

RESEARCH ARTICLE

Level set topology optimization for design-dependent pressure load problems

Hélio Emmendoerfer Jr.¹ | Eduardo Alberto Fancello²  | Emílio Carlos Nelli Silva¹ 

¹Department of Mechatronics and Mechanical Systems Engineering, University of Sao Paulo, São Paulo, SP, Brazil

²Department of Mechanical Engineering, Federal University of Santa Catarina, Campus Universitário - Trindade, Florianópolis, SC, Brazil

Correspondence

Eduardo Alberto Fancello, Department of Mechanical Engineering, Federal University of Santa Catarina, Campus Universitário - Trindade, Florianópolis, SC, Brazil.

Email: eduardo.fancello@ufsc.br

Summary

This work presents a level set framework to solve the compliance topology optimization problem considering design-dependent pressure loads. One of the major technical difficulties related to this class of problem is the adequate association between the moving boundary and the pressure acting on it. This difficulty is easily overcome by the level set method that allows for a clear tracking of the boundary along the optimization process. In the present approach, a reaction-diffusion equation substitutes the classical Hamilton-Jacobi equation to control the level set evolution. This choice has the advantages of allowing the nucleation of holes inside the domain and the elimination of the undesirable reinitialization steps. Moreover, the proposed algorithm allows merging pressurized (wet) boundaries with traction-free boundaries during level set movements. This last property, allied to the simplicity of the level set representation and successful combination with the reaction-diffusion based evolution applied to a design-dependent pressure load problem, represents the main contribution of this paper. Numerical examples provide successful results, many of which comparable with others found in the literature and solved with different techniques.

KEYWORDS

design-dependent load, level sets, reaction-diffusion equation, topology optimization

1 | INTRODUCTION

Traditionally, many applications of topology optimization problems fix the region where mechanical loads are applied. However, there is a class of problems, namely design-dependent loads, in which the loading is not frozen during the topology optimization procedure, that is, both direction and location of the loads can change with the structural design. One of the greatest difficulties in these problems is identifying and tracking the loaded moving boundaries during topology evolution mainly at intermediate designs. Design-dependent load problems in a continuum structure were firstly introduced in the topology optimization literature by Hammer and Olhoff,¹ Chen and Kikuchi,² and Bourdin and Chambolle,³ which used density methods (solid isotropic material with penalization models).

Although density methods do not allow directly representing the boundaries, they are widely used in the literature to solve design-dependent load problems. This class of problems represents a challenging problem for density methods since it is necessary to develop strategies to track the loaded boundary during topology changes. For example, Bézier curves are adopted in the works of Hammer and Olhoff¹ and Fuchs and Shemesh,⁴ a fictitious thermal loading is applied in the

work of Chen and Kikuchi,² isolines (iso-density line to define the boundary) are used by Du and Olhoff⁵ and Lee and Martins⁶ a pseudo potential function is introduced in the work of Zheng et al,⁷ and strategies of considering the void phase as an incompressible hydrostatic fluid transferring thus pressure loads are employed by Sigmund and Clausen⁸ and Bruggi and Cinquini.⁹ Some works use algorithm searching schemes (based on decision criterion) in the background grid to identify the load surfaces (see, for instance, the work of Zhang et al¹⁰). There are also papers mixing methods to recover the loading surface at each step of the minimization process. For example, the work of Wang et al¹¹ inherits the merits of the level set method and proposes a technique by associating the density approach and the distance regularized level set evolution.

Another approach commonly used for dealing with design-dependent load problems is based on level sets.^{12,13} An advantage of this approach over density methods is the clear representation of boundaries through a level set function. Within this context, Guo et al¹⁴ developed a sweep algorithm to apply the loading to a neighborhood of the zero level set curve. However, this algorithm may become computationally inefficient, and it may not operate for non-Cartesian design domains. Recently, Xia et al¹⁵ presented a methodology in which, to represent the pressure boundary separately, two level set functions are propagated via Hamilton-Jacobi evolution equation. The approach focuses on avoiding the crossing or touching of the free boundary and the pressure boundary. Moreover, other works¹⁶⁻¹⁸ present some applications to design-dependent loads by using level sets; however, implementation details of how to track the surface loading during topology changes are not discussed.

In all these papers, the level set function is represented by classical low-order Lagrangian functions on a fixed grid and its evolution guided by the solution of the Hamilton-Jacobi equation. As it is well known, this combination usually induces a degradation of the level set function forcing reinitialization schemes that preclude the appearing of new holes.

A possible way to overcome this difficulty is to use smoother level set representations. Example of this are the radial basis functions^{19,20} or B-splines.^{21,22} Another alternative consists of maintaining the simplicity of the nodal level set representation but including regularization (diffusion) terms on the evolution equation.^{23,24} This modification prevents the degradation of the level set function and eliminates the need of reinitialization procedures.

Another issue is related to the way the boundary defined by the level set representation is transferred to the numerical discretization of the physical problem. The simplest way is using the concept of ersatz material on the elements *cut* by the zero level curve (see, for example, the works of Allaire et al¹⁶ and Wang et al²⁵). Better representations at the cost of a more involved implementations are achieved by remeshing²⁶⁻²⁸ or discontinuous enrichments like those provided by the extended FEM.²⁹⁻³² For the particular case of design-dependent loads, this last alternative was used in the works of Jenkins and Maute³³ (fluid-solid interaction) and Coffin and Maute³⁴ (thermal conduction problems).

Up to the present moment, density and level set approaches are the most frequent techniques found in the literature to solve design-dependent load problems. However, very recently, other methods were used, providing promising results. For example, the bi-directional evolutionary structural optimization method for designing completely submerged buoyant modules is used in the work of Picelli et al,³⁵ and the topological derivative concept applied to optimize structures subject to hydrostatic pressure loading is proposed by Xavier and Novotny.³⁶

In this work, a method based on a level set framework is developed to solve the compliance topology optimization problem of structures subject to design-dependent pressure loads. The topology changes are controlled by a single level set function defined by conventional Lagrangian bases on a fixed grid. This level set evolves according to a reaction-diffusion equation analogous to that proposed by Yamada et al.²³ This approach has already shown to be a simple and efficient way to get rid of the reinitialization procedures and, consequently, to allow for the generation of new holes during the optimization sequence. The implicit boundary defined by the level set is transferred to a fixed Lagrangian finite element mesh using the concept of ersatz material on those elements sliced by the zero level curve. A simple and efficient element-based approximation is chosen to compute equivalent nodal forces consistent with the surface loads applied on the moving boundary. Moreover, analytical and algorithmically consistent sensitivity expressions can be retrieved from this approximation. Finally, the approach allows merging pressurized (wet) boundaries with traction-free boundaries during level set movements. This characteristic provides freedom for the required topological changes and more independence of the initial level set configuration. This last property allied to the simplicity of the level set representation and successful combination with the reaction-diffusion based evolution applied to a design-dependent pressure load problem are, in the opinion of the authors, the main contributions of the present manuscript.

The paper is organized as follows. The design-dependent load problem formulation by using a level set framework is presented in Section 2. The proposition of the level set evolution and tracking of the loaded moving boundary are discussed in Section 3. In Section 4, details on the discretization and numerical implementation of the methodology are provided, including the optimization algorithm. Section 5 is dedicated to the numerical examples. Finally, the conclusions are related in Section 6.

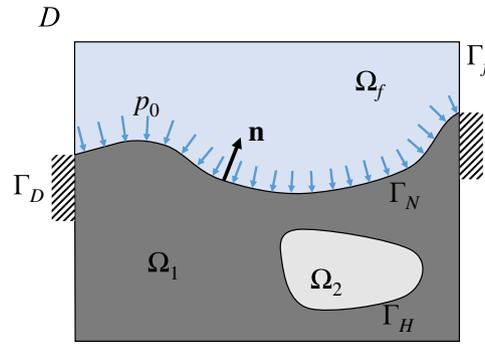


FIGURE 1 Representation of the domain composed of solid, void, and pressurized regions [Colour figure can be viewed at wileyonlinelibrary.com]

2 | DESIGN-DEPENDENT LOAD PROBLEM FORMULATION

2.1 | Level set-based problem definition

The mechanical problem comprises a bounded background domain $D \subseteq R^n$ ($n = 2, 3$) with smooth boundary ∂D (see Figure 1). Domain D is divided into a solid or strong region Ω_1 , a void or weak region Ω_2 , and a pressurized region $\Omega_f = D \setminus (\Omega_1 \cup \Omega_2)$. Each region is non-overlapping, ie, $\Omega_1 \cap \Omega_2 = \emptyset$, $\Omega_1 \cap \Omega_f = \emptyset$ and $\Omega_f \cap \Omega_2 = \emptyset$. The boundary of Ω_1 is split into three parts

$$\partial\Omega_1 = \Gamma_H \cup \Gamma_D \cup \Gamma_N, \quad \text{such that} \quad \Gamma_H \cap \Gamma_D \cap \Gamma_N = \emptyset, \quad (1)$$

where Dirichlet conditions are applied on the fixed boundary Γ_D and Γ_H is submitted to homogeneous Neumann conditions. Region Ω_f with boundary $\partial\Omega_f$ decomposed as

$$\partial\Omega_f = \Gamma_f \cup \Gamma_N, \quad \text{such that} \quad \Gamma_f \cap \Gamma_N = \emptyset, \quad (2)$$

exerts a constant pressure p_0 on the interface $\Gamma_N = \partial\Omega_1 \cap \partial\Omega_f$. Here, Γ_f is the part of ∂D that surrounds Ω_f . In design-dependent load problems, the free boundary Γ_H and the loading boundary Γ_N may change during the optimization procedure.

The present formulation consists in using a level set approach^{12,13} to represent the domain defined above. Thus, all regions and boundaries are implicitly defined by a conventional level set function $\phi : D \rightarrow R$ given as

$$\phi(\mathbf{x}) = \begin{cases} > 0 & \text{if } \mathbf{x} \in \Omega_1; \quad \mathbf{x} \in \Gamma_D \\ = 0 & \text{if } \mathbf{x} \in \Gamma_H \cup \Gamma_N \\ < 0 & \text{if } \mathbf{x} \in D \setminus \bar{\Omega}_1; \quad \mathbf{x} \in \Gamma_f. \end{cases} \quad (3)$$

Note that $\phi(\mathbf{x})$ is greater than zero everywhere within the solid domain Ω_1 .

Equilibrium is achieved by the displacement field $\mathbf{u} \in U$ that satisfies

$$a_\phi(\mathbf{u}, \mathbf{v}) = l_\phi(\mathbf{v}), \quad \forall \mathbf{v} \in V, \quad (4)$$

$$a_\phi(\mathbf{u}, \mathbf{v}) = \int_D \mathbf{C}(\phi) \boldsymbol{\varepsilon}(\mathbf{u}) \cdot \boldsymbol{\varepsilon}(\mathbf{v}) dD, \quad (5)$$

$$l_\phi(\mathbf{v}) = \int_{\Gamma_N} \boldsymbol{\tau}(\phi) \cdot \mathbf{v} \, d\partial\Omega = \int_D \boldsymbol{\tau}(\phi) \cdot \mathbf{v} \, \|\nabla\phi\| \delta(\phi) \, dD, \quad (6)$$

where U and V denote the sets of kinematically admissible displacements and admissible variations, respectively. Linear operators $a_\phi(\cdot, \cdot)$ and $l_\phi(\cdot)$ represent the virtual work of internal and external forces, respectively, and $\boldsymbol{\varepsilon}(\mathbf{u}) = \nabla^s \mathbf{u}$ denotes the linear strain tensor. For the sake of simplicity, no body forces are considered in state equation. $\boldsymbol{\tau}(\phi)$ is the level set-dependent (pressure) load and is defined in terms of ϕ . The isotropic elasticity tensor $\mathbf{C}(\phi)$ is given by

$$\mathbf{C}(\phi) = H(\phi)^p \mathbf{C}_1 + (1 - H(\phi)^p) \mathbf{C}_2, \quad H(\phi(\mathbf{x})) = \begin{cases} 1 & , \quad \text{if } \phi(\mathbf{x}) \geq 0, \\ 0 & , \quad \text{if } \phi(\mathbf{x}) < 0, \end{cases} \quad (7)$$

where $H(\phi(\mathbf{x}))$ is the Heaviside function and elasticity tensors \mathbf{C}_1 and \mathbf{C}_2 account for the properties of materials 1 and 2, respectively, such that $\|\mathbf{C}_2\| \ll \|\mathbf{C}_1\|$. Penalization parameter p is identical to that of the solid isotropic material with

penalization method. The reasons for the inclusion of an exponent $p > 1$ in Equation (7) are twofold: Firstly, although exponent p has no effect on the exact Heaviside function, its use on a discretized counterpart of $\mathbf{C}(\phi)$, shown in Section 4, will lead to the classical ersatz material model for those elements cut by the implicit boundary $\partial\Omega$. Secondly, it plays a formal role in the derivative of $\mathbf{C}(\phi)$ with respect to function ϕ as explained in Section 3.1.

The structural topology optimization problem considered here is the so-called compliance problem (strain energy minimization) formally stated as

Problem P₁:

$$\begin{aligned} \min_{\phi} \quad & W(\mathbf{u}, \phi) = \int_D \frac{1}{2} \mathbf{C}(\phi) \boldsymbol{\varepsilon}(\mathbf{u}) \cdot \boldsymbol{\varepsilon}(\mathbf{u}) \, dD, \\ \text{subject to :} \quad & \begin{cases} a_{\phi}(\mathbf{u}, \mathbf{v}) = l_{\phi}(\mathbf{v}), & \forall \mathbf{v} \in V, \\ g(\phi) = G(\phi) - \bar{V} \leq 0, \end{cases} \end{aligned} \quad (8)$$

where

$$G(\phi) = \int_D H(\phi) \, dD, \quad (9)$$

is the volume (or area) occupied by the solid material and \bar{V} the required volume (or area) at the end of the optimization process.

2.2 | Augmented Lagrangian approach

The volume constraint is included in the objective function based on an augmented Lagrangian approach.^{37,38} Consequently, problem (8) is rewritten as

Problem P₂: At iteration k , for given values of penalization factor c^k and Lagrange multiplier α^k , solve the minimization

$$\begin{aligned} \min_{\phi} \quad & J^k(\phi) = \int_D \frac{1}{2} \mathbf{C}(\phi) \boldsymbol{\varepsilon}(\mathbf{u}) \cdot \boldsymbol{\varepsilon}(\mathbf{u}) \, dD + \alpha^k h(\phi) + \frac{c^k}{2} h(\phi)^2, \\ \text{subject to :} \quad & a_{\phi}(\mathbf{u}, \mathbf{v}) = l_{\phi}(\mathbf{v}), \quad \forall \mathbf{v} \in V, \end{aligned} \quad (10)$$

where \mathbf{u} is the solution of Equation (4) and $h(\phi)$ is given by

$$h(\phi) = \max \left\{ g(\phi); -\frac{\alpha^k}{c^k} \right\}. \quad (11)$$

Once the minimum is achieved (or after a conveniently specified number of iterations), verify the condition

$$\left| \alpha^k h(\phi)^k \right| < \varepsilon, \quad (12)$$

where ε is small tolerance. If (12) is not satisfied, update the Lagrange multiplier and the penalization factor

$$\begin{aligned} \alpha^{k+1} &= \max \{ \alpha^k + c^k h(\phi)^k; 0 \}, \\ c^{k+1} &= \beta c^k, \quad \beta > 1, \quad 0 < c^k < c_{\max}. \end{aligned} \quad (13)$$

Let $k = k + 1$ and restart the process. The sequence $\{\alpha^k\}$, $k = 1, 2, \dots$ should converge to the Lagrange multiplier α that satisfies the necessary optimality conditions.

3 | LEVEL SET EVOLUTION VIA REACTION-DIFFUSION

The level set evolution technique proposed by Yamada et al²³ is used in the present study. The objective function $J(\phi)$ holds an additional regularization term, and the minimization problem to be solved becomes

$$\min_{\phi} \quad J_R(\phi) = J(\phi) + \frac{1}{2} \tau \int_D |\nabla \phi|^2 \, dD, \quad (14)$$

where $J_R(\phi)$ is named *regularized objective function* and $\tau > 0$ is a regularization (diffusion) weighting coefficient. The rate of ϕ is defined to be proportional to the derivative of $J_R(\phi)$

$$\frac{\partial \phi}{\partial t} = -\frac{dJ_R(\phi)}{d\phi} = -\frac{dJ(\phi)}{d\phi} + \tau \nabla^2 \phi. \quad (15)$$

The formal procedure to obtain $dJ_R(\phi)/d\phi$ in the above equation can be found in the work of Emmendoerfer and Fancello.³⁹ The resulting evolution problem with boundary and initial conditions is summarized as follows:

$$\begin{cases} \frac{\partial \phi}{\partial t} = \tau \nabla^2 \phi - \frac{dJ(\phi)}{d\phi}, & \text{in } D \\ \phi(t=0) = \phi_0, & \text{in } D \cup \partial D \\ \frac{\partial \phi}{\partial \mathbf{n}} = 0, & \text{on } \partial D \setminus (\Gamma_D \cup \Gamma_f). \\ \phi > 0, & \text{on } \Gamma_D \\ \phi < 0, & \text{on } \Gamma_f. \end{cases} \quad (16)$$

Expression (16)₁ is a reaction-diffusion equation, where $\tau \nabla^2 \phi$ and $dJ(\phi)/d\phi$ are known as diffusion and reaction terms, respectively. The derivative $dJ(\phi)/d\phi$ brings the essential information for convenient changes of shape and is usually named *velocity field* in level set-based topology optimization problems. Its calculation is detailed in the next section. One of the main advantages of using this proposition is to avoid reinitialization steps of the level set function. In addition, with the aid of some convenient choices, it allows creating new holes in the design domain.

3.1 | Sensitivity analysis

This section focuses on the calculus of derivative $dJ(\phi)/d\phi$ of the reaction-diffusion equation in (16). To this aim, a Lagrangian function \mathcal{L} defined as the sum of objective function $J(\phi)$ (10) and state equation constraint (4) is provided

$$\begin{aligned} \mathcal{L}(\phi, \mathbf{u}, \lambda) &= J(\phi) + a_\phi(\mathbf{u}, \lambda) - l_\phi(\lambda) \\ &= \int_D \frac{1}{2} \mathbf{C}(\phi) \boldsymbol{\varepsilon}(\mathbf{u}) \cdot \boldsymbol{\varepsilon}(\mathbf{u}) \, dD + \alpha h(\phi) + \frac{c}{2} h(\phi)^2 + a_\phi(\mathbf{u}, \lambda) - l_\phi(\lambda), \end{aligned} \quad (17)$$

where $\lambda \in V$ is a Lagrangian multiplier. Considering fields ϕ , \mathbf{u} , and λ as independent variables, the total variation of \mathcal{L} is then

$$\delta \mathcal{L}(\phi, \mathbf{u}, \lambda) = \frac{\partial \mathcal{L}}{\partial \phi} [\delta \phi] + \frac{\partial \mathcal{L}}{\partial \mathbf{u}} [\delta \mathbf{u}] + \frac{\partial \mathcal{L}}{\partial \lambda} [\delta \lambda], \quad (18)$$

where $\delta \phi, \delta \mathbf{u}, \delta \lambda \in V$ are admissible variations of their respective arguments. Satisfaction of conditions

$$\frac{\partial \mathcal{L}}{\partial \lambda}(\phi, \mathbf{u}, \lambda) [\delta \lambda] = 0 \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial \mathbf{u}}(\phi, \mathbf{u}, \lambda) [\delta \mathbf{u}] = 0 \quad (19)$$

retrieve state equation (4) and provide the adjoint field $\lambda = -\mathbf{u}$, respectively (see, for instance, the work of Allaire et al¹⁶).

Finally, the partial derivative of \mathcal{L} with respect to ϕ is

$$\frac{\partial \mathcal{L}}{\partial \phi} [\delta \phi] = \frac{\partial J(\phi)}{\partial \phi} [\delta \phi] + \frac{\partial a_\phi(\mathbf{u}, \lambda)}{\partial \phi} [\delta \phi] - \frac{\partial l_\phi(\lambda)}{\partial \phi} [\delta \phi]. \quad (20)$$

Calculating each term separately, we obtain

$$\begin{aligned} \frac{\partial J(\phi)}{\partial \phi} [\delta \phi] &= \int_D \frac{1}{2} p(\mathbf{C}_1 - \mathbf{C}_2) H(\phi)^{p-1} \boldsymbol{\varepsilon}(\mathbf{u}) \cdot \boldsymbol{\varepsilon}(\mathbf{u}) \delta(\phi) \delta \phi \, dD \\ &\quad + [\alpha + ch(\phi)] \frac{\partial h(\phi)}{\partial \phi} [\delta \phi], \end{aligned} \quad (21)$$

$$\frac{\partial a_\phi(\mathbf{u}, \lambda)}{\partial \phi} [\delta \phi] = \int_D p(\mathbf{C}_1 - \mathbf{C}_2) H(\phi)^{p-1} \boldsymbol{\varepsilon}(\lambda) \cdot \boldsymbol{\varepsilon}(\mathbf{u}) \delta(\phi) \delta \phi \, dD, \quad (22)$$

$$\frac{\partial l_\phi(\lambda)}{\partial \phi} [\delta \phi] = \int_D \left[\lambda \cdot \frac{\partial \boldsymbol{\tau}(\phi)}{\partial \phi} \|\nabla \phi\| + \text{div}((\boldsymbol{\tau}(\phi) \cdot \lambda) \mathbf{n}) \right] \delta(\phi) \delta \phi \, dD. \quad (23)$$

See details of the derivative of $l_\phi(\lambda)$ in Appendix A. Note that we used the definition

$$\frac{\partial H(\phi)}{\partial \phi} [\delta \phi] = \delta(\phi) \delta \phi,$$

where $\delta(\phi)$ is the Dirac function.

Since $h(\phi) = \max \left\{ g(\phi); -\frac{\alpha}{c} \right\}$, two possible expressions arise for its derivative

1. If $h(\phi) = g(\phi)$,

$$\frac{\partial h(\phi)}{\partial \phi} [\delta \phi] = \int_D \delta(\phi) \delta \phi \, dD. \quad (24)$$

2. If $h(\phi) = -\frac{\alpha}{c}$,

$$\frac{\partial h(\phi)}{\partial \phi} [\delta \phi] = 0. \quad (25)$$

For a given set $(\phi, \mathbf{u}, \lambda)$ satisfying state and adjoint equations in (19), the total variation of the Lagrangian function reduces to its partial derivative with respect to ϕ (see Equation (18)).¹⁶ Therefore,

$$\delta \mathcal{E}(\phi, \mathbf{u}, \lambda) = \frac{\partial \mathcal{E}}{\partial \phi} [\delta \phi] = \frac{dJ(\phi)}{d\phi} [\delta \phi]. \quad (26)$$

Substituting (21)-(25) in (20) and taking into account that $\lambda = -\mathbf{u}$, the required total derivative is given by

$$\frac{dJ(\phi)}{d\phi} [\delta \phi] = \int_D V(\phi) \delta(\phi) \delta \phi dD \quad (27)$$

and yields the expression (see the work of Gurtin⁴⁰)

$$\frac{dJ(\phi)}{d\phi} = V(\phi) \delta(\phi) \quad (28)$$

for the variational derivative, where

$$V(\phi) = \begin{cases} -\frac{1}{2} p(\mathbf{C}_1 - \mathbf{C}_2) H(\phi)^{p-1} \boldsymbol{\varepsilon}(\mathbf{u}) \cdot \boldsymbol{\varepsilon}(\mathbf{u}) + \operatorname{div}((\boldsymbol{\tau}(\phi) \cdot \mathbf{u}) \mathbf{n}) \\ \quad + \mathbf{u} \cdot \frac{\partial \boldsymbol{\tau}(\phi)}{\partial \phi} \|\nabla \phi\| + \alpha + cg(\phi), & \text{if } g(\phi) \geq -\frac{\alpha}{c}, \\ -\frac{1}{2} p(\mathbf{C}_1 - \mathbf{C}_2) H(\phi)^{p-1} \boldsymbol{\varepsilon}(\mathbf{u}) \cdot \boldsymbol{\varepsilon}(\mathbf{u}) + \operatorname{div}((\boldsymbol{\tau}(\phi) \cdot \mathbf{u}) \mathbf{n}) \\ \quad + \mathbf{u} \cdot \frac{\partial \boldsymbol{\tau}(\phi)}{\partial \phi} \|\nabla \phi\|, & \text{if } g(\phi) < -\frac{\alpha}{c} \end{cases} \quad (29)$$

is a (velocity) field defined over D . Observe that the Heaviside function is always present in the first term of the right side of (29) because of the exponent $p > 1$ in expression (7). Thus, the Heaviside function controls the inclusion/exclusion of this term, depending on whether the point is inside or outside the material region. As a consequence, islands of material from the design space that arise during optimization are eliminated easier. Note also that the expression for derivative $\partial \boldsymbol{\tau}(\phi) / \partial \phi$ in (29) is still missing. Its calculus should be consistent with the algorithmic expressions chosen for the calculus of the boundary-dependent pressure $\boldsymbol{\tau}(\phi)$. Corresponding details are found in Section 4.

3.2 | Treatment and regularization on the velocity field

In this section, a sequence of operations on the velocity field $V(\phi)$ is proposed to improve the convergence of the minimization sequence. By keeping the descent properties of $V(\phi)$, such operations consist of:

1. Eliminating the Dirac function in (28) to extend the reaction term to the whole domain

$$\frac{dJ}{d\phi} = V(\phi). \quad (30)$$

It is thus possible to generate changes in the level set function that leads to the nucleation of new holes.

2. Applying the logarithmic scaling,

$$V_{\log}(\phi) = \begin{cases} \ln(V(\phi) + 1), & \text{if } V(\phi) \geq 0, \\ -\ln(-V(\phi) + 1), & \text{if } V(\phi) < 0. \end{cases} \quad (31)$$

This step mitigates high variations in the amplitude of $V(\phi)$; as a result, the convergence of the optimization is accelerated.

3. Solving the variational problem: Find $v(\phi) \in H^1(D)$ such that

$$\int_D (\kappa \nabla v(\phi) \cdot \nabla X + v(\phi) X) dD = \int_D X V_{\log}(\phi) dD, \quad \forall X \in H^1(D), \quad (32)$$

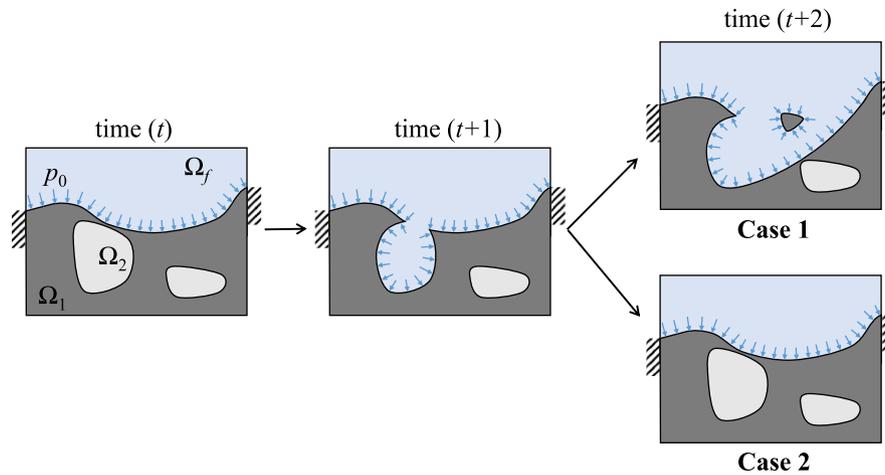


FIGURE 2 Possible cases of topology changes that can occur during level set movement [Colour figure can be viewed at wileyonlinelibrary.com]

where $\kappa > 0$ is a small smoothing regularization parameter and X is a weight function in H^1 space. Thus, $v(\phi)$ is smoother than $V_{\log}(\phi)$ because of the classical regularity theory for elliptic equations.⁴¹

Therefore, $v(\phi)$ is the velocity field finally used in the reaction-diffusion equation system (16) as the reaction term responsible for feasible topology changes that should decrease the objective function.

3.3 | Level set topology changes on the loading boundary

During optimization, the algorithm allows merging pressurized (wet) and traction-free boundaries. Consider the configuration of a structure at time (t) subject to the pressurized loading shown in Figure 2. The present approach allows the rupture of loaded boundary and consequent load transfer to the previous inner surface (see configuration at time $(t + 1)$). For a new configuration at time $(t + 2)$, two cases are allowed to occur.

- Case 1: The topology evolves to a design with islands of solid material within Ω_f .
- Case 2: The discontinued boundary at time $(t + 1)$ closes and go back to a configuration equal to that of time (t) , ie, a depressurized cavity. Creation of pressurized cavities are not within the goal of the present approach and therefore not considered in the present implementation.

4 | DISCRETIZATION AND NUMERICAL IMPLEMENTATION

Important aspects of the numerical implementation are discussed in this section. We proposed an approach to simulate design-dependent loads within the context of the classical topology optimization fixed-grid paradigm (ie, preserving the elements cut by the boundary) to reduce the computational complexity associated with the domain shape change. The same fixed mesh of quadrilateral bilinear elements is used for both, the level set description and the finite element analysis. Thus, the classical ersatz material model is used to compute integrals over all elements. This model relates the exact Heaviside function $H(\phi)$ with solid material fraction $a_e(\phi)$ based on the following integral over domain element D_e :

$$a_e(\phi) = \frac{\int_{D_e} H(\phi) dD}{\int_{D_e} dD} \quad (33)$$

This approach is significant to those elements cut by the boundary containing two different phases simultaneously. In this way, elasticity tensor $\mathbf{C}(\phi)$ defined in (7)₁ is approximated by $\mathbf{C}^e(\phi) = a^e(\phi)^p \mathbf{C}_1 + [1 - a^e(\phi)^p] \mathbf{C}_2$.

The same numerical scheme of Yamada et al²³ and Emmendoerfer and Fancello³⁹ is used here to solve the reaction-diffusion equation (16)₁, where the time discretization is performed by finite differences and domain D is

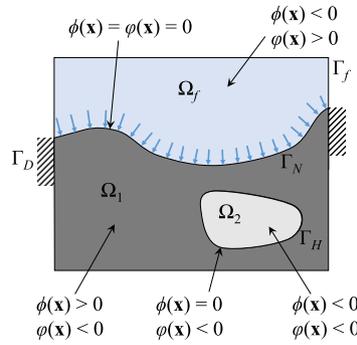


FIGURE 3 Implicit representation of functions $\phi(\mathbf{x})$ and $\varphi(\mathbf{x})$ on the working domain [Colour figure can be viewed at wileyonlinelibrary.com]

discretized based on the FEM. The updated level set field Φ_{t+1} is then obtained by solving the discretized evolution equation

$$(\mathbf{T}_1 + \Delta t(\tau \mathbf{T}_2)) \Phi_{t+1} = \mathbf{T}_1 (\Phi_t - \Delta t \mathbf{v}), \quad (34)$$

where all the boundary conditions of (16) are considered, Δt is the time increment, \mathbf{v} is the velocity array of the discrete version of $v(\phi)$, and

$$\mathbf{T}_1 = \bigcup_e \int_{D_e} \mathbf{N}^T \mathbf{N} dD, \quad (35)$$

$$\mathbf{T}_2 = \bigcup_e \int_{D_e} \nabla \mathbf{N}^T \nabla \mathbf{N} dD. \quad (36)$$

Here, \bigcup_e represents the union set of elements e , and \mathbf{N} is the interpolation function.

Similarly, problem (32) is also solved using FEM, and array \mathbf{v} is obtained from the linear equation system

$$(\mathbf{T}_1 + \kappa \mathbf{T}_2) \mathbf{v} = \mathbf{T}_1 \mathbf{V}_{\log}. \quad (37)$$

In numerical practice, $\kappa = 1-2 \times (\Delta x)^2$ is used, where Δx is the minimum grid size.

4.1 | Tracking of the moving loaded boundary

In order to track the part of the level set ϕ that indicates the moving loaded boundary, an auxiliary function φ is conveniently defined such that $\varphi(\mathbf{x}(t)) = \phi(\mathbf{x}(t)) = 0$ for all time t and $\forall \mathbf{x} \in \Gamma_N$ (see Figure 3)

$$\varphi(\mathbf{x}) = \begin{cases} > 0 & \text{if } \mathbf{x} \in \Omega_f; \mathbf{x} \in \Gamma_f \\ = 0 & \text{if } \mathbf{x} \in \Gamma_N \\ < 0 & \text{if } \mathbf{x} \in D \setminus \bar{\Omega}_f. \end{cases} \quad (38)$$

It is worth emphasizing that this function plays a merely operational role. It allows the identification of the portion of the level set function ϕ related to Γ_N . In this sense, this approach is completely different from that of Xia et al,¹⁵ where two level set functions evolve according a differential equation.

4.2 | Discretization of the pressure load on the moving boundary

Since the loaded boundary has an implicit level set-based representation, nodes are not generally attached to the boundary (see Figure 4). Therefore, a consistent discretization must be defined to transfer surface loads to equivalent nodal forces. To this aim, a scheme similar to that in the work of Lee and Martins⁶ is applied.

Figure 5A shows the constant pressure load p_0 acting on a zero level set curve within an element e . The boundary Γ_N intersects the element boundary at points (c_1, d_1) and (c_2, d_2) (see Figure 5B). A resultant force, located at middle point (\bar{c}, \bar{d}) of the segment of length L , can be calculated as

$$\mathbf{P}_e = -p_0 L \mathbf{n}. \quad (39)$$

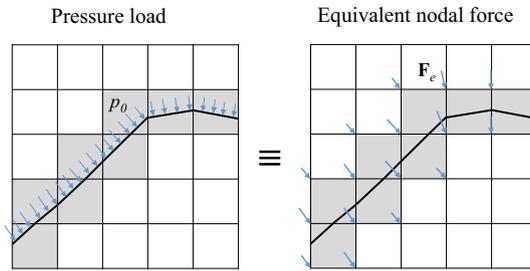


FIGURE 4 Determination of the equivalent nodal load from the pressure load by using a four-node quadrilateral element [Colour figure can be viewed at wileyonlinelibrary.com]

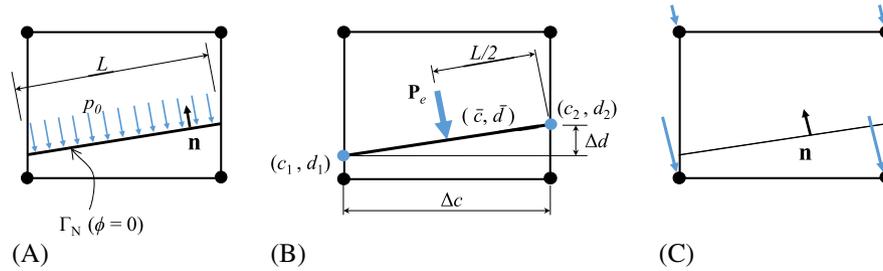


FIGURE 5 Pressure load and its equivalent concentrated force on a zero level set curve crossing a four-node element [Colour figure can be viewed at wileyonlinelibrary.com]

From Figure 5B, unit normal \mathbf{n} can be expressed by*

$$\mathbf{n} = \mathbf{S}_e \begin{bmatrix} \Delta d/L \\ \Delta c/L \end{bmatrix}, \quad \mathbf{S}_e = \begin{bmatrix} \text{sign}(n_x) & 0 \\ 0 & \text{sign}(n_y) \end{bmatrix}, \quad (40)$$

where matrix \mathbf{S}_e contains the sign of the components of normal \mathbf{n} . Substituting (40) in (39), the resultant force becomes

$$\mathbf{P}_e = p_0 \mathbf{S}_e \begin{bmatrix} \Delta d \\ \Delta c \end{bmatrix}. \quad (41)$$

The nodal loads \mathbf{F}_e equivalent to \mathbf{P}_e can be extrapolated to the nodes by⁴²

$$\mathbf{F}_e = \mathbf{N}^T(\bar{c}, \bar{d}) \mathbf{P}_e, \quad (42)$$

where $\mathbf{N}^T(\bar{c}, \bar{d})$ is obtained by evaluating interpolation function \mathbf{N} at the coordinates of point (\bar{c}, \bar{d}) to which force \mathbf{P}_e is applied. Thus, by substituting (41) into (42), the equivalent and consistent nodal load becomes

$$\mathbf{F}_e = p_0 \mathbf{N}^T(\bar{c}, \bar{d}) \mathbf{S}_e \begin{bmatrix} \Delta d \\ \Delta c \end{bmatrix}. \quad (43)$$

On the basis of this representation, the calculus of $\partial \tau(\phi)/\partial \phi$ in (29) is equivalent to the calculus of $\partial \mathbf{F}_e/\partial \phi$ given by the following expression:

$$\frac{\partial \mathbf{F}_e}{\partial \phi} = p_0 \frac{\partial \mathbf{N}^T(\bar{c}, \bar{d})}{\partial \phi} \mathbf{S}_e \begin{bmatrix} \Delta d \\ \Delta c \end{bmatrix} + p_0 \mathbf{N}^T(\bar{c}, \bar{d}) \mathbf{S}_e \frac{\partial}{\partial \phi} \begin{bmatrix} \Delta d \\ \Delta c \end{bmatrix}. \quad (44)$$

The procedures to obtain this derivative are found in Appendix B.

*To determine the direction which the loading is being applied to, we use the level set function to define

$$\mathbf{n} = -\frac{\nabla \phi}{\|\nabla \phi\|},$$

where \mathbf{n} is the outward unit normal of the boundary.

4.3 | Optimization algorithm

The minimization of problem $P_2(10)$ follows the classical procedure of the augmented Lagrangian method (see, for example, the works of Bertsekas³⁷ and Birgin and Martínez³⁸). The algorithm is summarized as follows.

External loop:

1. Initialize level set function ϕ_0 .
2. Map the elements containing boundary Γ_N and define φ .
3. Define $k = 1$, $c^k > 0$, $\alpha^k \in \mathbb{R}$.
4. Perform the *internal loop* to minimize function $J_R^k(\phi, \alpha^k, c^k)$ obtaining ϕ^k and φ^k .
5. Update c^k and α^k using (13).
6. $k = k + 1$. Return to step 4.

Step 4 of the *external loop*, called *internal loop*, is the stage in which J_R is minimized for fixed values of α^k and c^k . In this work, we used a fixed number of minimization iterations, say N_{iter} , to be accomplished prior to the penalization factor and Lagrange multiplier updating in step 5. The internal iterative procedure is the following.

Internal loop: for $j = 1$ to $j \leq N_{\text{iter}}$:

1. Obtain the discretized field \mathbf{u}_j by solving Equation (4).
2. Compute velocity field $v_j(\phi_j, \mathbf{u}_j)$.
3. Update ϕ using (34) during a time integration period $\Delta t_j = m\Delta t_{\text{CFL}}$ such that $J_R(\phi_{j+1}) \leq J_R(\phi_j)$.
 - (a) At each time step Δt_{CFL} , update φ at nodes i of elements e containing boundary Γ_N

$$\varphi_i^e = -\text{sign}(\phi_i^e), \quad \forall e \text{ such that } \Gamma_N \subset e, \quad i = 1, \dots, 4.$$
 - (b) Update the elements containing boundary Γ_N using φ .
 - (c) Limit ϕ between -1 and 1 .
4. If $|J_R(\phi_{j+1}) - J_R(\phi_j)| \leq \varepsilon$, then stop the iterative process; otherwise, continue.
5. $j = j + 1$. Go to step 1.

Some comments on the optimization algorithm above are necessary. In relation to time increment Δt , for a given velocity $v(\phi)$, the reaction-diffusion equation performs m update steps during time increment Δt , that is,

$$\Delta t_j = m\Delta t_{\text{CFL}}, \quad (45)$$

where Δt_{CFL} satisfies a Courant-Friedrichs-Lewy (CFL) condition. Small-time steps Δt_{CFL} are necessary for function φ to properly track the loading interface Γ_N along the level set movement. In this work, $\Delta t_{\text{CFL}} = 0.1 \times \min(\Delta x)$ is used. Number m is conveniently chosen to satisfy a descent condition given in step 3 of the internal minimization loop. Moreover, for numerical issues, the level set function has upper and lower limit constraints. In this work, ϕ is limited between $-1 \leq \phi(\mathbf{x}) \leq 1$ (step 3(c) of the internal loop).

5 | NUMERICAL EXAMPLES

The effectiveness of the optimization algorithm described above is evaluated in this section by means of numerical examples in 2D. In all the examples, the material properties are set as: Poisson's ratio $\nu = 0.3$, Young's modulus $E_1 = 1$ (for the solid material), and $E_2 = 1 \times 10^{-3}$ (for the void/weak material). We assume that pressurized region Ω_f has the same material properties of the compliant material in Ω_2 . The penalization parameter is fixed as $p = 3$. Lagrange multiplier α is initialized as zero and updated based on expression (13)₁. The number of iterations of the internal optimization loop was heuristically chosen as $N_{\text{iter}} = 15$ for all the examples. In the convergence diagrams, each iteration corresponds to a single solution of evolution equation (34). The volume ratio referred to along the text is the ratio between the final and total volume of the structure. We assume that the structures are under plane stress state.

5.1 | Example 1

The first example is a classical benchmark for topology optimization under pressure loads.^{1,2} The problem consists of a structure with a pressure applied at the bottom boundary as shown in Figure 6. Domain D is a rectangle of size 2.0×1.0 and clamped at the specified position according to Figure 6. The pressure is set to $p_0 = 1$. A mesh of 200×100 elements is used to discretize the fixed design domain. The optimization parameters are set as $c = 10$, $\beta = 2.5$, $\tau = 8 \times 10^{-3}$, and a number of $k = 8$ external iterations (therefore, $8 \times 15 = 120$ iterations). The initial level set domain is shown in Figure 7. In this example, some intermediate topologies are presented in Figure 8 to show the possible cases of level set topology

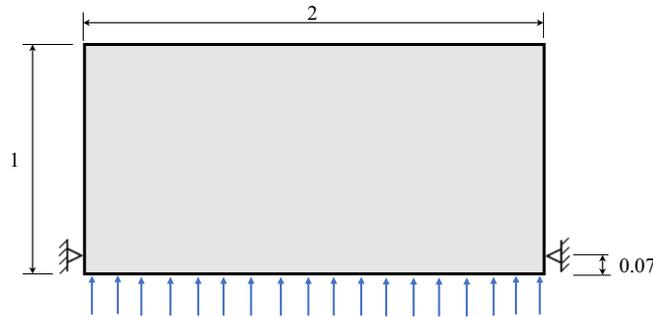


FIGURE 6 Example 1: Model [Colour figure can be viewed at wileyonlinelibrary.com]

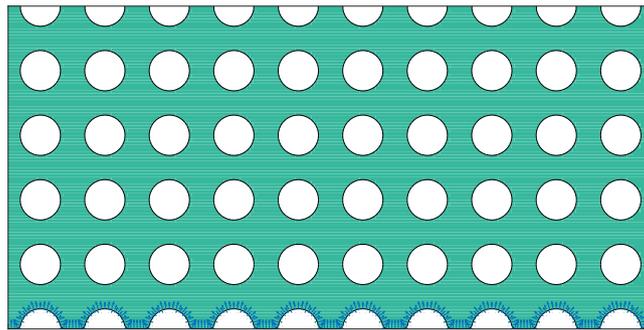


FIGURE 7 Example 1: Initial level set domain [Colour figure can be viewed at wileyonlinelibrary.com]

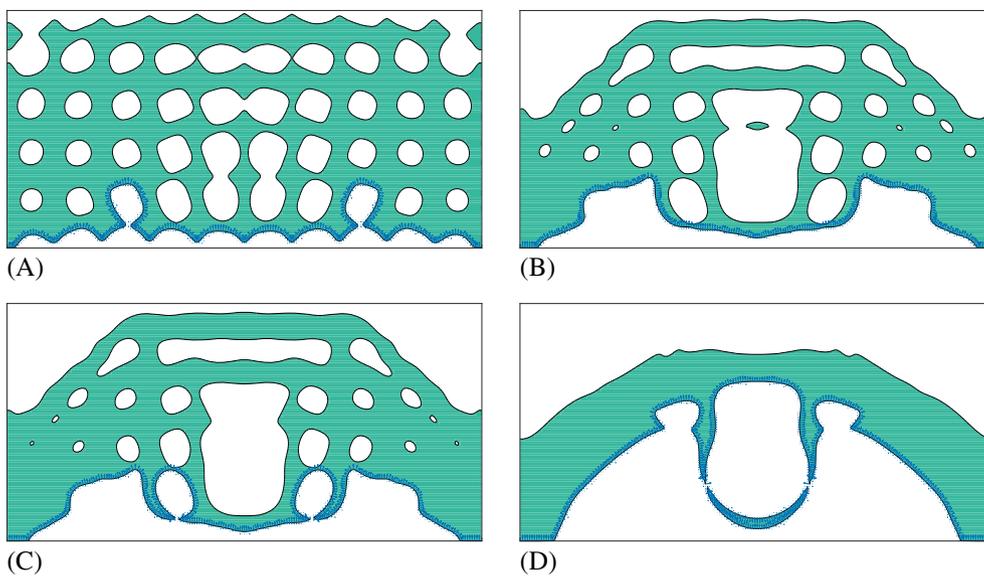


FIGURE 8 Example 1: Intermediate results. A, Iteration 2; B, Iteration 5; C, Iteration 6; D Iteration 20 [Colour figure can be viewed at wileyonlinelibrary.com]

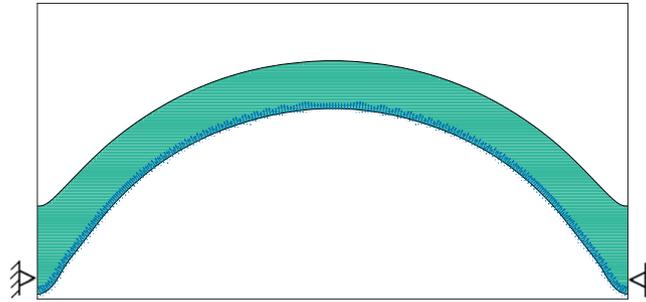


FIGURE 9 Example 1: Final result [Colour figure can be viewed at wileyonlinelibrary.com]

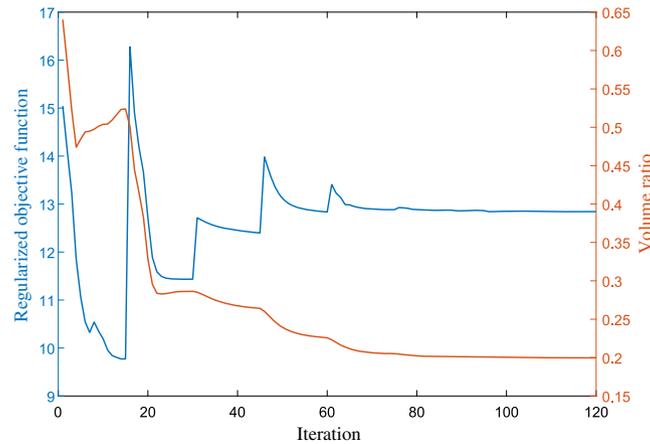


FIGURE 10 Example 1: Convergence history

changes considered along the optimization procedure (see Section 3.3). Note that the pressure boundary can evolve freely during optimization. Moreover, the figures show the pressure force vectors acting on the interface. The resulting arc-tie structure for a maximum volume ratio of 20% is illustrated in Figure 9. The optimized design has a compliance $W = 12.06$. Figure 10 shows the convergence history of the regularized objective function and volume ratio. Observe the fluctuations of the regularized objective function. They occur after each updating of Lagrange multiplier α and penalization factor c (mainly the first iterations), configuring the typical behavior of convergence in augmented Lagrangian approaches.

5.2 | Example 2

The second example deals with the problem described in Figure 11 and refers to the same rectangular domain (same finite element mesh) described in the previous example. Pressure $p_0 = 1$ is applied at the top and lateral boundaries. Two points at the bottom are clamped in different positions constituting models 1 and 2, as shown in Figure 11. The optimization

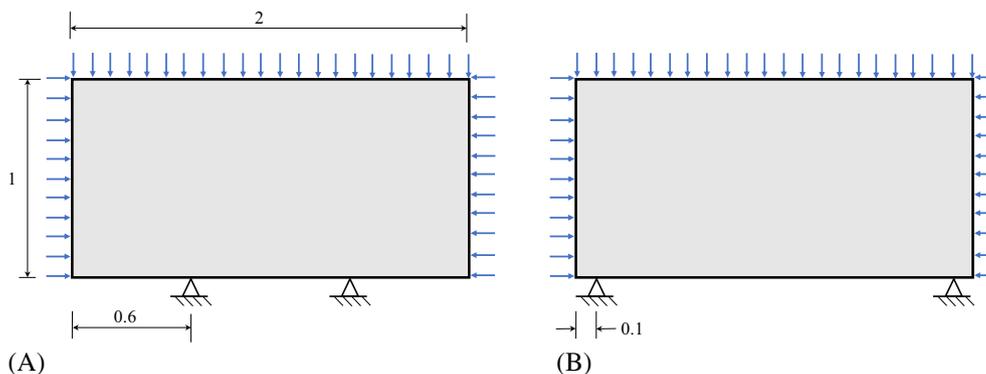


FIGURE 11 Example 2: Models 1 and 2 [Colour figure can be viewed at wileyonlinelibrary.com]

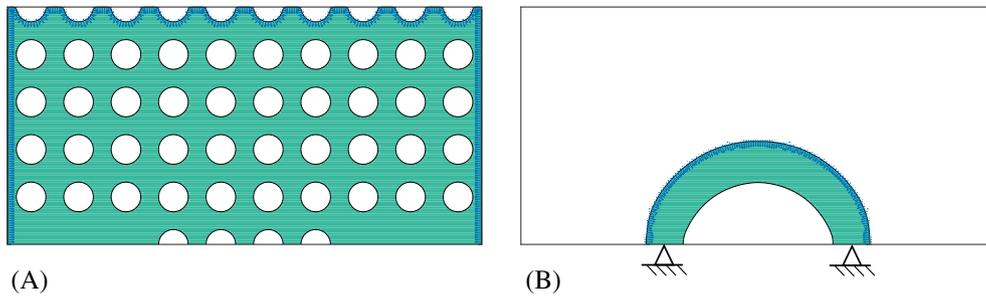


FIGURE 12 Example 2: Optimization of Model 1. A, Initial level set domain; B, Optimized topology for a volume ratio of 10% [Colour figure can be viewed at wileyonlinelibrary.com]

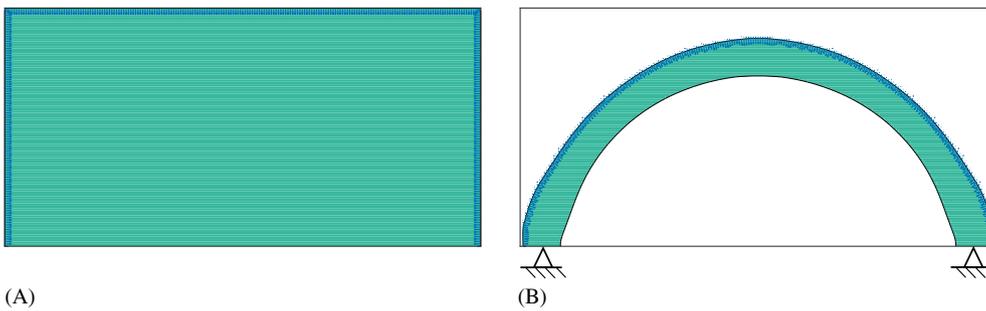


FIGURE 13 Example 2: Optimization of Model 2. A, Initial level set domain; B, Optimized topology for a volume ratio of 20% [Colour figure can be viewed at wileyonlinelibrary.com]

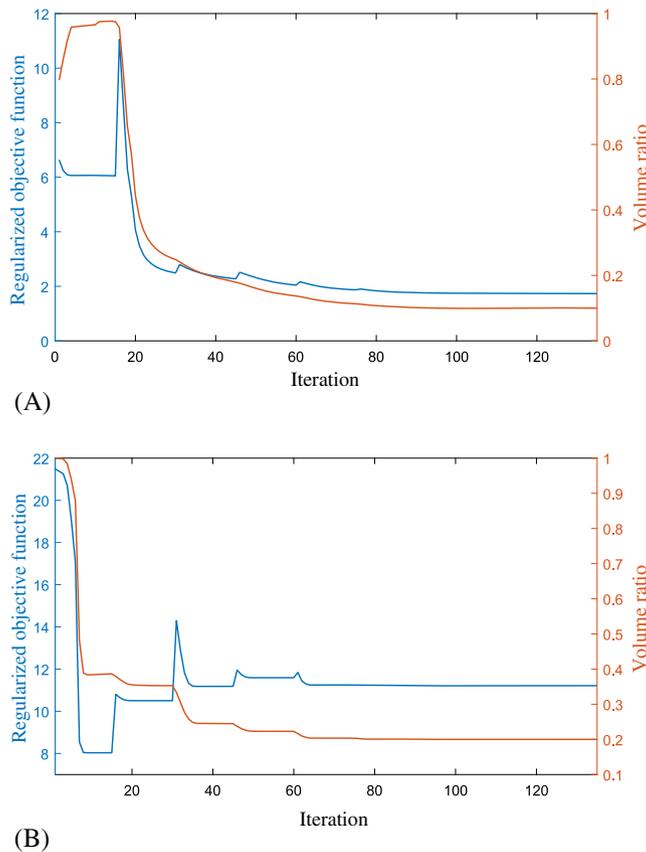


FIGURE 14 Example 2: Convergence history. A, Model 1; B, Model 2

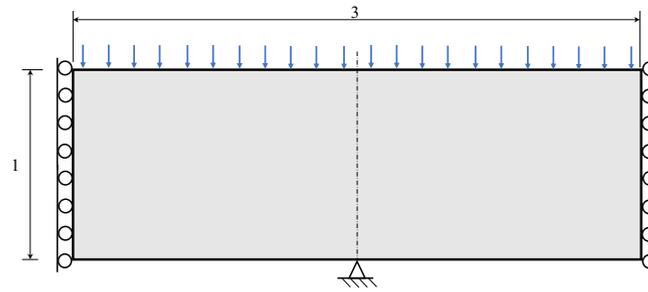


FIGURE 15 Example 3: Model [Colour figure can be viewed at wileyonlinelibrary.com]

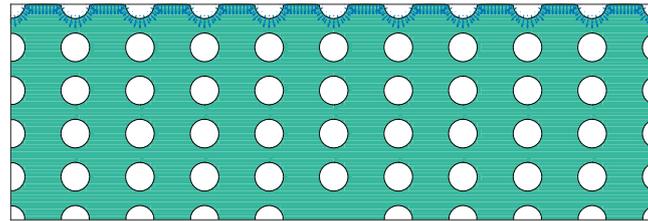


FIGURE 16 Example 3: Initial level set domain [Colour figure can be viewed at wileyonlinelibrary.com]

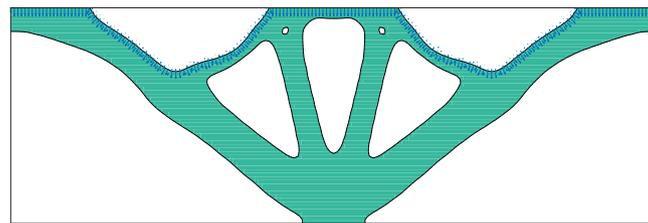


FIGURE 17 Example 3: Final result [Colour figure can be viewed at wileyonlinelibrary.com]

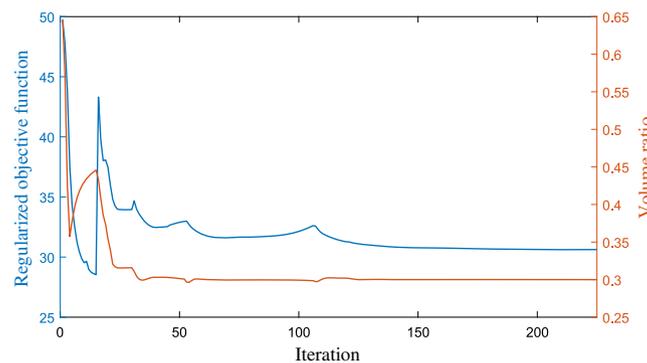


FIGURE 18 Example 3: Convergence history

is performed with penalization factor $c = 1$ (for model 1) and $c = 12$ (for model 2). The remaining parameters are the same for both models: $\beta = 2.5$, $\tau = 8 \times 10^{-3}$ and $k = 9$ external loop updatings (total of $9 \times 15 = 135$ iterations). The initial level set domain for model 1 is shown in Figure 12A. The optimized topology with volume constraint set to 10% of the total volume is presented in Figure 12B. In order to verify the capability of the proposed algorithm to provide local feasible solutions for arbitrary initializations, the optimization for model 2 starts from a solid initial configuration (that is, without voids) as shown in Figure 13A. The final topology with volume constraint set to 20% of the total volume is presented in Figure 13B. Note that the final designs converged to an arch shape as expected. The optimized designs have compliance values of $W = 1.28$ (for model 1) and $W = 10.28$ (for model 2). Figure 14 shows the convergence history of the regularized objective function and volume ratio for both models.

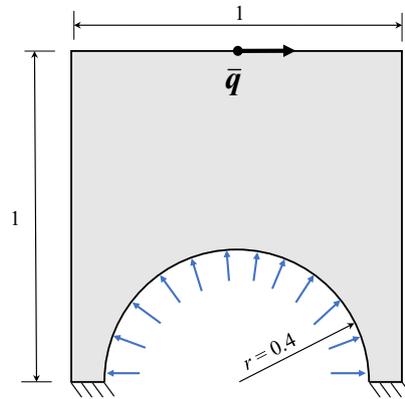


FIGURE 19 Example 4: Model [Colour figure can be viewed at wileyonlinelibrary.com]

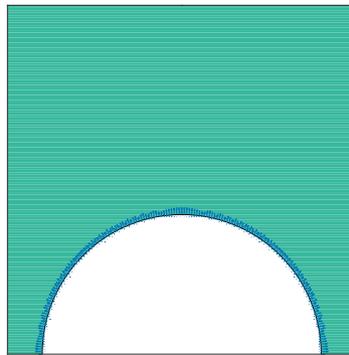


FIGURE 20 Example 4: Initial level set domain [Colour figure can be viewed at wileyonlinelibrary.com]

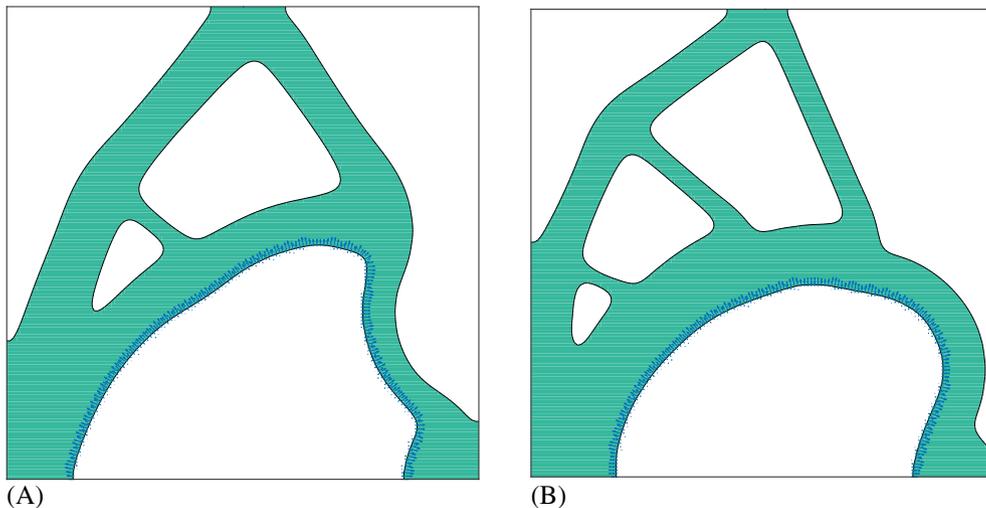


FIGURE 21 Example 4: Final results for an initial configuration with no holes. A, $p_0 = 2$, $\bar{q} = (1, 0)$; B, $p_0 = 5$, $\bar{q} = (1, 0)$ [Colour figure can be viewed at wileyonlinelibrary.com]

5.3 | Example 3

This case was introduced in the work of Sigmund and Clausen⁸ and consists of the design of a piston-like shape. Domain D is a rectangle of size 3.0×1.0 subject to a pressure load $p_0 = 1$ applied at the top boundary. On the left and right sides, the displacements along the horizontal direction are fixed. The central point of the bottom boundary is clamped as shown in Figure 15. A mesh of 180×60 elements is used to discretize D . The optimization parameters are set as: $c = 30$, $\beta = 5$, $\tau = 1 \times 10^{-3}$, and a number of $k = 15$ external iterations (total of 225 iterations). The initial level set domain is shown

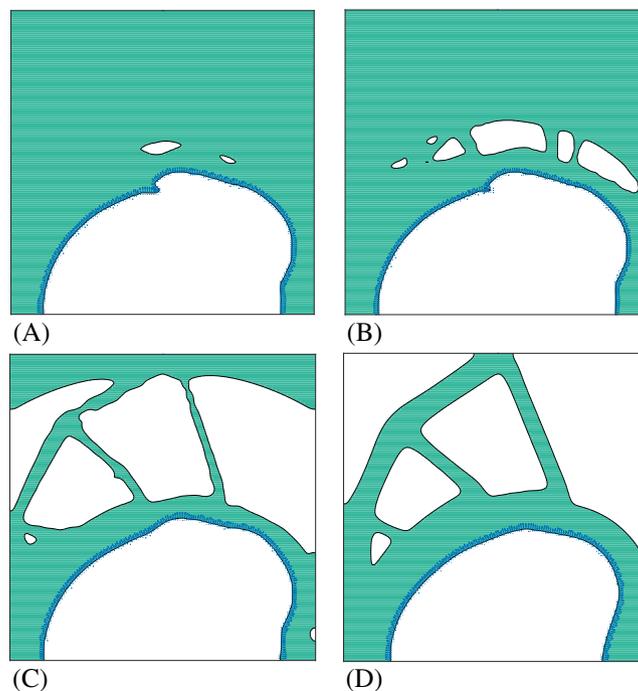


FIGURE 22 Example 4: Intermediate results of the final design in Figure 21B, with $p_0 = 5$, $\bar{\mathbf{q}} = (1, 0)$. A, Iteration 5; B, Iteration 13; C, Iteration 31; D, Iteration 80 [Colour figure can be viewed at wileyonlinelibrary.com]

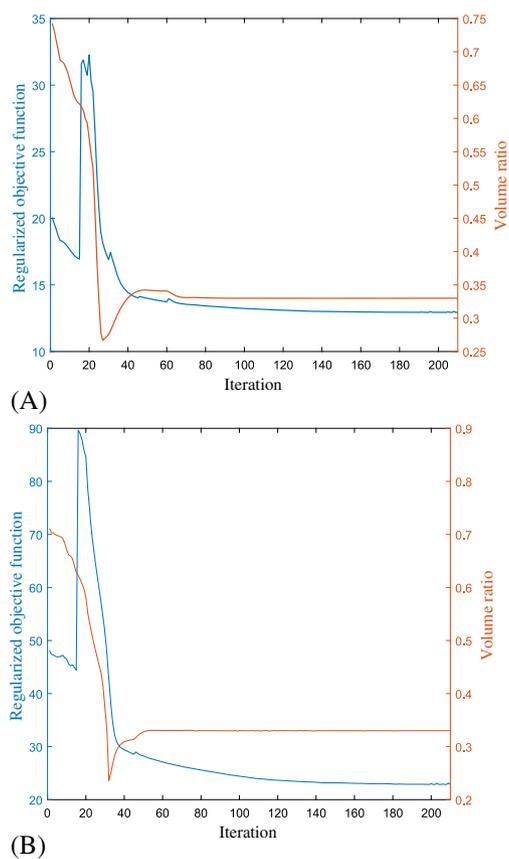


FIGURE 23 Example 4: Convergence history. A, $p_0 = 2$, $\bar{\mathbf{q}} = (1, 0)$; B, $p_0 = 5$, $\bar{\mathbf{q}} = (1, 0)$

in Figure 16. The final topology constrained to 30% of the total volume is presented in Figure 17. The optimized design has a compliance value of $W = 30.22$, and it is similar to those obtained in previous studies.^{6,8,9,15} Figure 18 shows the convergence history of the regularized objective function and volume ratio.

5.4 | Example 4

In the latter example, proposed by Xavier and Novotny,³⁶ domain D is given by a square structure of size 1.0×1.0 subjected to a pressure loading p_0 into a semicircle of radius $r = 0.4$ and center at the middle bottom of the square (see Figure 19). In addition to p_0 , a fixed horizontal load $\bar{\mathbf{q}} = (1, 0)$ is applied at the middle top of the structure. A clamped boundary condition is imposed at the bottom boundary. The mesh has 140×140 elements and the initial level set domain (no holes) is shown in Figure 20. This initialization is used to demonstrate the capacity of the present optimization algorithm to nucleate holes during the level set evolution. The volume constraint is set to 33% of the total volume, and different values of p_0 are tested. For $p_0 = 2$, Figure 21A shows the optimized design with final compliance $W = 12.76$, where the optimization parameters are: $c = 100$, $\beta = 2.5$ and $\tau = 7 \times 10^{-4}$. For $p_0 = 5$, Figure 21B presents the final topology with compliance $W = 22.73$, where the optimization parameters are: $c = 300$, $\beta = 2.5$, and $\tau = 5.5 \times 10^{-4}$. Both final results are obtained after $k = 14$ external iterations by totaling 210 iterations. Note that the solutions are comparable to those obtained by the aforementioned work,³⁶ including their observation regarding a change in the curvature of the right leg of the structure after increasing the pressure loading. Intermediate solutions of iterations 5, 13, 31, and 80 are shown in Figure 22. Here, it is possible to see the nucleation of new holes in the domain, which configures an important

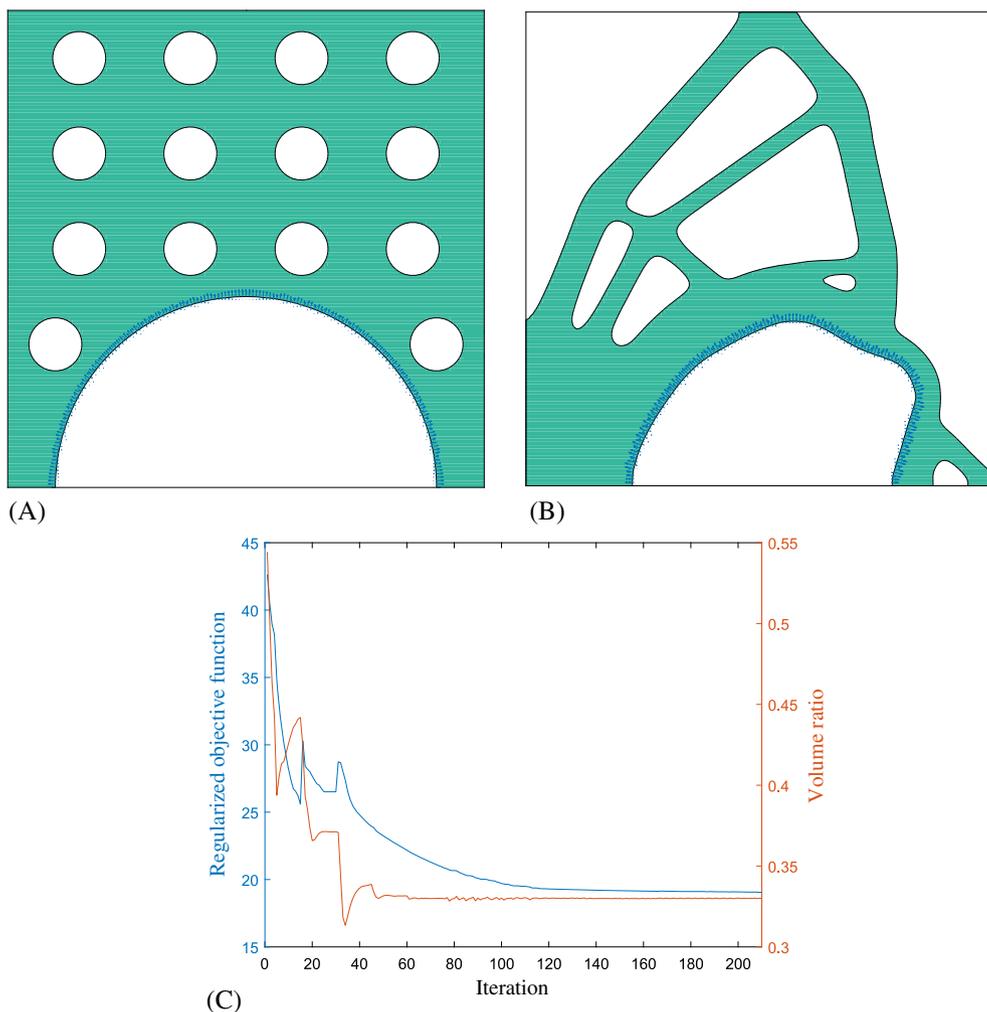


FIGURE 24 Example 4: Results for an initial level set domain with holes and $p_0 = 5$, $\bar{\mathbf{q}} = (1, 0)$. A, Initial configuration; B, Final topology; C, Convergence history

feature in a true topological optimization method. Figures 23A and 23B show the convergence history of the regularized objective function and volume ratio for pressure loads $p_0 = 2$ and 5, respectively.

The same problem for $p_0 = 5$ was again solved however starting from an initial domain with holes (see Figure 24A). The final design with compliance $W = 18.7$ is shown in Figure 24B. Note that the optimized design led to a topology significantly different including the shape of the pressurized cavity, showing the dependence of the solution on initial configurations. Moreover, the final compliance achieved was lower than that of the solid initial domain. Finally, Figure 24C refers to convergence issues, presenting the good behavior of the regularized objective function.

6 | CONCLUSION

A method to solve structural topology optimization problems with design-dependent (pressure) loads has been proposed. The approach relies on the clear identification of the moving boundaries provided by the level set technique. For the sake of simplicity, the well-established minimum compliance problem under volume constraint was analyzed. A reaction-diffusion evolutionary equation, analogous to that proposed by Yamada et al.,²³ was chosen for the level set evolution because of its capacity of avoiding reinitialization steps and, consequently, to allow for the nucleation of new holes in the domain. Just a single level set function is controlled by an evolution problem while an auxiliary function follows the level set on the loaded boundary. Being the pressure applied to an implicit boundary, a simple and efficient element-based approximation is chosen to compute equivalent nodal forces consistent with the surface load. Moreover, analytical and algorithmically consistent sensitivity expressions can be retrieved from this approximation. Finally, the proposed algorithm allows merging pressurized (wet) boundaries with traction-free boundaries during level set movements. This last property, allied to the simplicity of the level set representation and successful combination with the reaction-diffusion-based evolution applied to a design-dependent pressure load problem, represents the main contribution of this paper. All these features make the present approach quite efficient to deal with loaded moving boundaries as compared with the heuristics employed in density-based methods. Numerical examples provide successful results, many of which comparable with others found in the literature and solved with different techniques.

ACKNOWLEDGEMENTS

The first author thanks the financial support of FAPESP (The State of São Paulo Research Foundation) under grant 2016/09923-2 and CAPES (Coordination for the Improvement of Higher Education Personnel), the Brazilian Ministry of Education, Brasília, DF, 70040-020, Brazil. The first and the last authors gratefully acknowledge the support from BG/Shell Brasil and FAPESP through the Research Centre for Gas Innovation - RCGI (Fapesp Proc. 2014/50279-4), hosted by the University of Sao Paulo and the strategic importance of the support given by ANP (Brazil's National Oil, Natural Gas and Biofuels Agency) through the R&D levy regulation. The last author also acknowledges the financial support of CNPq (Brazilian National Council for Scientific and Technological Development) under grant 304121/2013-4. The second author would like to thank the financial support of CNPq under grant 309686/2013-0.

ORCID

Eduardo Alberto Fancello  <http://orcid.org/0000-0001-9085-3901>

Emilio Carlos Nelli Silva  <https://orcid.org/0000-0003-1715-1713>

REFERENCES

1. Hammer VB, Olhoff N. Topology optimization of continuum structures subjected to pressure loading. *Struct Multidisc Optim.* 2000;19:85-92.
2. Chen BC, Kikuchi N. Topology optimization with design-dependent loads. *Finite Elem Anal Des.* 2001;37:57-70.
3. Bourdin B, Chambolle A. Design-dependent loads in topology optimization. *ESAIM Control Optim Calc Var.* 2003;9:19-48. <https://doi.org/10.1051/cocv:2002070>
4. Fuchs MB, Shemesh NNY. Density-based topological design of structures subjected to water pressure using a parametric loading surface. *Struct Multidisc Optim.* 2004;28:11-19. <https://doi.org/10.1007/s00158-004-0406-z>
5. Du J, Olhoff N. Topological optimization of continuum structures with design-dependent surface loading – Part I: new computational approach for 2D problems. *Struct Multidisc Optim.* 2004;27:151-165. <https://doi.org/10.1007/s00158-004-0379-y>
6. Lee E, Martins JRRA. Structural topology optimization with design-dependent pressure loads. *Comput Methods Appl Mech Eng.* 2012;233-236:40-48.

7. Zheng B, Chang C-j, Gea C. Topology optimization with design-dependent pressure loading. *Struct Multidisc Optim*. 2009;38:535-543. <https://doi.org/10.1007/s00158-008-0317-5>
8. Sigmund O, Clausen PM. Topology optimization using a mixed formulation: an alternative way to solve pressure load problems. *Comput Methods Appl Mech Eng*. 2007;196:1874-1889. <https://doi.org/10.1016/j.cma.2006.09.021>
9. Bruggi M, Cinquini C. An alternative truly-mixed formulation to solve pressure load problems in topology optimization. *Comput Methods Appl Mech Eng*. 2009;198:1500-1512. <https://doi.org/10.1016/j.cma.2008.12.009>
10. Zhang H, Zhang X, Liu S. A new boundary search scheme for topology optimization of continuum structures with design-dependent loads. *Struct Multidisc Optim*. 2008;37:121-129. <https://doi.org/10.1007/s00158-007-0221-4>
11. Wang C, Zhao M, Ge T. Structural topology optimization with design-dependent pressure loads. *Struct Multidisc Optim*. 2016;53:1005-1018. <https://doi.org/10.1007/s00158-015-1376-z>
12. Osher S, Sethian J. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J Comput Phys*. 1988;79:12-49.
13. Sethian JA. *Level Set Method and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge, UK: Cambridge University Press; 1999.
14. Guo X, Zhao K, Gu Y. Topology optimization with design-dependent loads by level set approach. Paper presented at: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference; 2004; Albany, NY.
15. Xia Q, Wang MY, Shi T. Topology optimization with pressure load through a level set method. *Comput Methods Appl Mech Eng*. 2015;283:177-195.
16. Allaire G, Jouve F, Toader AM. Structural optimization using sensitivity analysis and a level-set method. *J Comput Phys*. 2004;194:363-393.
17. Liu Z, Korvink JG, Huang R. Structure topology optimization: fully coupled level set method via FEMLAB. *Struct Multidiscip Optim*. 2005;29(6):407-417. <https://doi.org/10.1007/s00158-004-0503-z>
18. Liu Z, Korvink JG. Adaptive moving mesh level set method for structure topology optimization. *Eng Optim*. 2008;40:529-558. <https://doi.org/10.1080/03052150801985544>
19. Wang S, Wang MY. Radial basis functions and level set method for structural topology optimization. *Int J Numer Methods Eng*. 2006;65(12):2060-2090. <https://doi.org/10.1002/nme.1536>
20. Wang SY, Lim KM, Khoo BC, Wang MY. An extended level set method for shape and topology optimization. *J Comput Phys*. 2007;221:395-421.
21. Chen J, Shapiro V, Suresh K, Tsukanov I. Shape optimization with topological changes and parametric control. *Int J Numer Methods Eng*. 2007;71:313-346. <https://doi.org/10.1002/nme>
22. Qian X. Full analytical sensitivities in NURBS based isogeometric shape optimization. *Comput Methods Appl Mech Eng*. 2010;199:2059-2071. <https://doi.org/10.1016/j.cma.2010.03.005>
23. Yamada T, Izui K, Nishiwaki S, Takezawa A. A topology optimization method based on the level set method incorporating a fictitious interface energy. *Comput Methods Appl Mech Eng*. 2010;199(45-48):2876-2891.
24. Choi JS, Yamada T, Izui K, Nishiwaki S, Yoo J. Topology optimization using a reaction-diffusion equation. *Comput Methods Appl Mech Eng*. 2011;200(29-32):2407-2420.
25. Wang MY, Wang XM, Guo DM. A level set method for structural topology optimization. *Comput Methods Appl Mech Eng*. 2003;192:217-224.
26. Xia Q, Shi T, Liu S, Wang MY. A level set solution to the stress-based structural shape and topology optimization. *Comput Struct*. 2012;90-91:55-64.
27. Allaire G, Dapogny C, Frey P. Shape optimization with a level set based mesh evolution method. *Comput Methods Appl Mech Eng*. 2014;282:22-53. <https://doi.org/10.1016/j.cma.2014.08.028>
28. Jeong S, Lim S, Min S. Level-set-based topology optimization using remeshing techniques for magnetic actuator design. *IEEE Trans Magn*. 2016;52(3):3-6.
29. Duysinx P, Miegroet LV, Jacobs T, Fleury C. Generalized shape optimization using X-FEM and level set methods. *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials*. Dordrecht, The Netherlands: Springer; 2006. *Solid Mechanics and its Applications*.
30. Wei P, Wang MY. A structural optimization method with XFEM and level set model. In: Proceedings of the TMCE 2008; 2008; İzmir, Turkey.
31. Wang MY, Li L. Shape equilibrium constraint: a strategy for stress-constrained structural topology optimization. *Struct Multidiscip Optim*. 2013;47(3):335-352.
32. Villanueva CH, Maute K. Density and level set-XFEM schemes for topology optimization of 3-D structures. *Comput Mech*. 2014;54:133-150. <https://doi.org/10.1007/s00466-014-1027-z>
33. Jenkins N, Maute K. Level set topology optimization of stationary fluid-structure interaction problems. *Struct Multidiscip Optim*. 2015;52:179-195. <https://doi.org/10.1007/s00158-015-1229-9>
34. Coffin P, Maute K. Level set topology optimization of cooling and heating devices using a simplified convection model. *Struct Multidiscip Optim*. 2016;53:985-1003. <https://doi.org/10.1007/s00158-015-1343-8>
35. Picelli R, Dijk RV, Vicente WM, Pavanello R, Langelaar M, van Keulen F. Topology optimization for submerged buoyant structures. *Eng Optim*. 2017;49(1):1-21. <https://doi.org/10.1080/0305215X.2016.1164147>

36. Xavier M, Novotny AA. Topological derivative-based topology optimization of structures subject to design-dependent hydrostatic pressure loading. *Struct Multidisc Optim*. 2017;56:47-57. <https://doi.org/10.1007/s00158-016-1646-4>
37. Bertsekas DP. *Constrained Optimization and Lagrange Multiplier Methods*. Belmont, MA: Athena Scientific; 1996.
38. Birgin EG, Martínez JM. *Practical Augmented Lagrangian Methods for Constrained Optimization*. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2014.
39. Emmendoerfer H, Fancello EA. Topology optimization with local stress constraint based on level set evolution via reaction-diffusion. *Comput Methods Appl Mech Eng*. 2016;305:62-88. <https://doi.org/10.1016/j.cma.2016.02.024>
40. Gurtin ME. Generalized Ginzburg-Landau and Cahn-Hilliard equations based on a microforce balance. *Physica D*. 1996;92:178-192.
41. Gournay F. Velocity extension for the level-set method and multiple eigenvalues in shape optimization. *SIAM J Control Optim*. 2006;45(1):343-367.
42. Cook R, Malkus DS, Plesha ME, Witt RJ. *Concepts and Applications of Finite Element Analysis*. 4th ed. New York, NY: Wiley; 2001.

How to cite this article: Emmendoerfer Jr H, Fancello EA, Silva ECN. Level set topology optimization for design-dependent pressure load problems. *Int J Numer Methods Eng*. 2018;115:825-848. <https://doi.org/10.1002/nme.5827>

APPENDIX A

DERIVATIVE OF THE LINEAR OPERATOR

By taking the variation of

$$l_\phi(\lambda) = \int_D \boldsymbol{\tau}(\phi) \cdot \lambda \|\nabla \phi\| \delta(\phi) \, dD$$

in relation to ϕ in direction $\delta\phi$, we obtain[†]

$$\frac{\partial l_\phi(\lambda)}{\partial \phi} [\delta\phi] = \int_D \lambda \cdot \frac{\partial \boldsymbol{\tau}(\phi)}{\partial \phi} [\delta\phi] \|\nabla \phi\| \delta(\phi) \, dD + \frac{\partial \mathcal{A}}{\partial \phi} [\delta\phi], \quad (\text{A1})$$

where

$$\begin{aligned} \frac{\partial \mathcal{A}}{\partial \phi} [\delta\phi] &= \int_D \boldsymbol{\tau}(\phi) \cdot \lambda \frac{\partial(\|\nabla \phi\| \delta(\phi))}{\partial \phi} [\delta\phi] \, dD \\ &= \int_D \boldsymbol{\tau}(\phi) \cdot \lambda \left\{ \frac{\partial(\delta(\phi))}{\partial \phi} [\delta\phi] \|\nabla \phi\| + \delta(\phi) \frac{\partial(\|\nabla \phi\|)}{\partial \phi} [\delta\phi] \right\} \, dD \\ &= \int_D \boldsymbol{\tau}(\phi) \cdot \lambda \left\{ \delta'(\phi) \|\nabla \phi\| \delta\phi + \delta(\phi) \frac{\nabla \phi \cdot \nabla(\delta\phi)}{\|\nabla \phi\|} \right\} \, dD \\ &= \int_D \boldsymbol{\tau}(\phi) \cdot \lambda \left\{ \nabla(\delta(\phi)) \cdot \frac{\nabla \phi}{\|\nabla \phi\|} \delta\phi + \delta(\phi) \frac{\nabla \phi \cdot \nabla(\delta\phi)}{\|\nabla \phi\|} \right\} \, dD. \end{aligned} \quad (\text{A2})$$

Here, $\delta'(\phi)$ is the first derivative of the Dirac delta function and we have used relationship $\nabla \delta(\phi) = \delta'(\phi) \nabla \phi$. We can further simplify (A2) by using the product rule to make the following substitutions:

$$\nabla(\delta(\phi)) \cdot \frac{\nabla \phi}{\|\nabla \phi\|} = \nabla \cdot \left(\delta(\phi) \frac{\nabla \phi}{\|\nabla \phi\|} \right) - \delta(\phi) \nabla \cdot \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right) \quad (\text{A3})$$

and

$$\delta(\phi) \frac{\nabla \phi \cdot \nabla(\delta\phi)}{\|\nabla \phi\|} = \nabla \cdot \left(\delta\phi \delta(\phi) \frac{\nabla \phi}{\|\nabla \phi\|} \right) - \delta\phi \nabla \cdot \left(\delta(\phi) \frac{\nabla \phi}{\|\nabla \phi\|} \right). \quad (\text{A4})$$

[†]Verify the distinction in notation between variation $\delta\phi$ and Dirac function $\delta(\phi)$.

Substituting (A3) and (A4) in (A2), we have

$$\begin{aligned}
\frac{\partial \mathcal{A}}{\partial \phi} [\delta \phi] &= - \int_D \boldsymbol{\tau}(\phi) \cdot \boldsymbol{\lambda} \nabla \cdot \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right) \delta(\phi) \delta \phi dD \\
&\quad + \int_D \boldsymbol{\tau}(\phi) \cdot \boldsymbol{\lambda} \nabla \cdot \left(\delta \phi \delta(\phi) \frac{\nabla \phi}{\|\nabla \phi\|} \right) dD \\
&= - \int_D \boldsymbol{\tau}(\phi) \cdot \boldsymbol{\lambda} \nabla \cdot \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right) \delta(\phi) \delta \phi dD \\
&\quad + \int_D \left[\nabla \cdot \left(\boldsymbol{\tau}(\phi) \cdot \boldsymbol{\lambda} \delta \phi \delta(\phi) \frac{\nabla \phi}{\|\nabla \phi\|} \right) \right. \\
&\quad \left. - \nabla (\boldsymbol{\tau}(\phi) \cdot \boldsymbol{\lambda}) \cdot \left(\delta \phi \delta(\phi) \frac{\nabla \phi}{\|\nabla \phi\|} \right) \right] dD.
\end{aligned} \tag{A5}$$

The Divergence theorem is used to obtain

$$\begin{aligned}
\frac{\partial \mathcal{A}}{\partial \phi} [\delta \phi] &= - \int_D \left[\boldsymbol{\tau}(\phi) \cdot \boldsymbol{\lambda} \nabla \cdot \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right) + \nabla (\boldsymbol{\tau}(\phi) \cdot \boldsymbol{\lambda}) \cdot \frac{\nabla \phi}{\|\nabla \phi\|} \right] \delta(\phi) \delta \phi dD \\
&\quad + \int_{\partial D} \boldsymbol{\tau}(\phi) \cdot \boldsymbol{\lambda} \frac{\delta(\phi)}{\|\nabla \phi\|} \frac{\partial \phi}{\partial \mathbf{n}} \delta \phi d\partial D.
\end{aligned} \tag{A6}$$

Now, by assuming that ϕ does not vary in the normal direction \mathbf{n} on ∂D , that is,

$$\frac{\partial \phi}{\partial \mathbf{n}} \Big|_{\partial D} = 0,$$

and by using the derivation rule, we conclude that

$$\begin{aligned}
\frac{\partial \mathcal{A}}{\partial \phi} [\delta \phi] &= - \int_D \nabla \cdot \left(\boldsymbol{\tau}(\phi) \cdot \boldsymbol{\lambda} \frac{\nabla \phi}{\|\nabla \phi\|} \right) \delta(\phi) \delta \phi dD \\
&= \int_D \operatorname{div} ((\boldsymbol{\tau}(\phi) \cdot \boldsymbol{\lambda}) \mathbf{n}) \delta(\phi) \delta \phi dD,
\end{aligned} \tag{A7}$$

where

$$\mathbf{n} = - \frac{\nabla \phi}{\|\nabla \phi\|}.$$

Finally, by substituting (A7) in (A1) and assuming that $\frac{\partial \boldsymbol{\tau}(\phi)}{\partial \phi} [\delta \phi]$ is Fréchet derivative, we found

$$\frac{\partial l_\phi(\boldsymbol{\lambda})}{\partial \phi} [\delta \phi] = \int_D \left[\boldsymbol{\lambda} \cdot \frac{\partial \boldsymbol{\tau}(\phi)}{\partial \phi} \|\nabla \phi\| + \operatorname{div} ((\boldsymbol{\tau}(\phi) \cdot \boldsymbol{\lambda}) \mathbf{n}) \right] \delta(\phi) \delta \phi dD. \tag{A8}$$

APPENDIX B

DERIVATIVE OF CONSISTENT NODAL FORCES

This section presents the procedure for computing the sensitivity as the loaded zero level set curve goes over the fixed grid of the finite element model.

In this work, we discretize the level set function ϕ with bilinear shape functions and the same as that used for the finite element analysis. Figure B1 shows four configurations of the zero level set isocontour intersecting an element in 2D. Note that because of the bilinear shape functions used for the interpolation of ϕ , the double intersection shown in Figure B1(D) is naturally avoided: if all nodes of the element have the same sign, it is assumed the element is not cut by the boundary. The case as shown in Figure B1(C) is algorithmically avoided. If, during the level set evolution, an element achieves Case 3 being Γ_N one the boundaries, the evolution equation is continued until this configuration is eliminated. In this way, it is arbitrarily enforced that each element is cut by a single boundary Γ_N .

A set of cases, however, deserves attention. Consider an element is in Case 1, and a new boundary, coming from the upper edge, modifies the sign of one of the upper nodes. In this case, the configuration changes immediately (discontinuously) to Case 2. Similar behavior may happen in Case 4. In these situations, we have a jump on the configuration

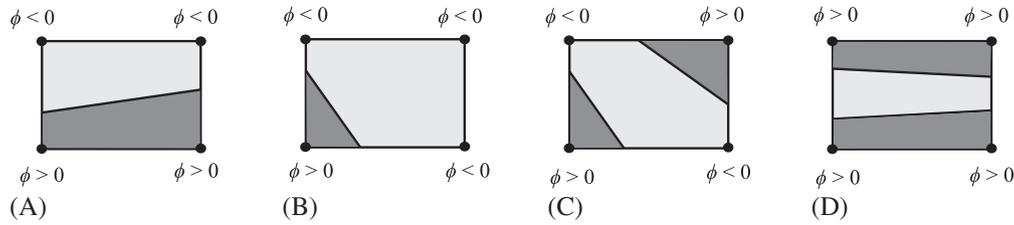


FIGURE B1 Intersection configurations in a four-node element cut by the zero level set function. A, Case 1: allowed; B, Case 2: allowed; C, Case 3: avoided; D, Case 4: unsupported

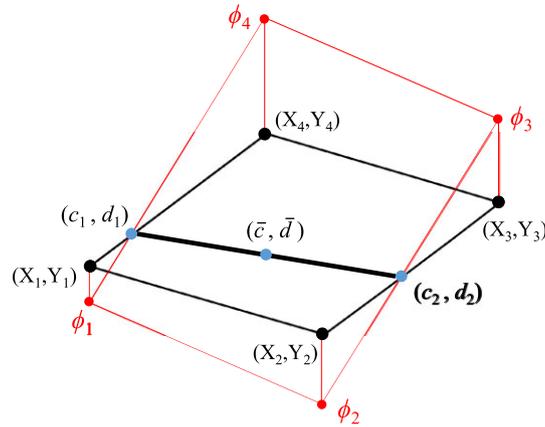


FIGURE B2 Figure shows a four-node element cut by the zero level set function [Colour figure can be viewed at wileyonlinelibrary.com]

and therefore on the operations: small discontinuity on the objective function. Despite, in numerical examples, not much sensitivity has been found because of these situations, these cases will be subject to analysis in future works.

Then, to accomplish the derivative of Equation (44), distances Δc and Δd (see Figure 5B) and shape function \mathbf{N} need to be written in terms of level set function ϕ . To this end, points (c_1, d_1) and (c_2, d_2) , which intercept the element, can be written as (see Figure B2)

$$c_w = \frac{X_i |\phi_{i+1}| + X_{i+1} |\phi_i|}{|\phi_i| + |\phi_{i+1}|}, \quad w = 1, 2, \quad (B1)$$

$$d_w = \frac{Y_i |\phi_{i+1}| + Y_{i+1} |\phi_i|}{|\phi_i| + |\phi_{i+1}|}$$

where w identifies the (two) intersecting points, i and $i + 1$ are the subsequent nodes of the element associated with each point w such that $\text{sign}(\phi_i) \neq \text{sign}(\phi_{i+1})$, and X and Y are global coordinates.

Thus, the middle point (\bar{c}, \bar{d}) is expressed by

$$\bar{c} = \bar{c}(\phi) = \frac{1}{2} \sum_{w=1}^2 c_w, \quad \bar{d} = \bar{d}(\phi) = \frac{1}{2} \sum_{w=1}^2 d_w, \quad (B2)$$

and distances Δc and Δd are defined as

$$\Delta c = \Delta c(\phi) = |c_1 - c_2| \quad \text{and} \quad \Delta d = \Delta d(\phi) = |d_1 - d_2|. \quad (B3)$$

By using the definitions above, the shape functions for a structured mesh of quadrilateral four-node element (see Figure B3) are

$$N_1 = \frac{(a - \bar{c} + \bar{X})(b - \bar{d} + \bar{Y})}{4ab}, \quad N_2 = \frac{(a + \bar{c} - \bar{X})(b - \bar{d} + \bar{Y})}{4ab} \quad (B4)$$

$$N_3 = \frac{(a + \bar{c} - \bar{X})(b + \bar{d} - \bar{Y})}{4ab}, \quad N_4 = \frac{(a - \bar{c} + \bar{X})(b + \bar{d} - \bar{Y})}{4ab}$$

where

$$\bar{X} = \frac{X_1 + X_2}{2}, \quad \text{and} \quad \bar{Y} = \frac{Y_1 + Y_2}{2},$$

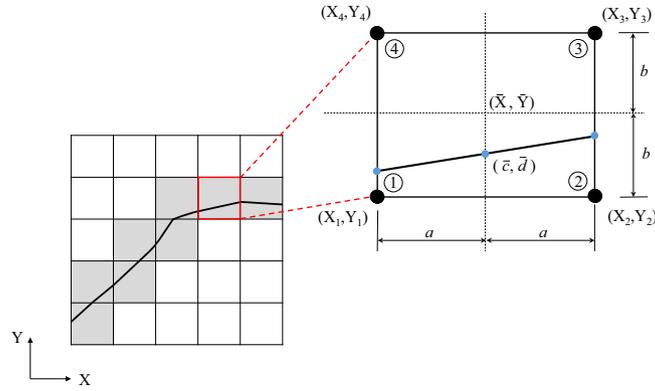


FIGURE B3 Bilinear quadrilateral element and its zero level set boundary [Colour figure can be viewed at wileyonlinelibrary.com]

and a and b represent the physical dimension of the element.

Now, we can compute the derivative of the loading given in (44), which consists in calculating derivatives $\partial \mathbf{N}^T(\bar{c}, \bar{d})/\partial \phi$, $\Delta c/\partial \phi$, and $\Delta d/\partial \phi$. To obtain the partial derivative of the shape function, we can use the chain rule as

$$\frac{\partial \mathbf{N}^T(\bar{c}, \bar{d})}{\partial \phi} = \frac{\partial \mathbf{N}^T(\bar{c}, \bar{d})}{\partial \bar{c}} \frac{\partial \bar{c}}{\partial \phi} + \frac{\partial \mathbf{N}^T(\bar{c}, \bar{d})}{\partial \bar{d}} \frac{\partial \bar{d}}{\partial \phi}. \quad (\text{B5})$$

By calculating each term of the equation above, we have the derivative of the interpolants (B4) in relation to \bar{c} given by

$$\frac{\partial \mathbf{N}^T(\bar{c}, \bar{d})}{\partial \bar{c}} = \begin{bmatrix} \frac{\partial N_1}{\partial \bar{c}} & 0 & \dots & \frac{\partial N_4}{\partial \bar{c}} & 0 \\ 0 & \frac{\partial N_1}{\partial \bar{c}} & \dots & 0 & \frac{\partial N_4}{\partial \bar{c}} \end{bmatrix}^T,$$

where

$$\begin{aligned} \frac{\partial N_1}{\partial \bar{c}} &= -\frac{1}{4ab}(b - \bar{d} + \bar{Y}), & \frac{\partial N_2}{\partial \bar{c}} &= \frac{1}{4ab}(b - \bar{d} + \bar{Y}), \\ \frac{\partial N_3}{\partial \bar{c}} &= \frac{1}{4ab}(b + \bar{d} - \bar{Y}), & \frac{\partial N_4}{\partial \bar{c}} &= -\frac{1}{4ab}(b + \bar{d} - \bar{Y}). \end{aligned}$$

The derivatives in relation to \bar{d} are easily obtained in a similar way. Derivatives $\partial \bar{c}/\partial \phi$ and $\partial \bar{d}/\partial \phi$ still have to be determined in (B5). From definitions in (B2), we have that

$$\frac{\partial \bar{c}}{\partial \phi} = \frac{1}{2} \sum_{w=1}^2 \frac{\partial c_w}{\partial \phi} \quad \text{and} \quad \frac{\partial \bar{d}}{\partial \phi} = \frac{1}{2} \sum_{w=1}^2 \frac{\partial d_w}{\partial \phi}. \quad (\text{B6})$$

Again, we use the chain rule and can write

$$\begin{aligned} \frac{\partial c_w}{\partial \phi} &= \frac{\partial c_w}{\partial \phi_i} + \frac{\partial c_w}{\partial \phi_{i+1}} = \frac{\partial c_w}{\partial |\phi_i|} \frac{\partial |\phi_i|}{\partial \phi} + \frac{\partial c_w}{\partial |\phi_{i+1}|} \frac{\partial |\phi_{i+1}|}{\partial \phi} \\ &= \frac{\partial c_w}{\partial |\phi_i|} \text{sign}(\phi_i) + \frac{\partial c_w}{\partial |\phi_{i+1}|} \text{sign}(\phi_{i+1}). \end{aligned} \quad (\text{B7})$$

By using expressions in (B1), we obtain the following derivatives:

$$\frac{\partial c_w}{\partial |\phi_i|} = \frac{|\phi_{i+1}|}{(|\phi_i| + |\phi_{i+1}|)^2} (X_{i+1} - X_i), \quad (\text{B8})$$

$$\frac{\partial c_w}{\partial |\phi_{i+1}|} = \frac{|\phi_i|}{(|\phi_i| + |\phi_{i+1}|)^2} (X_i - X_{i+1}). \quad (\text{B9})$$

Substituting (B8) and (B9) in (B7) and rearranging terms, we have

$$\frac{\partial c_w}{\partial \phi} = Z_w (X_{i+1} - X_i), \quad (\text{B10})$$

where

$$Z_w = \frac{|\phi_{i+1}| \text{sign}(\phi_i) - |\phi_i| \text{sign}(\phi_{i+1})}{(|\phi_i| + |\phi_{i+1}|)^2}. \quad (\text{B11})$$

Thus, substituting (B10) in (B6) and proceeding analogously for \bar{d} , the lacking derivatives are expressed as

$$\frac{\partial \bar{c}}{\partial \phi} = \frac{1}{2} \sum_{w=1}^2 Z_w (X_{i+1} - X_i), \quad (\text{B12})$$

$$\frac{\partial \bar{d}}{\partial \phi} = \frac{1}{2} \sum_{w=1}^2 Z_w (Y_{i+1} - Y_i). \quad (\text{B13})$$

Finally, the derivatives of Δc and Δd in relation to level set function ϕ are calculated as[‡]

$$\begin{aligned} \frac{\partial \Delta c}{\partial \phi} &= \frac{\partial(|c_1 - c_2|)}{\partial \phi} = \frac{(c_1 - c_2)}{|c_1 - c_2|} \left(\frac{\partial c_1}{\partial \phi} - \frac{\partial c_2}{\partial \phi} \right) \\ &= \text{sign}(c_1 - c_2) \left(\frac{\partial c_1}{\partial \phi} - \frac{\partial c_2}{\partial \phi} \right), \end{aligned} \quad (\text{B14})$$

and, analogously,

$$\frac{\partial \Delta d}{\partial \phi} = \text{sign}(d_1 - d_2) \left(\frac{\partial d_1}{\partial \phi} - \frac{\partial d_2}{\partial \phi} \right). \quad (\text{B15})$$

Therefore, all the necessary derivatives are determined, and we can now obtain the sensitivity of the loaded moving boundaries in a semianalytical manner.

[‡]Note that: $|c_1 - c_2| = \sqrt{(c_1 - c_2)(c_1 - c_2)}$.