_x}, \quad \delta^+_y \phi_{i,j} = \frac{\phi_{i,j+1} - \phi_{i,j}}{h_y}, \\
\delta^-_x \phi_{i,j} = \frac{\phi_{i,j} - \phi_{i-1,j}}{h_x}, \quad \delta^-_y \phi_{i,j} = \frac{\phi_{i,j} - \phi_{i,j-1}}{h_y}, \\
\delta^0_x \phi_{i,j} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h_x}, \quad \delta^0_y \phi_{i,j} = \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2h_y},
$$

(40)

respectively to be forward, backward and centered finite differences. Here, $h_x$ and $h_y$ are uniform steps in the $x$ and $y$ directions, respectively.

For the regularization term, we use the smooth approximation

$$
\int_U |\nabla \phi| \, dx \approx \int_U \sqrt{\phi_x^2 + \phi_y^2 + \varepsilon_1} \, dx,
$$

(41)

where $\varepsilon_1$ is defined as in (15). The partial derivatives $\phi_x$ and $\phi_y$ at $(x_i, y_j)$ are discretized using central differences, i.e.,

$$(\phi_x)_{i,j} \approx \delta^0_x \phi_{i,j} \quad \text{and} \quad (\phi_y)_{i,j} \approx \delta^0_y \phi_{i,j}.$$

For the curvature term, we have

$$
\nabla \cdot \left( \left( \frac{\nabla \phi}{|\nabla \phi|} \right)_{i,j} \right) = \left( \frac{\phi_x}{|\nabla \phi|} \right)_{x, i,j} + \left( \frac{\phi_y}{|\nabla \phi|} \right)_{y, i,j}
$$

$$
= \frac{1}{h_x} \left[ \left( \frac{\phi_x}{|\nabla \phi|} \right)_{i+1/2,j} - \left( \frac{\phi_x}{|\nabla \phi|} \right)_{i-1/2,j} \right] + \frac{1}{h_y} \left[ \left( \frac{\phi_y}{|\nabla \phi|} \right)_{i,j+1/2} - \left( \frac{\phi_y}{|\nabla \phi|} \right)_{i,j-1/2} \right],
$$

where

$$
\left( \frac{\phi_x}{|\nabla \phi|} \right)_{i+1/2,j} = \frac{\delta^+_x \phi_{i,j}}{\sqrt{(\delta^+_x \phi_{i,j})^2 + \frac{1}{4}(\delta^+_y \phi_{i,j} + \delta^0_y \phi_{i+1,j})^2 + \varepsilon_1}},
$$

$$
\left( \frac{\phi_x}{|\nabla \phi|} \right)_{i-1/2,j} = \frac{\delta^-_x \phi_{i,j}}{\sqrt{(\delta^-_x \phi_{i,j})^2 + \frac{1}{4}(\delta^-_y \phi_{i,j} + \delta^0_y \phi_{i-1,j})^2 + \varepsilon_1}}.
$$
\[
\begin{align*}
\left(\frac{\phi_y}{\nabla \phi}\right)_{i,j+1/2} &= \frac{\delta_y^i \phi_{i,j}}{\sqrt{\frac{1}{4} (\delta_x^i \phi_{i,j} + \delta_x^i \phi_{i,j+1})^2 + (\delta_y^i \phi_{i,j})^2 + \epsilon_1}}, \\
\left(\frac{\phi_y}{\nabla \phi}\right)_{i,j-1/2} &= \frac{\delta_y^i \phi_{i,j}}{\sqrt{\frac{1}{4} (\delta_x^i \phi_{i,j-1} + \delta_x^i \phi_{i,j})^2 + (\delta_y^i \phi_{i,j})^2 + \epsilon_1}}.
\end{align*}
\]

We observe from experiments that the level set function in the region far from the interface converges quickly to 1 and 2. Once the topology structure for optimal design is determined, the interface is usually smooth and it still takes rather many steps for \( \phi \) in the region near the interface to form the discontinuity. To speed up the projection Lagrangian algorithm, we propose an acceleration technique similar as that applied in the augmented Lagrangian algorithm under the binary level set frame [20]. We introduce the function

\[
\tilde{\phi} := \text{sign}(2\phi - 3) = \begin{cases} 
\frac{2\phi - 3}{|2\phi - 3|} & \text{if } \phi \neq \frac{3}{2} \\
0 & \text{else}
\end{cases}
\]

and replace \( \phi \) by the function \( \hat{\phi} = (\tilde{\phi} + 3)/2 \). Inserting \( \hat{\phi} \) into Eq. (17), we obtain

\[
\rho(\hat{\phi}) = \begin{cases} 
\frac{1}{2} (1 - \epsilon) \frac{2\phi - 3}{|2\phi - 3|} + \frac{1}{2} (1 + \epsilon) & \text{if } \phi \neq \frac{3}{2}, \\
\frac{1}{2} (1 + \epsilon) & \text{else}.
\end{cases}
\]

The chain rule implies

\[
\rho'(\hat{\phi}) = \rho'(\tilde{\phi}) \hat{\phi}'(\tilde{\phi}) \tilde{\phi}'(\phi) = \frac{1}{2} \delta(2\phi - 3)(1 - \epsilon),
\]

where \( \delta(x) \) denotes the Dirac delta function, i.e. \( \delta(0) = 1 \) and \( \delta(x) = 0 \ \forall x \neq 0 \).

In numerical implementations, \( \hat{\phi} \) is replaced by a smoothed approximation

\[
\hat{\phi} \approx \frac{2\phi - 3}{2 \sqrt{(2\phi - 3)^2 + \epsilon_2}} + \frac{3}{2},
\]

where \( \epsilon_2 \) is a small positive number which has to be chosen properly. As \( \phi \) is replaced by \( \hat{\phi} \), the gradient computation in (34) should also be changed using (44). In numerical experiments, however, we observe that satisfactory results can be obtained if we just replace \( \frac{1}{2} \delta(2\phi - 3) \) by 1. Therefore, we do not need to change the codes for the algorithm. The only modification for the codes is to replace \( \rho(\phi) \) by \( \rho(\hat{\phi}) \).

It should be noted that the acceleration procedure only starts shortly after the optimal shape is formed. In other words, the algorithm has nearly converged right before acceleration. The acceleration actually can only sharpen the smooth optimal interface.
In the following text, we refer to the ordinary and the accelerated implementation of the projection Lagrangian algorithm as the ordinary projection Lagrangian algorithm (OPLA) and the accelerated projection Lagrangian algorithm (APLA), respectively. For APLA, we have chosen $10^{-3}$ as the initial value for $\varepsilon_2$ and decreased it by a factor of 0.9 at each iteration.

5 Numerical experiments

We perform some numerical tests for our algorithms in this section. We choose $U = (0, 1) \times (0, 1)$ and divide it into $60 \times 60$ square elements. In order to make numerical comparisons between the presented algorithms and level set methods, we use numerical examples in [12]. At each iteration, the linear finite element approximation is employed to solve the state equation. Unless specified otherwise, we choose $\varepsilon_1 = 10^{-6}$, $\varepsilon = 10^{-3}$ and set $\theta = 2$ for the time step. To test robustness of the algorithm, we use four initial values of $\phi^0$ and their corresponding density distributions in Fig. 2. The initial guess in Fig. 2(b) is a uniform random distribution in $[-1, 1]$ with zero mean. Denote by $\Omega^*$ the optimal shape of $\Omega$ and $J^*_{\epsilon}$ its corresponding cost functional.

Example 1. $f(x, y) = 10 \sin^2(4\pi x)$, $\lambda = 0.5$, $\mu = 0$

We first demonstrate the necessity of acceleration by comparing the OPLA and APLA for this example. With initial design $\phi^0 = 1.5$, the iteration process for the APLA is exactly the same as that for the OPLA in the first 50 steps. Then we start to accelerate after 50 iterations. We show evolutions of the level set function and its corresponding density distribution in Fig. 3 and Fig. 4, respectively. By APLA, $\phi$ converges to 1 or 2 after 100 iterations shown in Fig. 3(f). But the discontinuity is still not formed by OPLA even after 300 steps as shown in Fig. 3(e). Although we observe from Fig. 5(a) that the convergence history of the objective functional by APLA visually agrees well with that by OPLA, Fig. 5(b) indicates that the APLA converges much faster than OPLA. In Table 1, the optimal value $J^*_{\epsilon}$ quantitatively shows that APLA is slightly superior to OPLA. We also can see that there is much reduction in the iteration number.

For this example, it is illustrated in [12] that the shape gradient based level set algorithm converges to a local maximum which is lower than that obtained by using the topological derivatives. The topology of the optimal shape obtained by the shape gradient based algorithm is largely different from that of the solution obtained by the shape-topological gradient based algorithm. Here we show our algorithm is less likely to fall into a local optimum than the classical shape gradient based level set algorithm. Our algorithm leads to a similar shape (Fig. 4(f)) as that using the topological derivative based algorithm. As shown in Table 1, although either of the present $J^*_{\epsilon}$ is slightly lower than that obtained by the level set method with shape and topological derivatives [12], it is much larger than that obtained using the shape gradient based level set algorithm. Furthermore, the comparison on the iteration number required demonstrates the efficiency of our algorithm.
All the following tests are performed using the APLA. To show that the algorithm is nearly independent of the initial design, other three initial designs in Fig. 2 are used for test. We see from Fig. 4(f) and Fig. 6 that the four final shapes are all similar as that by the shape-topological gradient based level set method [12]. The acceleration begins after 100 steps. See Fig. 7 for the comparisons on the convergence history from different initial designs. The four final values for $J_\epsilon^*$ or $\|H(\phi)\|_{L^2}$ are very close to each other.

The following tests are based on $\phi^0 = 1.5$. The acceleration procedure starts after 60 iterations. In Fig. 8, we show the minor effect of the ersatz material density on the optimal design and objective functional when $\epsilon$ is tiny.

Example 2. $f(x, y) = 10x + y$

Case 1. $\lambda = 1, \mu = 0$
See Figs. (9)-(10) for the optimization process and the curves of objective functional and $\|H(\phi)\|_2$. The optimized shape in Fig. 9(d) is similar as that in [12].

Case 2. $\lambda = 1.7, \mu = 0$
Increasing the parameter $\lambda$ for the area constraint leads to a different design process depicted on Fig. 11. Compared with Fig. 9(d), a relatively larger area for $\Omega^*$ is obtained as shown in Fig. 11(c).

Case 3. $\lambda = 1.7, \mu = 0.4$
We finally test the influence of the perimeter constraint. We choose $c = 5.0$. See Fig. 12. The perimeter of $\Omega^*$ in Case 2 turns from 1.27 to 0.96 in Case 3.

Finally, we compare APLA and the topology gradient based level set algorithm [12] for example 2 in Table 2, which illustrates the accuracy and efficiency of our algorithm.

6 Conclusions

Based on the PCLSM, we have presented a variational method for a class of elliptic boundary value problems in shape optimization. We introduce a weak phase and approximate the original model by a two-phase optimal shape design problem. Then by introducing a PCLSF, we transform this two-phase problem to a new constrained minimization problem. We use a projection Lagrangian algorithm to solve it. Compared with standard shape gradient based level set algorithms, we do not need to solve the Hamilton-Jacobi equation and perform the re-initialization process. Furthermore, the proposed method has a hole nucleation mechanism as topological gradient based algorithms. Numerical experiments demonstrate that the proposed accelerated algorithm is very efficient and robust with respect to the initial guess of the level set function.

However, PDE constrained shape optimization problems are generally nonconvex. Both our algorithm and the algorithm of [12] are gradient-type methods, which means that
they are dependent on the initial guess and likely to fall into local extrema. There is no guarantee that the converged solution obtained by our algorithm is globally optimal although the algorithm is robust and depends not too much on the initial guess. But from numerical comparisons in [12] and our paper, both the shape-topological gradient based level set method and the presented algorithm are at least less likely to fall into local extrema than the classical shape gradient based level set algorithm.

Acknowledgements

We would like to thank anonymous reviewers and the Editor for their valuable suggestions and constructive comments. The first author wishes to thank Prof. Cheng and Dr. Xiaoxia Dai for helpful discussions on the model problem. This work was supported by National Natural Science Foundation of China (No. 10871178) and Natural Science Foundation (No. Y606154) of Zhejiang Province.
References


Fig. 2. Four initial designs for the level set function and the density distributions.

Table 1
Comparison of different methods for example 1.

<table>
<thead>
<tr>
<th>Method</th>
<th>$J^*_\epsilon$</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPLA</td>
<td>0.277648</td>
<td>300</td>
</tr>
<tr>
<td>APLA</td>
<td>0.277783</td>
<td>100</td>
</tr>
<tr>
<td>SG-LSM$^a$</td>
<td>0.264070</td>
<td>331</td>
</tr>
<tr>
<td>STG-LSM$^b$</td>
<td>0.282319</td>
<td>331</td>
</tr>
</tbody>
</table>

$^a$ Shape gradient based level set method.
$^b$ Shape-topological gradient based based level set method.
Table 2  
Comparison on the optimal objective functional and the iteration number for example 2 between present work and [12].

<table>
<thead>
<tr>
<th>Case</th>
<th>Present $J^*$</th>
<th>$J^*$ [12]</th>
<th>Present Iterations</th>
<th>Iterations [12]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda = 1, \mu = 0$</td>
<td>0.597983</td>
<td>0.598434</td>
<td>100</td>
<td>176</td>
</tr>
<tr>
<td>$\lambda = 1.7, \mu = 0$</td>
<td>1.189580</td>
<td>1.190227</td>
<td>100</td>
<td>143</td>
</tr>
<tr>
<td>$\lambda = 1.7, \mu = 0.4$</td>
<td>1.184526</td>
<td>1.185849</td>
<td>100</td>
<td>143</td>
</tr>
</tbody>
</table>
Fig. 3. Comparison of APLA and OPLA for shape optimization of Example 1 from initial design $\phi^0 = 1.5$: evolution of the level set function.
Fig. 4. Comparison of APLA and OPLA for shape optimization of Example 1 from initial design $\phi^0 = 1.5$: evolution of the density distribution.
Fig. 5. Comparison of APLA and OPLA for shape optimization of Example 1. The objective functional (left) and $\| H(\phi) \|_{L^2}$ (right) vs. number of iterations.

Fig. 6. The final optimal design for Example 1 from different initial designs Figs. 2(b)-2(d). From left to right: $J_\epsilon^* = 0.277849, 0.276789$ and 0.276006.

Fig. 7. Shape optimization of Example 1 by APLA from four different initial designs: the objective functional and $\| H(\phi) \|_{L^2}$ vs. number of iterations.
Fig. 8. The final optimal design for Example 1 from different fictitious material densities. From left to right: \( \epsilon = 10^{-2} \) \( (J_{\epsilon}^* = 0.279197) \), \( \epsilon = 10^{-3} \) \( (J_{\epsilon}^* = 0.277783) \), \( \epsilon = 10^{-4} \) \( (J_{\epsilon}^* = 0.277734) \).

Fig. 9. Shape optimization for Case 1 of Example 2 by APLA: evolution of the density distribution.

(a) Iteration 20  
(b) Iteration 30  
(c) Iteration 60  
(d) Iteration 100
Fig. 10. Shape optimization for Case 1 of Example 2 by APLA: the objective functional (left) and $\|H(\phi)\|_{L^2}$ (right) vs. number of iterations.
Fig. 11. Shape optimization for Case 2 of Example 2 by APLA: evolution of the density distribution.
Fig. 12. Shape optimization for Case 3 of Example 2. Left: the optimal shape. Right: the objective functional vs. number of iterations.