# MULTIMATERIAL TOPOLOGY OPTIMIZATION BY VOLUME CONSTRAINED ALLEN-CAHN SYSTEM AND REGULARIZED PROJECTED STEEPEST DESCENT METHOD

R. TAVAKOLI

Abstract. A new computational algorithm is introduced in the present study to solve multimaterial topology optimization problems. It is based on the penalization of the objective functional by the multiphase volume constrained Ginzburg-Landau energy functional. The update procedure is based on the gradient flow of the objective functional by a fractional step projected steepest descent method. In the first step, the new design is found based on the projected steepest descent method to ensure the reduction in the objective functional, simultaneously satisfying the control constraints. In the second step, regularization step, an $H^1$ regularity of the solution is ensured while keeping the feasibility of solution with respect to the set of control constraints. The presented algorithm could be accounted as a constrained $H^1$ optimization algorithm, which, according to our knowledge, has not been reported to solve such kind of problems yet. The success and efficiency of the presented method are shown through several test problems. Numerical results show that the presented algorithm ends with a near 0-1 topology and its computational cost scales sub-linearly by the number of phases. For the sake of reader convenience and the ease of further extension, the MATLAB implementation of the presented algorithm is included in the appendix.

**Keywords.** Gradient projection; MATLAB code; Multiphase Allen-Cahn; Multiphase Cahn-Hilliard; Multiphase topology optimization; Sobolev gradient.

## Contents

1. Introduction 2
2. Problem Formulation 4
2.1. Multimaterial Minimum compliance topology optimization problem 4
2.2. Regularization by Perimeter Penalization 5
2.3. Constraint qualification 8
3. Optimality conditions 8
3.1. Optimality conditions based on the projected gradient approach 8
3.2. First order adjoint sensitivity analysis 10
3.3. First order necessary optimality conditions 10
4. Regularized projected gradient flow minimization algorithm 10
5. The spatial discretization 14
6. Projection onto the design space 14
7. Computer implementation 17
8. Numerical results and discussion 17
9. Summary 29
Acknowledgment 29
Appendix A. MATLAB code 30
References 32

---

*Date*: April 4, 2014.

R. Tavakoli, Materials Science and Engineering Department, Sharif University of Technology, Tehran, Iran, P.O. Box 11365-9466, tel: 982166165209, fax: 982166005717, email: rtavakoli@sharif.ir.

## 1. Introduction

Topology optimization [1] aims at finding the distribution of a material in a fixed design domain such that an objective functional is minimized under certain constraints. In recent years there has been a growing interest in solving topology optimization problems using multiple phases. The possibility of using multiple number of phases during design of engineering structures opens a new window toward the design of smart and advanced structures. It is particularly interesting for the optimal design of multifunctional structures, i.e., including a specific phase to manage a specific functionality. For instance, materials with high thermal conductivity usually have high thermal expansion coefficients and poor mechanical properties. On the other hand, high strength materials commonly have poor thermal conductivities. Therefore, using multiple number of phases appears to be a reasonable choice for the design of structures under thermomechanical loading. Moreover, multimaterial design strategy can reduce the total price or weight of structures, in contrast to the traditional single phase design paradigm. The later issue has been addressed in [2] by the mass constraint reformulation of multimaterial topology optimization problems. Solution of multimaterial topology optimization problems however includes several challenges. The lack of physically-based materials interpolation scheme [3] and efficient numerical method to solve the discritized form of the original problem is of these challenges. The later issue is our main goal in the present work.

The material interpolation here means computing the local value of desired physical properties based on the volume fractions of contributing phases. The solid isotropic material with penalization model (SIMP) [1], averaged Hashin-Shtrikman bounds [1] and homogenization methods [4, 1] have been extensively used in the literature for (void-material) binary-phase topology optimization problems. The extension of these methods to manage arbitrary number of phases is not an easy job. For instance it is well known that the extended version of SIMP approach to three or more phases does not essentially follow the Hashin-Shtrikman bounds [1]. There are several attempts, e.g. [5, 6], to develop general multimaterial interpolation schemes, there is however no progress toward developing a physically based material interpolation approach to mange arbitrary number of materials. Due to the simplicity and satisfactory results, the extended version of SIMP method has been extensively used in literature, e.g. [2, 6–10], to perform the material interpolation. In practice, the material interpolation is an artificial construction in the context of the topology optimization, because a successful procedure results in a distinct phase within each computational element at the final solution [11]. Therefore, the interpolation model influences the optimization path in terms of the computational efficiency and the final design (note that the problem conconvexity leads to existence of different local optima). Appropriate schemes have to be developed for interpolation of the physical properties of multiphase materials. For instance, the existence of such interpolation schemes is a prerequisite for optimal design of multiphase functionally graded materials [12–14]. It is worth to mention that the sharp interface tracking approaches, like the level set method [15], can eliminate the need of material interpolation schemes. It is important to note that this issue is only correct if inter-phase interfaces are actually tracked explicitly (i.e. one does not have a computational cell including more than one phase), which is not the case of matter in many level set based methods.

Using the homogenization method, the design of three-phase composites with extremal thermal expansion, piezoelectricity and bulk modulus have been considered in [16, 17], [18] and [19] respectively (cf. [20]). To avoid the topological instability (cf. [21]) and posing control on the minimum length scale, the low-pass sensitivity filtering [1] has been included in these works. Using combined multiphase SIMP approach and sensitivity filtering, optimal design of piezoelectric actuators has been considered in [9]. In these works, authors however did not attend to numerical solution of resulted systems of optimality conditions and passed it to a general purpose optimization black-box. In [22], the optimization of the position of fuel assemblies in a nuclear reactor core has been formulated as a four-phase topology optimization problem based on the homogenization theory. Then, it has been solved numerically using a simple constrained gradient descent approach. Using the block coordinate descent approach, the classical optimality criteria approach [23, 1] has been extended in [10] to solve multimaterial topology optimization problems.

The level set and the variational level set approaches have been adapted in [22, 24, 11, 25–27] to solve multimaterial topology optimization problems. The piecewise-constant variational level set method has been used in [28, 29] to improve the computational cost of multiple level set based approaches. The dependency of the final topology to the initial design appears to be the main shortcoming of the level-set based methods.

Optimization of laminated composite shell structures using multimaterial topology optimization strategy, called discrete material optimization (DMO), has been introduced in [30, 31]. In this method, every virtual material is defined as a fiber reinforced composite with a priori known fiber volume fraction and orientation, consequently with a priori known elasticity tensor. The design variables are defined as the local volume fractions of these virtual materials and the local elasticity tensor is computed based on the mixture rule modified by the SIMP penalization approach. Solution of optimal design problem up to 12 virtual phases has been considered by the authors in this work. DMO approach has been used in [2] to solve multimaterial topology optimization with the mass constraint. Using interpolation based on Lagrangian shape functions, commonly used in finite element analysis, a new interpolation approach, called shape function parameterization (SFP), has been introduced in [32, 33] to reduce the number of design variables in DMO. SFP approach has been modified in [34] by introducing bi-value coding parameterization (BCP), to further reduce the number of design variables; for instance using only 4 design vectors to solve 16-phase optimization problems. They also reported numerical results for optimal design problems up to 36 candidate material orientations.

A phase-field approach coupled to a special material interpolation method has been used in [35, 36] to solve three-phase topology optimization problems. In this method one field variable has been used to determine the volume fraction of different phases. Because of using mono-variable material interpolation method, it permits only the existence of at most two phases at interphase boundaries, i.e., triple junctions are mandatory filtered by this approach. Using multimaterial phase-field approach based on Cahn-Hilliard equation, a general method to solve multiphase structural topology optimization problems has been introduced in [7, 8]. The intrinsic volume preserving property is one of the most important benefit of this method, i.e., the iterations will be kept strictly feasible with respect to the design domain without any further effort. The slow convergence of the phase field method is however the main drawback of this approach. For instance over $10^4$ iterations are commonly required to find a suitable solution. This limitation stems in the severe time-step restriction due to the stability of nonlinear forth order parabolic PDEs corresponding to the evolution of phase-field variables. The volume constrained Allen-Cahn equation has been suggested in [37, 38] to overcome the slow convergence of Cahn-Hilliard based topology optimization methods. The asymptotic convergence of this method to sharp interface limit has been analyzed in [39] by the method of matched asymptotic. They demonstrated that the sharp interface limit of the optimality conditions corresponding to the phase-field approach asymptotically converges to that of the shape calculus.

In the present work a novel computational approach is introduced to solve multimaterial volume constrained topology optimization problems. For this purpose, at first the objective functional is penalized by a new multiphase version of Ginzburg-Landau energy functional (cf. [40]. Then the evolution of constrained gradient flow of the penalized objective functional is used to find the optimal solution. By splitting the penalized objective functional into two parts and managing the overall procedure by an operator-splitting like approach (cf. [41]), the miss of regularity due to the $L^2$ projection is bypassed while iterations are still kept inside the feasible domain. We believe that our approach is somewhat equivalent to constrained $H^1$ gradient flow of the penalized objective functional. It is worth to mention that an alternative $H^1$-type topology optimization algorithm has been developed in [42] for the solution of binary phase topology optimization problems. However, it significantly differs from the method presented in the present study. The first step of their algorithm is an unconstrained $H^1$ optimization by the (regularized) steepest descent method. Then, in the second step, the feasibility of the solution will be ensured by the projection of the temporary solution on to the feasible domain with respect to the $H^1$ norm. However, it is computationally expensive, in particular in the case of multimaterial topology optimization problems. In contrast, in the present work, the first step includes the optimization on the $L^2$ space by the projected steepest descent algorithm. The $L^2$-projection onto the feasible domain will be performed efficiently with a practically time-linear algorithm. Then the $H^1$ regularity will be ensured by the regularization step, lifting from $L^2$ space to $H^1$ space by the inverse Helmholtz operator, while the feasibility of the solution will be kept in this step without any extra effort. Moreover, the presented method manages arbitrary number of phases without extra complexity in the implementation phase. Although the multiphase version of SIMP approach is used for the material interpolation in the present study, we are hopeful that our approach works successfully using other material interpolation methods.

## 2. Problem Formulation

2.1. **Multimaterial Minimum compliance topology optimization problem.** The volume constrained (binary-phase) minimum compliance topology optimization problem is one of the most common model problem in the literature of topology optimization. In the minimum compliance topology optimization problem, the unknown material distribution is determined such that the stiffness of structure is maximized, or equivalently the structure's compliance is minimized. To avoid the trivial solution, a global volume constraint is applied on the total volume occupied by the stiff phase in the final design. Following [8, 10], we present the multimaterial version of this model problem in this section.

Consider $\Omega$ as the design domain. It is assumed that $\Omega$ is a fixed nonempty and sufficiently regular subset of $\mathbb{R}^d$ $(d = 1, 2, 3)$. It is assumed that $\Omega$ is occupied by $p \in \mathbb{N}$ $(p \geqslant 2)$ number of distinct isotropic linearly elastic materials. Possibly, the void could be considered as a separate phase in our formulation. The optimal design problem goals to find the optimal material distribution inside $\Omega$. The materials distribution is determined by the local volume fraction fields, $\varrho_i$ $(i = 1, \dots, p)$, corresponding to the contributing phases; note that $\varrho_i = \varrho_i(\mathbf{x})$. It is assumed that $\varrho_i \in \mathcal{V}(\Omega)$, where $\mathcal{V}$ is a sufficiently regular function space. Whenever there is no confusion, we will denote all design vectors (scalar fields) into a single vector field denoted by $\boldsymbol{\varrho}$, i.e., $\boldsymbol{\varrho} = \{\varrho_1, \dots, \varrho_p\}$.

Assume that the displacement field in $\Omega$ is denoted by vector filed $\mathbf{u} \in \mathcal{U}(\Omega)$, where $\mathcal{U}$ is a sufficiently regular function space. For a given materials distribution $\boldsymbol{\varrho}$, $\mathbf{u}$ should satisfy the following PDE:

$$\left\{ \begin{array}{rcll} \nabla \cdot (\mathbf{C} : \mathcal{D}(\mathbf{u})) & = & \mathbf{f}(\mathbf{x}) & \texttt{in} \quad \Omega \\ \mathbf{u}(\mathbf{x}) & = & \hat{\mathbf{u}}(\mathbf{x}) & \texttt{on} \quad \Gamma_u \\ (\mathbf{C} : \mathcal{D}(\mathbf{u}))\mathbf{n} & = & 0 & \texttt{on} \quad \Gamma_f \\ (\mathbf{C} : \mathcal{D}(\mathbf{u}))\mathbf{n} & = & \hat{\mathbf{t}}(\mathbf{x}) & \texttt{on} \quad \Gamma_t \end{array} \right. \tag{1}$$

where $\mathbf{f}$ denotes the volumetric body force, $\hat{\mathbf{u}}$ denotes the prescribed displacement on boundaries $\Gamma_u$, $\Gamma_f$ denotes the traction free parts of boundaries, $\hat{\mathbf{t}}$ denotes the specified traction force applied on a part of boundaries of $\Omega$ denoted by $\Gamma_t$, $\partial\Omega := \Gamma = \Gamma_u \cup \Gamma_f \cup \Gamma_t$ and $\mathcal{D}(\mathbf{u}) := \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) = \frac{1}{2}(\partial_i u_j + \partial_j u_i)_{i,j}$, $i, j = 1 \dots, d$. Note that the double dot operator, " : ", denotes the usual contraction over two sets of indices. Moreover, $\mathbf{C} = \mathbf{C}(\mathbf{x}) = \mathbf{C}(\varrho(\mathbf{x}))$ denotes the elasticity tensor which is a forth order supersymmetric tensor (symmetric in both the right and the left Cartesian index pair, together with symmetry under the interchange of the pairs). Following [8, 10], the SIMP approach is used for the purpose material interpolation in the present study. Therefore we have:

$$\mathbf{C}(\boldsymbol{\varrho}) = \sum_{i=1}^{p} \varrho_i^q \ \mathbf{C}_i \tag{2}$$

where $q \geqslant 3$ is the SIMP penalization power and $\mathbf{C}_i$ is the constant stiffness tensor corresponding to phase $i$-th. The use of SIMP approach in this study is because of its simplicity, ease of extension to arbitrary number of phases and satisfactory results. For these reasons, it has been extensively used in literature, for instance see [2, 6–10, 43]. As it is mentioned in section 1, the interpolation of the elasticity tensor based on (2) is not always feasible. This choice could result in nonphysical data, e.g., for large values of SIMP-power $q$, the elasticity tensor could violate the Hashin-Shtrikman bounds. According to [1], a reasonable interpolation scheme is to compute the Hashin-Shtrikman lower and upper bounds for each of the physical property. Then, the interpolated elasticity tensor will be computed by replacing its entries with their corresponding averaged Hashin-Shtrikman bounds. However, it is very difficult to find the explicit form of Hashin-Shtrikman bounds when number of phases is greater than 3. Moreover, Hashin-Shtrikman bounds are attainable under certain conditions on volume fractions and physical properties of original phases [19]. Because a topology optimization solver commonly ends with a near 0-1 topology, by virtue, the material interpolation scheme 2 does not effect on the final solution. However it could change the intermediate optimization path. Considering the fact that our goal in the present study is to develop a computationally efficient solver for multimaterial volume constrained topology optimization problems, we limit ourself to SIMP approach here.

The objective functional, to be minimized, is defined as follows:

$$\mathcal{J}(\boldsymbol{\varrho},\mathbf{u}) = \frac{1}{2}\int_{\Omega} (\mathbf{C}(\boldsymbol{\varrho}) : \mathcal{D}(\mathbf{u})) : \mathcal{D}(\mathbf{u}) \; d\mathbf{x} \tag{3}$$

In addition to PDE (1), we have several constraints on the design vector $\varrho$. Obviously, we have the following pointwise bound constraints on every $\varrho_i$,

$$l_i \leqslant \varrho_i \leqslant h_i, \quad i = 1, \ldots, p \tag{4}$$

where $l_i, h_i \in \mathcal{V}(\Omega)$ and $0 \leqslant l_i \leqslant h_i \leqslant 1$. The inequalities are understood componentwise here. It is worth to mention that to fix the topology of a material at a specific portion of $\Omega$, it is sufficient to use the desired fixed value instead of lower and upper bounds in (4). Since no overlap and gap are allowed in the desired design, the sum of volume fractions at every point $x \in \Omega$ should be equal to unity, i.e.,

$$\sum_{i=1}^{p} \varrho_i = 1 \tag{5}$$

where the summation operator is to be understood componentwise (local) here. The intersection of above mentioned control constraints is commonly called at the Gibbs simplex in materials science literature. In every volume constrained topology optimization problem, we further have a global volume constraint on the total volume of each material inside $\Omega$. Without loss of generality, we restrict ourselves to equality resource constraints in the present study, i.e.,

$$\int_{\Omega} \varrho_i \; d\mathbf{x} = \Lambda_i |\Omega|, \quad i = 1, \ldots, p \tag{6}$$

where $\Lambda_i$ $(i = 1, \ldots, p)$ are user defined parameters; obviously $0 \leqslant \Lambda_i \leqslant 1$ and $\sum_{i=1}^{p} \Lambda_i = 1$. These constraints are commonly called the resource constraints. Therefore, the admissible design space, denoted by $\mathcal{A}$ here, could be expressed as follows:

$$\mathcal{A} := \left\{ \boldsymbol{\varrho} \in \{\; \varrho_i \in \mathcal{V}(\Omega) \;\}_{i \in \{1,\ldots,p\}} \;\middle|\; \begin{array}{ll} \sum_{i=1}^{p} \varrho_i = 1, & \\ \int_{\Omega} \varrho_i \; d\mathbf{x} = \Lambda_i |\Omega|, & i = 1, \ldots, p \\ l_i \leqslant \varrho_i \leqslant h_i, & i = 1, \ldots, p \end{array} \right\}$$

Putting all together, we are going to solve the following optimization problem:

$$\min_{(\boldsymbol{\varrho},\mathbf{u}) \in (\mathcal{A},\mathcal{U})} \mathcal{J}\big(\boldsymbol{\varrho}, \mathbf{u}(\boldsymbol{\varrho})\big) \quad \textit{subject to}: \quad \textit{equation} \;\; (1) \tag{7}$$

It is important to note that due to the local incompressibility constraints, we only need to determine $\varrho_i(\mathbf{x})$ for $i = 1, \ldots, p-1$.

### 2.2. **Regularization by Perimeter Penalization.**

Mesh dependency, topological instability and checkerboard patterns [21] are of the famous problems during the numerical solution of topology optimization problems. The density filtering, sensitivity filtering and perimeter penalization are commonly used to overcome these difficulties [21]. The penalization of objective functional with the interphase perimeter has been firstly used in [44] to avoid the topological instability during two phase compliant based topology optimization. In this subsection, we shall introduce an approximation of the total interphase perimeter based on the extended Modica and Mortola [45, 46] approach.

Assuming the domain $\Omega$ is occupied by $p$ number of phases and $\chi_i \in BV(\omega)$ $(i = 1, \dots, p)$ denotes the characteristic function of the phase $i$-th ($\chi_i \in \{0, 1\}$), the total perimeter, $P$, could be computed as follows,

$$P(\chi_1, \dots, \chi_p) = \sum_{i=1}^{p} \|\chi_i\|_{BV(\Omega)} = \sum_{i=1}^{p} \int_\Omega |D\chi_i| \ d\mathbf{x} \tag{8}$$

where $BV(\Omega)$ is the normed space of functions of bounded variations and $D(\cdot)$ denotes the distributional (weak) derivative operator. To use (8), it is required to determine the characteristic function of each phase in $\Omega$, i.e., having a 0-1 topology. However, working with 0-1 topologies is computationally very expensive in practice (needs to solve a large scale integer programming problem). An approximate form of (8) will be used in the present study. In [45, 46], it has been shown that for $p = 2$, $P^\varepsilon$ $\Gamma$-converges to $P$ as $\varepsilon \to 0$,

$$P^\varepsilon(\varrho) = \int_\Omega \frac{\varepsilon}{2} |\nabla \varrho|^2 \ d\mathbf{x} + \int_\Omega \frac{1}{\varepsilon} W(\varrho) \ d\mathbf{x} \tag{9}$$

where $\varrho$ is the regularized characteristic function which varies continuously between 0 and 1, $W : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is a scalar function with exactly two minimizers at 0 and 1 satisfying $W(0) = W(1) = 0$ and $W$ assumes $\infty$ value outside of $[0, 1]$. Moreover, $\int_\Omega \varrho \ du = \Lambda$ and $\int_0^1 \sqrt{W(r)} \ dr = 1/2$. Note that in the case of $p = 2$, only a single field variable is sufficient to compute the total perimeter (as used in (9)). Therefore, the solution will be attained by continuation on $\varepsilon$, i.e., start with a sufficiently large $\varepsilon$ and solve a series of problems by sequential decreasing of $\varepsilon$. Common choices for $W$ are [47]: either the double-well potential $W(r) = r^2(1 - r)^2$ $(r \in \mathbb{R})$, or the double-obstacle potential $W(r) = r(1-r)$ ($r \in [0, 1]$, $W$ assumes the $\infty$ value outside of $[0, 1]$ interval). Formulae (9) has been extensively used in the literature of binary phase (void-material) topology optimization, for instance see [47, 48, 43, 49–51].

There are a few works on extension of (9) for multiphase problems. The following irreducible form has been suggested in [52–54],

$$P^\varepsilon(\varrho_1, \dots, \varrho_p) = \sum_{i=1}^{p} \sum_{j=i+1}^{p} \int_\Omega \frac{\varepsilon}{2} |\varrho_i \nabla \varrho_j - \varrho_j \nabla \varrho_i|^2 \ d\mathbf{x} + \int_\Omega \frac{1}{\varepsilon} F(\varrho_1, \dots, \varrho_p) \ d\mathbf{x} \tag{10}$$

where $F$ is a $p$-well function in the above formulation. From a thermodynamical point of view, $F$ could be written as follows:

$$F(\varrho_1, \dots, \varrho_p) = \sum_{i=1}^{p} \varrho_i \ln \varrho_i + c \sum_{i=1}^{p} \sum_{j=i+1}^{p} \varrho_i \varrho_j \tag{11}$$

where $c \geqslant 0$ is a constant. However, due to the singularity of $\ln(x)$ near $x = 0$, working with (11) is difficult in practice. Several polynomial based choices for $F$ have been suggested in the literature. For instance [53, 54],

$$F(\varrho_1, \dots, \varrho_p) = 9 \sum_{i=1}^{p} \sum_{j=i+1}^{p} \varrho_i^2 \varrho_j^2 + 6 \sum_{i=1}^{p} \sum_{j=i+1}^{p} \sum_{k=j+1}^{p} \varrho_i^2 \varrho_j^2 \varrho_k^2 \tag{12}$$

or [8],

$$P^\varepsilon(\varrho_1, \dots, \varrho_p) = \sum_{i=1}^{p} \sum_{j=i+1}^{p} \int_\Omega \frac{\varepsilon}{2} \ \nabla \varrho_i \cdot \nabla \varrho_j \ d\mathbf{x} + \sum_{i=1}^{p} \sum_{j=i+1}^{p} \int_\Omega \frac{c}{\varepsilon} \ \varrho_i^2 \varrho_j^2 \ d\mathbf{x} \tag{13}$$

where $c \in \mathbb{R}^+$ is a constant parameter. In contrast to (10), model (13) possesses a lower nonlinearity. The rigorous mathematical theory for extension of the Modica-Mortola approximation to multiphase systems has been developed in [55]. In this work, it has been proved that if the function $F$ has $p$ minimizers and assumes zero only at its minimizing points, and if we have global volume and local bound constraints on $\boldsymbol{\varrho} = (\varrho_1, \dots, \varrho_p)$, the following functional will $\Gamma$-converge (up to a constant scaling) to the perimeter of $p$-phase system as $\varepsilon \to 0$,

$$P^{\varepsilon}(\varrho_1, \ldots, \varrho_p) = \sum_{i=1}^{p} \int_{\Omega} \frac{\varepsilon}{2} |\nabla \varrho_i|^2 \, d\mathbf{x} + \int_{\Omega} \frac{1}{\varepsilon} \, F(\varrho_1, \ldots, \varrho_p) \, d\mathbf{x} \tag{14}$$

Very recently, in [56] the convergence theory of [55] has been extended to the following simplified form of (14):

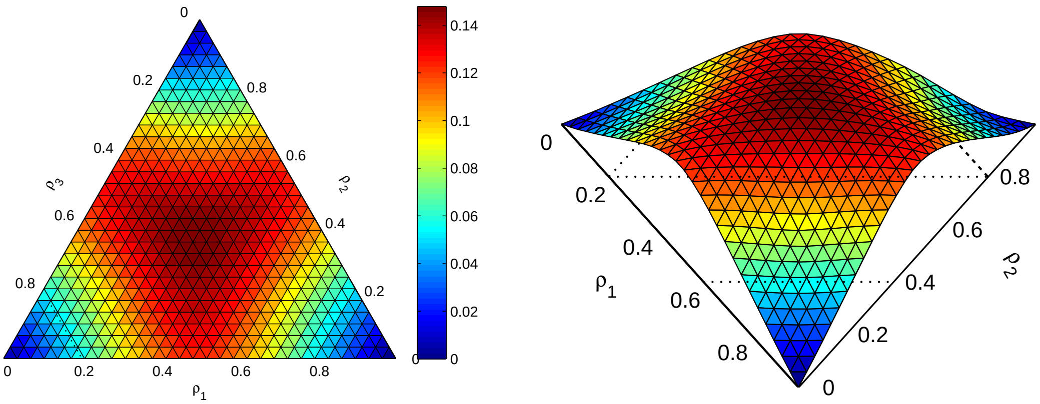
$$P^{\varepsilon}(\varrho_1, \ldots, \varrho_p) = \sum_{i=1}^{p} \int_{\Omega} \frac{\varepsilon}{2} \, |\nabla \varrho_i|^2 \, d\mathbf{x} + \sum_{i=1}^{p} \int_{\Omega} \frac{1}{\varepsilon} \, W(\varrho_i) \, d\mathbf{x} \tag{15}$$

where $\boldsymbol{\varrho} = (\varrho_1, \ldots, \varrho_p)$ has to lie inside the volume constrained Gibbs simplex $\mathcal{A}$. The approximation (15) has several benefits in contrast to (12) and (13), for instance: lower nonlinearity, ease of implementation and a rigorous convergence theory.

The penalization of the objective functional by the perimeter approximation (15) with the following double-well potential will be used in the present study:

$$W(r) = r^2(1-r)^2 \tag{16}$$

To give a better insight into the behavior of the p-well function used in the present study, the plot of 3-well function $W_3(\varrho) = \sum_{i=1}^{3} \varrho_i^2(1-\varrho_i)^2$ on the Gibbs 3-simplex is shown in the figure 1.



(a) top view (b) side view

FIGURE 1. The plot of 3-well function, $W_3(\varrho) = \sum_{i=1}^{3} \varrho_i^2(1-\varrho_i)^2$, on the Gibbs 3-simplex (contours illustrates the value of function).

Due to the existence of local incompressibility constraints in the present study, we redefine (15) to the following form:

$$P^{\varepsilon}(\varrho_1, \ldots, \varrho_{p-1}) = \sum_{i=1}^{p-1} \int_{\Omega} \frac{\varepsilon}{2} \, |\nabla \varrho_i|^2 \, d\mathbf{x} + \sum_{i=1}^{p-1} \int_{\Omega} \frac{1}{\varepsilon} \, W(\varrho_i) \, d\mathbf{x} \tag{17}$$

Therefore, the regularized form of (7) could be expressed as follows,

$$\min_{(\boldsymbol{\varrho}, \mathbf{u}) \in (\mathcal{A}, \mathcal{U})} \mathcal{J}^{\varepsilon}\big(\boldsymbol{\varrho}, \mathbf{u}(\boldsymbol{\varrho})\big) \quad \textit{subject to:} \quad \textit{equation} \ \ (1) \tag{18}$$

where $\zeta \geqslant 0$ is the penalization parameter and,

$$\mathcal{J}^{\varepsilon}\big(\boldsymbol{\varrho}, \mathbf{u}(\boldsymbol{\varrho})\big) := \mathcal{J}\big(\boldsymbol{\varrho}, \mathbf{u}(\boldsymbol{\varrho})\big) + \zeta P^{\varepsilon}(\boldsymbol{\varrho})$$

Prior to closing this section, it is worth to mention some of the recent progresses in regard to the topology optimization with perimeter penalization approach. In [57], the binary phase topology optimization has been considered with perimeter penalization. In this work, an explicit inequality bound has been imposed on the total perimeter of the inter-phase boundaries. The solution algorithm was based on the lagrange multiplier approach. In [58], the similar problem has been considered, however, instead of using a double well function, like that of ours, a double obstacle function has been included in the corresponding functional. Furthermore, the optimization problem has been reformulated as a min-max optimization problem and has been solved by the Howard's algorithm. In [59], a new regularized perimeter penalization has been introduced to solve binary phase topology optimization problems by the topological asymptotic approach. In the traditional perimeter penalization approaches, when a small hole with radius $B > 0$ is introduced in the body, the total perimeter and compliance scale by factors $B^{d-1}$ and $B^d$ respectively. While in the method presented in [59], the regularized perimeter scales by factor $B^d$. According to [59], this behavior results in the better convergence of the numerical method.

### 2.3. Constraint qualification.

The constraint qualification is a common requirement for the success of optimization algorithms from both the theoretical and practical points of view. To obtain the qualification of constraints for linearly constrained optimization problems, it is sufficient to show that the equality constraints are linearly independent and that there exists a feasible point satisfying all inequalities strictly. However, the set of equality constraints in $\mathcal{A}$ are not linearly independent. Assuming that $\mathcal{A}$ is nonempty, to ensure the constraint qualification, we could remove $\varrho_p$ from the set of unknown vectors using equality constraints (5) ($\varrho_p = 1 - \sum_{i=1}^{p-1} \varrho_i$). Therefore, a new version of problem (7) satisfying the constraint qualification conditions has the following form:

$$\min_{(\boldsymbol{\varrho},\mathbf{u})\in(\mathcal{A}',\mathcal{U})} \mathcal{J}^{\varepsilon}\big(\boldsymbol{\varrho},\mathbf{u}(\boldsymbol{\varrho})\big) \quad \textit{subject to}: \quad \textit{equation}\ \ (1) \tag{19}$$

where the design vector $\boldsymbol{\varrho}$ is redefined as $\boldsymbol{\varrho} := \{\varrho_1, \dots \varrho_{p-1}\}$ and the feasible set $\mathcal{A}'$ is defined as follows:

$$\mathcal{A}' := \left\{ \boldsymbol{\varrho} \in \{\ \varrho_i \in \mathcal{V}(\Omega)\ \}_{i\in\{1,\dots,p-1\}} \ \middle|\ \begin{array}{ll} \int_\Omega \varrho_i \, d\mathbf{x} = \Lambda_i |\Omega|, & i = 1,\dots,p-1 \\ l_i \leqslant \varrho_i \leqslant h_i, & i = 1,\dots,p-1 \\ 1 - h_p \leqslant \sum_{i=1}^{p-1} \varrho_i \leqslant 1 - l_p, & \end{array} \right\}$$

## 3. Optimality conditions

The first order necessary optimality conditions for problem (19) will be presented in this section. Following [60], we shall express optimality conditions based on the projected gradient approach. For the sake of reader convenience, we shall recall some known results related to the projected gradient approach in the next subsection. Then, we shall present the first order necessary optimality conditions in subsection 3.3 accordingly.

### 3.1. Optimality conditions based on the projected gradient approach.

**Theorem 3.1.** *(orthogonal projection over a convex set, theorem 12.1.10 of [61]) Let $\mathcal{W}$ be a Hilbert space and $\mathcal{K}$ as a convex closed nonempty subset of $\mathcal{W}$. For all $w \in \mathcal{W}$ , there exists a unique $w_{\mathcal{K}} \in \mathcal{K}$ such that*

$$\|w - w_{\mathcal{K}}\|_2^2 = \arg\min_{v\in\mathcal{K}} \|w - v\|_2^2.$$

*The orthogonal projection of $w$ onto set $\mathcal{K}$ is shown by operator $\mathcal{P}_{\mathcal{K}}[w]$ henceforth in this paper, i.e., $w_{\mathcal{K}} = \mathcal{P}_{\mathcal{K}}[w]$. Equivalently, $w_{\mathcal{K}}$ is characterized by the following property:*

$$w_{\mathcal{K}} \in \mathcal{K}, \quad \langle w_{\mathcal{K}} - w,\ v - w_{\mathcal{K}}\rangle \geqslant 0, \quad \forall \eta \in \mathcal{K} \tag{20}$$

*where $\langle \cdot,\cdot \rangle$ denotes the inner product on $V$.*

**Theorem 3.2.** *(Euler inequality for convex sets, theorem 10.2.1 of [61]) Let $\mathcal{W}$ be a Hilbert space and $\mathcal{K}$ as a convex closed nonempty subset of $\mathcal{W}$. Assume functional $J(\theta) : \mathcal{K} \to \mathbb{R}$ is differentiable at $w \in \mathcal{K}$ with the directional derivative denoted by $J'(w)$. If $w^*$ denotes a local minimum point of $J(w)$ over $\mathcal{K}$, then:*

$$\langle J'(w^*),\ v - w^*\rangle \geqslant 0, \quad \forall v \in \mathcal{K} \tag{21}$$

**Corollary 3.3.** *(necessary optimality conditions based on the projected gradient) Let $\mathcal{W}$ be a Hilbert space and $\mathcal{K}$ as a convex closed nonempty subset of $\mathcal{W}$. Assume the functional $J(w) : \mathcal{K} \to \mathbb{R}$ is differentiable at $w \in \mathcal{K}$ with the directional derivative denoted by $J'(w)$. If $w^*$ denotes a local minimizer of $J(w)$ over $\mathcal{K}$, then:*

$$J'_{\mathcal{K},\mu}(w^*) = \mathcal{P}_{\mathcal{K}}[w^* - \mu J'] - w^* = 0 \quad \text{a.e.} \tag{22}$$

*where $\mu \in \mathbb{R}^+$. Since $\mathcal{P}_{\mathcal{K}}(w - \mu J') - w$ is equal to the scaled projected gradient of $J$ at $w$, constrained stationary points of $J$ are roots of the scaled projected gradient with respect to set $\mathcal{K}$. Therefore we call* (22) *the necessary optimality conditions based on the projected gradient.*

*Proof.* Considering an arbitrary $\mu \in \mathbb{R}^+$, by (21) we have:

$$\langle \mu J'(w^*),\ v - w^* \rangle \geqslant 0, \quad \forall \eta \in \mathcal{K}$$

Simple algebra results in:

$$\Big\langle w^* - \big(w^* - \mu J'(w^*)\big),\ v - w^* \Big\rangle \geqslant 0, \quad \forall v \in \mathcal{K} \tag{23}$$

Comparing (20) and (23) results in:

$$w^* = \mathcal{P}_{\mathcal{K}}\big(w^* - \mu J'(w^*)\big)$$

almost everywhere, which completes the proof. □

**Corollary 3.4.** *(descent property of the scaled projected gradient) Let $\mathcal{W}$ be a Hilbert space and $\mathcal{K}$ as a convex closed nonempty subset of $\mathcal{W}$. Assume functional $J(w) : \mathcal{K} \to \mathbb{R}$ is differentiable at $w \in \mathcal{K}$ with the directional derivative denoted by $J'(w)$. Assume that the scaled projected gradient at $w \in \mathcal{K}$ is denoted by $J'_{\mathcal{K},\mu}(w)$, i.e., $J'_{\mathcal{K},\mu}(w) = \mathcal{P}_{\mathcal{K}}[w - \mu J'] - w$. Then for all $w \in \mathcal{K}$ and $\mu \in \mathbb{R}^+$ we have:*

$$\langle J'(w),\ J'_{\mathcal{K},\mu}(w) \rangle \leqslant -\frac{1}{2\mu} \|J'_{\mathcal{K},\mu}(w)\|_2^2 \tag{24}$$

*Proof.* According to the definition of the projection operator we have:

$$\mathcal{P}_{\mathcal{K}}[w - \mu J'(w)] = \arg\min_{v \in \mathcal{K}} \|v - (w - \mu J'(w))\|_2^2$$

since $w, v \in \mathcal{K}$ we have,

$$\|v - (w - \mu J'(w))\|_2^2 \leqslant \|\mu J'(w)\|_2^2 \tag{25}$$

the expansion of left hand side of (25) results in,

$$2\mu \big\langle J'(w),\ v - w \big\rangle \leqslant -\|v - w\|_2^2$$

since $v$ is an arbitrary member of $\mathcal{K}$, we replace it by $\mathcal{P}_{\mathcal{K}}(w - \mu J'(w))$. Therefore,

$$2\mu \big\langle J'(w),\ \mathcal{P}_{\mathcal{K}}[w - \mu J'(w)] - w \big\rangle \leqslant -\|\mathcal{P}_{\mathcal{K}}[w - \mu J'(w)] - w\|_2^2$$

which completes the proof. □

Corollary 3.4 suggests a simple iterative method to find local minimums of convex constrained optimization problems, as follows:

$$w_{k+1} = w_k + \alpha_k J'_{\mathcal{K},\mu_k}(w_k), \quad k = 0, 1, \ldots, \tag{26}$$

where subscript $k$ denotes the iterations counter, $w_0$ denotes the initial guess and $\alpha_k \in (0, 1]$ is computed based on a globalization strategy, such as the line search or trust region methods (cf. [62]). If we look at $w_k$ and $w_{k+1}$ as the values of design vector in two consecutive (virtual) timesteps, and look at $\alpha_k$ as the time-step size, (26) will be equivalent to the time discretized form of the following parabolic ordinary differential equation,

$$\frac{dw}{dt} = J'_{\mathcal{K},\mu}, \quad w(t = 0) = w_0 \tag{27}$$

The equation (27) is in fact the constrained gradient flow of the objective functional $J$. For the sake of convenience, this form of update procedure will be used during the presentation of optimization algorithm in the current study (see section 4).

3.2. **First order adjoint sensitivity analysis.** To use the projected gradient approach discussed in subsection 3.1, we should compute the gradient of the objective functional $\mathscr{J}^{\varepsilon}$ with respect to the design vector $\boldsymbol{\varrho}$ with considering the PDE (state) constraint (1). This procedure is commonly called the adjoint sensitivity analysis in the computational engineering literature (cf. [1, 61, 63]). By straightforward derivation, the first variation (directional derivative) of the objective functional $\mathscr{J}^{\varepsilon}$ with respect to $\boldsymbol{\varrho}$, denoted by $\boldsymbol{J'} = \{J'_1, \dots, J'_{p-1}\}$ here, could be computed as follows:

$$\begin{aligned} J'_i(\mathbf{x}) \quad &= \quad \Big(\mathbf{C}_{\varrho_i}\big(\boldsymbol{\varrho}(\mathbf{x})\big) : \mathcal{D}\big(\mathbf{u}(\mathbf{x})\big)\Big) : \Big(\tfrac{1}{2}\mathcal{D}\big(\mathbf{u}(\mathbf{x})\big) - \mathcal{D}\big(\mathbf{v}(\mathbf{x})\big)\Big) \\ &- \quad \Big(\mathbf{C}_{\varrho_p}\big(\boldsymbol{\varrho}(\mathbf{x})\big) : \mathcal{D}\big(\mathbf{u}(\mathbf{x})\big)\Big) : \Big(\tfrac{1}{2}\mathcal{D}\big(\mathbf{u}(\mathbf{x})\big) - \mathcal{D}\big(\mathbf{v}(\mathbf{x})\big)\Big) \\ &- \quad \zeta\varepsilon\, \Delta\big(\varrho_i(\mathbf{x})\big) + \tfrac{\zeta}{\varepsilon}\, W'\big(\varrho_i(\mathbf{x})\big) \quad \texttt{in} \quad \Omega, \quad \texttt{for} \;\; i = 1, \dots, p-1, \end{aligned} \tag{28}$$

together with the following boundary conditions:

$$\nabla\varrho_i(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = 0 \;\; \texttt{on} \;\; \Gamma, \quad \texttt{for} \;\; i = 1, \dots p-1 \tag{29}$$

where $\mathbf{u}$ is solution of the forward PDE (1) for a given value of $\boldsymbol{\varrho}$, and $\mathbf{v}$ is the solution of the following adjoint PDE for given values of $\mathbf{u}$ and $\boldsymbol{\varrho}$:

$$\left\{ \begin{array}{rcll} \nabla\cdot(\mathbf{C} : \mathcal{D}(\mathbf{v})) - \nabla\cdot(\mathbf{C} : \mathcal{D}(\mathbf{u})) & = & 0 & \texttt{in} \;\; \Omega \\ \mathbf{v}(\mathbf{x}) & = & 0 & \texttt{on} \;\; \Gamma_u \\ (\mathbf{C} : \mathcal{D}(\mathbf{v}))\mathbf{n} & = & 0 & \texttt{on} \;\; \Gamma_f \\ (\mathbf{C} : \mathcal{D}(\mathbf{v}))\mathbf{n} - (\mathbf{C} : \mathcal{D}(\mathbf{u}))\mathbf{n} & = & 0 & \texttt{on} \;\; \Gamma_t \end{array} \right. \tag{30}$$

Moreover, $W'(r) = 4r^3 - 6r^2 + 2r$ and $\mathbf{C}_{\varrho_i}$ denotes the partial derivative of the elasticity tensor with respect to $\varrho_i$. The operator $\Delta$ in (28) denotes the classical Laplacian operator, i.e., $\Delta(\cdot) := \nabla\cdot(\nabla(\cdot))$. By differentiating from 2 and considering $\varrho_p = 1 - \sum_{i=1}^{p-1}\varrho_i$, we have:

$$\mathbf{C}_{\varrho_i}(\boldsymbol{\varrho}) = q\big(\varrho_i^{q-1}\mathbf{C}_i - \varrho_p^{q-1}\mathbf{C}_p\big), \quad \texttt{for} \;\; \texttt{i} = \texttt{1}, \dots, \texttt{p} - \texttt{1} \tag{31}$$

According to (30), when $\hat{\mathbf{u}}(\mathbf{x}) = 0$ in the forward PDE, i.e., we have homogeneous Dirichlet boundary contitions on $\Gamma_u$, the forward problem is self adjoint, i.e., $\mathbf{u}(\mathbf{x}) = \mathbf{v}(\mathbf{x})$. Therefore, in these cases there is no need to solve the adjoint PDE (30).

3.3. **First order necessary optimality conditions.** Considering results presented in subsections 3.1 and 3.2, the first order necessary optimality conditions based on the projected gradient approach could be expressed as follows:

$$\mathcal{P}_{\mathcal{A}'}[\boldsymbol{\varrho}^* - \boldsymbol{J'}] - \boldsymbol{\varrho}^* = 0 \;\; in \;\; \Omega, \quad subject\ to: \;\; (1) \;\; and \;\; (30), \tag{32}$$

where $\boldsymbol{J'} = \{J'_1, \dots, J'_{p-1}\}$, $\boldsymbol{\varrho}^* = \{\varrho^*_1, \dots, \varrho^*_{p-1}\}$ denotes a constrained stationary point (a point which satisfies the first order necessary optimality conditions) of the objective functional $\mathscr{J}^{\varepsilon}$ and $J'_i$ (for $i = 1, \dots, p-1$) is computed based on (28).

## 4. Regularized projected gradient flow minimization algorithm

Considering (26) and (32), the following iterative scheme appears to be a straightforward algorithm to solve the problem (19):

$$\boldsymbol{\varrho_{k+1}} = \boldsymbol{\varrho_k} + \alpha_k\Big(\mathcal{P}_{\mathcal{A}'}[\boldsymbol{\varrho_k} - \boldsymbol{J_k}'] - \boldsymbol{\varrho_k}\Big), \quad k = 0, 1, \dots, \tag{33}$$

where $\boldsymbol{\varrho_0} \in \mathcal{A}'$ denotes a feasible initial guess. Similar to (27), (33) could be written as the following gradient flow like form:

$$\frac{d\boldsymbol{\varrho}}{dt} = \mathcal{P}_{\mathcal{A}'}[\boldsymbol{\varrho} - \boldsymbol{J'}] - \boldsymbol{\varrho}, \quad \boldsymbol{\varrho}(t = 0) = \boldsymbol{\varrho_0} \tag{34}$$

However, (33) or (34) could lead to unsatisfactory results like checkerboard instability and slow convergence due to the missing of gradient regularity by the projection operator $\mathcal{P}_{\mathcal{A}'}$. More precisely, the projection of gradient with respect to $L^2$ metric results in a projected gradient which is a member of $L^2(\Omega)$ space. Therefore, by using formulae (33) (or (34)), the new design vector $\boldsymbol{\varrho_{k+1}}$ will be a member of $L^2(\Omega)$ space. However, an $L^2$ regularity of the solution is not essentially sufficient in topology optimization problems. For instance, is not amenable to checkerboard instability.

To avoid possible problems using the update formulae (34) (or (33)), we shall adapt the operator splitting approach to solve the problem (19). Our approach is inspired from [41], in which a different continuous optimization problem has been solved by the operator splitting approach (note that our operator splitting approach differs significantly from that of [41]). For this purpose, we decompose the objective functional into two parts as follows:

$$\mathcal{J}^{\varepsilon}\big(\boldsymbol{\varrho}, \mathbf{u}(\boldsymbol{\varrho})\big) = \mathcal{J}^{\varepsilon}_N\big(\boldsymbol{\varrho}, \mathbf{u}(\boldsymbol{\varrho})\big) + \mathcal{J}^{\varepsilon}_C\big(\boldsymbol{\varrho}\big)$$

where,

$$\mathcal{J}^{\varepsilon}_N\big(\boldsymbol{\varrho}, \mathbf{u}(\boldsymbol{\varrho})\big) := \mathcal{J}\big(\boldsymbol{\varrho}, \mathbf{u}(\boldsymbol{\varrho})\big) + \frac{\zeta}{\varepsilon} \sum_{i=1}^{p-1} \int_{\Omega} W(\varrho_i)\ d\mathbf{x}$$

$$\mathcal{J}^{\varepsilon}_C\big(\boldsymbol{\varrho}\big) := \frac{\zeta\varepsilon}{2} \sum_{i=1}^{p-1} \int_{\Omega} |\nabla \varrho_i|^2\ d\mathbf{x}$$

Similar to approach discussed in section 3, the first variation of $\mathcal{J}^{\varepsilon}_N$ and $\mathcal{J}^{\varepsilon}_C$, denoted by $\boldsymbol{J'}_N = \{J'_{N,1}, \ldots, J'_{N,p-1}\}$ and $\boldsymbol{J'}_C = \{J'_{C,1}, \ldots, J'_{C,p-1}\}$ respectively here, could be computed as follows:

$$\begin{aligned} J'_{N,i}(\mathbf{x}) \quad &= \quad \Big(\mathbf{C}_{\varrho_i}\big(\boldsymbol{\varrho}(\mathbf{x})\big) : \mathcal{D}\big(\mathbf{u}(\mathbf{x})\big)\Big) : \Big(\tfrac{1}{2}\mathcal{D}\big(\mathbf{u}(\mathbf{x})\big) - \mathcal{D}\big(\mathbf{v}(\mathbf{x})\big)\Big) \\ &- \quad \Big(\mathbf{C}_{\varrho_p}\big(\boldsymbol{\varrho}(\mathbf{x})\big) : \mathcal{D}\big(\mathbf{u}(\mathbf{x})\big)\Big) : \Big(\tfrac{1}{2}\mathcal{D}\big(\mathbf{u}(\mathbf{x})\big) - \mathcal{D}\big(\mathbf{v}(\mathbf{x})\big)\Big) \\ &+ \quad \tfrac{\zeta}{\varepsilon}\, W'\big(\varrho_i(\mathbf{x})\big) \;\; \texttt{in}\;\; \Omega, \;\; \texttt{for}\;\; i = 1, \ldots, p-1 \end{aligned} \tag{35}$$

where $\mathbf{u}$ and $\mathbf{v}$ are solutions of (1) and (30) respectively.

$$J'_{C,i}(\mathbf{x}) = -\zeta\varepsilon\ \Delta\varrho_i(\mathbf{x}) \;\; \texttt{in}\;\; \Omega, \;\; \texttt{for}\;\; i = 1, \ldots, p-1 \tag{36}$$

with boundary conditions identical to (29).

Now, we express our operator splitting approach in a gradient flow like language, i.e., a virtual time is considered as the continuous analog to iteration count. Assume that the current value of the design vector at $t = t_k$ is denoted by $\varrho_k$, then the update procedure to compute the new design vector at time $t_{k+1}$, denoted by $\varrho_{k+1}$, has the following two steps. First, we solve the following (discrete) gradient flow equation (gradient flow of objective function $\mathcal{J}^{\varepsilon}_N$) in time windows $t \in (t_k, t_{k+1}]$ with boundary conditions (29):

$$\tilde{\boldsymbol{\varrho_k}} = \boldsymbol{\varrho_k} + \alpha_k\Big(\mathcal{P}_{\mathcal{A}'}[\boldsymbol{\varrho_k} - \boldsymbol{J'}_{N\boldsymbol{k}}] - \boldsymbol{\varrho_k}\Big) \tag{37}$$

where the step size $\alpha_k$ is equivalent to the virtual time-step size, i.e., $\alpha_k = t_{k+1} - t_k$. Analogous to (34), (37) can be expressed in the following continuous format:

$$\frac{d\tilde{\boldsymbol{\varrho}}}{dt} = \mathcal{P}_{\mathcal{A}'}[\tilde{\boldsymbol{\varrho}} - \boldsymbol{J'}_N] - \tilde{\boldsymbol{\varrho}}, \quad \tilde{\boldsymbol{\varrho}}(t_k) = \boldsymbol{\varrho_k}, \quad t_k < t \leqslant t_{k+1} \tag{38}$$

Then, we solve the following gradient flow equation (gradient flow of objective function $\mathcal{J}^{\varepsilon}_C$) in time windows $t \in (t_k, t_{k+1}]$ with boundary conditions (29):

$$\frac{d\boldsymbol{\varrho}}{dt} = -\boldsymbol{J'}_C, \quad \boldsymbol{\varrho}(t_k) = \tilde{\boldsymbol{\varrho_{k+1}}}, \quad t_k < t \leqslant t_{k+1} \tag{39}$$

Note that the solution of (39) is equivalent to the solution of $p-1$ decoupled linear heat equations with homogeneous Neumann boundary conditions, as follows:

$$\frac{d\boldsymbol{\varrho}}{dt} = \zeta\varepsilon\ \boldsymbol{\Delta}\boldsymbol{\varrho}, \quad \boldsymbol{\varrho}(t_k) = \tilde{\boldsymbol{\varrho}_{\boldsymbol{k+1}}}, \quad t_k < t \leqslant t_{k+1} \tag{40}$$

The time discretization of (40) by the Euler implicit approach with the step size $\alpha_k = t_{k+1} - t_k$ results in:

$$\boldsymbol{\varrho_{k+1}} = \tilde{\boldsymbol{\varrho_k}} + \alpha_k\zeta\varepsilon\ \boldsymbol{\Delta}\boldsymbol{\varrho_{k+1}} \tag{41}$$

Rearranging term in (41) results in the following Helmholtz equation with homogeneous boundary condition:

$$\boldsymbol{\varrho_{k+1}} = \big(\mathbf{I} - \alpha_k\zeta\varepsilon\ \boldsymbol{\Delta}\big)^{-1}\tilde{\boldsymbol{\varrho_k}} \tag{42}$$

where $'\mathbf{I}'$ denotes the (vectorized) identity operator. Interestingly, equation (42) is similar to the PDE used for sensitivity filtering based on the Helmholtz equation (cf. [64]). However, the design vector is filtered here instead of the gradient vector. In fact, the second step in our operator splitting approach acts as the lifting of the design vector from $L^2(\Omega)$ to $H^1(\Omega)$, as it is illustrated in the following proposition:

**Proposition 4.1.** *The second step, the regularization step, of the introduced operator splitting approach, i.e., the solution of* (42)*, is equivalent to the transition of the design vector* $\tilde{\boldsymbol{\varrho_k}}$ *from* $L^2(\Omega)$ *to the more regular function space* $H^1(\Omega)$ *occupied with the inner product* $(f,g)_{H^1(\Omega) := \int_\Omega (fg + \vartheta\nabla f\cdot\nabla g)\ d\mathbf{x}}$ *for* $f, g \in H^1(\Omega)$*, where* $\vartheta \in \mathbb{R}^+$.

*Proof.* Because the projected gradient approach is used at the first step of the algorithm and the projection is performed with respect to the $L^2(\Omega)$ metric, $\tilde{\boldsymbol{\varrho_k}} \in L^2(\Omega)$. From (42), for every $\boldsymbol{\psi} \in H^1(\Omega)$, we have:

$$\int_\Omega (\boldsymbol{\varrho_{k+1}} - \vartheta\ \boldsymbol{\Delta}\boldsymbol{\varrho_{k+1}})\ \boldsymbol{\psi}\ d\mathbf{x} = \int_\Omega \tilde{\boldsymbol{\varrho_k}}\ \boldsymbol{\psi}\ d\mathbf{x}$$

where $\vartheta = \alpha_k\zeta\varepsilon$. Using the divergence theorem and the homogeneous Neumann boundary conditions results in:

$$\int_\Omega (\boldsymbol{\varrho_{k+1}}\ \boldsymbol{\psi} + \vartheta\ \boldsymbol{\nabla}\boldsymbol{\varrho_{k+1}}\cdot\boldsymbol{\nabla}\boldsymbol{\psi})\ d\mathbf{x} = \int_\Omega \tilde{\boldsymbol{\varrho_k}}\ \boldsymbol{\psi}\ d\mathbf{x}$$

i.e., $(\boldsymbol{\varrho_{k+1}}, \boldsymbol{\psi})_{H^1(\Omega)} = (\tilde{\boldsymbol{\varrho_k}}, \boldsymbol{\psi})_{L^2(\Omega)}$, which completes the proof. □

Therefore, our solution strategy could be considered as a constrained $H^1$ topology optimization algorithm. It is interesting to note that this regularization procedure will not change the feasibility of the solution with respect to the design space $\mathcal{A}'$. In the other words, the solution of (42) will lie inside the design space, i.e., $\boldsymbol{\varrho_{k+1}} \in \mathcal{A}'$. This property will be proved in the following proposition. It is important to note that an explicit solution of (40) (e.g., using Euler explicit time discretization) does not ensure the $H^1$ regularity of the solution.

**Proposition 4.2.** *By starting from an initial guess* $\boldsymbol{\varrho_0} \in \mathcal{A}'$*, if the (time) step size,* $\alpha_k$*, is selected in interval* $(0,1]$*, the update procedure based on time discretized form of* (38) *and* (39) *will generate iterations which are strictly feasible with respect to* $\mathcal{A}'$.

*Proof.* Considering the properties of projected gradient method, mentioned in section 3, every solution of (38) lies in $\mathcal{A}'$. Therefore, we should prove that every solution of (39) lies inside $\mathcal{A}'$. Because the linear heat equation follows the maximum principles and the initial value lies inside $\mathcal{A}'$, the solution of (39) will satisfy the pointwise bound constraints. Let us define $\varphi$ as $\varphi := \sum_{i=1}^{p-1}\varrho_i$. Summing over all $p-1$ heat equations in (39) results in:

$$\frac{d\varphi}{dt} = \zeta\varepsilon\ \Delta\varphi, \quad \varphi(t_k) = \varphi_k, \quad t_k < t \leqslant t_{k+1}$$

Because $1 - h_p \leqslant \varphi_k \leqslant 1 - l_p$, the maximum principles again ensure the feasibility of (39) solutions with respect to inequality constraints $1 - h_p \leqslant \sum_{i=1}^{p-1}\varrho_i \leqslant 1 - l_p$. Finally, if we integrate the heat equations over $\Omega$, we will have:

$$\int_\Omega \frac{d\varrho_i}{dt}\, d\mathbf{x} = \zeta\varepsilon \int_\Omega \Delta\varrho_i \, d\mathbf{x}, \quad \varrho_i(t_k) = \varrho_{i,k}, \quad t_k < t \leqslant t_{k+1}$$

for $i = 1, \ldots, p-1$. By applying the divergence theorem and using boundary conditions (29), we have:

$$\frac{d}{dt}\int_\Omega \varrho_i \, d\mathbf{x} = \zeta\varepsilon \int_\Gamma \nabla\varrho_i \cdot \mathbf{n}\, d\mathbf{x} = 0, \quad \varrho_i(t_k) = \varrho_{i,k}, \quad t_k < t \leqslant t_{k+1}$$

Because $\int_\Omega \varrho_{i,k}\; d\mathbf{x} = \Lambda_i|\Omega|$, every solution of (39) satisfies costraints $\int_\Omega \varrho_i \; d\mathbf{x} = \Lambda_i|\Omega|$ (for $i = 1, \ldots, p-1$). $\square$

According to the proposition 4.2, the generated design vectors by iterations of the presented algorithm in this study not only satisfy the pointwise bound constraints, but also their total volumes are invariant during the iterations. Therefore, the two-step gradient flow approach introduced in the present study is exactly volume preserving, similar to the Cahn-Hilliard based gradient flow. The statement of our optimization algorithm at continuous level is as follows:

**Algorithm 1. Regularized projected steepest descent**

**Step 0.** Initialization

0.1 Given the model problem parameters as follows: the spatial domain $\Omega$ and its boundaries $\Gamma_u$, $\Gamma_f$, $\Gamma_t$, boundary conditions $\hat{\mathbf{u}}$, $\hat{\mathbf{t}}$, volumetric source term $\mathbf{f}$, $p$, elasticity tenors $\mathbf{C}_i$ ($i = 1, \ldots, p$), SIMP penalization power $q$, volume fraction data $\Lambda_i$ ($i = 1, \ldots, p$), perimeter regularization parameter $\varepsilon$ and penalization factor $\zeta$.

0.2 Given initial guess $\boldsymbol{\varrho}_0$ and stopping criteria parameters. If $\boldsymbol{\varrho_0} \notin \mathcal{A}$ then $\boldsymbol{\varrho_0} = \mathcal{P}_{\mathcal{A}'}[\boldsymbol{\varrho_0}]$, set $k \to 0$.

**Step 1.** Iterations

1.1 For given $\boldsymbol{\varrho_k}$, solve the forward PDE (1) for $\mathbf{u}_k$, and then solve the adjoint PDE (30) for $\mathbf{v}_k$. Since the homogeneous Dirichlet boundary conditions will be used in the present study, as it is mentioned in section 3.2, the forward PDE will be self-adjoint. Therefore, $\mathbf{u}(\mathbf{x}) = \mathbf{v}(\mathbf{x})$, and it is not necessary to solve the adjoint PDE in this step.

1.2 For given $\mathbf{u}_k$ and $\mathbf{v}_k$, compute $\boldsymbol{J}'_{N\boldsymbol{k}}$ by (35). Then compute $\tilde{\boldsymbol{\varrho}}_{\boldsymbol{k}}$ by the following equation:

$$\tilde{\boldsymbol{\varrho}}_{\boldsymbol{k}} = \boldsymbol{\varrho_k} + \alpha_k\Big(\mathcal{P}_{\mathcal{A}'}[\boldsymbol{\varrho_k} - \boldsymbol{J}'_{N\boldsymbol{k}}] - \boldsymbol{\varrho_k}\Big) \tag{43}$$

1.3 For given $\tilde{\boldsymbol{\varrho}}_{\boldsymbol{k}}$, compute the new design vector $\boldsymbol{\varrho_{k+1}}$ by the following equation with homogeneous Neumann boundary condition:

$$\boldsymbol{\varrho_{k+1}} = \big(\mathbf{I} - \alpha_k\zeta\varepsilon\, \boldsymbol{\Delta}\big)^{-1}\tilde{\boldsymbol{\varrho}}_{\boldsymbol{k}} \tag{44}$$

**Step 2.** Stopping criteria

2.1 If the solution does not meet the stopping criteria set $k \to k+1$ and goto step 1, else return $\boldsymbol{\varrho_{k+1}}$ as the optimal solution.

The step size $\alpha_k$ in algorithm 1 should be computed based on a globalization strategy to ensure the global convergence to a local solution. In the present study however $\alpha_k = 0.5$ is used in all of our numerical experiments. The infinity norm of the projected gradient is commonly used in the mathematical programming literature as the stopping criteria in the projected gradient algorithm (cf. [65]). However, because our algorithm has two steps this stopping criteria may not work well in practice. On the other hand, the distance between two consecutive design vectors under the infinity norm, i.e., $\|\boldsymbol{\varrho}_{\boldsymbol{k+1}} - \boldsymbol{\varrho}_{\boldsymbol{k}}\|_\infty$, is commonly used in the topology optimization literature as a convergence criteria. However, our numerical experiments illustrate that this distance does not essentially vary monotonically as iterations proceed. In fact it is not a monotonically decreasing measure. Therefore, it is not a reliable termination threshold in the present study. We believe that this behavior has two sources, the two step nature of our algorithm and bi-objective nature of our functional. To drive a reasonable termination criterion, it is required to establish the convergence theory of the presented algorithm (and possibly modify it accordingly). Because, this issue is beyond the scop of this paper, we simply limit the number of iterations to the upper bound value $k_{max}$, and terminate iterations when $k > k_{max}$. In the numerical experiments presented in this study $k_{max} = 1000$ which is selected based on our numerical experiments (according to our observation, all topologies reach to a sufficient maturity after 1000 iterations).

Because the step size $\alpha_k$ acts as a virtual time-step size in our computational algorithm, similar to [43], using the adaptive time-stepping method [66] can reduce the number of iterations to find desired solutions.

## 5. The spatial discretization

Without loss of generality, it is assumed that the spatial domain $\Omega$ is a rectangle with edges of $nx$ and $ny$ units along $x$ and $y$ axis respectively. For the purpose of numerical solution, $\Omega$ is discretized into a uniform $nx \times ny$ grid with square shaped elements. The state variables, $\mathbf{u}$, $\mathbf{v}$, are defined at the vertices of computational cells and the design vector, $\boldsymbol{\varrho} = \{\varrho_1, \dots, \varrho_{p-1}\}$, is defined at the cell centers. Let us use superscript $(\cdot)^h$ to denote the finite dimensional counterpart of each field variable. Therefore, after the spatial discretization $\mathbf{u}^h, \mathbf{v}^h \in \mathbb{R}^{n1}$ and $\boldsymbol{\varrho}^h \in \mathbb{R}^{n2}$, where $n1 = 2 \times (nx+1) \times (ny+1)$ and $n2 = n3\ (p-1)$, $n3 = nx \times ny$. The classical finite element method with Lagrangian elements is employed in the present study to solve the forward and adjoint PDEs. The displacement fields $\mathbf{u}^h$ and $\mathbf{v}^h$ are interpolated on P1 elements (bilinear interpolation on squared elements). Every volume fraction field is interpolated on P0 finite elements (piecewise constant interpolation).

To solve (44) by the classical lagrangian based finite element method, it is required to have $\tilde{\boldsymbol{\varrho}_{\boldsymbol{k}}}$ on P1 finite elements. On the other hand, as it is mentioned above, our density field is represented element-wise using P0 finite elements. The method described in section 4.2 of [67] is used in the present study to solve (44). In this method, the density field is firstly interpolated from elements centers to nodal points by the linear operator $\mathbf{T}_F$, i.e. the representation of $\tilde{\boldsymbol{\varrho}_{\boldsymbol{k}}}$ on P1 finite elements. Then, (44) is solved by the classical finite element method. Finally, the solution of (44) is interpolated back to the elements centers by the operator $\mathbf{T}_F^T$. Interested readers are referred to [67] for further details.

## 6. Projection onto the design space

After the spatial discretization, the feasible domain $\mathcal{A}'^h$ will has the following discrete structure,

$$\mathcal{A}'^h = \left\{ \boldsymbol{\varrho}^h \in \{\ \varrho_i^h \in \mathbb{R}^{n3}\ \}_{i\in\{1,\dots,p-1\}} \ \middle|\ \begin{array}{ll} 1^h \cdot \varrho_i^h = s_i, & i = 1, \dots, p-1 \\ l_i^h \leqslant \varrho_i^h \leqslant h_i^h, & i = 1, \dots, p-1 \\ 1 - h_p^h \leqslant \sum_{i=1}^{p-1} \varrho_i^h \leqslant 1 - l_p^h, & \end{array} \right\}$$

where $s_i = n3 \times \Lambda_i$. The projection of a trail point $\boldsymbol{\xi}^h = \{\xi_1^h, \dots, \xi_{p-1}^h\}$ onto $\mathcal{A}'^h$ is equivalent to the solution of the following quadratic programming (QP) problem:

$$\min_{\boldsymbol{\varrho}^h \in \mathcal{A}'^h} \sum_{i=1}^{p-1} \frac{1}{2} \|\varrho_i^h - \xi_i^h\|_2^2 \tag{45}$$

Because (45) is strictly convex, when ${\mathcal{A}'}^h \neq \emptyset$, (45) has a unique solution. Therefore, one could use a general purpose QP solver to find the unique solution of (45). However, when $n3$ is sufficiently large, as it is the case in our study, the solution of a QP problem is very expensive (in general) in terms of the computational cost and computer memory. Roughly speaking, its computational cost is an order of magnitude higher than the cost of solving a discretized elasticity problem. Several customized solvers have been developed in the literature for the efficient solution of (45) when $p = 2$. Some instances are: breakpoints searching [68, 69], variable fixing [70], bracketing [71] and bisection [60] algorithms. However, to the best of our knowledge, there is no archived work on the development of a customized solver for the efficient solution of (45), but [72]. In [72], three practically time-linear algorithms have been introduced by the current author to solve QP problems with structures similar to that of (45). It has been shown in [72] that the computational cost of these algorithms scales linearly by the number of design parameters and/or number of phases. The third algorithm of [72] will be used in the present study to solve (45). In this section, we briefly describe the outline of this algorithm and refer interested readers to [72] for further details. To reduce the complexity of the notation, we remove the superscript $\square^h$ in the remainder of this section.

Consider the decomposition of $\mathcal{A}'$ into the following sets:

$$\mathcal{A}_1 := \left\{ \boldsymbol{\varrho} \in \{ \ \varrho_i \in \mathbb{R}^{n3} \ \}_{i\in\{1,\dots,p-1\}} \ \middle| \ 1 - h_p \leqslant \textstyle\sum_{i=1}^{p-1} \varrho_i \leqslant 1 - l_p \right\}$$

$$\mathcal{A}_2 := \left\{ \boldsymbol{\varrho} \in \{ \ \varrho_i \in \mathbb{R}^{n3} \ \}_{i\in\{1,\dots,p-1\}} \ \middle| \ \mathbf{1} \cdot \varrho_i = s_i, \quad i = 1, \dots, p-1 \right\}$$

$$\mathcal{A}_3 := \left\{ \boldsymbol{\varrho} \in \{ \ \varrho_i \in \mathbb{R}^{n3} \ \}_{i\in\{1,\dots,p-1\}} \ \middle| \ l_i \leqslant \varrho_i \leqslant h_i, \quad i = 1, \dots, p-1 \right\}$$

$$\mathcal{A}_{12} := \mathcal{A}_1 \cap \mathcal{A}_2$$

Obviously $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_{12}$ are convex and $\mathcal{A}' = \mathcal{A}_{12} \cap \mathcal{A}_3$. The third algorithm of [72] can be expressed as follows:

**Algorithm 2 (Alternating projection method):**

**Step 0.** Initialization: Given $\delta > 0$, $n_{max}$, $\mathcal{A}_{12}$, $\mathcal{A}_3$, and $\boldsymbol{\varrho_0}$. Set $n \to 0$.

**Step 1.** Iterations

1.1. $\hat{\boldsymbol{\varrho_n}} = \mathcal{P}_{\mathcal{A}_{12}}[\boldsymbol{\varrho_n}]$, $\quad \boldsymbol{\varrho_{n+1}} = \mathcal{P}_{\mathcal{A}_3}[\hat{\boldsymbol{\varrho_n}}]$, $\quad n \to n+1$.

1.2. If $\|\boldsymbol{\varrho_{n+1}} - \boldsymbol{\varrho_n}\|_\infty \leqslant \delta$ or $n \geqslant n_{max}$ stop and return $\boldsymbol{\varrho_{n+1}}$ else goto step 1.

where $\delta$, $n_{max}$ and $\boldsymbol{\varrho_0}$ denote the convergence threshold, the maximum number of iterations and the initial guess (starting point) respectively.

**Proposition 6.1.** *(Convergence theory of algorithm 2) Algorithms 2 is well defined and when $\mathcal{A} \neq \emptyset$, by starting from an arbitrary initial guess, it converges to the unique minimizer of problem* (45)*.*

*Proof.* The proof is directly followed from the convergence theory of alternating projection algorithm, see [73, 74]. □

Obviously, a key for the efficiency of algorithm 2 is the existence of efficient methods for the projection of trial points onto sets $\mathcal{A}_{12}$ and $\mathcal{A}_3$. The projection of a trail point $\boldsymbol{\xi} = \{\xi_1, \dots, \xi_{p-1}\}$ onto $\mathcal{A}_3^h$ leads to solution of a convex separable optimization problem, which has the following explicit solution [72]:

$$\mathcal{P}_{\mathcal{A}_3}[\boldsymbol{\xi}] = (\xi_1^*, \dots, \xi_{p-1}^*), \quad \xi_j^* = \texttt{mid}\ (l_j,\ \xi_j^*,\ h_j)$$

where $\texttt{mid}(\cdots)$ denotes the classical median operator, i.e., $\texttt{mid}\ (a,\ b,\ c) := \max\ \big(a,\ \min\ (c,\ b)\big)$, and is understood componentwise here.

The problem of projection onto $\mathcal{A}_{12}$ is a nonseparable convex optimization problem. However, as it will be shown here, we shall compute its semi-implicit solution in expense of solving a $(p-1)\times(p-1)$ system of nonlinear equations. Because $p \ll n3$, the computational cost of solving this system of equations is negligible in contrast to the overall computational cost of the solution algorithm.

Assuming $\mathcal{A}_{12} \neq \emptyset$, finding the projection of trial point $\boldsymbol{\varrho} = (\varrho_1, \ldots, \varrho_{m-1}) \in \mathbb{R}^{n2}$ onto $\mathcal{A}_{12}$, denoted by $\boldsymbol{\xi}^* = (\xi_1^*, \ldots, \xi_{m-1}^*) \in \mathbb{R}^{n2}$, is equivalent to find the unique stationary point, denoted by $(\boldsymbol{\xi}^*, \lambda_l^*, \lambda_u^*, \mu^*) \in (\mathbb{R}^{n2}, \mathbb{R}^{n3}, \mathbb{R}^{n3}, \mathbb{R}^{p-1})$, of the following augmented lagrangian,

$$\mathcal{L}(\boldsymbol{\xi}, \lambda_l, \lambda_u, \mu) = \frac{1}{2}\sum_{i=1}^{p-1}(\xi_i\cdot\xi_i - \varrho_i\cdot\xi_i) - \lambda_u\cdot\big(1 - l_p - \sum_{i=1}^{p-1}\xi_i\big) + \lambda_l\cdot\big(1 - h_p - \sum_{i=1}^{p-1}\xi_i\big) + \sum_{i=1}^{p-1}\big(\mu_i\,(1\cdot\xi_i - s_i)\big)$$

where $\lambda_u, \lambda_l, 1 \in \mathbb{R}^{n3}$, $\mu \in \mathbb{R}^{p-1}$ are lagrange multipliers corresponding to inequalities and resource constraints. Using the first order KKT necessary optimality conditions results in:

$$\xi_i^* - \varrho_i + \lambda_u^* + \lambda_l^* + \mu_i^* 1 = 0, \quad i = 1, \ldots, p-1 \tag{46}$$

$$1\cdot\xi_i^* - s_i = 0, \quad i = 1, \ldots, p-1 \tag{47}$$

$$\lambda_u^* \geqslant 0, \quad \lambda_u^*\cdot\big(1 - l_p - \sum_{i=1}^{p-1}\xi_i^*\big) = 0, \tag{48}$$

$$\lambda_l^* \geqslant 0, \quad \lambda_l^*\cdot\big(1 - h_p - \sum_{i=1}^{p-1}\xi_i^*\big) = 0, \tag{49}$$

$$1 - l_p - \sum_{i=1}^{p-1}\xi_i^* \geqslant 0, \tag{50}$$

$$1 - h_p - \sum_{i=1}^{p-1}\xi_i^* \leqslant 0, \tag{51}$$

Computing $\xi_i^*$ from (46) and substituting it into (48) and (49), followed by simple algebra, results in:

$$\lambda_u^* + \lambda_l^* = \frac{1}{p-1}\Big[\max\Big(\sum_{i=1}^{p-1}\varrho_i - 1 + l_p - \sum_{i=1}^{p-1}\mu_i^*, 0\Big) \;+\; \min\Big(\sum_{i=1}^{p-1}\varrho_i - 1 + h_p - \sum_{i=1}^{p-1}\mu_i^*, 0\Big)\Big] \tag{52}$$

and so,

$$\xi_j^* = \varrho_j - \mu_j^* - \frac{1}{p-1}\Big[\max\Big(\sum_{i=1}^{p-1}\varrho_i - 1 + l_p - \sum_{i=1}^{p-1}\mu_i^*, 0\Big) \;+\; \min\Big(\sum_{i=1}^{p-1}\varrho_i - 1 + h_p - \sum_{i=1}^{p-1}\mu_i^*, 0\Big)\Big] \tag{53}$$

for $j = 1, \ldots, p-1$. Substituting the above relation into (47) leads to the following $(p-1)\times(p-1)$ system of equations:

$$\mu_j^*(1\cdot 1) + \Big(\frac{1}{p-1}\Big)\cdot\Big[\max\Big(\sum_{i=1}^{p-1}\varrho_i - 1 + l_p - \sum_{i=1}^{p-1}\mu_i^*, 0\Big) + \min\Big(\sum_{i=1}^{p-1}\varrho_i - 1 + h_p - \sum_{i=1}^{p-1}\mu_i^*, 0\Big)\Big] = 1\cdot\varrho_j - s_j \tag{54}$$

for $j = 1, \ldots, p-1$. Obviously, there is a unique solution for the above system of nonlinear equations. Although it is possible to solve it by modern solvers, e.g. semi-smooth Newton, the Picard linearization approach is used in the present study.

## 7. Computer implementation

The computer code used in the present study was written by modifying the 82-line MATLAB code introduced in [67] to solve the binary phase topology optimization problems. For the sake of reader convenience and ease of further extension, the complete listing of MATLAB code used in the present study is included in the appendix A. In the remainder of this section, we shall briefly review the structure of our code. Interested readers are referred to [23, 67] for further details.

The presented MATLAB code in appendix A possesses 95 lines of MATLAB scripts. The main function is "`topopt_multi`". The input parameters are defined in lines 3-5. They are follows. `nx` and `ny`: number of elements along $x$ and $y$ directions respectively; `q`: SIMP penalization power; `p`: number of phases; `epsi`: regularization parameter; `alpha`: virtual time-step size; `zeta`: penalization parameter corresponding to the perimeter regularization; `maxit`: maximum number of optimization iterations; *nu*: poisson ration; `e`: vector of Young's modulus of all phases; `v`: vector of overall volume fractions of all phases. The design vector is initialized to a uniform field based on the overall volume fraction of each phase in line 7. Lines 9-18 compute the element's stiffness matrix and node numbering for efficient global stiffness matrix assembly (cf. section 3.1 of [67]). Lines 20-31 include finite element implementation of the Helmholtz equation solver to solve (42) (cf. section 4.2 of [67]). Lines 33 and 34 include the implementation of boundary conditions. The specific implementation of the boundary conditions in the code is related to the cantilever beam test case in the present study, cf. [23, 67] to implement other boundary conditions. Lines 39-51 includes the main optimization loop. The objective function and its gradient are computed by function "`eval_fg`" in line 40. Line 41 performs the first step of our optimization approach, i.e., solves (43), and lines 42-44 perform the second step, regularization step, of our optimization algorithm, i.e., solve (44). Line 49 visualizes the outcome of optimization after every 10 iterations by making a bitmap image based on the distribution of phases. Function "`proj`" performs the orthogonal projection of trial vectors on to the design space, $\mathcal{A}'$, based on algorithm 3. Lines 81-87 in this function solve (54).

## 8. Numerical results and discussion

The success and performance of the presented algorithm will be studied in this section by means of numerical experiments. A personal computer with an AMD 2.4 GHz and 2.5 GB DDR2 RAM is used as the computational resource in the present work. Two minimum compliance topology optimization problems, named as MBB and Cantilever beams, are considered here to evaluate the presented algorithm. Geometries and boundary conditions corresponding to these benchmarks are shown in figure 2. The load `F` in figure 2 is equal to unity in all of our test cases here. To save the computational cost, the symmetry of problem is taken into account in the case of MBB beam (results however will be presented for the full domain in this case). Following [8, 10, 67], the poisson ratio of all phases is assumed to be equal, 0.3 here. Therefore, the elasticity tensor of each phase could be written as $E\mathbf{C}_0$, where $E$ denotes the Young's modulus and $\mathbf{C}_0$ denotes the elasticity tensor of a material with the same poisson ratio and unit Young's modulus. The computational domain is divided into a uniform $96 \times 48$ grid with unit edge length (for the case of MBB beam the computational domain here is equal to one half of the physical domain by exploiting the domain symmetry). The penalization factor, $q$, and the regularization parameter, $\varepsilon$, are respectively equal to 3 and 1 in our numerical experiments in this section. In fact $\varepsilon$ is equal to the mesh size in our numerical experiment. Note that smaller values for $\varepsilon$ is not recommended in the related literature.

Table 1 shows the simulation data corresponding to our test cases in this section. Note that test cases 1-12 and 25-32 are identical to references [8, 10]. It makes it possible for reader to compare our resulted topologies to those of [8, 10]. In all cases the optimization is terminated after 1000 iterations. A feasible design with the uniform material distribution is considered as the initial design in all test cases here, i.e., for all $\mathbf{x} \in \Omega$ we have $\varrho_i^0(\mathbf{x}) = \Lambda_i$ for $i = 1, \ldots, p$. Following [10], a bitmap image will be made to show multi-phase topologies. In all cases the background phase (the weakest phase) will be shown by white color and other phases will be shown by black, red, blue, green, yellow, pink and turquoise colors respectively based on $E$ order. For instance when $p = 4$, we shall use black, red, blue and white colors.

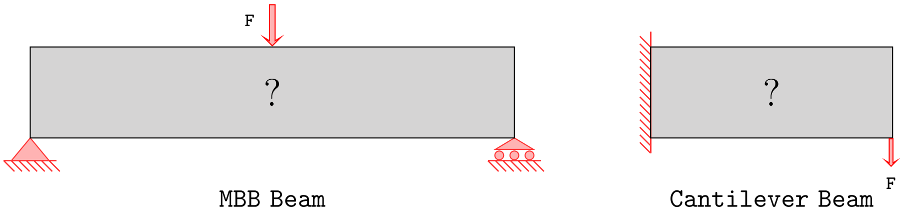


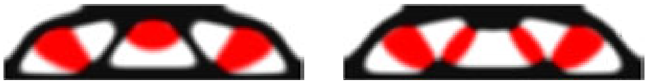
FIGURE 2. Geometry and boundary conditions corresponding to MBB and Cantilever beams problems in the present study.

Figures 3 and 4 show the final topologies (after 1000 iterations) corresponding to test cases 1-48 in the present study. They illustrate that the presented algorithm was successful to generate near 0-1 topologies without regard to the number of phases. The production of near 0-1 topologies reduces the effect of applied material interpolation scheme on the validity/feasibility of the final design. According to the plots, by increasing the perimeter penalization parameter $\zeta$, the minimum length scale in the final topology is increased. Moreover, the total perimeter in the final design decreases by increasing of $\zeta$. In the test cases #11 and #12, individual (non-load bearing) droplets are formed at the final topologies. Therefore, $\zeta$ should be selected carefully to avoid undesirable results (for instance by numerical experiments). A clue in this regard is to check the value of parameter $\frac{\mathcal{J}}{\zeta P^{\varepsilon}}$ during the optimization. If this parameter is too small, $\zeta$ should be decreased. On the other hand, $\zeta$ should be increased when it is too large. To study the dynamical behavior of the presented algorithm, the evolution of topologies during the optimization cycles (pseudo time-stepping) for selective test cases #5, #24, #29 and #44 are shown in figures 5 and 6. According to the plots, during the first 100 iterations the phases are well separated and near 0-1 topologies are attained. Furthermore, the outline of structures and overall distribution of phases are almost determined within the first 100 iterations. Subsequent iterations are mainly related to the reduction of the total perimeter by the migration of inter-phase boundaries. Moreover, the rate topological evolution is decreased as iterations proceed, in particular the variations of topologies are negligible during the last 500 iterations in all cases.

According to figures 3 and 4, the final topologies are qualitatively comparable to the results reported in other phase-field based topology optimization methods (e.g. see: [8, 39]). Comparing our results to those of [8] shows that the evolution of topology by our regularized Allen-Cahn dynamics is at least an order of magnitude faster than that of the generalized Cahn-Hilliard method introduced in [8]. For instance, topologies in [8] did not reach to a sufficient maturity after 1000 iterations, and in some cases over $10^5$ iterations were used to approach the local equilibrium. Comparing our results to those of the generalized Allen-Cahn dynamics introduced in [39] led us to the similar conclusion; the faster convergence of the presented method. For instance, $3 \times 10^4$, $6 \times 10^3$ and $3 \times 10^5$ iterations were used to produce the results presented in figures 2, 3 and 6 of [39]. Moreover, topologies are reached to a relative maturity after about 1000 iterations in [39]. Comparison of our results to other available literatures that used Cahn-Hilliard based dynamics and the traditional time-stepping scheme led us to the similar conclusion.

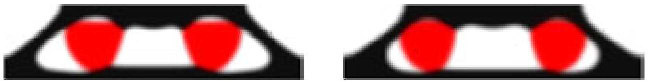
Table 1. Design parameters corresponding to test cases #1-#48 in the present study.

| ID of Test Case | p | $\zeta$ | Young's modulus | Volume Fraction |
|---|---|---|---|---|
| MBB Beam | | | | |
| #1 | 3 | 0.125 | [2 1 1e − 9] | [.4 .2 .4] |
| #2 | 3 | 0.25 | [2 1 1e − 9] | [.4 .2 .4] |
| #3 | 3 | 0.5 | [2 1 1e − 9] | [.4 .2 .4] |
| #4 | 3 | 1.0 | [2 1 1e − 9] | [.4 .2 .4] |
| #5 | 4 | 0.125 | [4 2 1 1e − 9] | [.2 .15 .15 .5] |
| #6 | 4 | 0.25 | [4 2 1 1e − 9] | [.2 .15 .15 .5] |
| #7 | 4 | 0.5 | [4 2 1 1e − 9] | [.2 .15 .15 .5] |
| #8 | 4 | 1.0 | [4 2 1 1e − 9] | [.2 .15 .15 .5] |
| #9 | 4 | 0.125 | [9 3 1 1e − 9] | [.16 .08 .08 .68] |
| #10 | 4 | 0.25 | [9 3 1 1e − 9] | [.16 .08 .08 .68] |
| #11 | 4 | 0.5 | [9 3 1 1e − 9] | [.16 .08 .08 .68] |
| #12 | 4 | 1.0 | [9 3 1 1e − 9] | [.16 .08 .08 .68] |
| #13 | 5 | 0.125 | [4 3 2 1 1e − 9] | [0.125 0.125 0.125 0.125 0.5] |
| #14 | 5 | 0.25 | [4 3 2 1 1e − 9] | [0.125 0.125 0.125 0.125 0.5] |
| #15 | 5 | 0.5 | [4 3 2 1 1e − 9] | [0.125 0.125 0.125 0.125 0.5] |
| #16 | 5 | 1.0 | [4 3 2 1 1e − 9] | [0.125 0.125 0.125 0.125 0.5] |
| #17 | 6 | 0.125 | [5 4 3 2 1 1e − 9] | [0.1 0.1 0.1 0.1 0.1 0.5] |
| #18 | 6 | 0.25 | [5 4 3 2 1 1e − 9] | [0.1 0.1 0.1 0.1 0.1 0.5] |
| #19 | 6 | 0.5 | [5 4 3 2 1 1e − 9] | [0.1 0.1 0.1 0.1 0.1 0.5] |
| #20 | 6 | 1.0 | [5 4 3 2 1 1e − 9] | [0.1 0.1 0.1 0.1 0.1 0.5] |
| #21 | 7 | 0.125 | [6 5 4 3 2 1 1e − 9] | [.5/6 .5/6 .5/6 .5/6 .5/6 .5/6 0.5] |
| #22 | 7 | 0.25 | [6 5 4 3 2 1 1e − 9] | [.5/6 .5/6 .5/6 .5/6 .5/6 .5/6 0.5] |
| #23 | 7 | 0.5 | [6 5 4 3 2 1 1e − 9] | [.5/6 .5/6 .5/6 .5/6 .5/6 .5/6 0.5] |
| #24 | 7 | 1.0 | [6 5 4 3 2 1 1e − 9] | [.5/6 .5/6 .5/6 .5/6 .5/6 .5/6 0.5] |
| Cantilever Beam | | | | |
| #25 | 3 | 0.125 | [2 1 1e − 9] | [0.4 0.2 0.4] |
| #26 | 3 | 0.25 | [2 1 1e − 9] | [0.4 0.2 0.4] |
| #27 | 3 | 0.5 | [2 1 1e − 9] | [0.4 0.2 0.4] |
| #28 | 3 | 1.0 | [2 1 1e − 9] | [0.4 0.2 0.4] |
| #29 | 4 | 0.125 | [4 2 1 1e − 9] | [0.2 0.1 0.1 0.6] |
| #30 | 4 | 0.25 | [4 2 1 1e − 9] | [0.2 0.1 0.1 0.6] |
| #31 | 4 | 0.5 | [4 2 1 1e − 9] | [0.2 0.1 0.1 0.6] |
| #32 | 4 | 1.0 | [4 2 1 1e − 9] | [0.2 0.1 0.1 0.6] |
| #33 | 5 | 0.125 | [4 3 2 1 1e − 9] | [0.125 0.125 0.125 0.125 0.5] |
| #34 | 5 | 0.25 | [4 3 2 1 1e − 9] | [0.125 0.125 0.125 0.125 0.5] |
| #35 | 5 | 0.5 | [4 3 2 1 1e − 9] | [0.125 0.125 0.125 0.125 0.5] |
| #36 | 5 | 1.0 | [4 3 2 1 1e − 9] | [0.125 0.125 0.125 0.125 0.5] |
| #37 | 6 | 0.125 | [5 4 3 2 1 1e − 9] | [0.1 0.1 0.1 0.1 0.1 0.5] |
| #38 | 6 | 0.25 | [5 4 3 2 1 1e − 9] | [0.1 0.1 0.1 0.1 0.1 0.5] |
| #39 | 6 | 0.5 | [5 4 3 2 1 1e − 9] | [0.1 0.1 0.1 0.1 0.1 0.5] |
| #40 | 6 | 1.0 | [5 4 3 2 1 1e − 9] | [0.1 0.1 0.1 0.1 0.1 0.5] |
| #41 | 7 | 0.125 | [6 5 4 3 2 1 1e − 9] | [.5/6 .5/6 .5/6 .5/6 .5/6 .5/6 0.5] |
| #42 | 7 | 0.25 | [6 5 4 3 2 1 1e − 9] | [.5/6 .5/6 .5/6 .5/6 .5/6 .5/6 0.5] |
| #43 | 7 | 0.5 | [6 5 4 3 2 1 1e − 9] | [.5/6 .5/6 .5/6 .5/6 .5/6 .5/6 0.5] |
| #44 | 7 | 1.0 | [6 5 4 3 2 1 1e − 9] | [.5/6 .5/6 .5/6 .5/6 .5/6 .5/6 0.5] |
| #45 | 8 | 0.125 | [7 6 5 4 3 2 1 1e − 9] | [.5/7 .5/7 .5/7 .5/7 .5/7 .5/7 .5/7 0.5] |
| #46 | 8 | 0.25 | [7 6 5 4 3 2 1 1e − 9] | [.5/7 .5/7 .5/7 .5/7 .5/7 .5/7 .5/7 0.5] |
| #47 | 8 | 0.5 | [7 6 5 4 3 2 1 1e − 9] | [.5/7 .5/7 .5/7 .5/7 .5/7 .5/7 .5/7 0.5] |
| #48 | 8 | 1.0 | [7 6 5 4 3 2 1 1e − 9] | [.5/7 .5/7 .5/7 .5/7 .5/7 .5/7 .5/7 0.5] |



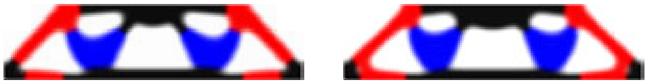
(a) #1 at iteration 1000

(b) #2 at iteration 1000



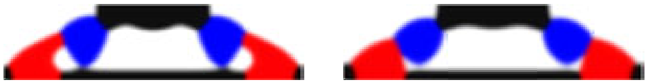
(c) #3 at iteration 1000

(d) #4 at iteration 1000



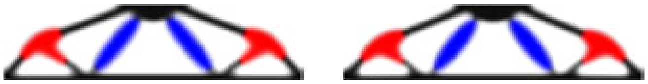
(e) #5 at iteration 1000

(f) #6 at iteration 1000



(g) #7 at iteration 1000

(h) #8 at iteration 1000

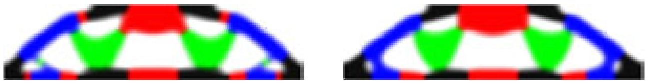


(i) #9 at iteration 1000

(j) #10 at iteration 1000

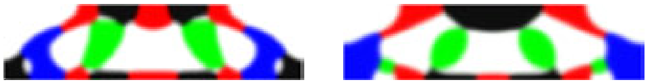
(k) #11 at iteration 1000

(l) #12 at iteration 1000



(m) #13 at iteration 1000

(n) #14 at iteration 1000



(o) #15 at iteration 1000

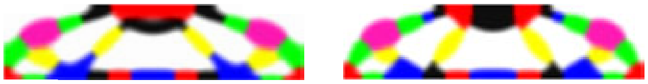
(p) #16 at iteration 1000

(q) #17 at iteration 1000

(r) #18 at iteration 1000

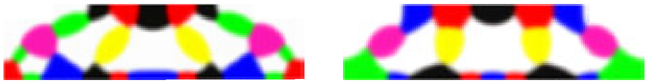
(s) #19 at iteration 1000

(t) #20 at iteration 1000



(u) #21 at iteration 1000

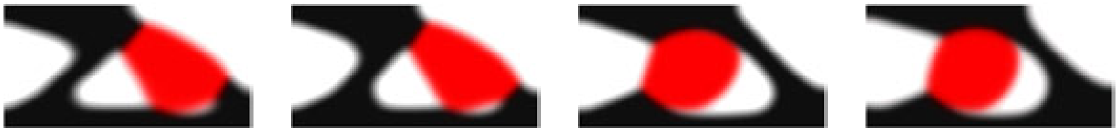
(v) #22 at iteration 1000



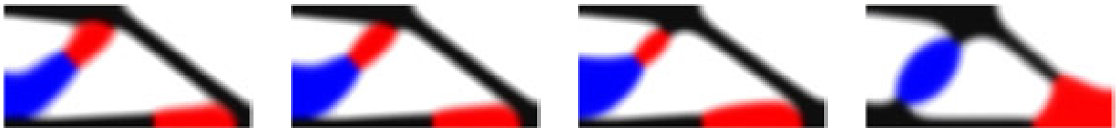
(w) #23 at iteration 1000

(x) #24 at iteration 1000

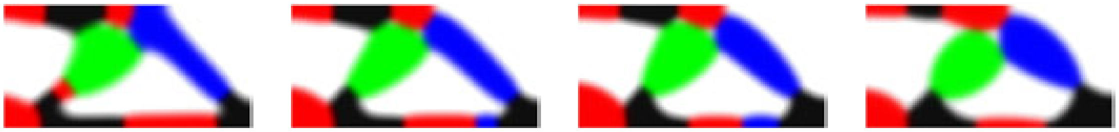
Figure 3. Final topologies (at iteration 1000) corresponding to test cases #1-#24.



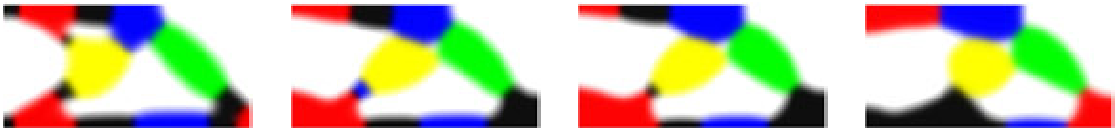
(a) #25 at iteration 1000 (b) #26 at iteration 1000 (c) #27 at iteration 1000 (d) #28 at iteration 1000



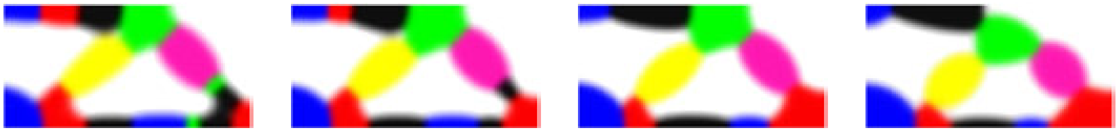
(e) #29 at iteration 1000 (f) #30 at iteration 1000 (g) #31 at iteration 1000 (h) #32 at iteration 1000



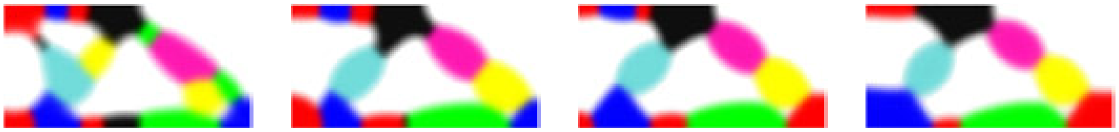
(i) #33 at iteration 1000 (j) #34 at iteration 1000 (k) #35 at iteration 1000 (l) #36 at iteration 1000



(m) #37 at iteration 1000 (n) #38 at iteration 1000 (o) #39 at iteration 1000 (p) #40 at iteration 1000



(q) #41 at iteration 1000 (r) #42 at iteration 1000 (s) #43 at iteration 1000 (t) #44 at iteration 1000

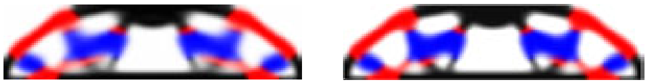


(u) #45 at iteration 1000 (v) #46 at iteration 1000 (w) #47 at iteration 1000 (x) #48 at iteration 1000

Figure 4. Final topologies (at iteration 1000) corresponding to test cases #25-#48.

(a) #5 at iteration 10

(b) #5 at iteration 25

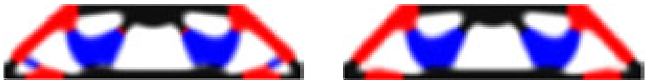


(c) #5 at iteration 50

(d) #5 at iteration 75

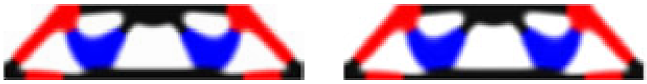
(e) #5 at iteration 100

(f) #5 at iteration 200



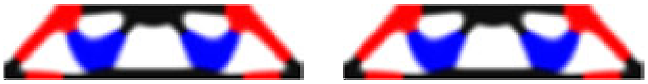
(g) #5 at iteration 300

(h) #5 at iteration 400



(i) #5 at iteration 500

(j) #5 at iteration 600

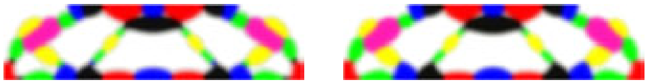


(k) #5 at iteration 800

(l) #5 at iteration 1000

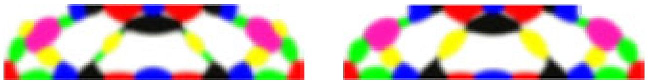
(m) #24 at iteration 10

(n) #24 at iteration 25



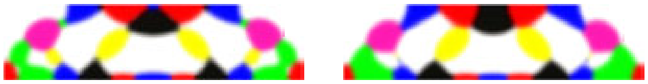
(o) #24 at iteration 50

(p) #24 at iteration 75



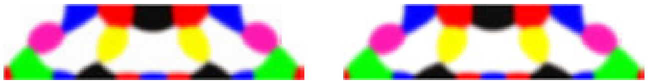
(q) #24 at iteration 100

(r) #24 at iteration 200



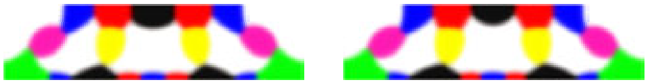
(s) #24 at iteration 300

(t) #24 at iteration 400



(u) #24 at iteration 500

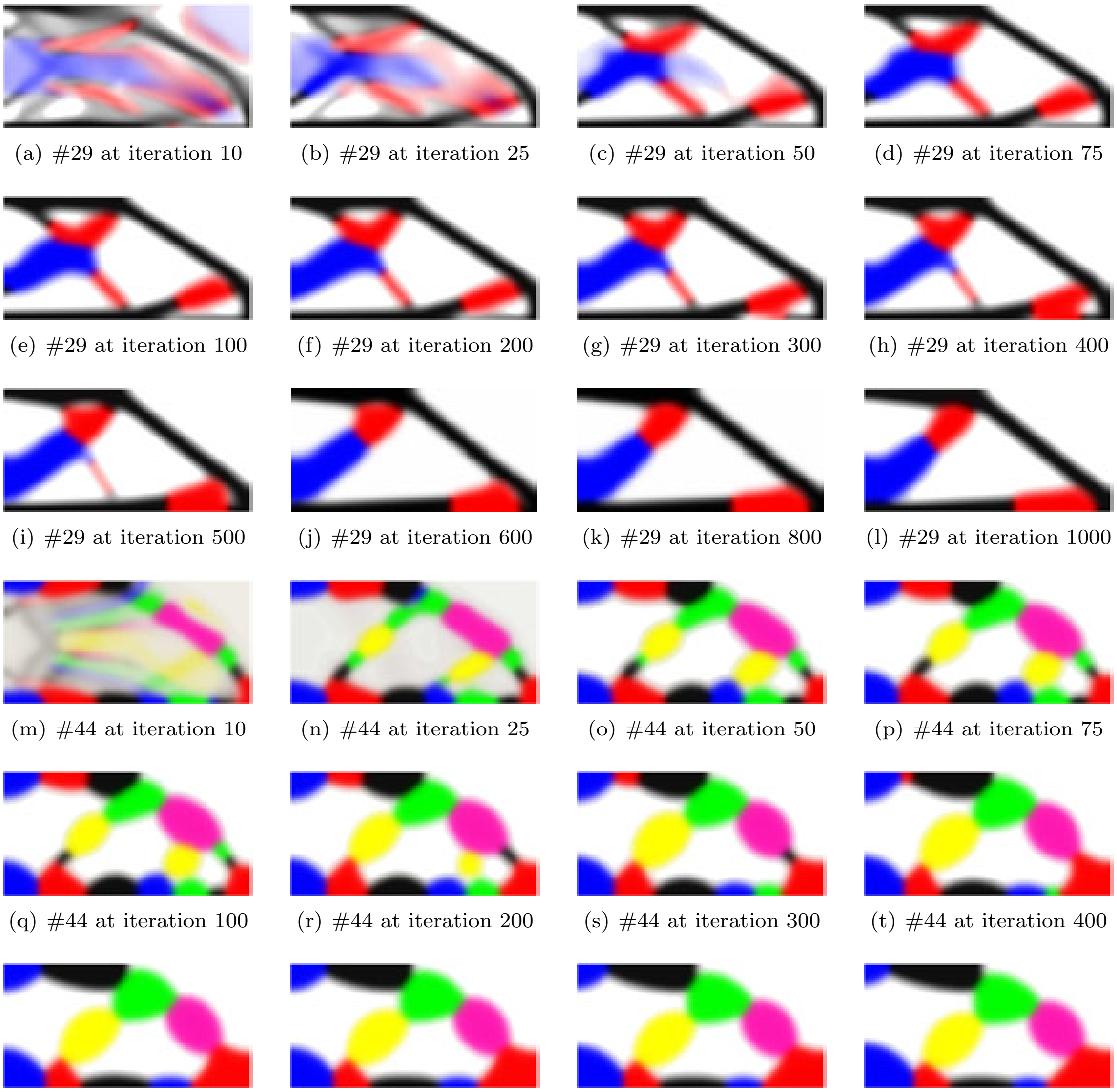
(v) #24 at iteration 600



(w) #24 at iteration 800

(x) #24 at iteration 1000

Figure 5. Evolution of topologies during the optimization corresponding to test cases #5 and #24.



(a) #29 at iteration 10 (b) #29 at iteration 25 (c) #29 at iteration 50 (d) #29 at iteration 75

(e) #29 at iteration 100 (f) #29 at iteration 200 (g) #29 at iteration 300 (h) #29 at iteration 400

(i) #29 at iteration 500 (j) #29 at iteration 600 (k) #29 at iteration 800 (l) #29 at iteration 1000

(m) #44 at iteration 10 (n) #44 at iteration 25 (o) #44 at iteration 50 (p) #44 at iteration 75

(q) #44 at iteration 100 (r) #44 at iteration 200 (s) #44 at iteration 300 (t) #44 at iteration 400

(u) #44 at iteration 500 (v) #44 at iteration 600 (w) #44 at iteration 800 (x) #44 at iteration 1000

Figure 6. Evolution of topologies during the optimization corresponding to test cases #29 and #44.

Now, let us to briefly justify the above mentioned observation by connecting the rate of topological evolution in phase-field based methods to the metric of interpolation space used to compute the gradient of the objective functional. In general, phase-field based methods use pseudo time-stepping by moving along the descent direction to find local minimums. The zero limit of the pseudo time-step size leads to a parabolic PDE to update the design vector. This PDE is commonly called the gradient flow equation. The steepest descent direction, the reverse direction of the objective function gradient, is a common choice for the descent direction in the phase-field based methods. The essential difference in the phase-field based topology optimization methods stems in the choice of interpolation space metric, denoted by $V$ here, used to compute the objective functional gradient. For instance, when $V = L^2$, the gradient flow equation is commonly categorized in the Allen-Cahn category. In the Allen-Cahn category, the corresponding equation is commonly a nonlinear second order parabolic PDE. When $V = H^{-1}$, it is classified in the Cahn-Hilliard category. The corresponding PDE of the Cahn-Hilliard category is commonly a nonlinear forth order parabolic PDE. Assume that for the purpose of the numerical solution, the spatial element size is denoted by $h$, time-step size is denoted by $\Delta t$ and an explicit time integration method is employed to solve the gradient flow equation. Using the classical linear stability analysis illustrates that $\Delta t$ is of $O(h^2)$ and $O(h^4)$ for Allen-Cahn and Cahn-Hilliard like gradient flow respectively. Therefore, the dynamics of Cahn-Hilliard flow is significantly slower than that of the Allen-Cahn flow. It implies the slower convergence rate of Cahn-Hilliard flow in contrast to the Allen-Cahn flow. This conclusion is well documented in the literature, e.g. see: [38, 37]. When $V = H^1$, the gradient flow is unconditionally stable based on the linear stability analysis. Therefore, we expect the possibility of using larger time increments, or equivalently faster convergence rate of the corresponding gradient flow, in contrast to formerly mentioned flows. It is well supported by our numerical results presented here. It is however important to note that the practical gradient flow is a nonlinear function of the design vector. Therefore, the time-step size based on the linear stability analysis can not be directly used in practice, but it can give us some idea about the possible maximum value of the allowable time-step size. The $H^1$-gradient is commonly called the Sobolev gradient in the optimization literature [75]. The faster convergence rate of the $H^1$ gradient flow in contrast to that of the $L^2$ gradient flow is demonstrated in literature for unconstrained and constrained optimization problems, for instance see: [42, 75–81].

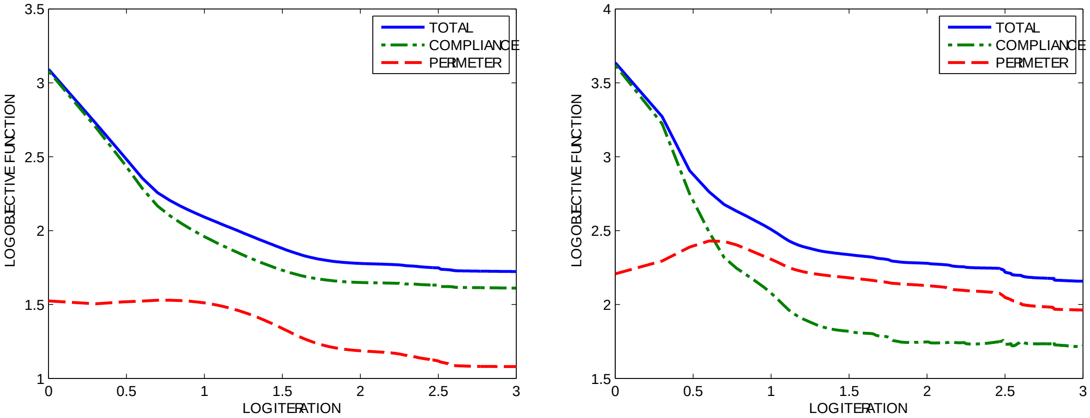
The variation of the objective functional, $\mathcal{J}^\varepsilon$, structure's compliance, $\mathcal{J}$ and the scaled total inter-phase perimeter, $\zeta P^\varepsilon$, by iterations are plotted in figure 7 for test cases #5, #24, #29 and #44. Because the variation of the objective functional is large at initial iterations and its variation decays as iterations proceed, the graphs are plotted in $\log - \log$ scale for the purpose of better visualization. According to figure 7, in all cases, the objective functional $\mathcal{J}^\varepsilon$ decreases monotonically as iterations proceed. Moreover, in all cases, the variation of objective functional decays by iterations and its graph reaches to a plateau shape after about 500 iterations. The compliance of structure shows the same trend in all cases. The total inter-phase perimeter increases during early iterations (because of the phase separation), and after about 10 iterations it starts to decrease monotonically by iterations. These observations confirm the convergence and feasibility of the presented algorithm in practice.

Now, let us to study the convergence of the presented algorithm by examining the $L^2$-norm of the objective functional gradient. Because our algorithm has two steps per iteration, the following function will be used for this purpose:

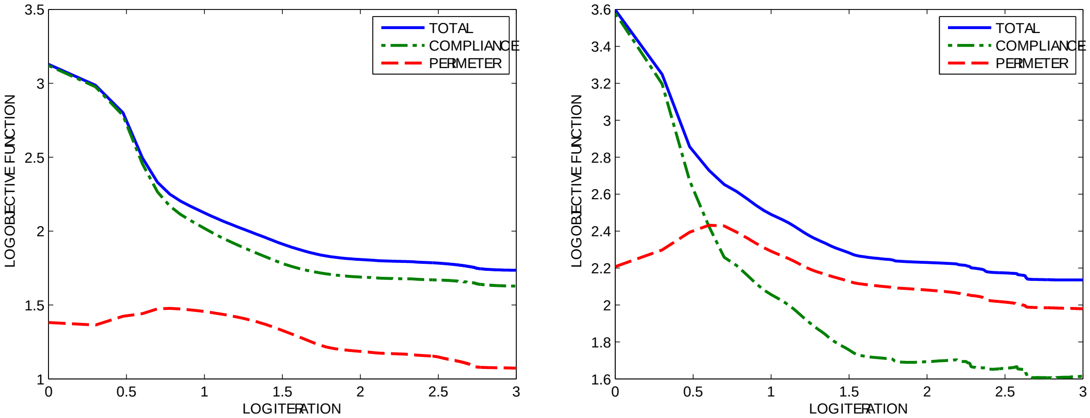
$$g(\boldsymbol{\varrho}) = \left\|\mathcal{P}_{\mathcal{A}'}[\tilde{\boldsymbol{\varrho}} - \boldsymbol{J}'_N] - \tilde{\boldsymbol{\varrho}}\right\|_{L^2(\Omega)} + \left\|\zeta\varepsilon\, \boldsymbol{\Delta\varrho}\right\|_{L^2(\Omega)} \tag{55}$$

To compute the second term in (55), it is required that $\boldsymbol{\varrho} \in H^2(\Omega)$. However, $\boldsymbol{\varrho} \in H^1(\Omega)$ in our algorithm. To resolve this problem, we further regularize the density field (only) to evaluate the second term in (55). As it has been proved in [82] (see also section 5.4 of [83]), if we apply the inverse Helmholtz operator (with the homogeneous boundary conditions) m-times to a field with the $L^2(\Omega)$ regularity, then the resulting field will has the $H_0^m(\Omega)$ regularity. Therefore, to increase the regularity of $\boldsymbol{\varrho}$ from $H^1(\Omega)$ to $H^2(\Omega)$, it is sufficient to solve (44) with $\boldsymbol{\varrho}$ as the input argument. In fact, $g$ sums the $L^2$-norm of gradients used during two steps of the presented algorithm. Obviously, when $g(\boldsymbol{\varrho}^*) = 0$, vector $\boldsymbol{\varrho}^*$ can be accounted as a stationary point of our (constrained) gradient flow. Because no line-search or any other globalization strategy are used in the present study, we will not expect that $g$ decreases monotonically by iterations, but an almost monotonically decreasing behavior could be expected. Figure 8 shows the variation of $g$ as a function of iterations for the test cases #5, #24, #29 and #44. For the purpose of better illustration, the graphs are plotted in $\log - \log$ scale here. According to the plots $g$ does not essentially decay monotonically during early iterations, however, after a few iterations, it decreases almost monotonically in all cases.

To study the volume preserving property of the presented algorithm, the variation of total volume fractions of contributing phases by iterations are plotted in figure 9 for the test cases #5 and #29. It demonstrates the volume preserving property of the presented algorithm in practice. Examining other test cases led us to the same conclusion in this regard. In fact, the presented algorithm preserves the volumes of phases up to the machine precision.



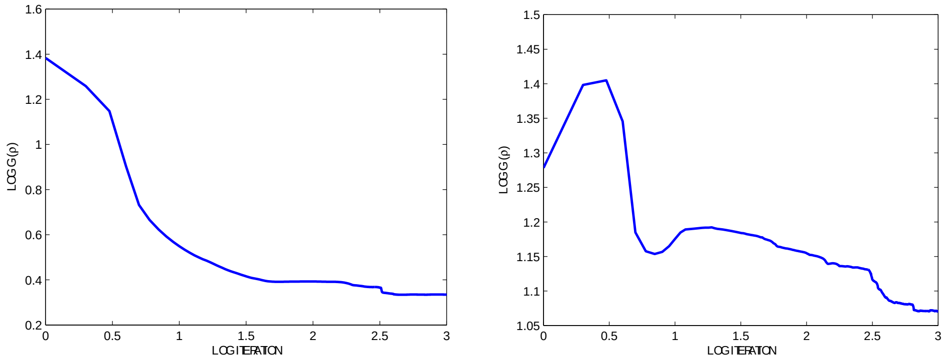
(a) test case #5 (b) test case #24



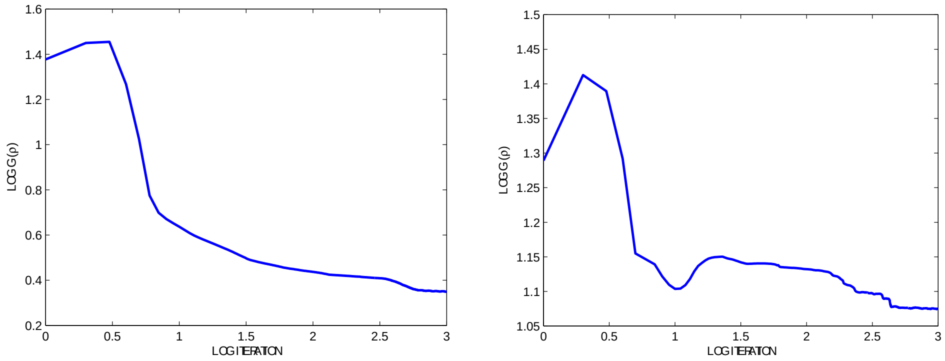
(c) test case #29 (d) test case #44

Figure 7. Variation of the objective functional, $\mathcal{J}^{\varepsilon}$ (denoted by "Total" in plots), structure's compliance, $\mathcal{J}$ (denoted by "Compliance" in plots) and the scaled total inter-phase perimeter, $\zeta P^{\varepsilon}$ (denoted by "Perimeter" in plots), as a function of iterations for test cases #5, #24, #29 and #44 (all axis are plotted in the logarithmic scale for the purpose of better illustration).

The CPU-times corresponding to test cases 1-48 are listed in table 2. It shows that the computational cost of the topology optimization increases slightly by increasing the number of contributing phases. For instance by increasing the number of phases from 3 in test case #25 to 8 in test case #45, the CPU-time is increased by factor 2.2. Roughly speaking, the CPU-time scales sub-linearly by the number of phases. It is an important benefit of the presented algorithm in contrast to the alternating active phase algorithm [10] in which the computational complexity scales quadratically by the number of phases.

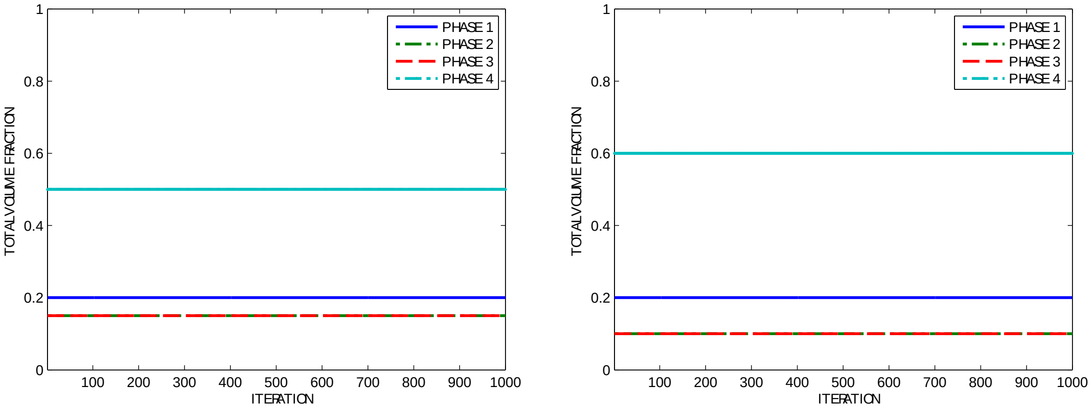


(a) test case #5 (b) test case #24



(c) test case #29 (d) test case #44

FIGURE 8. Variation of $g$, see equation (55), as a function of iterations for test cases #5, #24, #29 and #44 (all axis are plotted in the logarithmic scale for the purpose of better illustration).

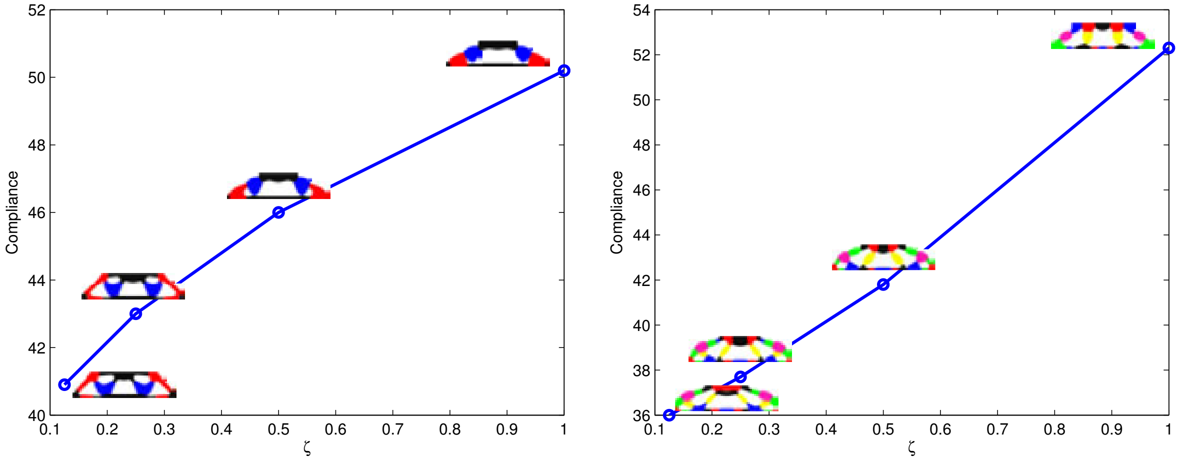


(a) test case #5 (b) test case #29

FIGURE 9. Variation of the total volume fractions of phases by iterations for test cases #5 and #29.

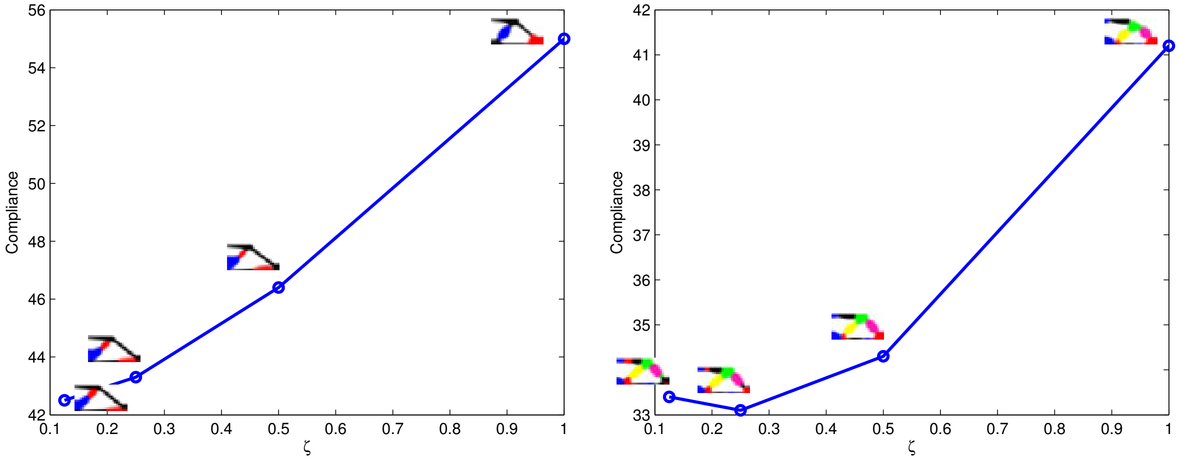
| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 | #9 | #10 | #11 | #12 |
| 308.4 | 308.4 | 310.7 | 318.2 | 362.7 | 366.8 | 382.3 | 384.7 | 416.7 | 414.5 | 430.3 | 385.5 |
| #13 | #14 | #15 | #16 | #17 | #18 | #19 | #20 | #21 | #22 | #23 | #24 |
| 424.4 | 427.3 | 438.6 | 466.4 | 485.7 | 498.5 | 520.4 | 539.5 | 579.4 | 585.1 | 654.6 | 705.2 |
| #25 | #26 | #27 | #28 | #29 | #30 | #31 | #32 | #33 | #34 | #35 | #36 |
| 292.2 | 287.6 | 293.8 | 296.9 | 362.8 | 356.1 | 369.6 | 424.1 | 401.5 | 446.5 | 415.7 | 451.8 |
| #37 | #38 | #39 | #40 | #41 | #42 | #43 | #44 | #45 | #46 | #47 | #48 |
| 473.3 | 555.0 | 563.3 | 636.2 | 597.4 | 580.3 | 693.6 | 736.5 | 641.6 | 662.2 | 707.0 | 929.3 |

TABLE 2. CPU times corresponding to test cases #1-#48 (in second).



(a) test cases #5, #6, #7, #8

(b) test cases #21, #22, #23, #24



(c) test cases #29, #30, #31, #32

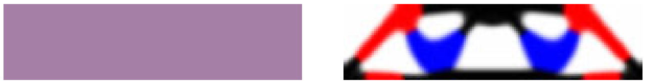
(d) test case #41, #42, #43, #44

FIGURE 10. Variation of final compliance as a function of $\zeta$.

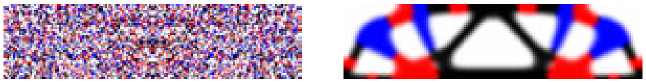
| #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 | #9 | #10 | #11 | #12 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 47.5 | 47.8 | 50.0 | 58.5 | 40.9 | 43.0 | 46.0 | 50.2 | 33.8 | 35.8 | 42.5 | 51.7 |
| #13 | #14 | #15 | #16 | #17 | #18 | #19 | #20 | #21 | #22 | #23 | #24 |
| 45.7 | 46.6 | 54.4 | 67.1 | 40.3 | 42.7 | 46.5 | 54.7 | 36.0 | 37.7 | 41.8 | 52.3 |
| #25 | #26 | #27 | #28 | #29 | #30 | #31 | #32 | #33 | #34 | #35 | #36 |
| 42.5 | 43.2 | 43.9 | 49.6 | 42.5 | 43.3 | 46.4 | 55.0 | 39.5 | 41.0 | 44.9 | 55.0 |
| #37 | #38 | #39 | #40 | #41 | #42 | #43 | #44 | #45 | #46 | #47 | #48 |
| 36.0 | 36.6 | 39.7 | 45.7 | 33.4 | 33.1 | 34.3 | 41.2 | 30.7 | 30.4 | 32.9 | 34.9 |

Table 3. Final values of compliance corresponding to test cases #1-#48.

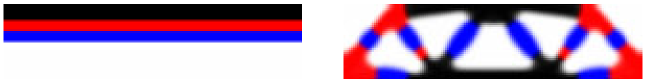
The final value of compliance corresponding to test cases 1-48 are listed in the table 3. To get more idea about the effect of $\zeta$ on the final results, the values of final compliance in conjunction with the final topologies are plotted in figure 10 for test cases 5-8, 21-24, 29-32 and 41-44 (note that in every series of these test cases, except $\zeta$ all parameters are similar). It shows that, almost in all cases, increasing the value of $\zeta$ increases the final value of the compliance. However, the outline of final topology is almost independent to the value of $\zeta$.



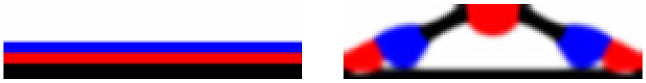
(a) #5-1 at iteration 0 (b) #5-1 at iteration 1000



(c) #5-2 at iteration 0 (d) #5-2 at iteration 1000

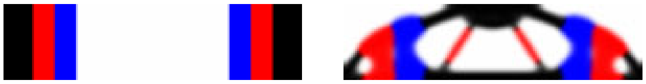


(e) #5-3 at iteration 0 (f) #5-3 at iteration 1000



(g) #5-4 at iteration 0 (h) #5-4 at iteration 1000

(i) #5-5 at iteration 0 (j) #5-5 at iteration 1000



(k) #5-6 at iteration 0 (l) #5-6 at iteration 1000

Figure 11. Effect of the variation of initial materials distributions on the final topologies (at iteration 1000) corresponding to test case #5. The case of #5-1 and #5-2 denote the case of uniform and random 0-1 initial materials distributions).

| #5-1 | #5-2 | #5-3 | #5-4 | #5-5 | #5-6 |
|---|---|---|---|---|---|
| 40.9 | 46.7 | 45.5 | 43.7 | 44.9 | 43.3 |

TABLE 4. Final values of compliance corresponding to test cases #5-1-#5-6.

Because both parts of our regularized objective functional are non-convex, it is a strongly non-convex in practice and the corresponding optimal design problem has possibly many local solutions. Therefore, the result of our optimization algorithm can be a function of the initial design. To study this effect, test case #5 is solved using five different initial materials distribution. Figure 11 shows the outcome of this numerical experiment. According to this figure, the final topology, and consequently the optimization path, varies significantly by the variation of the initial design. The final value of the compliance corresponding to these test cases, are listed in the table 4. According to this table, the final value of the compliance can be varied significantly by altering the initial design. Moreover, using the uniform initial materials distribution leads to the best performance of resulted structure in this case.

## 9. SUMMARY

A new computational algorithm based on the volume constrained Allen-Cahn system is introduced in the present study to solve multimaterial topology optimization problems. Increasing the number of phases does not increase the algorithmic complexity of the presented method. An interesting feature of the presented algorithm is to solve the topology optimization problems on an $H^1$ Sobolev space. According to our knowledge, it is the first report on the solution of topology optimization problems on an actual $H^1$ space. It is important to note that there are many works in which authors computed the gradient of the objective functional on an $H^1$ space, for instance see [64]. However, their overall computational algorithm did not ensure an $H^1$ regularity, because no specific attention has been paid to keep the regularity of solution while satisfying the set of constraints. The success and efficiency of the presented algorithm are illustrated numerically by solution of several test problems. The computational cost of the presented algorithm scales sub-linearly by the number of phases. Numerical results of the presented method illustrates that it is an order of magnitude faster than the generalized Cahn-Hilliad based topology optimization algorithm introduced in [8] (solving problem on $H^{-1}$ function space).

As it is mentioned in section 1, multimaterial topology optimization is particularly interesting for the optimal design of multiphysics/multifunctional structures. Moreover, the mass or price constraint reformulation of multimaterial topology optimization problems (cf. [2]) are interesting applications of design optimization with multiple number of phases. The extension of our algorithm to solve these kinds of problems are recommended for further extension of the present work.

## ACKNOWLEDGMENT

We would like to thanks anonymous reviewers for their constructive comments which improves the presentation of our paper.

## Appendix A. MATLAB code

The following code includes a 95 lines MATLAB code to solve multimaterial minimum compliance topology optimization problems based on the introduced algorithm in the present study. The loading and boundary conditions included in this code are related to the Cantilever beam test case.

```
%119 LINES MATLAB CODE MULTIPHASE MINIMUM COMPLIANCE TOPOLOGY OPTIMIZATION
function topopt_multi
 clear all; clc; nx = 96; ny = 48; q = 3; p = 4;
 eps = 1; alpha = 0.5; zeta = 0.125; maxit = 1000;
 nu = 0.3; e = [4 2 1 1e-9]'; v = [0.2 0.1 0.1 0.6]';
 % INIT DESIGN VECTOR
 x = zeros(nx*ny,p);
 for i = 1:p
  x (:,i) = v(i);
 end
 sx = sum(x);
 %PREPARE FINITE ELEMENT TOOLS FOR ELASTICITY
 A11 = [12 3 -6 -3; 3 12 3 0; -6 3 12 -3; -3 0 -3 12];
 A12 = [-6 -3 0 3; -3 -6 -3 -6; 0 -3 -6 3; 3 -6 3 -6];
 B11 = [-4 3 -2 9; 3 -4 -9 4; -2 -9 -4 -3; 9 4 -3 -4];
 B12 = [ 2 -3 4 -9; -3 2 9 -2; 4 9 2 3; -9 -2 3 2];
 KE = 1/(1-nu^2)/24*([A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);
 nodenrs = reshape(1:(1+nx)*(1+ny),1+ny,1+nx);
 edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nx*ny,1);
 edofMat = repmat(edofVec,1,8)+repmat([0 1 2*ny+[2 3 0 1] -2 -1],nx*ny,1);
 iK = reshape(kron(edofMat,ones(8,1))',64*nx*ny,1);
 jK = reshape(kron(edofMat,ones(1,8))',64*nx*ny,1);
 %PREPARE FINITE ELEMENT TOOLS FOR HELMHOLTZ SOLVER
 D = alpha*eps*zeta;
 KEF = D*[4 -1 -2 -1; -1 4 -1 -2; -2 -1 4 -1; -1 -2 -1 4]/6 + ...
         [4 2 1 2; 2 4 2 1; 1 2 4 2; 2 1 2 4]/36;
 edofVecF = reshape(nodenrs(1:end-1,1:end-1),nx*ny,1);
 edofMatF = repmat(edofVecF,1,4)+repmat([0 ny+[1:2] 1],nx*ny,1);
 iKF = reshape(kron(edofMatF,ones(4,1))',16*nx*ny,1);
 jKF = reshape(kron(edofMatF,ones(1,4))',16*nx*ny,1);
 sKF = reshape(KEF(:)*ones(1,nx*ny),16*nx*ny,1);
 KF = sparse(iKF,jKF,sKF); LF = chol(KF,'lower');
 iTF = reshape(edofMatF,4*nx*ny,1); jTF = reshape(repmat([1:nx*ny],4,1)',4*nx*ny,1);
 sTF = repmat(1/4,4*nx*ny,1);
 TF = sparse(iTF,jTF,sTF);
 %DEFINE LOADS AND SUPPORTS FOR CANTILEVER BEAM
 F = sparse(2*(ny+1)*(nx+1),1,-1,2*(ny+1)*(nx+1),1);
 fixeddofs = [1:2*(ny+1)];
 U = zeros(2*(ny+1)*(nx+1),1);
 alldofs = [1:2*(ny+1)*(nx+1)];
 freedofs = setdiff(alldofs,fixeddofs);
 % OPTIMIZATION LOOP
 for iter = 1:maxit
  [obj objc g] = eval_fg (nx,ny,eps,zeta,p,q,e,KE,U,edofMat,freedofs,iK,jK,F,x);
  x = x + alpha * (proj (sx, x - g) - x);
  for i = 1:p-1
   x(:,i) = TF'*(LF'\(LF\(TF*x(:,i))));
  end
  x(:,p) = 1 - sum(x(:,1:p-1),2);
  fprintf('It:%5i c:%8.4f obj:%8.4f\n',iter, objc, obj);
  % VISUALIZATION OF CURRENT TOPOLOGY
  if mod(iter,10) == 0
   I = make_bitmap (p,nx,ny,x); image(I), axis image off, drawnow;
  end
 end
end
%COMPUTE OBJECTIVE FUNCTION AND ITS GRADIENT
```

## References

[1] M.P. Bendsøe and O. Sigmund. *Topology Optimization: theory, methods and applications*. Springer, 2004.

[2] T. Gao and W. Zhang. A mass constraint formulation for structural topology optimization with multiphase materials. *Int J Numer Meth Eng*, 88(8):774–796, 2011.

[3] M.P. Bendsøe and O. Sigmund. Material interpolation schemes in topology optimization. *Arch Appl Mech*, 69(9):635–654, 1999.

[4] G. Allaire. Shape optimization by the homogenization method. *New York, Springer-Verlag, 2002.*, 2002.

[5] L Yin and G.K. Ananthasuresh. Topology optimization of compliant mechanisms with multiple materials using a peak function material interpolation scheme. *Struct Multidisc Optim*, 23(1):49–62, 2001.

[6] C.F. Hvejsel and E. Lund. Material interpolation schemes for unified topology and multi-material optimization. *Struct Multidisc Optim*, 43(6):811–825, 2011.

[7] S. Zhou and M.Y. Wang. 3d multi-material structural topology optimization with the generalized cahn-hilliard equations. *CMES: Comput Model in Eng Sci*, 16(2):83–102, 2006.

[8] S. Zhou and M.Y. Wang. Multimaterial structural topology optimization with a generalized cahn–hilliard model of multiphase transition. *Struct Multidisc Optim*, 33(2):89–111, 2007.

[9] L. Zhen, G. Wei, and S. Chongmin. Design of multi-phase piezoelectric actuators. *J Intel Mater Sys Struct*, 21(8):1851–1865, 2010.

[10] R. Tavakoli and S.M. Mohseni. Alternating active-phase algorithm for multimaterial topology optimization problems – a 115-line matlab implementation. *Struct Multidisc Optim, in press, DOI 10.1007/s00158-013-0999-1*, 2013.

[11] M.Y. Wang and X. Wang. color level sets: a multi-phase method for structural topology optimization with multiple materials. *Comput Meth Appl Mech Engng*, 193(6):469–496, 2004.

[12] S. Turteltaub. Functionally graded materials for prescribed field evolution. *Comput Meth Appl Mech Engng*, 191(21):2283–2296, 2002.

[13] S. Turteltaub. Optimal control and optimization of functionally graded materials for thermomechanical processes. *Int J Solids Struct*, 39(12):3175–3197, 2002.

[14] S. Turteltaub. Optimal non-homogeneous composites for dynamic loading. *Struct Multidisc Optim*, 30(2):101–112, 2005.

[15] G. Allaire, F. Jouve, and A. Toader. Structural optimization using sensitivity analysis and a level-set method. *J comput phys*, 194(1):363–393, 2004.

[16] O. Sigmund and S. Torquato. Composites with extremal thermal expansion coefficients. *Appl Phys Lett*, 69(21):3203–3205, 1996.

[17] O. Sigmund and S. Torquato. Design of materials with extreme thermal expansion using a three-phase topology optimization method. *J Mech Phys Solids*, 45(6):1037–1067, 1997.

[18] O. Sigmund and S. Torquato. Design of smart composite materials using topology optimization. *Smart Mater Struct*, 8:365, 1999.

[19] L.V. Gibiansky and O. Sigmund. Multiphase composites with extremal bulk modulus. *J Mech Phys Solids*, 48(3):461–498, 2000.

[20] O. Sigmund. Recent developments in extremal material design. *in Trends in Computational Mechanics, W.A. Wall, K.-U. Bletzinger and K. Schweizerhof (eds)*, pages 228–232, 2001.

[21] O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Struct Multidisc Optim*, 16(1):68–75, 1998.

[22] G. Allaire and C. Castro. Optimization of nuclear fuel reloading by the homogenization method. *Struct Multidisc Optim*, 24(1):11–22, 2002.

[23] O. Sigmund. A 99 line topology optimization code written in matlab. *Struct Multidisc Optim*, 21(2):120–127, 2001.

[24] M. Yulin and W. Xiaoming. A level set method for structural topology optimization and its applications. *Adv Eng Softw*, 35(7):415–441, 2004.

[25] M.Y. Wang and X. Wang. A level-set based variational method for design and optimization of heterogeneous objects. *Computer-Aided Design*, 37(3):321–337, 2005.

[26] E. Dombre, G. Allaire, O. Pantz, and D. Schmitt. Shape optimization of a sodium fast reactor core. In *ESAIM Proc*, volume 38, pages 319–334. EDP Sciences, 2012.

[27] G. Allaire, C. Dapogny, and G. Delgado. Multi-phase structural optimization via a level set method. *preprint, hal-00839464*, 2013.

[28] P. Wei and M.Y. Wang. Piecewise constant level set method for structural topology optimization. *Int J Numer Methods Eng*, 78(4):379–402, 2009.

[29] Z. Luo, L. Tong, J. Luo, P. Wei, and M.Y. Wang. Design of piezoelectric actuators using a multiphase level set method of piecewise constants. *J Comput Phys*, 228(7):2643–2659, 2009.

[30] J. Stegmann and E. Lund. Discrete material optimization of general composite shell structures. *Int J Numer Meth Eng*, 62(14):2009–2027, 2005.

[31] E. Lund and J. Stegmann. On structural optimization of composite shell structures using a discrete constitutive parametrization. *Wind Energy*, 8(1):109–124, 2005.

[32] M. Bruyneel. Sfpa new parameterization based on shape functions for optimal material selection: application to conventional composite plies. *Struct Multidisc Optim*, 43(1):17–27, 2011.

[33] M. Bruyneel, P. Duysinx, C. Fleury, and T. Gao. Extensions of the shape functions with penalization parameterization for composite-ply optimization. *AIAA Journal*, 49(10):2325–2329, 2011.

[34] T. Gao, W. Zhang, and P. Duysinx. A bi-value coding parameterization scheme for the discrete optimal orientation design of the composite laminate. *Int J Numer Meth Eng*, 91(1):98–114, 2012.

[35] B. Bourdin and A. Chambolle. Design-dependent loads in topology optimization. *ESAIM COCV*, 9:19–48, 2003.

[36] M.Y. Wang and S. Zhou. Synthesis of shape and topology of multi-material structures with a phase-field method. *J Comput-Aided Mater Des*, 11(2):117–138, 2004.

[37] L. Blank, H. Garcke, L. Sarbu, and V. Styles. Primal-dual active set methods for allen–cahn variational inequalities with nonlocal constraints. *Numer Meth Partial Differential Equations*, 2012.

[38] L. Blank, H. Garcke, L. Sarbu, T. Srisupattarawanit, V. Styles, and A. Voigt. Phase-field approaches to structural topology optimization. In *Constrained Optimization and Optimal Control for Partial Differential Equations*, pages 245–256. Springer, 2012.

[39] L. Blank, H. Farshbaf-Shaker, H. Garcke, and V. Styles. Relating phase field and sharp interface approaches to structural topology optimization. *preprint*, 2013.

[40] J.W. Cahn and J.E. Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *J Chem Phys*, 28:258, 1958.

[41] Q. Du and F. Lin. Numerical approximations of a norm-preserving gradient flow and applications to an optimal partition problem. *Nonlinearity*, 22(1):67–83, 2009.

[42] C. Talischi and G.H. Paulino. An operator splitting algorithm for tikhonov-regularized topology optimization. *Comput Meth Appl Mech Eng*, 253:599–608, 2013.

[43] L. Dedè, M. Borden, and T. Hughes. Isogeometric analysis for topology optimization with a phase field model. *Arch Comput Meth Eng*, 19(3):427–465, 2012.

[44] R.B. Haber, C.S. Jog, and M.P. Bendsøe. A new approach to variable-topology shape design using a constraint on perimeter. *Struct Optim*, 11(1-2):1–12, 1996.

[45] L. Modica and S. Mortola. Un esempio di $\gamma$-convergenza. *Boll. Un. Mat. Ital. B*, 14(1):285–299, 1977.

[46] L. Modica. The gradient theory of phase transitions and the minimal interface criterion. *Arch Rat Mech Anal*, 98(2):123–142, 1987.

[47] M. Burger and R. Stainko. Phase-field relaxation of topology optimization with local stress constraints. *SIAM J Control Optim*, 45(4):1447–1466, 2006.

[48] J.S. Choi, T. Yamada, K. Izui, S. Nishiwaki, and J. Yoo. Topology optimization using a reaction–diffusion equation. *Comput Meth Appl Mech Eng*, 200(29):2407–2420, 2011.

[49] M. Wallin, M. Ristinmaa, and H. Askfelt. Optimal topologies derived from a phase-field method. *Struct Multidisc Optim*, 45(2):171–183, 2012.

[50] A.L. Gain and G.H. Paulino. Phase-field based topology optimization with polygonal elements: a finite volume approach for the evolution equation. *Struct Multidisc Optim*, 46(3):327–342, 2012.

[51] P. Penzler, M. Rumpf, and B. Wirth. A phase-field model for compliance shape optimization in nonlinear elasticity. *ESAIM: Control, Optim Calc Var*, 18(01):229–258, 2012.

[52] I. Steinbach, F. Pezzolla, B. Nestler, M. Seeßelberg, R. Prieler, G. Schmitz, and J. Rezende. A phase field concept for multiphase systems. *Physica D*, 94(3):135–147, 1996.

[53] H. Garcke, B. Nestler, and B. Stoth. A multiphase field concept: numerical simulations of moving phase boundaries and multiple junctions. *SIAM J Appl Math*, 60(1):295–315, 1999.

[54] H. Garcke, B. Nestler, B. Stinner, and F. Wendler. Allen–cahn systems with volume constraints. *Math Mod Meth Appl Sci*, 18(08):1347–1381, 2008.

[55] S. Baldo. Minimal interface criterion for phase transitions in mixtures of cahn-hilliard fluids. *Ann Inst H. Poincaré (C) Anal Non Linéaire*, 7(2):67–90, 1990.

[56] E. Oudet. Approximation of partitions of least perimeter by $\gamma$-convergence: around kelvins conjecture. *Exp Math*, 20(3):260–270, 2011.

[57] C.S. Jog. Topology design of structures using a dual algorithm and a constraint on the perimeter. *Int J Numer Meth Eng*, 54(7):1007–1019, 2002.

[58] M. Wallin and M. Ristinmaa. Howard's algorithm in a phase-field topology optimization approach. *Int J Numer Meth Eng*, 94(1):43–59, 2013.

[59] S. Amstutz. Regularized perimeter for topology optimization. *SIAM J Control Optim*, 51(3):2176–2199, 2013.

[60] R. Tavakoli and H. Zhang. A nonmonotone spectral projected gradient method for large-scale topology optimization problems. *Numer Algebra, Control Optim*, 2(2):395–412, 2012.

[61] G. Allaire. *Numerical analysis and optimization: an introduction to mathematical modelling and numerical simulation*. Translated by: Craig, A., Oxford University Press, USA, 2007.

[62] J. Nocedal and S. Wright. *Numerical optimization*. Springer, 2006.

[63] F. Tröltzsch. *Optimal control of partial differential equations: Theory, methods and applications*, volume 112. AMS Bookstore, 2010.

[64] B. Lazarov and O. Sigmund. Filters in topology optimization based on helmholtz-type differential equations. *Int J Numer Meth Eng*, 86(6):765–781, 2011.

[65] E. Birgin, J.M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J Optim*, 10(4):1196–1211, 2000.

[66] H. Gómez, V.M. Calo, Y. Bazilevs, and T. Hughes. Isogeometric analysis of the cahn–hilliard phase-field model. *Comput Meth Appl Mech Eng*, 197(49):4333–4352, 2008.

[67] E. Andreassen, A. Clausen, M. Schevenels, B.S. Lazarov, and O. Sigmund. Efficient topology optimization in matlab using 88 lines of code. *Struct Multidisc Optim*, 43(1):1–16, 2011.

[68] K. Kiwiel. On linear-time algorithms for the continuous quadratic knapsack problem. *J Optim Theory Appl*, 134(3):549–554, 2007.

[69] K. Kiwiel. Breakpoint searching algorithms for the continuous quadratic knapsack problem. *Math Prog*, 112(2):473–491, 2008.

[70] K. Kiwiel. Variable fixing algorithms for the continuous quadratic knapsack problem. *J Optim Theory Appl*, 136(3):445–458, 2008.

[71] Y. Dai and R. Fletcher. New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds. *Math Prog*, 106(3):403–421, 2006.

[72] R. Tavakoli. On the coupled continuous knapsack problems: projection onto the volume constrained Gibbs N-simplex. *Optimization Online preprint*, 2013. is available online at: `www.optimization-online.org/DB_HTML/2013/09/4026.html`.

[73] S. Boyd and J. Dattorro. Alternating projections. online note, 2003. is available online at: `www.stanford.edu/class/ee392o/alt_proj.pdf`.

[74] R. Escalante and M. Raydan. *Alternating Projection Methods*, volume 8. SIAM, 2011.

[75] J.W. Neuberger. Sobolev gradients and differential equations. *Lecture Notes in Mathematics*, 1670, 2010.

[76] S. Sial, J. Neuberger, T. Lookman, and A. Saxena. Energy minimization using sobolev gradients: application to phase separation and ordering. *J Comput Phys*, 189(1):88–97, 2003.

[77] N. Raza, S. Sial, and S. Siddiqi. Sobolev gradient approach for the time evolution related to energy minimization of ginzburg–landau functionals. *J Comput Phys*, 228(7):2566–2571, 2009.

[78] G. Von Winckel and A. Borzì. Computational techniques for a quantum control problem with h1-cost. *Inverse Probl*, 24(3):034007, 2008.

[79] B. Protas. Adjoint-based optimization of pde systems with alternative gradients. *J Comput Phys*, 227(13):6490–6510, 2008.

[80] B. Protas, T.R. Bewley, and G. Hagen. A computational framework for the regularization of adjoint analysis in multiscale pde systems. *J Comput Phys*, 195(1):49–89, 2004.

[81] I. Danaila and P. Kazemi. A new sobolev gradient method for direct minimization of the gross-pitaevskii energy with rotation. *SIAM J Sci Comput*, 32(5):2447–2467, 2010.

[82] J. Calder, A. Mansouri, and A. Yezzi. New possibilities in image diffusion and sharpening via high-order sobolev gradient flows. *J Math Imaging Vis*, 40(3):248–258, 2011.

[83] P. Kazemi and I. Danaila. Sobolev gradients and image interpolation. *SIAM J Imaging Sci*, 5(2):601–624, 2012.