A discrete adjoint based level set topology optimization method for stress constraints
Sandilya Kambampati\textsuperscript{a,1,}\textsuperscript{*}, Hayoung Chung\textsuperscript{b,2}, H. Alicia Kim\textsuperscript{a,3,}\textsuperscript{3}

\textsuperscript{a} University of California San Diego, San Diego, CA 92093, USA
\textsuperscript{b} Ulsan National Institute of Science and Technology, South Korea
\textsuperscript{c} Cardiff University, Cardiff, UK

Received 16 May 2020; received in revised form 18 September 2020; accepted 8 November 2020
Available online xxxx

Abstract
This paper proposes a new methodology for computing boundary sensitivities in level set topology optimization using the discrete adjoint method. The adjoint equations are constructed using the discretized governing field equations. The objective function is differentiated with respect to the boundary point movement for computing boundary sensitivities using the discrete adjoint equations. For this purpose, we present a novel approach where we perturb the boundary implicitly by locally modifying the level set function around a given boundary point. These local perturbations are combined with the derivatives of the objective function with respect to the volume fractions of individual elements to compute boundary sensitivities. This enables the circumvention of smoothing or interpolation methods typically used in level set topology optimization to compute sensitivities; and improves the accuracy of the sensitivities and the convergence characteristics. We demonstrate the effectiveness of our method in the context of stress minimization and stress constrained topology optimization problems for orthogonal bracket design under multiple load cases.

\textcopyright 2020 Elsevier B.V. All rights reserved.

Keywords: Level set method; Topology optimization; Discrete adjoint sensitivities; Stress constraints

1. Introduction
The level set method (LSM) is a popular method used for topology optimization, which incorporates an implicit representation of the boundary. Specifically, the structure is described implicitly using the level set function $\phi$, as $\{ x \mid \phi(x) \geq 0 \}$, where $x$ is any point that belongs to structure, and the boundary is described as $\{ x \mid \phi(x) = 0 \}$. In classical level set topology optimization methods \cite{1,2}, the boundary is updated via the level set equation (LSE \cite{3}), given by $d\phi / dt + V_n |\nabla \phi| = 0$, where $V_n$ is the normal velocity field, and $t$ is the pseudo time. The LSE is a type of a Hamilton–Jacobi equation, and is numerically solved using an upwinding scheme \cite{3}. Other level set methods which update the level set function using strategies other than the LSE can be found in topology optimization literature.

\textsuperscript{*} Corresponding author.
E-mail addresses: sakambampati@ucsd.edu (S. Kambampati), hychung@unist.ac.kr (H. Chung), alicia@ucsd.edu (H.A. Kim).

1 Postdoctoral Researcher, Structural Engineering.
2 Assistant Professor, Mechanical Engineering.
3 Jacobs Scholars Chair Professor, Structural Engineering.

https://doi.org/10.1016/j.cma.2020.113563
0045-7825/© 2020 Elsevier B.V. All rights reserved.
Fig. 1. A schematic of the continuous and discrete adjoint methods. In the discrete adjoint method, the governing field equations are first discretized, and then differentiated for the derivation of the adjoint equations. Discrete adjoint method is used in parametric level set topology optimization. In the continuous adjoint method, the governing field equations are differentiated and for the derivation of the adjoint equations in the continuous space, and then discretized. The continuous adjoint method is popularly used in the classical level set topology optimization.

such as the topological derivative method [4], the parametric level set method [5], the level set evolution based on a generalization of topological gradient [6], or the reaction–diffusion equation method [7]. A comprehensive literature review on the different level set methods used in topology optimization can be found in [8].

In the classical level set topology optimization methods, the continuous adjoint method is popularly used to compute sensitivities [1,2,9,10]. Fig. 1 is a schematic that helps in describing the continuous adjoint method. The governing field equations are first differentiated to derive the adjoint variable in the continuous space, using an augmented functional constructed in the continuous space. This is followed by the discretization of the field and adjoint equations using the finite element analysis (FEA) for sensitivity computation. Thus, the continuous adjoint method follows the differentiate-then-discretize philosophy [11]. The continuous adjoint method, however, has a major obstacle during the computational implementation. When shape functions with $C^0$ continuity are used to approximate the displacement field, the sensitivity expressions are discontinuous at the finite element nodes and edges. Typically, interpolation and smoothing methods are used to get around this discontinuity in the sensitivity field. This can lead to a reduction in the accuracy of the sensitivities, forcing the optimization process to diverge for higher move limits, as we demonstrate in this paper.

The discrete adjoint method (schematic shown in Fig. 1) follows a different philosophy. The field equations are first discretized using the finite element method. The discretized field equations are then differentiated to derive the adjoint equations using an augmented functional constructed in the discrete space. Thus, the discrete adjoint method follows the discretize-then-differentiate philosophy [11]. The parametric level set methods (PLSM) [5,12–15] use the discrete adjoint method for sensitivity computation, and they do not require any interpolation or smoothing of the sensitivity fields. Nonetheless, PLSM can suffer from oscillations in the boundary and require regularization for the stabilization of the level set function evolution [14]. On the other hand, the classical level set method does not need to be regularized if the sensitivities and velocities are computed at the boundaries and are extended into the nodes using fast extension method [16]. To the best of our knowledge, the discrete adjoint method has not been used for sensitivity computation in classical level set topology optimization. Such a methodology can improve the accuracy of the sensitivities significantly as it helps in the circumvention of smoothing or interpolation of the sensitivity field. In addition, it can facilitate the use of algorithmic differentiation [17] in classical level set topology optimization.
In this paper, we present a discrete adjoint method for computing boundary sensitivities in the classical level set topology optimization method. The augmented functional and the adjoint equations are constructed using the discretized governing field equations. The design variables for the optimization are the discretized boundary point movements. We present a semi-analytical formulation where the Lagrangian function is differentiated with respect to the movement of the boundary points to compute the sensitivities using discrete adjoint equations. The novelty of our approach is that we perturb the boundary implicitly via appropriately modifying the level set function in a narrow band around a given boundary point. This allows for the circumvention of interpolation or smoothing of the sensitivity field, resulting in more accurate sensitivities. We demonstrate the developed method using stress minimization and stress constrained design examples, since stress based optimization is considered to be challenging in level set topology optimization. Using the finite difference method, we compare the accuracy of the proposed discrete adjoint boundary sensitivity against a continuous adjoint implementation presented in [10]. Finally, we demonstrate our method in the design of minimum-mass topologies of a 3D orthogonal bracket under stress constraints for multiple load cases.

2. Optimization method

In this section, we outline our level set topology optimization method. The boundary of the structure is implicitly described as

\[
\phi(x) \geq 0, \quad x \in \Omega
\]

\[
\phi(x) = 0, \quad x \in \Gamma
\]

\[
\phi(x) < 0, \quad x \notin \Omega
\]

where \(\phi(x)\) is the level set function, \(\Omega\) is the domain, and \(\Gamma\) is the domain boundary [18]. The goal of the optimization is to solve the following problem

\[
\min_{\Omega} F = \int_{\Omega} f(\Omega) d\Omega
\]

subject to \(G_j = \int_{\Omega} g_j(\Omega) d\Omega \leq g^0_j, \quad j = 1, 2, \ldots, N_g,\)

where \(F\) is the objective function, \(G_j\) is the \(j\)th constraint function, \(g^0_j\) is the \(j\)th constraint value, and \(N_g\) is the number of constraints. For a given level set function topology, the objective and constraint functions are linearized using the boundary sensitivities, and the following sub-optimization problem is constructed

\[
\min_{\Delta t} \Delta F = \sum_i (s^b_i A_i z_i)
\]

subject to \(G_j = \sum_i (s^b_{j,i} A_i z_i) \leq g^j_j, \quad j = 1, 2, \ldots, N_g,\)

\[-m_i \leq z_i \leq m_i\]

where \(z_i = V_n, \Delta t\) is the movement of the \(i\)th boundary point, \(\Delta t = 1\) is the pseudo time step, \(g^j_j\) is the target constraint value for the \(j\)th constraint, \(A_i\) is the area of the discretized boundary that belongs to the \(i\)th boundary point, and \(m_i\) is the move limits on the boundary. Further details on the linearization of Eqs. (2) to (3) can be found in [19].

The boundary point sensitivities of the objective function and the \(j\)th constraint function are \(s^b_i\) and \(s^b_{j,i}\), respectively. In this paper, we present a discrete adjoint method for computing the boundary point sensitivities, and compare them against a continuous adjoint method presented in [10,19]. The sub-optimization problem in Eq. (3) is solved using the Simplex method [20], resulting in the optimum boundary point velocities. The Simplex method is implemented using the GLPK library (gnu.org/software/glpk/).

The obtained boundary point velocities are extended to all the grid points in a narrow band around the boundary, by solving the following equation

\[
\nabla \phi \cdot \nabla V_n = 0
\]
using the fast marching method [16]. The reason for using Eq. (4) for the velocity extension via fast marching method is that the signed-distance property ($|\nabla \phi| = 1$) is always preserved. During optimization, the level set function can become too steep or too flat at times, and it needs to be reinitialized to a signed distance function frequently for avoiding numerical instabilities [21]. Therefore, one can choose not to reinitialize the level set function to a signed distance function if it is updated using Eq. (4).

2.1. Fast marching method

In this section, we briefly present the details of the fast marching method. The complete details of the theory and implementation of the fast marching method can be found in [16].

The optimum boundary velocities which are the solution of the optimization problem Eq. (3), describe the velocity of the boundary. These boundary velocities are extended into the grid points by solving Eq. (4). The key steps of the fast marching algorithms for velocity extension can be briefly summarized as follows

1. Begin the algorithm by marking all the grid points that have a neighbor on the opposite side of the boundary as close. Mark all the other grid points in the narrow band as far. Fig. 2a illustrates the marking of grid points as close and far.
2. Find the grid point in close that has the least value of $\phi$, and mark it as trial.
3. Mark the trail grid point as accepted. Compute the extended velocity of this point using the following equation for 2D grids

\[
(\phi_{r,s} - \phi_{a,s}) \cdot (V_{r,s} - V_{a,s}) + (\phi_{r,s} - \phi_{r,\beta}) \cdot (V_{r,s} - V_{r,\beta}) = 0
\]  

where

\[
\alpha = r - 1 \text{ if } \phi_{r-1,s} < \phi_{r+1,s} ; \text{ else } \alpha = r + 1
\]

\[
\beta = s - 1 \text{ if } \phi_{r,s-1} < \phi_{r,s+1} ; \text{ else } \beta = s + 1
\]

where $\phi_{r,s}$ and $V_{r,s}$ represent the level set function value and velocity at the grid point $(r, s)$. Eqs. (5) and (6) can be similarly constructed for 3D grids.
4. Mark all the neighboring narrow band grid points of the recently accepted point as close. Compute the extended velocities of these close points using Eq. (5). Fig. 2b illustrates the marking of the trial grid point as accepted and its neighboring grid points as close.
5. Repeat Steps 2 to 4 until all the points in the narrow band are tagged accepted and their extension velocities are computed.
The central idea of the fast marching method is that the velocity needs to be extended by marching from low values of $\phi$ to higher values. We note the velocity is extended separately for grid points outside the front ($\phi < 0$) and inside the front ($\phi > 0$).

### 2.2. Level set update

The level set function is then numerically updated using the following discretization scheme

$$\phi_i^{k+1} = \phi_i^k - \Delta t |\nabla \phi_i^k| V_{n,i}$$

where $i$ is a nodal point in the domain, $k$ is the iteration number, and $\nabla \phi_i$ is the gradient of the level set function.

The gradient $\nabla \phi_i$ is computed using the Hamilton–Jacobi weighted essentially non-oscillatory scheme (HJ-WENO). The HJ-WENO is a gradient stencil designed specifically for the Hamilton–Jacobi type equations such as the level set advection equation. The magnitude of the gradient $|\nabla \phi|$ at a grid point $(r, s)$ is computed for 2D grids as

$$|\nabla \phi_{r,s}| = \sqrt{\phi^2_x + \phi^2_y}$$

where

$$\phi_x = \phi^+_x \text{ if } V_{r,s} \geq 0; \text{ else } \phi_x = \phi^-_x$$

$$\phi_y = \phi^+_y \text{ if } V_{r,s} \geq 0; \text{ else } \phi_y = \phi^-_y$$

$\phi^+_x, \phi^-_x, \phi^+_y, \phi^-_y$ are the upwind derivatives. The upwind derivatives in the $x$ directions ($\phi^+_x$ and $\phi^-_x$) are given by closed form expressions of $\phi$ at the grid point $(r, s)$ and the neighboring grid points $\{\phi(r-3, s), \phi(r-2, s), \ldots, \phi(r+3, s)\}$. Similarly, the upwind derivatives in the $y$ directions ($\phi^+_y$ and $\phi^-_y$) are given by closed form expressions of $\phi$ at the grid point $(r, s)$ are functions of $\{\phi(r, s-3), \phi(r, s-2), \ldots, \phi(r, s+3)\}$. The details of HJ-WENO and the complete expressions for the upwind derivatives can be found in [3].

This design cycle consisting of boundary sensitivity computation – optimization – level set update is iteratively performed until convergence in the objective function obtained.

### 3. Sensitivity computation

In this section, we present a discrete adjoint method for computing the boundary point sensitivities, which are used to solve the sub-optimization problem in Eq. (3) in the classical level set topology optimization method. We present our analysis in the context of stress based level set topology optimization. For the sake of comparison and perspective, we also briefly discuss the continuous adjoint formulations used in the classical level set methods, and also the discrete adjoint formulations used in the parametric level set methods.

#### 3.1. Continuous adjoint method

In this section, we briefly present the continuous adjoint for stress constraints in level set topology optimization. The complete derivation can be found in [10]. In stress based optimization, the $p$-norm of the von mises stress is used to approximate the maximum stress. The $p$-norm of the stress field $\sigma_{pn}$ is given by

$$\sigma_{pn} = \left( \int_{\Omega} \sigma_{vm}^p d\Omega \right)^{1/p}$$

where $\sigma_{vm}$ is the von Mises stress. For convenience, we introduce the auxiliary function

$$J = \int_{\Omega} \sigma_{vm}^p d\Omega$$

so that the shape derivative of the $p$-norm of the stress field $\sigma'_{pn}$ can be expressed as

$$\sigma'_{pn} = \frac{\sigma_{pn}^{p-1} \sigma'}{p} J'$$
where $J'$ is the shape derivative of the auxiliary function $J$. The linear elasticity field equation can be written as

$$a(u, v) = l(v)$$  \hspace{1cm} (14)

where

$$a(u, v) = \int_{\Omega} C \varepsilon(u) \cdot \varepsilon(v) d\Omega$$  \hspace{1cm} (15)

and

$$l(v) = \int_{\Omega} bv d\Omega + \int_{\Gamma_N}fv d\Gamma$$  \hspace{1cm} (16)

$b$ is the body force, $u$ is the displacement, $v$ is a test function, $f$ is the traction force applied on $\Gamma_N$, $C$ is the elasticity tensor, and $\varepsilon(u)$ is the strain tensor under the displacement field $u$. The augmented functional $L$ for the $J$ considering the linear elasticity equation is given by

$$L = J + a(u, \lambda) - l(\lambda)$$  \hspace{1cm} (17)

where $\lambda$ is the adjoint variable. The adjoint variable is computed such that the variation of the augmented functional is stationary with respect to the variation in the field variable $u$, which yields

$$\int_{\Omega} C \varepsilon(w) \cdot \varepsilon(\lambda) d\Omega = \int_{\Omega} f_{\lambda} w d\Omega$$  \hspace{1cm} (18)

where $f_{\lambda} = d\sigma_p / du$ indicates the adjoint pseudo-load, and $w$ is a test function. The shape derivative $L'$ for a given boundary perturbation $\theta$ is given by

$$L' = \int_{\Gamma} j' \cdot n d\Gamma + \int_{\Gamma} C \varepsilon(u) \cdot \varepsilon(\lambda) \cdot n d\Gamma$$  \hspace{1cm} (19)

Finally, the derivative of the $p$-norm $\sigma_{pn}$ for a given boundary perturbation $\theta$ is given by

$$\sigma_{pn}' = 1 - p \frac{\sigma_{pn}^{1-p}}{p} \int_{\Gamma} (\sigma_{cm}^p + C \varepsilon(u) \cdot \varepsilon(\lambda)) \cdot n d\Gamma$$  \hspace{1cm} (20)

In summary, for the continuous adjoint method, the governing field equations (Eq. (14)) and the augmented functional (Eq. (17)) are differentiated with respect to a continuous boundary perturbation $\theta$, resulting in a continuous adjoint equation (Eq. (18)). Both, the field equations and the adjoint equations are discretized and solved using FEA. This is followed by the boundary sensitivity computation using Eq. (20). However, the sensitivity expression in Eq. (20) involves computing the strain $\varepsilon(u)$ and the adjoint pseudo-strain $\varepsilon(\lambda)$ fields. When using bi-linear shape functions in 2D and tri-linear in 3D for the field variable, the strain and the adjoint strain field is discontinuous and incompatible at the boundary between two elements. Therefore, evaluating the sensitivity along the element edges and vertices poses a tough challenge when using the continuous adjoint method for level set topology optimization. To overcome this challenge, smooth stress recovery methods are used: the sensitivity expressions are evaluated at the element centroids and are extrapolated to the nodes in [1], Gaussian smoothing functions are used in [2], a combination of interpolation and extrapolation schemes is used in [9], and a least squares interpolation method is used in [10]. This interpolation and extrapolation based stress recovery used in the continuous adjoint methods is not consistent with the boundary movement, and can lead to reduced accuracy of the sensitivities, ultimately resulting in slow convergence of the level set method.

### 3.2. Discrete adjoint method

In the discrete adjoint sensitivity method, the linear elasticity field equation (Eq. (14)) is first discretized as

$$K u = f$$  \hspace{1cm} (21)

where $K$ is the stiffness matrix, $u$ is the discretized displacement, and $f$ is the discretized force. The stiffness matrix is assembled using elemental stiffness matrices as follows

$$K = \sum_{i}^{N_e} K_{e,i}$$  \hspace{1cm} (22)
where $K_{e,i}$ is the elemental stiffness matrix corresponding to an element $i$, constructed using Ersatz material as

$$K_{e,i} = (E_{min} + x_i(E - E_{min}))K_{e,0}$$  \hspace{1cm} (23)

where $x_i$ is the volume fraction of the element $i$ cut by the level set, $E_{min}$ is the elasticity modulus of a void element (where $x_i = 0$), $E$ is the elasticity modulus of the material, and $E_{min} + x_i(E - E_{min})$ is the elasticity modulus of a partially cut element, and $K_{e,0}$ is the stiffness matrix computed assuming a unit value for the elasticity modulus.

The adjoint variable is computed by solving

$$\frac{\partial}{\partial L}$$  \hspace{1cm} (24)

The augmented functional $L$ is given by

$$L = \sigma_{pn} + \lambda^T(Ku - f)$$  \hspace{1cm} (25)

The adjoint variable is computed by solving $\frac{\partial L}{\partial u} = 0$, which yields

$$K^T\lambda = f_\lambda$$  \hspace{1cm} (26)

where $f_\lambda = \sum_{i}^{N_e} f_i^\lambda$ is the adjoint pseudo-force computed by assembling the elemental adjoint forces $f_i^\lambda$, given by

$$f_i^\lambda = \frac{\sigma_{pm}}{p} \sum_{j=1}^{Ng} p\sigma_{vm,ij}^{-2}CB_{ij}^TVCB_{ij}$$  \hspace{1cm} (27)

The augmented functional is differentiated with respect to the volume fractions of each element, to compute the elemental sensitivities $s_i$, given by

$$s_i = \frac{\partial L}{\partial x_i} = \frac{\partial \sigma_{pn}}{\partial x_i} + \lambda^T\frac{\partial K}{\partial x_i}u$$  \hspace{1cm} (28)

In parametric level set methods, the elemental sensitivities $s_i$ are coupled to level set function sensitivities through the chain rule. For instance, the sensitivity of the augmented functional $L$ with respect to the nodal level set function value $\phi_j$ at the node $j$, is given by

$$\frac{\partial L}{\partial \phi_j} = \sum_i^{N_e} \frac{\partial L}{\partial x_i} \cdot \frac{\partial x_i}{\partial \phi_j} = \sum_i^{N_e} s_i \cdot \frac{\partial x_i}{\partial \phi_j}$$  \hspace{1cm} (29)

where the term $\frac{\partial x_i}{\partial \phi_j}$ corresponds to the change in the volume fraction of the $i$th element with respect to the change in the nodal level set function value $\phi_j$ of the $j$th node. In parametric level set methods, the level set functions are generally described using radial basis functions [5,12–14], and Eq. (31) is used to compute the sensitivities of the augmented functional with respect to the level set function. More importantly, unlike the continuous adjoint method, there are no needs for interpolation or extrapolation of the sensitivity field in the discrete adjoint method, which leads to more accurate sensitivities. However, parametric level set methods can suffer from fluctuations of the boundary in the design process, and often they need to be regularized to a signed-distance function [14].
The classical level set method on the other hand does not need to be explicitly regularized to a signed-distance function. Specifically, if

(a) the sensitivities and velocities are computed at the boundary and are extended into the level set function grid points using the extension scheme proposed in [23], and
(b) the level set function is updated using the Hamilton–Jacobi level set equation (Eq. (7))

then the signed distance property of the level set function will always be maintained.

This approach of computing the sensitivities via boundary perturbation is similar to some of the shape optimization methods, such as Kim and Chang [24], which uses an explicit representation of the boundary. However, explicit representations cannot easily incorporate topological changes in the design process, while the level set method can. Additionally, significantly fewer design variables are generally used to describe the geometry in explicit representations. For instance, Kim and Chang used a maximum of twelve design variables in their study, (compared to $\sim 10^3 - 10^5$ design variables used in this study). To the best of our knowledge, an accurate, consistent, and computationally efficient methodology for computing the sensitivity of the discretized Lagrangian function with respect to the change in the level set function field (an expression similar to that of Eq. (31) for the classical level set method) does not exist in literature. Such a methodology can circumvent the need for the sensitivity interpolation and filtering which are typically needed in the classical level set topology optimization methods. This leads to stable and robust convergence.

### 3.2.1. Methodology for using discrete adjoint method in LSM

In our level set topology optimization method, the design variables are the boundary point movements $z_j$ in Eq. (3). We use the following expression to compute the sensitivity of the augmented functional $\mathcal{L}$ with respect to the boundary point movements $z_j$

$$\frac{\partial \mathcal{L}}{\partial z_j} = \sum_i^{N_e} \frac{\partial \mathcal{L}}{\partial x_i} \cdot \frac{\partial x_i}{\partial z_j} = \sum_i^{N_e} s_i \cdot \frac{\partial x_i}{\partial z_j}$$

(32)

where $s_i = \frac{\partial \mathcal{L}}{\partial x_i}$ is the derivative of the Lagrangian with respect to the volume fraction, and is given in Eq. (30). The term $\frac{\partial x_i}{\partial z_j}$ corresponds to the change in the volume fraction of the $i$th element due to the $j$th boundary point movement. The expression $\frac{\partial x_i}{\partial z_j}$ is computed by perturbing the boundary implicitly via appropriately modifying the level set function in a narrow band around the boundary point. In this study a narrow band width of four grid points is used on either side of the boundary.

Fig. 3 shows the schematic that we use to illustrate the computation of $\frac{\partial x_i}{\partial z_j}$. For a given boundary point of interest, we construct a mini level set grid around the boundary point by copying the level set function values around the boundary point on to the mini grid, as shown in Fig. 3b. Next, we assign a small perturbation boundary velocity $\delta$ to the boundary point of interest, in the normal direction to the boundary at the point. The magnitude of the perturbation velocity is of the order of $\delta \sim 10^{-8} \Delta x$, where $\Delta x$ is the width of the level set grid element. The perturbation normal velocity is extended into the mini grid points using Eq. (4). In Fig. 3c, we show the extended normal velocity field. The extended velocities are non-zero at the nodes shown in yellow. This implies that the level set function values at the nodes in yellow are affected due to the perturbation. The level set function is advected under the extended perturbation velocity field by solving Eq. (7) on the mini level set grid. This results in a new perturbed level set function and a perturbed boundary on the mini grid as shown in Fig. 3d. Finally, the term $\frac{\partial x_i}{\partial z_j}$ can be approximated as follows

$$\frac{\partial x_i}{\partial z_j} \approx \frac{x_i - x_i^b}{\delta}$$

(33)

where $x_i^b$ is the fraction of the $i$th element volume cut by the perturbed level set on the mini grid. In other words, the term $\frac{\partial x_i}{\partial z_j}$ accounts for the change in the volume fraction of the $i$th element that is affected by the implicit perturbation of the $j$th boundary point. In Fig. 3e, we show in gray the elements whose volume fractions are affected by the perturbation of the $j$th boundary point. The computational cost associated with the evaluation of the term $\frac{\partial x_i}{\partial z_j}$ is small, as the velocity extension and advection are performed on a mini-grid around the boundary point.
In summary, this section presents the computationally efficient and easy-to-implement formulation of the boundary point sensitivity (Eq. (32)) using the discrete adjoint method by combining the volume fraction sensitivities computed using Eq. (30), and the boundary point implicit perturbation via modifying the level set function on the mini grid (Eq. (33)).

4. Numerical examples

In this section, we present the numerical examples in 2D and 3D that demonstrate the discrete adjoint method in level set topology optimization for stress constraints. The 2D examples are run on a workstation with 16 GB RAM and 8 processors. A direct solver [25] is used for the FE analysis for the 2D examples. The 3D examples are run on a workstation with 384 GB RAM and 64 processors. The FE analysis for the 3D examples is done using in-house C++ codes based on a parallel implementation presented in [26].

4.1. 2D L-bracket

The schematic of the L-bracket commonly used to benchmark stress based topology optimization is shown in Fig. 4a. The FE mesh used to model the L-bracket comprises of $100 \times 100$ quad elements with a $60 \times 60$ elements sized square section removed on the top-right side. The L-bracket is clamped on the top and a vertical force of 1.2 units is applied on the right hand side. The structure is assumed to be having a unit elastic modulus and Poisson’s ratio of 0.3. Fig. 4b shows the stress distribution inside the structure, illustrating the stress concentration at the re-entrant corner of the L-bracket.

4.1.1. Finite difference validation

In this section, we validate the boundary point sensitivity computation for the $p$-norm of the maximum stress $\sigma_{pn}$ against the finite difference sensitivities. A perturbation velocity $V_{\delta} = 1.0 \times 10^{-4}$ for a given boundary point of interest $i$ on a topology $\Omega$. This perturbation velocity is extended into the grid points by solving Eq. (4), and the
Fig. 4. (a) A schematic of an L-bracket that is clamped on the top and a vertical force applied on the right side. (b) The stress distribution (capped at 0.65) of the L-bracket showing a stress concentration at the re-entrant corner.

Fig. 5. Initial topology used for the finite difference analysis of the boundary point sensitivities.

level set function is updated to compute the new topology $\Omega_{b,i}$. The finite difference sensitivity $s_{f,i}^b$ of a boundary point $i$ is given by

$$s_{f,i}^b = \frac{\sigma_{pn}(\Omega_{b,i}) - \sigma_{pn}(\Omega)}{V_b} \tag{34}$$

For this validation study, $p = 12$ is used. The sensitivity verification is performed on a topology with five holes, as shown in Fig. 5. The relative error $\epsilon_i^r$ of a boundary point $i$, defined as

$$\epsilon_i^r = \frac{|s_i^b - s_{f,i}^b|}{|s_{f,i}^b|} \tag{35}$$

where $s_i^b$ is the sensitivity of the $i$th boundary point computed using analytical adjoint methods. Fig. 6 shows the relative error $\epsilon_i^r$ computed using the discrete adjoint method. For the sake of comparison, we also show the relative errors computed using a continuous adjoint method developed by [10]. We can see from Fig. 6 that the errors are significantly high for the continuous adjoint method, with many points having errors over 10%. On the other hand, the discrete adjoint method has lower errors, with almost all points having an error less than 1%. The errors are over 1% for the boundary points close to the bottom-right corner when using the discrete adjoint method. This is because the relative error formula has the finite difference sensitivity in the denominator, and the finite difference sensitivity is almost zero for the points in this region — leading to errors over 1%.

Fig. 7 shows the absolute errors of the boundary points for the discrete and continuous adjoint methods. The absolute error $\epsilon_i^a$ of a boundary point $i$, defined as

$$\epsilon_i^a = \frac{|s_i^b - s_{f,i}^b|}{\max_i |s_{f,i}^b|} \tag{36}$$
Fig. 6. The relative errors of the boundary point sensitivities in percentages for the continuous and the discrete adjoint methods. The discrete adjoint method based sensitivities have significantly low relative errors compared to the continuous adjoint method based sensitivities.

Fig. 7. The absolute errors (the errors normalized by the maximum absolute sensitivity) of the boundary point sensitivities in percentages for the continuous and the discrete adjoint methods. The discrete adjoint method based sensitivities have significantly low relative errors. The continuous adjoint method based sensitivities also have significantly low error for most of the points, except for the points close to the re-entrant corner.

The difference between the relative errors and absolute errors is that the absolute error formulation is normalized by the same value for all the boundary points. We can see from Fig. 7 that the errors are significantly higher for
the continuous adjoint method (over 10%) near the re-entrant corner. On the other hand, the absolute errors for the discrete adjoint method are significantly lower, with the errors being less than 0.1% for all the boundary points.

Similar trends are obtained for the boundary sensitivities of global quantities such as compliance sensitivities (as opposed to local quantities such as stress). Fig. 8 shows the histograms of relative and absolute errors of compliance sensitivities computed using discrete and continuous adjoint methods. We can see from Fig. 8 that the relative errors for the continuous adjoint method are high. On the other hand, the discrete adjoint method has significantly lower errors, with all the points having an error of less than 1%.

4.1.2. Stress minimization subject to a volume constraint

In this section, the $p$-norm stress of the L-bracket is minimized subject to a volume constraint of $V_0 = 48\%$ of the square bounding box of the L-bracket. This volume constraint corresponds to 75\% of the volume of the L-shaped domain volume. The optimization problem can be framed as

$$\min \sigma_{pn} \quad \text{s.t. } V \leq V_0$$

(37)

There are no initial holes used for this example and the boundary is fixed on the top region of the design domain. Fig. 9 shows the results obtained using the discrete adjoint method for different $p$ values ranging from 6 to 12. We can see from Fig. 9 that the optimized topologies resemble a hook-like structure, and the optimum $p$-norm stress decreases as $p$ increases. Furthermore, the maximum stress $\sigma_{max}$ is reduced from 1.27 for the initial design (Fig. 4b) to 0.71, 0.62, 0.65, 0.62 for $p = 6, 8, 10, 12$, respectively.

Fig. 10 shows the iteration history of the $p$-norm stress $\sigma_{pn}$ and the volume $V$ for the stress minimization example obtained for a move limit ($m_i$ in Eq. (3)) of 0.25. From Fig. 10, we can see that $p$-norm stress converges smoothly in approximately 70 iterations for all values of $p$.

Fig. 11 shows the iteration history of the $p$-norm stress and the volume for the stress minimization example solved using the continuous adjoint method. Specifically, Fig. 11a shows the convergence history when a small move limit of 0.1 is used. We can see that for $p = 6, 8$, the algorithm converges smoothly. For $p = 10, 12$, the $p$-norm of the stress increases every iteration after approximately iteration 100, even though the volume constraint has been satisfied. Fig. 11b shows the convergence history when a higher move limit of 0.25 is used. We can see from Fig. 11b that the $p$-norm of the stress diverges for all values of $p$. The divergence of the continuous adjoint algorithm for higher move limits can be attributed to the reduced accuracy of the boundary point sensitivities, as discussed in Section 4.1.1.
The minimum stress topologies obtained using the discrete adjoint based sensitivities and no initial holes. The $p$-norm of the stress $\sigma_{pn}$ is minimized subject to a volume constraint for different values of $p$. The topologies resemble hook-like structures. The stress distribution is capped at 0.65.

The iteration history of the $p$-norm stress and the volume for the stress minimization example solved using the discrete adjoint based sensitivities. The algorithm converges smoothly in approximately 70 iterations for all values of $p$.

Fig. 12 shows the results obtained when higher resolution meshes of $200 \times 200$ and $300 \times 300$ elements are used for the optimization. The corresponding convergence history of the volume and $p$-norm stress normalized by the $p$-norm stress at iteration 0 ($\sigma_{pn,0}$), is shown in Fig. 13. A move limit of 0.4 is used for the first 100 iterations and the move limit is reduced to 0.2 for the rest of the iterations for both the meshes. From Fig. 13, we can see that the convergence is slower for the higher resolution mesh of $300 \times 300$ elements, than the lower resolution mesh of $200 \times 200$ elements.

Fig. 14 shows the time consumed by different parts of the analysis per iteration. From Fig. 14a, we can see that the time consumed by the FEA solver is the highest, with over 5000 ms when the mesh is $300 \times 300$ elements. The time consumed for boundary sensitivity computation via local perturbations is approximately two orders of magnitude smaller than the FEA solver — with less than 50 milli seconds for the mesh $300 \times 300$ elements. The time taken by the simplex solver and the advection (which includes fast marching method and HJ-WENO based level set update) are even smaller, — with less than 5 milli seconds for the mesh with $300 \times 300$ elements (Fig. 14b)).

In this next example, we investigate the stress minimization subject to a volume constraint with a few holes seeded in the initial design. Fig. 15 shows the optimization results obtained using the discrete adjoint method, for a volume constraint of 32% of the volume of the bounding box of the L-bracket, and for different values of $p$ from 6 to 12. The initial topology has five initial holes seeded as shown on the left hand side of Fig. 15. A move
Fig. 11. The iteration history of the $p$-norm stress and the volume for the stress minimization example solved using the continuous adjoint based sensitivities. When the move limit is 0.1, the algorithm converges smoothly for $p = 6, 8$, while it diverges for $p = 10, 12$. When the move limit is 0.25, the algorithm diverges for all values of $p$.

limit of 0.25 is used. We can see that the maximum stress $\sigma_{\text{max}}$ is reduced from 1.66 for the initial design to 0.72, 0.61, 0.60, 0.61 for $p = 6, 8, 10, 12$, respectively. We can also see that for the optimized designs, the $p$-norm of the stress decreases as $p$ increases. The iteration history of the $p$-norm stress and the volume is shown in Fig. 16, where we can see that the algorithm converges smoothly in approximately 70 iterations.

Fig. 17 shows the iteration history of the $p$-norm stress and the volume for the stress minimization example obtained using the continuous adjoint for the same example. When a small move limit of 0.1 is used, the algorithm converges smoothly for $p = 6, 8$ as shown in Fig. 17a. We can also see from Fig. 17a that for $p = 10, 12$, the $p$-norm of the stress increases every iteration after approximately iteration 100, even though the volume constraint has been satisfied. When a higher move limit of 0.25 is used, the $p$-norm of the stress diverges for $p = 8, 10, 12$. The divergence of the continuous adjoint algorithm for higher move limits is also observed for this example. This divergence can also be attributed to the reduced accuracy of the boundary point sensitivities, as discussed in Section 4.1.1.

4.1.3. Volume minimization subject to a stress constraint

In this section, the volume of the L-bracket is minimized subject to a stress constraint $\sigma^*$. The optimization problem can be framed as

\[
\begin{align*}
\text{min} & \quad V \\
\text{s.t.} & \quad \max(\sigma) \leq \sigma^*
\end{align*}
\]  

(38)
The formulation in Eq. (38) is a popular optimization problem statement used in structural optimization where the stress constraint is the maximum yield stress of the material. The maximum stress function, however, is not differentiable with respect to change in topology. In topology optimization, the $p$-norm stress is used to approximate the maximum stress. However, the $p$-norm stress can overshoot the maximum stress as evidenced by Figs. 9, 15, and 18. Therefore, in this study, we define the following correction factor at every iteration to approximate the maximum stress $\bar{\sigma}$ as

$$\max(\sigma) \approx \bar{\sigma} = \eta \sigma_{pn}$$

where $\eta$ is the correction factor defined at every iteration as

$$\eta = \frac{\max(\sigma)}{\sigma_{pn}}$$
Fig. 14. Time consumed (in milliseconds) by different parts of the analysis per iteration.

Fig. 15. The minimum stress topologies obtained using the discrete adjoint based sensitivities and initial holes. The $p$-norm of the stress $\sigma_{pn}$ is minimized for different values of $p$ subject to a volume constraint.

Fig. 16. The iteration history of the $p$-norm stress and the volume solved using the discrete adjoint based sensitivities. The $p$-norm stress $\sigma_{pn}$ is minimized subject to a volume constraint, and the algorithm converges smoothly in approximately 70 iterations for all values of $p$. The stress distribution is capped at 0.65.
Fig. 17. The iteration history for the stress minimization example solved using the continuous adjoint based sensitivities. The $p$-norm stress is minimized subject to a volume constraint. When the move limit is 0.1, the algorithm converges smoothly for $p = 6, 8$ and diverges for $p = 10, 12$. When the move limit is 0.25, the algorithm diverges for all values of $p = 8, 10, 12$.

The approximated maximum stress $\tilde{\sigma}$ is a differentiable function, with its derivatives given as

$$
\frac{d\tilde{\sigma}}{dz_i} = \eta \frac{d\sigma_{pn}}{dz_i}
$$

(41)

Fig. 18 shows the optimized results obtained using for different values of $p = 6, 8, 10, 12$, and for the stress constraint $\sigma^* = 0.65, 0.75, 0.85$. Fig. 19 shows the stress distribution corresponding to the different topologies shown in Fig. 18. From Figs. 18 and 19, we can see that all optimized designs have rounded the re-entrant corner. For a given value of $p$, the optimized volume decreases as the stress constraint increases. Moreover, for a given stress constraint, the optimized volume obtained for higher values of $p = 10, 12$ is lower than the optimized volume obtained for $p = 6$. The iteration history of the maximum stress ($\max(\sigma)$) and the volume is shown in Fig. 20 for the stress constraint $\sigma^* = 0.65$ and $p = 8, 10, 12$.

4.2. 3D orthogonal brackets

In this section, volume minimization subject to stress constraints for 3D problems is investigated. Specifically, we optimize the topologies of 3D orthogonal brackets for three load cases, as shown in Fig. 21. The optimization
Fig. 18. Minimum volume topologies obtained for different stress constraints using the discrete adjoint method. For a given value of $p$, the optimized volume decreases as the stress constraint increases. For a given stress constraint, the optimized volume obtained for higher values of ($p = 10, 12$) is better than the optimized volume obtained for $p = 6$.

Fig. 19. Stress distribution for the optimal topologies shown in Fig. 18. The stress distribution is capped at 0.65.

The problem is stated as

$$\begin{align*}
\min \quad & V \\
\text{s.t.} \quad & \max(\sigma) \leq \sigma^*
\end{align*}$$

(42)
Fig. 20. Iteration history of the maximum stress \( \max(\sigma) \) and the volume for the stress constraint \( \sigma^* = 0.65 \) and \( p = 8, 10, 12 \).

Fig. 21. Schematics of the design domains of 3D orthogonal brackets that are clamped on the top subjected to forces as shown. The forces are applied vertically up for load case — A. The forces are applied vertically but in opposite directions for load case — B. For load case — C, the forces are applied in the same plane but in orthogonal directions.

The orthogonal bracket (dimensions are \( 2 \times 2 \times 2 \)) is clamped on the top surface and the forces are applied at the tips as shown in Fig. 21. For the first load case (Load case — A in Fig. 21), the forces on the orthogonal bracket are applied in the same direction (vertically up). For the second load case (Load case — B), the loads are acting in opposite directions (vertically up on one side and vertically down on the other). In Load case — C, the loads are applied in the same plane, but in orthogonal directions on either side. A finite element mesh of \( 100 \times 100 \times 100 \) hexahedral elements is used for the analysis, where the entire bounding box of the design domain is meshed. The initial design is a solid geometry covering the entire design domain.

Fig. 22 shows the optimization results obtained for load case corresponding to Load case — A, for \( F = 2.5 \times 10^{-3} \). Under this loading the initial design has a stress concentration at one of the re-entrant corners as shown in Fig. 22. The maximum stress of the initial design is 0.38. The optimized topology obtained for a stress constraint of \( \sigma^* = 0.28 \) is also shown in Fig. 22. The optimized design satisfies the stress constraint, and the stress concentration is eliminated. The volume of the optimized solution is 31\% of that of the design domain.

Fig. 23 shows the optimization results obtained for Load case — B where \( F = 2.5 \times 10^{-3} \). The initial design has a stress concentration at the two corners as shown in Fig. 24. The maximum stress of the initial design is 0.33. The stress constraint used for this example is again \( \sigma^* = 0.28 \). The optimized design satisfies the stress constraint and the stress concentration is eliminated. The optimized design has a volume of 25\% of that of the design domain.

Fig. 24 shows the optimization results obtained for Load case — C where \( F = 2.5 \times 10^{-3} \). The initial design has a stress concentration along the edges as shown in Fig. 24. The maximum stress of the initial design is 0.22. The stress constraint used for this example is \( \sigma^* = 0.18 \). The optimized design satisfies the stress constraint removes the stress concentrations. The optimized design has a volume equal to 47\% of that of the design domain.

Finally, we present the results when all the load cases are applied simultaneously. The optimization problem is

\[
\begin{align*}
\min & \quad V \\
\text{s.t.} & \quad \max(\sigma_A) \leq \sigma^* \\
& \quad \max(\sigma_B) \leq \sigma^* \\
& \quad \max(\sigma_C) \leq \sigma^* 
\end{align*}
\]

(43)
Fig. 22. The results obtained for a stress constraint of $\sigma^* = 0.28$ for load case — A. The optimized topology has a volume of approximately 31% of that of the initial design. The optimized stress distribution does not have any high stress concentrations that are present in the initial stress distribution. The figures in each row are views from the same vantage point.

Fig. 23. The results obtained for a stress constraint of $\sigma^* = 0.28$ for load case — B. The optimized topology has a volume of approximately 25% of that of the initial design. The optimized stress distribution does not have any high stress concentrations that are present in the initial stress distribution. The figures in each row are views from the same vantage point.

where $\sigma_A$, $\sigma_B$, and $\sigma_C$ are the stress for load cases A, B, and C, respectively. Fig. 25 shows the results obtained when all the load cases are applied together. The stress constraint used is $\sigma^* = 0.3$. The optimized topology has a volume of 55% of the initial design. The maximum stress for load cases A, B, and C are 0.3, 0.27, and 0.3, respectively. The iteration history of the maximum stress ($\max(\sigma)$) for the three load cases and the volume of the topology is shown in Fig. 26, where we can see that the optimization converges in under 200 iterations.
Fig. 24. The results obtained for a stress constraint of $\sigma^* = 0.18$ for load case — C. The optimized topology has a volume of approximately 47% of that of the initial design. The optimized stress distribution does not have any high stress concentrations that are present in the initial stress distribution. The figures in each row are views from the same vantage point.

Fig. 25. The results obtained for a stress constraint of $\sigma^* = 0.3$ for all load cases. The optimized topology has a volume of approximately 53% of that of the initial design. The optimized stress distribution for different load cases. The figures in each row are views from the same vantage point.

5. Conclusion

A discrete adjoint method for computing boundary point sensitivities in classical level set topology optimization is presented. Stress minimization and stress constrained examples, which are considered to be challenging in level set topology optimization, are chosen to benchmark the developed method. The adjoint equations are derived using the discretized governing equations. A semi-analytical sensitivity formulation is presented, where the sensitivity of the Lagrangian function with respect to the volume fraction of an individual element is combined with a local perturbation of the level set function around a boundary point to compute the boundary point sensitivity. The
novelty of our method lies in the fact that we do not perturb the boundary point directly for computing the local perturbations. Instead, we perturb the boundary implicitly via modifying the level set function locally. In other words, the developed method moves the level set locally by perturbing the boundary point with a given velocity. This enables us to obtain the sensitivities without using any smoothing or interpolation typically used in classical level set topology optimization, resulting in improved accuracy and fast convergence when compared to using the continuous adjoint formulation. Finally, we demonstrate the developed method for obtaining minimum mass topologies of 3D orthogonal brackets subjected to stress constraints under multiple load cases.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We acknowledge the support from NASA’s Transformation Tools and Technologies Project, USA (grant number 80NSSC18M0153) and the U.S. National Science Foundation (grant number CMMI-1762530). We like to thank Prof. Julian Norato for the discussions we had on stress constraint formulation. We also would like to thank Dr. Justin Gray for the discussions we had on the discrete adjoint method.

References