Computational Mechanics
https://doi.org/10.1007/s00466-023-02325-8

ORIGINAL PAPER

Robust, strong form mechanics on an adaptive structured grid: efficiently solving variable-geometry near-singular problems with diffuse interfaces

Vinamra Agrawal[1] · **Brandon Runnels**[2]

Received: 5 December 2022 / Accepted: 22 March 2023

Abstract
Many solid mechanics problems on complex geometries are conventionally solved using discrete boundary methods. However, such an approach can be cumbersome for problems involving evolving domain boundaries due to the need to track boundaries and constant remeshing. The purpose of this work is to present a comprehensive strategy for efficiently solving such problems on an adaptive structured grid, while expositing some of the basic yet important nuances associated with solving near-singular problems in strong form. We employ a robust smooth boundary method (SBM) that represents complex geometry implicitly, in a larger and simpler computational domain, as the support of a smooth indicator function. We present the resulting semidefinite equations for mechanical equilibrium, in which inhomogeneous boundary conditions are replaced by source terms. In this work, we present a computational strategy for efficiently solving near-singular SBM-based solid mechanics problems. We use the block-structured adaptive mesh refinement method, coupled with a geometric multigrid solver for an efficient solution of mechanical equilibrium. We discuss some of the practical numerical strategies for implementing this method, notably including the importance of grid versus node-centered fields. We demonstrate the solver's accuracy and performance for three representative examples: (a) plastic strain evolution around a void, (b) crack nucleation and propagation in brittle materials, and (c) structural topology optimization. In each case, we show that very good convergence of the solver is achieved, even with large near-singular areas, and that any convergence issues arise from other complexities, such as stress concentrations.

Keywords Finite differences · Elasticity · Plasticity · Fracture · Topology optimization

1 Introduction

Many computational mechanics problems involve analyzing mechanical systems with highly variable geometry. Such problems require that the mechanical deformations, and resulting stresses, be resolved subject to a set of complex, time-varying, and sometimes unknown topologies. Such examples include, but are not limited to, fracture mechanics, problems involving material growth or removal (such as dendrite growth), or structural topology optimization. In all of these, it is essential to accurately solve mechanical equilibrium equations. Computational mechanics has historically been overwhelmingly dominated by the finite element method (FEM) due to its ability to conform to arbitrary geometry through iso-parametric elements. Indeed, FEM is nearly synonymous with computational elasticity. However, in the case of variable topology, the key advantage of FEM—conformal meshing with isoparametric elements—is less beneficial. This may necessitate costly mid-simulation remeshing, the use of an explicitly meshed and overlayed boundary, or the use of excessive refinement in anticipation of topological change.

The strong form method with finite differences is an attractive alternative for such problems. Recently, it was shown by the authors that this method may be coupled to the block-structured adaptive mesh refinement (BSAMR) method, along with the geometric multigrid method, to produce a highly efficient linear elastic solver [1]. The solver has been applied to numerous small strain [2–4] and finite deformation [5] mechanics problems. Phase field methods have also been implemented using this method, albeit with

✉ Brandon Runnels
brunnels@uccs.edu

[1] Department of Aerospace Engineering, Auburn University, Auburn, AL, USA

[2] Department of Mechanical and Aerospace Engineering, University of Colorado, Colorado Springs, CO, USA

Published online: 24 April 2023

problematically slow convergence rates due to limitations of the method that will be addressed in this work [6]. Many of the problems of interest are reducible to representative volume elements (RVE), for which the finite difference method is ideally suited. We seek to apply this method to problems in which the geometry may be considered to be variable.

We let the "smooth boundary method" (SBM) refer to the approach in which a complex geometry is defined within a simpler computational domain as the support of some smooth indicator function $\phi > 0.5$. Smoothness requires that the transition from solid to void is continuous, and we assume in general that the ϕ varies smoothly from 0 to 1 over some finite interval. We consider SBM to refer specifically to the technique of replacing discrete boundary conditions with equivalent source terms, such that the boundary conditions are recovered exactly in the sharp interface limit. By thus embedding the complex geometry through a diffused interface, SBM circumvents the challenges associated with domain meshing encountered in discrete interface approaches. The SBM has been used to solve partial differential equations with general boundary conditions on complex boundaries and can easily be coupled with diffuse boundary methods (such as phase field) by evolving the order parameters using a thermodynamic equation. Some examples of SBM applications include the use of phase field methods to study corrosion in Mg alloys [7], mass flux boundary conditions in fluids [8], and general partial differential equations [9, 10].

The SBM's efficiency relies heavily on using BSAMR to resolve the diffuse boundary. When suitably coupled, SBM effectively eliminates the need for explicitly defining the mesh since the interface can be resolved with an appropriate resolution and the mesh can be updated to track the evolving interface. The above-mentioned BSAMR strategy has been widely used for high-performance computational fluid mechanics problems [11–14] and, to some extent, for solid mechanics problems [1, 3, 6]. BSAMR stores and evolves each mesh level independently, evolving finer levels with smaller time steps to avoid overly restrictive CFL conditions on coarser levels. The information between levels is communicated through averaging (fine level to coarse level) and ghost cells (coarse level to fine level).

Application of SBM in solid mechanics applications can lead to semi-definite problems due to the lack of uniqueness resulting from a mesh-resolved "void" region. This situation often arises in topology optimization problems where the simulation domain is an output rather than the input but is endemic to any implicit boundary method. Without properly addressing the semi-definiteness of the operator, the result can be poor (or no) convergence and, worse, an incorrect solution.

In this work, we present computational techniques to allow for the efficient solution of semi-definite smooth boundary problems using the finite difference method with BSAMR. The paper is structured in the following way. In Sect. 2, the SBM is formalized for elasticity, and the specific challenges of solving semidefinite problems are addressed. This section also describes some of the computational methods and challenges unique to solving problems of this nature. In Sect. 3, three representative examples are presented that demonstrate the model's effectiveness: (1) plasticity with variable geometry, (2) phase field fracture mechanics, and (3) structural topology optimization. Each example is somewhat self-contained, so the disparate applications will find relevance in their respective communities. We conclude by highlighting some limitations of the framework in its current form.

2 Computational methods

In this section, we present the key elements of the SBM method and practical strategies for its implementation. We first present the formulation of the equations of linear elasticity with traction boundary conditions with SBM. We then outline the reflux-free multigrid implementation of the solver using BSAMR. Next, we emphasize the need for choosing a cell-based indicator field for the stability of the solver. Finally, we outline the implementation of material models as a vector space to efficiently work with the solver.

2.1 Diffuse boundary method for linear elasticity

In this section we present the diffuse boundary formulation of mechanical equilibrium for a linear elastic material. We consider a body of interest occupying some region $\Omega \subset \mathbb{R}^3$, with a natural boundary $\partial\Omega$ upon which surface tractions $\boldsymbol{t}^0(\boldsymbol{y}) \quad \boldsymbol{y} \in \partial\Omega$ are prescribed as boundary conditions. (We do not consider the diffuse formulation of essential, i.e. displacement, boundary conditions at this time.) In the discrete setting, the problem is posed as a differential equation with boundary conditions prescribed at the domain boundary. In the diffuse setting, we represent boundary effects implicitly. To construct the diffuse problem, we first replace the explicit domain Ω with a continuous function, called an order parameter ϕ^ϵ, that represents Ω. In the limit as $\epsilon \to 0$, the support of ϕ^ϵ is identical to the discrete-boundary domain, Ω; this is called the sharp interface limit. For $\epsilon > 0$, we define the diffuse domain and boundary to be, respectively: the support of $\phi^\epsilon = 1$, denoted Ω_ϵ; the support of $\nabla\phi^\epsilon$, denoted $\partial_\epsilon\Omega_\epsilon$. We require that $\phi^\epsilon = 0$ outside of $\partial_\epsilon\Omega_\epsilon \cup \Omega_\epsilon$, and that $\partial_\epsilon\Omega_\epsilon = \partial\Omega \times (-\epsilon/2, \epsilon/2)$, bounding the diffuse region to the ϵ neighborhood of the discrete boundary. As long as $\epsilon << r_{\min}$, the smallest radius of curvature of $\partial\Omega$, we require that there exist some parameterization of ϕ^ϵ within the diffuse boudnary by

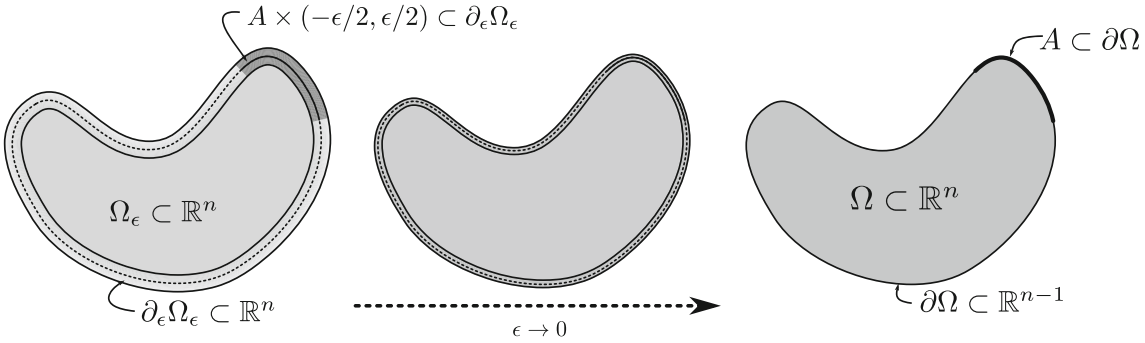


Fig. 1 Illustration showing the diffuse boundary construction approaching the sharp interface limit. The diffuse domain Ω_ϵ is the support of $\phi^\epsilon = 1$, equal to Ω in the limit. The diffuse boundary $\partial_\epsilon \Omega_\epsilon$ is the ϵ an open set defined in the region where $0 < \phi^\epsilon < 1$, and vanishes in the limit. It is also defined as the $\epsilon/2$-neighborhood of $\partial\Omega$. The shaded region on the left is $A \times (-\epsilon/2, \epsilon/2)$, a subset of $\partial_\epsilon \Omega_\epsilon$ corresponding to an arbitrary subset A of $\partial\Omega$

$$\phi^\epsilon(\boldsymbol{y} + s\boldsymbol{n}) = \hat{\phi}(s) \quad \boldsymbol{y} \in \partial\Omega, \quad s \in (-\epsilon/2, \epsilon/2) \tag{1}$$

where $\boldsymbol{n}$ is the normal to the discrete interface. $\hat{\phi}$ is some Lipschitz function describing the behavior of ϕ over the diffuse interface, and is generally a regularized step function over the interval $(-\epsilon/2, \epsilon/2)$. Its derivative is not defined in the limit as $\epsilon \to 0$, but approaches the Dirac delta distribution (Fig. 1). With these definitions and restrictions in hand, it is possible to establish the following theorem, which was presented in [8]:

Theorem 1 *Let ϕ^ϵ be an idealized order parameter with length scale ϵ, and let f and g be either scalar or vector-valued bounded functions, with $\boldsymbol{n} \cdot \nabla g$ bounded in $\partial_\epsilon \Omega_\epsilon$.*[1] *Then the following holds:*

$$\lim_{\epsilon \to 0} \int_A \int_{-\epsilon/2}^{\epsilon/2} \left(f\phi^\epsilon + g|\nabla\phi^\epsilon| \right) ds\, dA = \int_A g\, dA \quad \forall A \subset \partial\Omega \tag{2}$$

We can now present the diffuse interface formulation of mechanical equilibrium. Recall the usual sharp-interface equations of momentum conservation in the context of elasticity, with kinematics linearized about some eigenstrain $\boldsymbol{\varepsilon}^0$, are:

$$\mathbb{C}(\boldsymbol{x}) \left(\operatorname{grad} \boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{\varepsilon}^0(\boldsymbol{x}) \right) - \boldsymbol{\sigma}(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Omega$$
$$\boldsymbol{u}^0(\boldsymbol{y}) - \boldsymbol{u}(\boldsymbol{y}) = 0, \quad \boldsymbol{y} \in \partial_1\Omega \tag{3a}$$
$$\operatorname{div} \boldsymbol{\sigma}(\boldsymbol{x}) - \boldsymbol{b}(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Omega$$
$$\boldsymbol{t}^0(\boldsymbol{y}) - \boldsymbol{\sigma}(\boldsymbol{y})\hat{\boldsymbol{n}}(\boldsymbol{y}) = 0, \quad \boldsymbol{y} \in \partial_2\Omega. \tag{3b}$$

Equations (3a) and (3b) are the constitutive and mechanical equilibrium conditions. Here $\boldsymbol{u}(\boldsymbol{x})$ is the displacement field, $\mathbb{C}(\boldsymbol{x})$ is the fourth order elastic modulus tensor, $b(\boldsymbol{x})$ is the body force, $\boldsymbol{u}^0(\boldsymbol{x})$ is the displacement specified at the Dirichlet boundary $\partial_1\Omega$, $\boldsymbol{t}^0(\boldsymbol{x})$ is the traction specified in the traction boundary $\partial_2\Omega$, and $\hat{\boldsymbol{n}}(\boldsymbol{x})$ is the normal vector at any point on the boundary. We also assume the conventional major and minor symmetries of $\mathbb{C}(\boldsymbol{x})$ to allow us to directly work with the displacement $\boldsymbol{u}(\boldsymbol{x})$. While the Dirichlet boundary condition is essential to solid mechanics problems, imposing them on the diffused boundary has limited use cases. Therefore, we limit our attention to the traction boundary condition. To move to the diffuse boundary setting, we introduce the diffuse traction $\hat{\boldsymbol{t}}^0 : \partial_\epsilon \Omega_\epsilon \to \mathbb{R}^3$,

$$\hat{\boldsymbol{t}}^0(\boldsymbol{x} = \boldsymbol{y} + s\boldsymbol{n}) = \boldsymbol{t}^0(\boldsymbol{y}). \tag{4}$$

The diffuse traction is defined everywhere in the diffuse boundary as the value of the discrete traction at the closest point on the discrete boundary. (Recall that this is only valid as long as ϵ is smaller than the smallest radius of curvature of the discrete boundary; otherwise, the diffuse traction is multiply defined.) Now, consider the following diffuse-boundary modification of equation (3b):

$$\left(\operatorname{div} \boldsymbol{\sigma} - \boldsymbol{b} \right) \phi^\epsilon = \left(\hat{\boldsymbol{t}}^0 - \boldsymbol{\sigma}\boldsymbol{n} \right) |\nabla\phi^\epsilon|. \tag{5}$$

It is straightforward to show that the interior momentum equation holds by considering the weak form of the above equation. Integrate both sides over an arbitrary interior, measurable region $V \subset \Omega_\epsilon$,

$$\int_V \left(\operatorname{div} \boldsymbol{\sigma} - \boldsymbol{b} \right) \phi^\epsilon\, dV = 0 \quad \forall \operatorname{meas}.\, V \subset \Omega_\epsilon. \tag{6}$$

The right hand side vanishes since $|\nabla\phi^\epsilon|$ is zero in Ω_ϵ, by construction. But since equation (6) holds for all subsets of the diffuse interior, the integrand itself is zero for all $\boldsymbol{x} \in \Omega_\epsilon$. To show recovery of the boundary condition, we once again

[1] We note that the $\boldsymbol{n} \cdot \nabla g$ boundeness restriction is erroneously absent from the original presentation of the theorem in [8].

take the weak form of equation (5), this time over an arbitrary region within the diffuse boundary:

$$\int_A \int_{-\epsilon/2}^{\epsilon/2} \left(\left(\operatorname{div} \boldsymbol{\sigma} - \boldsymbol{b} \right) \phi^\epsilon - \left(\hat{\boldsymbol{t}}^0 - \boldsymbol{\sigma} \boldsymbol{n} \right) |\nabla \phi^\epsilon| \right) ds \, dA = 0 \quad \forall A \subset \partial\Omega \tag{7}$$

Applying Theorem 1 in the sharp interface limit reduces the above expression to

$$\int_A \left(\boldsymbol{t}^0 - \boldsymbol{\sigma} \boldsymbol{n} \right) dA = 0 \quad \forall A \subset \partial\Omega, \tag{8}$$

which shows that the integrand must be true for all $\boldsymbol{y} \in \partial\Omega$ since the above weak form holds for all subsets of $\partial\Omega$. This confirms that the traction boundary conditions are exactly recovered as $\epsilon \to 0$ for the diffuse interface formulation in Eq. (5). Finally, rearranging and noting that $\boldsymbol{n} = \nabla\phi^\epsilon / |\nabla\phi^\epsilon|$., Eq. (5) simplifies to

$$\operatorname{div} \left(\phi^\epsilon \boldsymbol{\sigma} \right) - \phi^\epsilon \boldsymbol{b} = \hat{\boldsymbol{t}}^0 |\nabla \phi^\epsilon|. \tag{9}$$

In summary, the boundary conditions associated with the discrete natural boundary $\partial\Omega$ are replaced by an equivalent source term that mimics the effect of the discrete natural boundary, exactly recovering its behavior in the sharp interface limit. The selection of ϕ can be determined by construction (for instance, explicitly prescribing an indicator function based on a predetermined geometry) or by coupling to a separate set of equations that describe the behavior of ϕ (such as phase field). In both cases, care must be taken that the behavior of ϕ does not deviate far from the requirements necessary for the validity of the diffuse boundary method to hold.

2.2 Reflux-free multigrid implementation

We implemented equation (9) in an in-house code, Alamo [1], a finite-difference based multi-level, multi-grid, and multi-component solver. Alamo uses AMReX libraries for block-structured adaptive mesh refinement (BSAMR) [11]. BSAMR divides the mesh into levels such that each level contains cells of the same size. Each level is treated independently, and the information between levels is communicated through restriction, relaxation, and restriction operations using ghost cells. As a result, BSAMR is highly scalable and enables massive parallelism across CPU and GPU cores. BSAMR can be naturally combined with standard multigrid methods by treating refined levels as an extension to the multigrid method's coarse/fine level sequence. The mesh refinement and coarsening are triggered and performed at regular intervals using the Berger–Rigoutsos algorithm [15].

Multigrid methods often require special treatment at the coarse-fine level interface during restriction operations. Improper handling of the coarse-fine interface can result in spurious forces at the interface and overall poor convergence of the solver. The coarse-fine interface can be handled using a "reflux" operation [16] where the operator is updated at the interface to use the information at both levels. However, this process can be difficult for a complicated operator such as the one for linear elasticity. An alternate "reflux-free" procedure was proposed by [1] where the levels are padded with an extra layer of ghost nodes/cells to ensure the translational symmetry of the restriction operator and that information at the coarse/fine boundary is updated with the current information. This circumvents the need for a special stencil at the coarse/fine boundary and results in good convergence.

2.3 Node-based and cell-based fields in strong form multigrid elasticity

The method discussed in Sect. 2.2 works well for elasticity problems but can behave very poorly unless care is taken to respect the proper placement of the relevant fields on the grid. In fluid mechanics, it is often necessary to place some quantities at nodes, some in cells, and some on cell faces, etc., with the exact scheme differing between methods [17–20]. On the other hand, in solid mechanics, values such as displacements are typically stored at nodes, whereas quantities governing material response are generally located at quadrature points within the element. Some solid mechanics methods, such as optimal transport meshfree [21], smoothed particle hydrodynamics [22], and the material point method [23–25], though not strictly finite element, still carefully distinguish between nodes and material points. The finite volume method has been used for solid mechanics, though not nearly as extensively as in fluid mechanics, and it is known that a staggered grid approach is needed to avoid the phenomenon of checkerboarding [26].

In the present method, which uses a regular cartesian grid, values may be stored at points, edges, faces, or cells. At first glance, there is no obvious reason for storing values at one location over another; indeed, it is possible to develop a solver in which all values are stored at faces or all values at nodes. In previous work, nodal locations were chosen for all quantities of interest [1]. Interestingly, this choice had no previous negative impact on the solver. However, as we will discuss here, the location of values is, in fact, quite essential to the performance of the solver when considering near-singular problems. Specifically, it is important that displacements be stored at the nodes, whereas the order parameter must be stored as a cell-based field. In this section, we provide two explanations for this: the first, from a practical perspective, and the second, by considering the geometry of the problem.

Our analysis considers the key aspect of the multigrid solver as applied to the elasticity problem: smoothing. Multigrid methods work primarily through the use of a smoother

between levels. The choice of smoothing algorithm can vary, but the preeminent solvers are generally Gauss–Seidel, Jacobi, or a variant of one of these methods. These methods are popular due to their ease of implementation, their parallel efficiency. For geometric multigrid methods, they are particularly attractive because they smooth high frequency error faster than low frequency error (unlike, for instance, the conjugate gradient method) [27].

Here, we consider the Jacobi method, which for an operator A is given by

$$\boldsymbol{u}^{n+1} = D^{-1}(\boldsymbol{b} - (A - D)\boldsymbol{u}^n), \tag{10}$$

where $\boldsymbol{u}^n$ is the solution at iteration n, $D = \mathrm{diag}(A)$, and $\boldsymbol{b}$ is the right hand side. We see that the inverse of the diagonal is of central importance: most notably, that if any of the diagonal elements of D are zero, the corresponding rows of $(\boldsymbol{b} - (A - D)\boldsymbol{u}^n)$ must be zero as well. To see the significance of this, consider the discretized elastic operator for a constant modulus C in one dimension, in which all values are stored at nodes:

$$\begin{aligned}\mathrm{div}\left(\phi^\epsilon \mathbb{C}\,\mathrm{grad}\,\boldsymbol{u}\right)_i = C\left(\frac{\phi^\epsilon_{i+1} - \phi^\epsilon_{i-1}}{2\Delta x}\right)\left(\frac{u_{i+1} - u_{i-1}}{2\Delta x}\right)\\ + C\phi^\epsilon_i\left(\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}\right)\end{aligned} \tag{11}$$

The diagonal of the operator, then, is nothing other than the coefficient of the u_i term, that is,

$$\mathrm{diag}(\mathrm{div}[\phi_\epsilon \mathbb{C}\,\mathrm{grad}])_i = -2\frac{C\phi^\epsilon_i}{\Delta x^2}, \tag{12}$$

and so the corresponding Jacobi update is thus given by

$$\begin{aligned}u^{n+1}_i &= -2\frac{\Delta x^2}{C\phi^\epsilon_i}\left\{b_i - \left[C\left(\frac{\phi^\epsilon_{i+1} - \phi^\epsilon_{i-1}}{2\Delta x}\right)\left(\frac{u^n_{i+1} - u^n_{i-1}}{2\Delta x}\right)\right.\right.\\ &\quad \left.\left. + C\phi^\epsilon_i\left(\frac{u^n_{i+1} + u^n_{i-1}}{\Delta x^2}\right)\right]\right\} \qquad (13)\\ &= -2\Delta x^2\left\{\frac{b_i}{\phi^\epsilon_i C} - \left[\frac{1}{\phi^\epsilon_i}\left(\frac{\phi^\epsilon_{i+1} - \phi^\epsilon_{i-1}}{2\Delta x}\right)\left(\frac{u^n_{i+1} - u^n_{i-1}}{2\Delta x}\right)\right.\right.\\ &\quad \left.\left. + \left(\frac{u^n_{i+1} + u^n_{i-1}}{\Delta x^2}\right)\right]\right\} \qquad (14)\end{aligned}$$

By inspection, it is clear that any nonzero body force will produce divergent behavior if applied where $\phi^\epsilon_i = 0$; this is natural since such a problem would be ill-defined. However, an inspection of the next term shows a second vulnerability: a point at which $\phi^\epsilon_i = 0$ may still induce instability depending on the values at *the adjacent nodes*. Therefore this can (and does) induce divergence at the nodes where the order parameter is zero, but the solution is well-defined: i.e., at the boundaries of the support of ϕ^ϵ.

On the other hand, consider the corresponding Jacobi update if the field ϕ^ϵ is stored in cells rather than at nodes, where fractional indices are used to denote locations of cells:

$$\begin{aligned}u^{n+1}_i = -2\Delta x^2\left\{\frac{2b_i}{(\phi^\epsilon_{i+1/2} + \phi^\epsilon_{i-1/2})C}\right.\\ - \left[\frac{2}{(\phi^\epsilon_{i+1/2} + \phi^\epsilon_{i-1/2})}\left(\frac{\phi^\epsilon_{i+1/2} - \phi^\epsilon_{i-1/2}}{\Delta x}\right)\right.\\ \left.\left.\times\left(\frac{u^n_{i+1} - u^n_{i-1}}{2\Delta x}\right) + \left(\frac{u^n_{i+1} + u^n_{i-1}}{\Delta x^2}\right)\right]\right\}.\end{aligned} \tag{15}$$

One can see by inspecting the second term (assuming, again, that the body force is responsibly applied) that the problem of instability is eliminated. Divergent behavior can now only occur when ϕ^ϵ is zero at both $i + 1/2$ and $i - 1/2$; but if this happens, then the difference between the two values would be zero as well. One may apply this same exercise to this problem with non-uniform elastic modulus, or to the problem in 2D or 3D, with the same result. This underscores the importance of a staggered grid approach that, though commonly used in other finite difference methods, was absent from prior finite difference implementations of the SBM on BSAMR grids.

One may take this analysis several steps further by considering the geometric significance of the solution and ϕ^ϵ fields. Within the past couple of decades, the tools of exterior calculus have been applied to the problem of linear elasticity [28–30], which identifies displacement fields as vector-valued 1-forms, body forces as vector-valued 3-forms, etc. This has been extended to the field of computational mechanics through the emerging sub-discipline of discrete differential geometry (DDG), which allows the explicit realization of exterior calculus constructs in the context of discrete mesh elements [31–33]. While a thorough treatment of DDG in the context of mechanics is outside the scope of the paper, we outline the underlying concept. The traditional fields (displacements, stress, etc) can be replaced by the DDG construct of forms, where an n-form is a field that can be integrated over an n-dimensional manifold. That is, n-forms contain a notion of geometry that is absent from raw fields, and which correspond to the proper integration domain. For instance, stress are integrated over surfaces, which are two-dimensional manifolds; therefore, stress is a 2-form; body forces are integrated over volumes, which are three-dimensional manifolds; therefore, body forces are 3-forms; displacements are evaluated at points, which are zero-dimensional manifolds; therefore, displacements are 0-forms; and so on. In the present work, the order parameter ϕ is integrated over a voluem; therefore, ϕ is a 3-form. In the discrete setting on a regular grid, 0-forms may be identified as nodal fields, 1-forms as edge fields, 2-forms as face fields, and 3-forms as cell fields. (The study of DDG has produced

analagous interpretations for non-regular and unstructured meshes as well.) Thus, we see that by representing displacements using a node-based grid (0-forms) and ϕ using a cell-based grid (3-forms), we are preserving the geometric structure of the these fields. For a more thorough discussion of the geometric interpretation of classical continuum theory, we refer the reader to the above references and to the excellent book by Frenkel [34].

2.4 Material model vector space

One of the key advantages of the BSAMR approach is its ability to adapt the mesh rapidly. The block-structured multilevel data structures afford rapid regridding in a parallel-efficient manner. Regridding, and inter-level communication, relies on the ability to rapidly transfer information between AMR (or multigrid) levels, usually in the form of interpolation or prolongation. When working with primitive fields such as velocity, density, pressure, etc., interpolations and prolongations are easy to compute. However, such operations are not always obvious in the context of material modeling. Materials often exhibit highly anisotropic behavior, with material response often depending on numerous parameters and time-evolving internal variables. In order for BSAMR to function correctly with such models as these, it is necessary to address the constraints and requirements needed for material modeling. Moreover, as it is a commonplace in any solid mechanics code to allow for modular material models, an *ad hoc* implementation is insufficient. Therefore, we prescribe the minimum requirements needed for a versatile implementation of material models in a BSAMR context.

The aforementioned requirements for BSAMR data structures are equivalent to those for the mathematical structure of a vector space. Specifically, BSAMR requires the consistency of solid models between AMR and multigrid levels, which is achieved through interpolation and restriction operations, which require the definition of addition and scalar multiplication of solid model objects. Therefore we impose the requirement on material models that they must satisfy the properties of a vector space. Let $\mathcal{M}$ denote the vector space corresponding to a certain material model. The salient properties are: (1) the existence of a "vector addition" operation, typically denoted $+$, such that $a + b \in \mathcal{M}\ \forall a, b \in \mathcal{M}$; and (2) the existence of a "scalar multiplication" operation denoted by "$*$" or concatenation, such that $\alpha * a = \alpha a \in \mathcal{M}$ $\forall \alpha \in \mathbb{R}, \forall a \in \mathcal{M}$, and (3) the existence of an identity element $0 \in \mathcal{M}$ such that $0 + a = a\ \forall a \in \mathcal{M}$. Other requirements include associativity and commutativity of $+$, the inverse of $+$, compatibility of $*$ and identity under $*$, and distributivity of $*$; generally, these are not troublesome to enforce, and they furnish a valuable framework for unit testing at the implementation phase. Properties 1–3 must hold not only for the internal variables stored in each material model but for their functions as well: specifically, the zeroth, first, and second derivatives of energy (W, DW, DDW), as well as any functions defining the evolution of internal variables. For instance, $(a + b).DW(\text{''}) \stackrel{!}{=} a.DW(\boldsymbol{\varepsilon}) + b.DW(\boldsymbol{\varepsilon})$, where $\boldsymbol{\varepsilon}$ is the local strain tensor. The structure also allows for the inverse of models to exist, which allows for derivatives of models to be calculated, e.g. $(da/dx).W(\varepsilon)$. For instance, the calculation of the gradient in the x_1 direction of a model field $a(\boldsymbol{x})$ using finite difference would be

$$\Big(\frac{da}{dx_1}\Big).DW(\boldsymbol{\varepsilon}) \approx \Big(\frac{a(x_1 + \frac{1}{2}\Delta x_1, x_2, x_3) - a(x_1 - \frac{1}{2}\Delta x_1, x_2, x_3)}{\Delta x_1}\Big). DW(\boldsymbol{\varepsilon}), \tag{16}$$

which makes use of the full agebraic structure of a: namely, algebraic operations of scalar multiplication and addition, as well as inversion.

The model $-a$ is the inverse of a with negative (and consequently unphysical) material properties. We emphasize the importance of placing checks in place to ensure that unphysical models are not accidentally used to calculate real physical properties.

The vector space material model requirement has immediate implications on the implementation of material models. Consider the simple case of linear elastic isotropic, which is generally parameterized by two properties, often chosen as Young's modulus E and Poisson's ratio ν. One can construct a material model based on these two properties (E, ν) along with the addition operation $(E_1, \nu_1) + (E_2, \nu_2) = (E_1 + E_2, \nu_1 + \nu_2)$, and scalar multiplication $\alpha(E, \nu) = (\alpha E, \alpha\nu)$. However, such a model violates the vector space behavior of W, DW, and DDW, since the energy, stress, and strain depend on the ratio ν/E rather than bilinearly on ν and E separately. Thus, one can instead store the Lamé constants λ, μ, on which the dependence of W, DW, DDW is bilinear.

Another salient example is the implementation of cubic elasticity, which requires the storage of rotational information along with elastic moduli C_{11}, C_{12}, C_{44}. One may also include an eigenstrain $\boldsymbol{\varepsilon}_0$, reflecting plastic evolution, thermal expansion, etc. Euler angles are sometimes used to store the local rotation but are clearly a poor choice here, as Euler angles do not form a vector space. Instead, we used quaternions to store rotation information, as they possess an algebraic structure that is relatively easy to implement. One complication is that quaternions must be normalized to obtain rotation information, meaning that the W, DW, DDW functions for the zero element are ill-defined. However, in practice, it is generally never the case that the zero element would be called upon to return those values, and if it did, it would always return zero anyway by necessity. The remaining values in the model readily admit an algebraic structure and are easily implemented.

Operator overloading (that is, the definition of functions using the syntax of operators such as +, −, +=, etc.) was used to provide both unary and binary vector addition and scalar multiplication operations, and unit tests can enforce associativity, commutativity, etc. We used forced code inlining (compile-time restructuring to place called code "inline" with the calling code) to ensure that there is no function call overhead, and C preprocessor macros can be used to implement all operators with minimal boilerplate code required automatically. We required functions such as `Zero` to furnish the zero element, with template metaprogramming used to enforce that all models comply with the vector space requirement. We refer the interested reader to [35–37] for further discussion of relevant high performance computational techniques.

3 Examples

In this section, we demonstrate the performance and accuracy of the solver and the SBM implementation within Alamo using problems within solid mechanics.

3.1 Plastic deformation due to spherical void

As a first step, we solved a standard canonical problem to validate the accuracy of the near singular solver. We considered the two-dimensional problem of a large linear elastic plate with a circular hole subjected to uniaxial stress. The stress fields around the hole are well-defined and can be analytically computed using the Airy stress function approach [38]. We chose a two-dimensional domain of $\boldsymbol{x} \in [-16, 16] \times [16, 16]$ and introduced a circular hole of radius 1.0 at the center. We used a range of regularization length scales 0.01, 0.05, 0.1, 0.5, and 1.0. We subject the domain to a uniaxial stress condition by fixing the left edge and applying a displacement in the x direction on the right edge. Figure 2a (top left) shows the ϕ^{ϵ} field with a regularization length scale of 0.01 along with the refined grid. The corresponding stress distributions σ_{xx}, σ_{xy}, and σ_{yy} are shown in Fig. 2a in the top right, bottom left and bottom left respectively. We present the comparison of the numerical solution with the analytical solution at $y = 1$ line (tangent to the hole) as a function of the regularization scale in Fig. 2b–d. We note that the solution predicted by the near-singular solver converges to the analytical solution [38] as the regularization length scale decreases. The normalized L2 error (i.e. norm of the difference divided by the norm of the reference) indicates convergence with respect to ε (Fig. 2e). The increased error for the smallest 1–2 values of ε indicates an error from the discretization since a constant mesh resolution (that is, the same number of AMR levels) was used for each case for consistency.

Having validated the elastic solver, we demonstrate the effectiveness of the method by considering the plastic deformation of a cuboidal object with an embedded spherical void subject to uniaxial loading. We represented the material using the order parameter ϕ_{ϵ} which takes the value 0 within the void and 1 outside, with a length scale ϵ. The stress in the SBM equation (9) is expressed as $\sigma = \mathbb{C}\left(\text{grad}\,\boldsymbol{u} - \boldsymbol{\varepsilon}_p\right)$, where $\boldsymbol{\varepsilon}_p$ is the plastic strain. We used a staggered approach to solve the elastic equilibrium equation (9) and plastic evolution, and modeled the evolution of plastic strain $\boldsymbol{\varepsilon}_p$ using the J_2-plasticity model for a linear elastic isotropic material with Lamé constants λ and μ. We chose a J_2 plastic strength model with isotropic hardening. The yield strength for isotropic hardening is given by

$$K(\alpha) = \sigma_Y + \theta \bar{H}\alpha, \quad \theta \in [0, 1] \tag{17}$$

where σ_Y is the flow stress, α is the equivalent plastic strain, $\bar{H}$ is the hardening modulus and θ is a parameter governing the hardening slope. We used the internal variables $\boldsymbol{q} = \{\alpha, \boldsymbol{\varepsilon}_p, \boldsymbol{\beta}\}$ for the plasticity model, where β is the center of the von-Mises yield surface in the stress deviator space. Following are the yield condition flow rule and hardening rule for the J_2 plasticity model.

$$\begin{aligned} &\boldsymbol{\eta} := \text{dev}[\boldsymbol{\sigma}] - \boldsymbol{\beta}, \quad \text{tr}\,\boldsymbol{\beta} := 0, \quad f(\boldsymbol{\sigma}, \boldsymbol{q}) = ||\boldsymbol{\eta}|| - \sqrt{\frac{2}{3}} K(\alpha), \\ &\dot{\boldsymbol{\varepsilon}}_p = \gamma \frac{\boldsymbol{\eta}}{||\boldsymbol{\eta}||}, \quad \dot{\alpha} = \gamma \sqrt{\frac{2}{3}}, \quad \dot{\boldsymbol{\beta}} = \gamma \frac{2}{3}(1-\theta)\bar{H}\frac{\boldsymbol{\eta}}{||\boldsymbol{\eta}||} \end{aligned} \tag{18}$$

We solved the above equations using the following radial return algorithm described in detail in [39]. Given a stress and strain state at time t_n as $\boldsymbol{\sigma}_n$ and $\boldsymbol{\varepsilon}_n$, and the strain $\boldsymbol{\varepsilon}_{n+1}$ at time t_{n+1}, the algorithm involves following steps.

Algorithm 1 J2 plasticity model update

$\boldsymbol{e}_n \leftarrow \text{dev}\,\boldsymbol{\varepsilon}_n$, $\boldsymbol{e}_{n+1} \leftarrow \text{dev}\,\boldsymbol{\varepsilon}_{n+1}$, $\boldsymbol{s}_n \leftarrow \text{dev}\,\boldsymbol{\sigma}_n$, ▷ Compute deviatoric strain and stress

$\boldsymbol{s}^{\text{trial}}_{n+1} \leftarrow \boldsymbol{s}_n + 2\mu(\boldsymbol{e}_{n+1} - \boldsymbol{e}_n)$, $\boldsymbol{\eta}^{\text{trial}}_{n+1} \leftarrow \boldsymbol{s}^{\text{trial}}_{n+1} - \boldsymbol{\eta}_n$ ▷ Compute trial states

$\boldsymbol{n}_{n+1} \leftarrow \boldsymbol{\eta}^{\text{trial}}_{n+1} / |\boldsymbol{\eta}^{\text{trial}}_{n+1}|$ ▷ Compute new yield surface normal

Solve $-\sqrt{\frac{2}{3}}K\left(\alpha_n + \sqrt{\frac{2}{3}}\Delta\gamma\right) + ||\boldsymbol{\eta}^{trial}_{n+1}|| = 0$ for $\Delta\gamma$ ▷ Consistency condition

$\boldsymbol{\varepsilon}^p_{n+1} = \boldsymbol{\varepsilon}^p_n + \Delta\gamma\boldsymbol{n}_{n+1}$

$\alpha_{n+1} = \alpha_n + \sqrt{\frac{2}{3}}\Delta\gamma$

$\boldsymbol{\beta}^p_{n+1} = \boldsymbol{\beta}^p_n + \sqrt{\frac{2}{3}}\theta\,\bar{H}(\alpha_{n+1} - \alpha_n)\boldsymbol{n}_{n+1}$ ▷ Update internal variables

A 3D domain $\boldsymbol{x} = (x_1, x_2, x_3) \in [-16, 16] \times [-16, 16] \times [-16, 16]$ (arbitrary units) with an ellipsoidal void centered at the origin and radii $\boldsymbol{r} = (r_x, r_y, r_z)$ was used. The order parameter ϕ was set to 1 outside the inclusion and 0 inside with a length scale $\epsilon = 0.4$. We chose the material parameters as Young's modulus $E = 210\,\text{GPa}$, Poisson's ratio $\nu = 0.3$, yield strength $\sigma_Y = 200\,\text{MPa}$, and hardening parameters $\bar{H} = 50\,\text{GPa}$ and $\theta = 1$. We performed a tension test with a fixed $x_1 = -16$ face and a cyclic displacement applied in the x_1 direction on the $x_1 = 16$ face. We chose the applied displacements in the increments of 0.004 going from total applied displacement from 0.0 to 0.1, then from 0.1 to -0.1, and finally from -0.1 to 0.0.

Figure 3 shows the stress–strain curves obtained for six different ellipsoid shapes and sizes. These are $\boldsymbol{r}_1 = (2.0, 0.5, 0.5)$, $\boldsymbol{r}_2 = (2.0, 7.07, 7.07)$, $\boldsymbol{r}_3 = (7.5, 4.08, 4.08)$, $\boldsymbol{r}_4 = (10, 10, 10)$, $\boldsymbol{r}_5 = (2.5, 14.14, 14.14)$, and $\boldsymbol{r}_6 = (14.14, 8.16, 8.16)$. We calculated the strains using the applied displacement and stresses from the total traction on the $x = 16$ face where the displacement is applied. As expected, the stress–strain curve for each case exhibits the classic hysteresis loop. As the size and aspect ratio of the void change, the plastic evolution within the domain changes leading to different stress–strain curves. The total plastic strain is higher for larger void with higher aspect ratios aligned with the loading directions, making the stress–strain curve flatter. Figure 4 shows the magnitude of plastic strain deviator at the applied displacement of 0.08 during the unloading cycle.

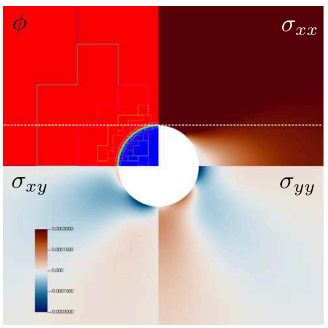
We performed these simulations on the UCCS INCLINE cluster using 128 cores on a single node. We chose a base mesh of $32 \times 32 \times 32$ with 5 levels of refinement. For the six cases presented, the solver took 4 minutes to 8 hours, depending on the size of the inclusion and the size of the portion of the domain refined with high resolution. The solver converged linearly for all cases despite large regions of voids within the domain. Therefore the solver performed well in predicting the stress fields and plastic strains due to voids.

3.2 Brittle fracture

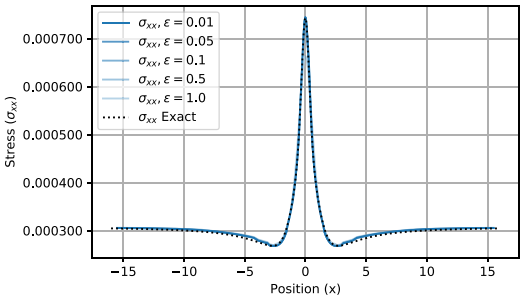
Fracture is one of the most prominent causes of failure for engineering structures. As such computational modeling of crack nucleation and propagation in engineering materials is critical for evaluating their performance. Computational methods for modeling fracture can be broadly classified as either discrete boundary or diffused boundary methods. Among the discrete boundary methods, there are two main approaches: the eXtended Finite Element Method (XFEM) [40–42] and the Scaled Boundary Finite Element Method (SBFEM) [43–46]. XFEM involves enriching classical finite elements with specialty elements designed specifically for capturing singularities at crack tips. On the other hand, SBFEM uses a dimensional reduction technique to reduce the problem domain to the boundary of the solid and scales the solution to the crack tip analytically. A detailed review of discrete methods can be found here [47, 48]. While these methods have been widely successful, they suffer from limitations when explicitly tracking crack fronts for complex crack patterns.

Diffuse boundary methods, or "phase field methods" use a smoothly varying scalar damage field $c(\boldsymbol{x}, t)$ to diffuse the sharp crack over a length scale ξ [49]. The differential equation governing the evolution of $c(\boldsymbol{x}, t)$ is based on a rigorous variational approach to fracture which uses an energy functional regularized over the length scale ξ [50]. The variational method has been shown to be consistent with linear elastic fracture mechanics under quasi-static loading [50] for brittle materials. Recently, phase field fracture methods have been extended to study heterogeneous materials [6, 51, 52], anisotropic materials [53, 54], functionally graded materials [55–57], dynamic fracture [58–60], ductile fracture [61–63], and fatigue loading [64]. Phase field fracture has also been used to study interfacial strength in composites by incorporating cohesive zone elements into the formulation [65–67]. Modifications to the traditional formulation have also been studied with different damage degradation functions and damage energy penalty terms [68, 69].

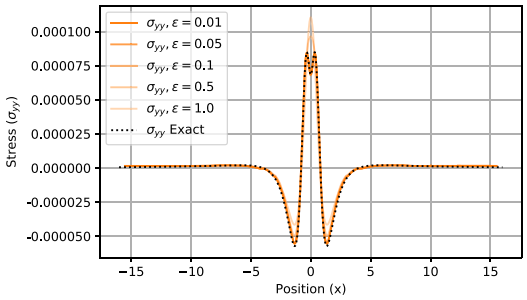
While phase field methods have gained wide adoption, they suffer from high computational costs due to typically small values of ξ. One way to circumvent this challenge is to use spectral methods, typically the Fast Fourier Transform (FFT), for memory efficiency [70, 71]. Another more widely used approach is to couple the phase field fracture implementation to AMR with higher resolutions near the



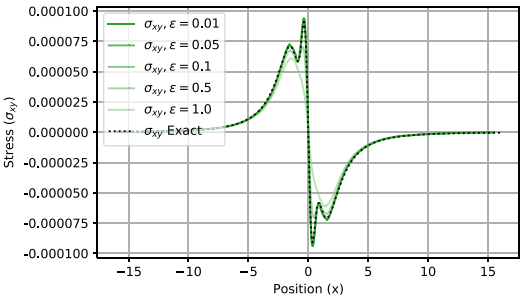
(a) Plot of ϕ (top left) and stress fields σ_{xx} (top right), σ_{yy} (bottom right), σ_{xy} (bottom left) for uniaxial tension in the x direction.



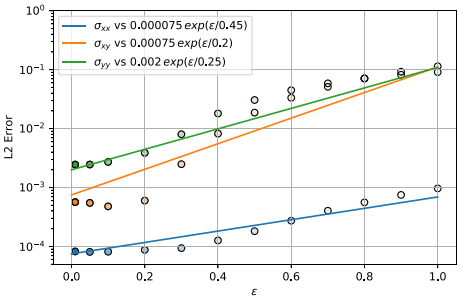
(b) σ_{xx}



(c) σ_{yy}



(d) σ_{xy}



(e) L2 error (as compared to the exact solution) with respect to diffuse boundary thickness ε.

Fig. 2 Verification of the method by comparison to the exact solution for a plate with a hole under uniaxial tension. Convergence with decreasing ε (corresponding to increasing decreasing line opacity), compared to the exact solution (dashed lines), along $y = R = 1$, which is the tangent to the hole (dashed white line)

crack tip. Among the AMR approaches, the discontinuous Galerkin approach has been used for single-level AMR [72] and the finite cell method together with h and p refinement has been used to achieve multiple levels of refinement on a regular grid [73]. Other attempts include hybridizing phase field method with XFEM [74].

In this work, we implemented a hybrid model of phase field brittle fracture to study crack propagation in Mode-I loading. We use a regularized field $c(\boldsymbol{x}, t)$ with values 1 outside the crack and 0 inside the crack and length scale parameter ξ. The phase field fracture energy functional is given by,

$$\mathcal{L} = \int_{\Omega} (g(c) + \eta)\, W_0(\boldsymbol{\varepsilon}(\boldsymbol{u}))dV + \int_{\Omega} G_c \left[\frac{w(c)}{4\xi} + \xi|\nabla c|^2\right] dV, \tag{19}$$

where W_0 is the elastic strain energy of the material without a crack field, G_c is the fracture energy, and $\eta = 10^{-4}$ chosen for computational stability. The interpolation function $g(c)$ takes the value 0 inside the crack and 1 outside. The interpolation function $w(c)$ takes the value 1 inside the crack and 0 outside. In this work, we choose a quartic interpolation function [69] with $g(c) = 4c^3 - 3c^4$ and $w(c) = 1 - g(c)$. The crack and displacement fields are evolved using the variational derivative of $\mathcal{L}$.

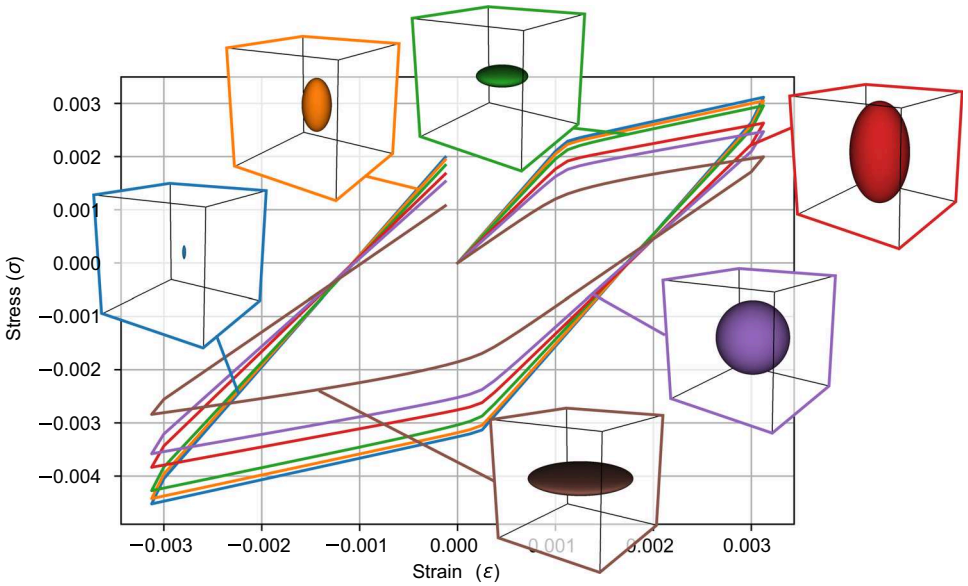


Fig. 3 Stress–strain hysteresis results for a variety of void shapes and sizes

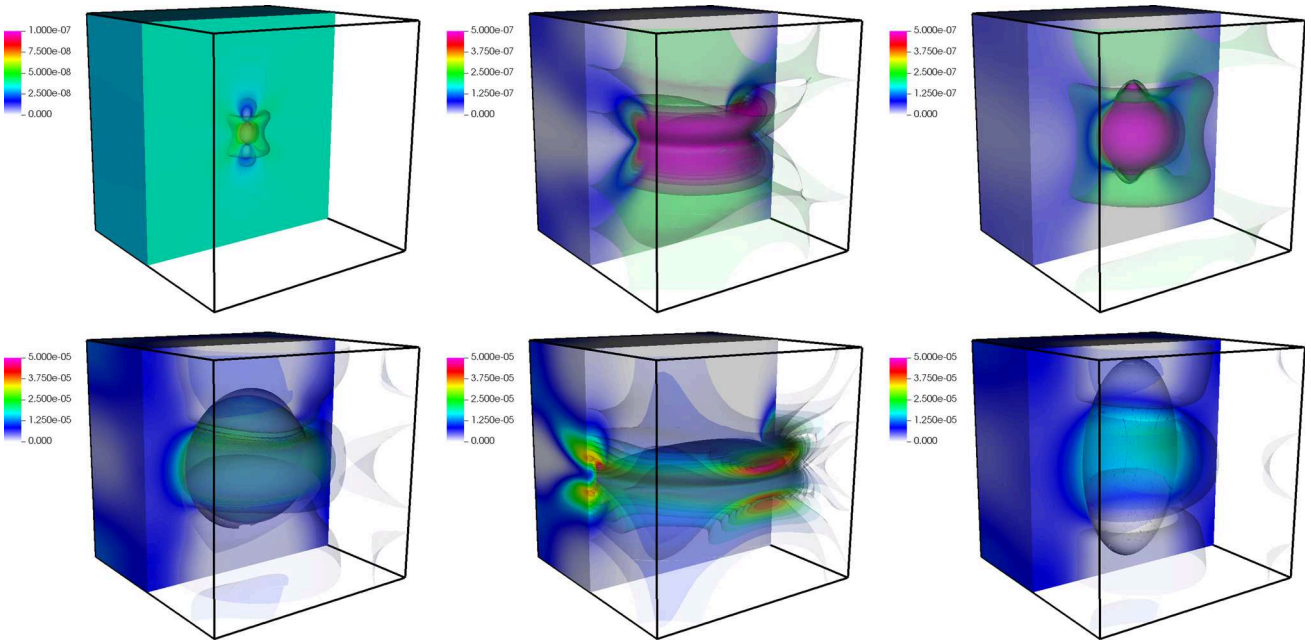


Fig. 4 Isocontours showing the magnitude of the plastic strain deviator at $t = 0.3$. From top right to bottom left, the void radii are $\boldsymbol{r}_1 = (2.0, 0.5, 0.5)$, $\boldsymbol{r}_2 = (2.0, 7.07, 7.07)$, $\boldsymbol{r}_3 = (7.5, 4.08, 4.08)$, $\boldsymbol{r}_4 = (10, 10, 10)$, $\boldsymbol{r}_5 = (2.5, 14.14, 14.14)$, and $\boldsymbol{r}_6 = (14.14, 8.16, 8.16)$

We chose a linear elastic isotropic material with Lamé constants λ and μ with the strain energy density and stress as

$$W_0 = \frac{1}{2}\lambda(\operatorname{tr}\boldsymbol{\varepsilon})^2 + \mu \operatorname{tr}\left(\boldsymbol{\varepsilon}^2\right), \quad \boldsymbol{\sigma} = \frac{\partial W_0}{\partial \boldsymbol{\varepsilon}} \tag{20}$$

To account for the tension-compression asymmetry, we assume an additive decomposition of the strain energy $W_0 = W_0^+ + W_0^-$ which uses the spectral decomposition of the strain tensor $\boldsymbol{\varepsilon} = \sum_{i=1}^{d} \varepsilon_i \hat{\boldsymbol{v}}_i \otimes \hat{\boldsymbol{v}}_i$. The strain energies are given by

$$W_0^{\pm} = \frac{1}{2}\lambda(\operatorname{tr}\boldsymbol{\varepsilon}_{\pm})^2 + \mu \operatorname{tr}\left(\boldsymbol{\varepsilon}_{\pm}^2\right), \quad \boldsymbol{\varepsilon}_{\pm} = \sum_{i=1}^{d} (\varepsilon_i)_{\pm}\, \hat{\boldsymbol{v}}_i \otimes \hat{\boldsymbol{v}}_i, \tag{21}$$

and the energy functional is updated to

$$\mathcal{L} = \int_{\Omega} \left[(g(c) + \eta)\, W_0^+ + W_0^- \right] dV + \int_{\Omega} G_c \left[\frac{w(c)}{4\xi} + \xi |\nabla c|^2 \right] dV. \tag{22}$$

We note that the above strain-based decomposition of energy does not always ensure that the compressive states do not contribute to crack growth [75, 76] and a stress based approach has been used recently to circumvent this issue [77]. The equations for the hybrid formulation of phase-field fracture are obtained by taking the variational derivative of the energy functional above.

$$\begin{aligned}
&\operatorname{div}\boldsymbol{\sigma} = 0, \quad \boldsymbol{\sigma} = (g(c) + \eta)\,\frac{\partial W_0}{\partial \boldsymbol{\varepsilon}},\\
&0 = g'(c)\mathcal{H}^+ - G_c\left[2\xi\,\Delta c + \frac{w'(c)}{2\xi}\right],\\
&\quad \text{where } \mathcal{H}^+ := \max_{\tau\in[0,t]} W_0^+(\boldsymbol{\varepsilon}(x,t))\\
&\forall \boldsymbol{x} : W_0^+ < W_0^- \;\Rightarrow c(\boldsymbol{x}) = 1
\end{aligned} \tag{23}$$

We replace Eq. (23b) with a Ginzburg–Landau type evolution law by introducing crack mobility M as

$$\dot{c} = -M\left[g'(c)\mathcal{H}^+ - G_c\left(2\xi\,\Delta c + \frac{w'(c)}{2\xi}\right)\right]. \tag{24}$$

Using Eqs. (23a), (23c), and (24), we solve a classic Mode-I fracture propagation problem using a staggered scheme within Alamo. The order parameter ϕ_ϵ from the SBM equation (9) corresponds to $g(c)$ in the phase fracture equations. At each time step, as the crack field c evolves, we update the order parameter ϕ_ϵ for the next iteration of the staggered solver.

Figure 5 (right) shows snapshots of mode-I crack propagation over a domain of $\boldsymbol{x} \in [-0.01, 0.01] \times [-0.01, 0.01]$. We chose a material with $\lambda = 121.15$ GPa, $\mu = 80.77$ GPa, $\xi = 1.0 \times 10^{-5}$, $G_c = 2700$ Pa, and $M = 1.0 \times 10^{-5}$. We initialized a notch of length 1.5×10^{-4} at the center of the left edge. We then fixed the bottom boundary, and apply a fixed y displacement of 1.5×10^{-5} on the top edge. We used six levels of refinement on a base grid of 64×64 to appropriately capture the interface. The refinement criteria was based on the gradient of the crack field as $|\nabla c|\nabla x| > 0.01$. We performed mesh-regridding every 10 time-steps, with a single time-step being $\Delta t = 10^{-4}$. As expected, we obtain a steadily propagating crack in the x direction with an adaptively refining grid following the crack field.

We ran this simulation on Auburn University's Easley computing cluster using 32 cores on a single node which took a total of 4.5 h. We note that a major portion of the simulation time was used to perform Ginzberg Landau evolution of the crack field (Eq. 24). Figure 5 (left) shows the number of multigrid iterations needed by the near-singular solver to solve Eq. (23a). The solver required 108 iterations for the first elastic solve. Since the solver uses the previous solution as a starting point, the number of iterations sharply declined immediately after the first elastic solve. We note a steady increase in required iterations as the crack progresses followed by a peak and slow decline. We attribute this trend to the changing nature of the mesh as the crack propagates and the fraction of the near-singular domain. Overall the solver never took more than 160 iterations throughout the entire simulation.

We further illustrate the performance of the near-singular solver by studying crack nucleation and propagation due to stress concentration in an L-shaped domain. We initialized the domain $\Omega := \boldsymbol{x} \in [-0.01, 0.01] \times [-0.01, 0.01]$ using a smooth differentiable function with length scale 4×10^{-5} that takes value 0 in $\Omega_1 := \boldsymbol{x} \in [0, 0.01] \times [0, 0.01]$ and 1 in $\Omega \backslash \Omega_1$. We fixed the bottom edge and applied a constant displacement of 1.5×10^{-5} on the top edge in the y direction. Figure 6 (right) shows the propagation of crack along with Von-Mises stress distribution in the domain. As expected, the crack nucleated at the corner $\boldsymbol{x} = (0, 0)$ with the highest stress concentration and propagated upwards towards the top free surface. This is confirmed by the Fig. 6 (left) where we plot the force (in non-dimensional units) on the top edge. We observe a linear decline in the force as the crack propagates, indicating the weakening of the material. Eventually a secondary crack nucleates at the top left corner, which coalesces with the primary crack causing the final failure of the material. This is indicated by a sharp decline in measured traction and the snapshot of crack at $t = 3.2$.

We performed this simulation on the UCCS INCLINE high-performance computing cluster using 128 cores on a single node. The simulation took a total of 6.5 h most of it, once again, was the Ginzberg Landau evolution of the crack field. We observe an increase in solver iterations after an initial decline. The maximum number of iterations required was 500, while the smallest was 48. Once again, we attribute this pattern to the evolving crack field and near-complete failure of the material. Overall, we observed results as expected and the solver performed well even near complete failure.

3.3 Structural topology optimization

Topology optimization refers to the computational method of determining the geometry of a material or set of materials that produce the optimal result subject to constraints. Topology optimization generally implies the minimization over a very high dimensional space, the space of all admissible geometries. Topology optimization has been applied to myriad fields of study, ranging from battery design [78] to fluid–structure interaction [79]. Structural topology optimization refers specifically to the problem of designing load-bearing structures, subject to constraints typically on the amount of material allowed in a certain volume, that minimizes compliance and maximizes the stiffness of the structure. Topology optimization has existed as a popular field of study for more than thirty years, stemming from ideas

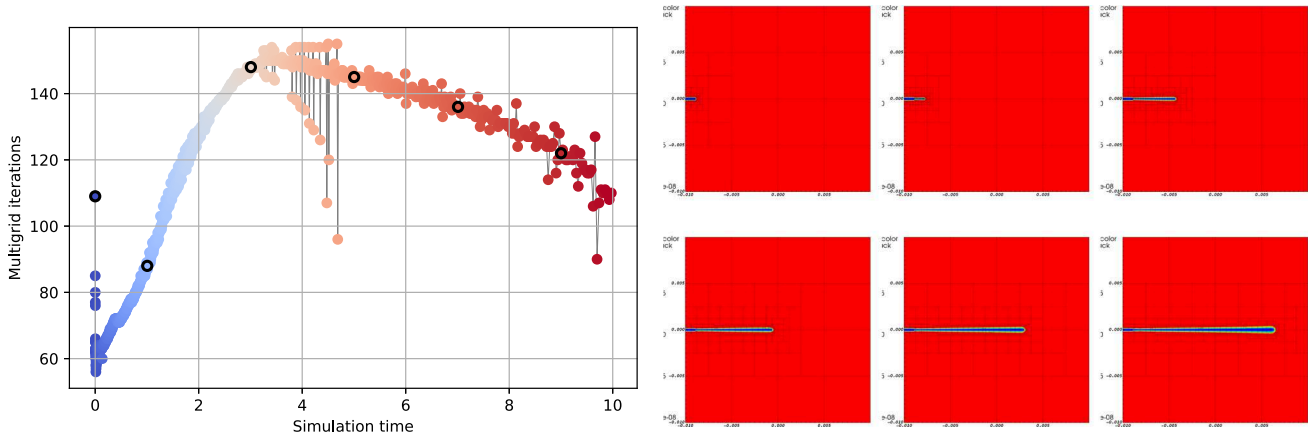


Fig. 5 Phase field fracture—Canonical Mode I loading. (Left) Performance of the MLMG solver during Mode I crack showing the number of MLMG iterations required during simulation time (indicated by position on x axis and by color). (Right) Snapshots showing the phase field evolution at indicated times ($t = 0, 1, 3, 5, 7, 9$ indicated by black circles) on the performance plot

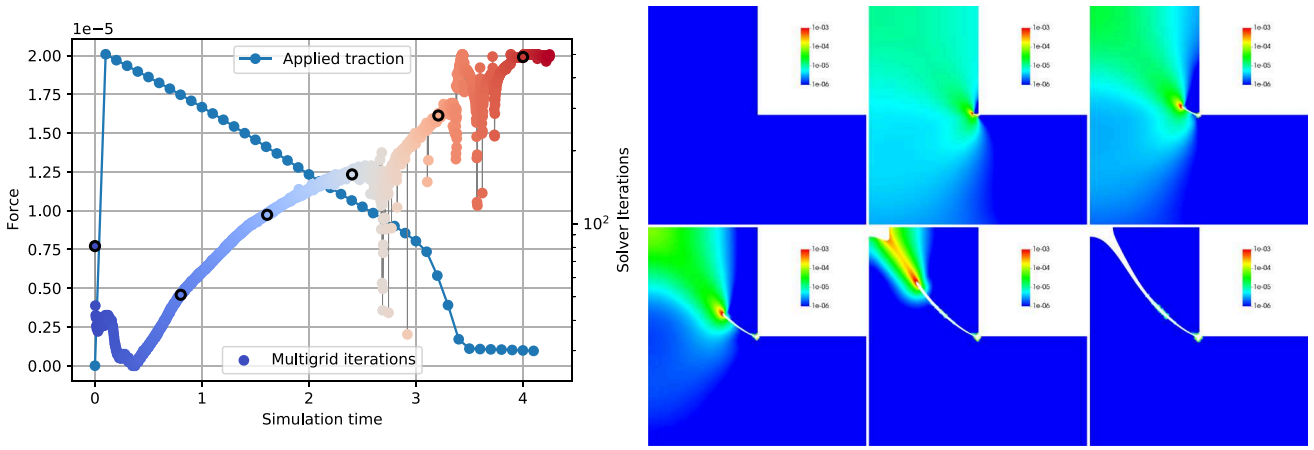


Fig. 6 Phase field fracture—Crack nucleated by stress concentration. (Left) Performance of the MLMG solver along with the measured applied traction due to the imposed displacement as a function of simulation time (indicated by position on x axis and by color in the iteration plot). (Right) Snapshots of the stress state in time ($t = 0, 0.8, 1.6, 2.4, 3.2, 4.0$) as the crack propagates. The region where $\psi < 0.1$ is colored white

originally proposed more than 150 years ago [80]; today, topology optimization is an entire sub-discipline in its own right. Topology optimization methods have even found their way into some commercial codes and consequently experienced accelerating usage, partly due to the recent interest in additive manufacturing.

There are a number of prevailing methods for solving structural topology optimization problems. Common to all structural topology optimization methods is (i) the need to solve the stress equilibrium problem, and (ii) the ability to resolve arbitrary geometry, without a priori knowledge, with resolution sufficient to resolve lengthscales of interest, and without excessive computational cost. Following the seminal work by Bendsoe [81, 82], other methods have included shape derivatives [83], the level set method [84–86], and evolutionary methods [87] as techniques for solving the optimization problem. Recently surging interest in machine learning has led to artificial intelligence-based newcomers, such as generative adversarial networks, that are not necessarily based in physics but are nonetheless capable of generating optimal or near-optimal structures very quickly [88–90]. A full review of structural topological optimization is well outside this work's scope, so we refer the reader to [91, 92] for a more comprehensive overview.

The phase field method is yet another option for solving the structural topology optimization problem. It is, in some sense, a natural choice, as the phase field method is used specifically for problems involving variable topology. Phase field was applied to topology optimization by [93], where the free energy functional contains the elastic strain

energy for the given configuration. Volumetric constraints can be applied by using a conservative Cahn–Hilliard equation that preserves volume [94], or by a constrained gradient descent method [95, 96]. Jeong et al. [97] implemented volume constraints, as well as additional design constraints, using augmented Lagrange multipliers. A limitation of current phase field methods is the need for high resolution across the diffuse boundary to prevent mesh dependence. Here, adaptive mesh refinement is needed; while AMR has been applied to phase field topology optimization [98, 99], the application has been limited.

The finite element method is used nearly universally in topology optimization. However, the smoothed boundary method for solving near-singular problems presented in this work is ideally suited for solving phase field topology optimization problems. The method's ability to rapidly regrid, and efficiently solve the near-singular mechanical equilibrium equation, ideally suit it for this application. In this section, we present results for a basic phase field topology optimization problem.

We use η as the order parameter to represent the topology over a domain Ω. Next, we define the free energy in terms of η (where square brackets implicitly indicate a functional over the value and its derivatives) to be:

$$\begin{aligned} W[\eta] = &\int_\Omega \left(\alpha\eta^2(1-\eta)^2 + \frac{\beta}{2}|\nabla\eta|^2\right) dx \\ &+ \inf_{\boldsymbol{u}} \left[\frac{1}{2}\int_\Omega \nabla\boldsymbol{u} \cdot ((\eta+\zeta)^2\mathbb{C})\,\nabla\boldsymbol{u}\, dx \right. \\ &\left. - \int_{\partial_2\Omega} \boldsymbol{u}\cdot\boldsymbol{t}^0\, dx\right], \end{aligned} \tag{25}$$

subject to the constraint

$$\int_\Omega \eta\, dx = V_0 \tag{26}$$

where α and β are numerical parameters controlling segregation and boundary energy, $\mathbb{C}$ is the fourth order elasticity tensor, $\boldsymbol{t}_0$ is prescribed surface traction, and V_0 is the allowable volume of material. (We note that $\alpha \sim \frac{1}{\epsilon}$, $\beta \sim \epsilon$ where ϵ controls the boundary width. We retained α and β for simplicity.) As discussed above, the volume requirement induces a constraint on the optimization problem. We adopt a straightforward regularization, allowing the optimum to be found by a modified version of the Allen-Cahn equation:

$$\frac{\partial\eta}{\partial t} = -L\left(\frac{\delta}{\delta\eta}W + \lambda(t)\left(\int_\Omega \eta\, dx - V_0\right)\right) \tag{27}$$

where L is a mobility parameter and $\partial/\partial\eta$ is the variational derivative. The function $\lambda(t)$ is a Lagrange multiplier that enforces the constraint in Eq. (26). In general, we let $\lambda(t)$ tend towards infinity as $t \to \infty$, to allow the constraint to be approached at a gradual rate. In problems with multiple optima, the form for $\lambda(t)$ may determine which local minimum is found. Each evaluation of Eq. (27) requires the evaluation of the elastic minimization problem in Eq. (25), which is solved using the proposed method. As in all examples, η is a cell-centered field while the displacement is node-centered. Node-to-cell averages are computed for each iteration.

We considered the classical problem of a load supported by a cantilevered structure. We selected dimensionless values for all quantities. For the two-dimensional results, we chose the domain to be 1 in height, and ranging from 1 to 4 in length. We used base-level grids of 32×32, 64×32, and 128×32 corresponding to the different aspect ratios. We used the powers of two for all grid dimensions, at the cost of some non-square unit cells, to optimize the multigrid solver's performance. We used a total of three AMR levels, with a refinement criterion $|\nabla\eta||\Delta\boldsymbol{x}| \geq 0.05$. We used a less restrictive refinement criterion for the strain, ε, although it did not contribute significantly to the results. We performed mesh regridding at each phase field iteration step.

For the cases considered here, we constrained the volume to 25% of the domain volume. We chose the phase field parameters as $\alpha = 200$, $\beta = 0.01$, $\lambda = 400.0$. We used an isotropic material model with $E = 1480$, and $\nu = 0.22$. We applied a point load of magnitude -0.1 per unit length at the center of the right face over a region 0.01 in height. We specified Neumann conditions for the order parameter, and it is possible to see slight artifacts at the edges of the domain that result from this condition. We chose the regularization parameter ζ to be 0.01. While we considered smaller values, they did not affect the performance of the solver. However, they did affect the nucleation behavior of the phase field method, causing spurious material segments to be generated and eventually resulting in some instability that caused the solution to diverge. Therefore, we attribute this to limitations of the phase field model and leave further optimization of the model to future work.

The algorithm's results were generally as expected for models of this type (Fig. 7). For the 1×1 case, the result is a fairly simple triangular brace structure. Varying the volume fraction and boundary width terms generally did not produce a substantial change. Increasing the domain to 1.5×1 produced a truss-like structure that is generally in line with the canonical result for this standard problem. Increasing again to 2×1 produced an irregular, asymmetric structure, likely due to the inaccessibility of an optimal symmetric structure for that aspect ratio. Asymmetries were common in this work, especially for cases with higher boundary penalization terms. Since we did not initialize the problem asymmetric perturbations, any deviation generally stemmed from perturbations induced by the regridding algorithm. For the 2.5×1 case,

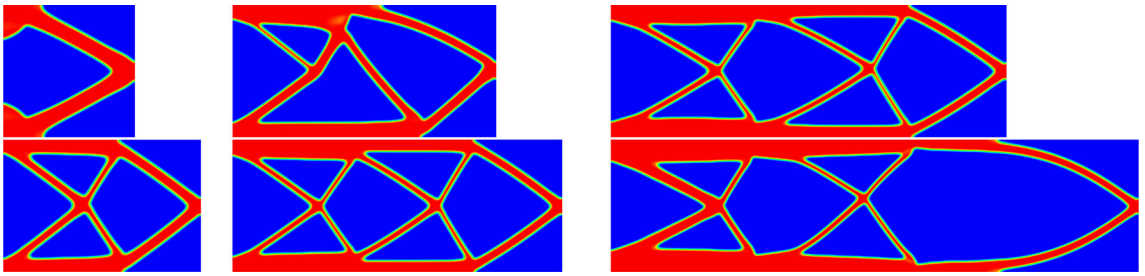


Fig. 7 Topology optimization results for point load supported by a cantilever, wtih 25% fill and increasing aspect ratio

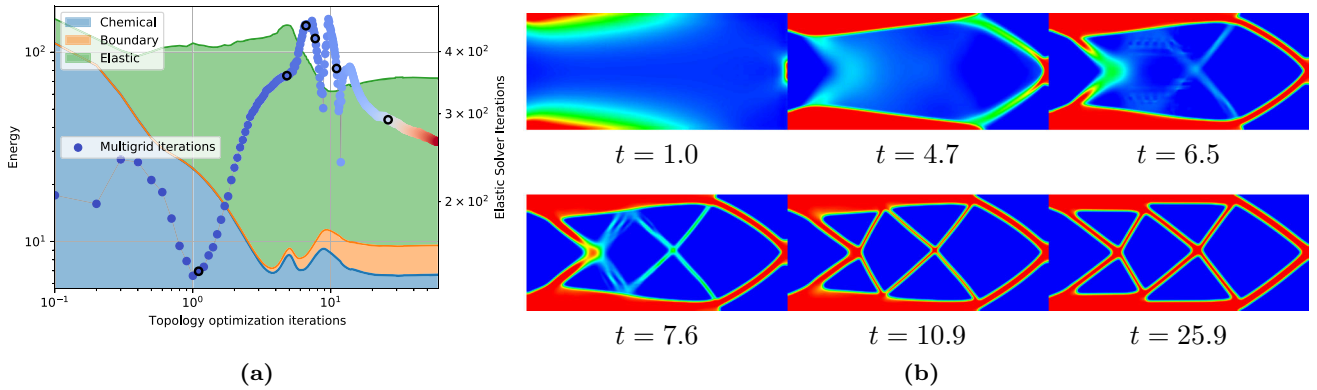


Fig. 8 Progression of the energy and the corresponding performance of the elastic solver during topology optimization. **a** Contributions of chemical potential, boundary energy, and elastic energy to the objective function. Number of solver iterations required are indicated by the position along the x axis and by color. **b** Snapshots corresponding to the black dots in **a**

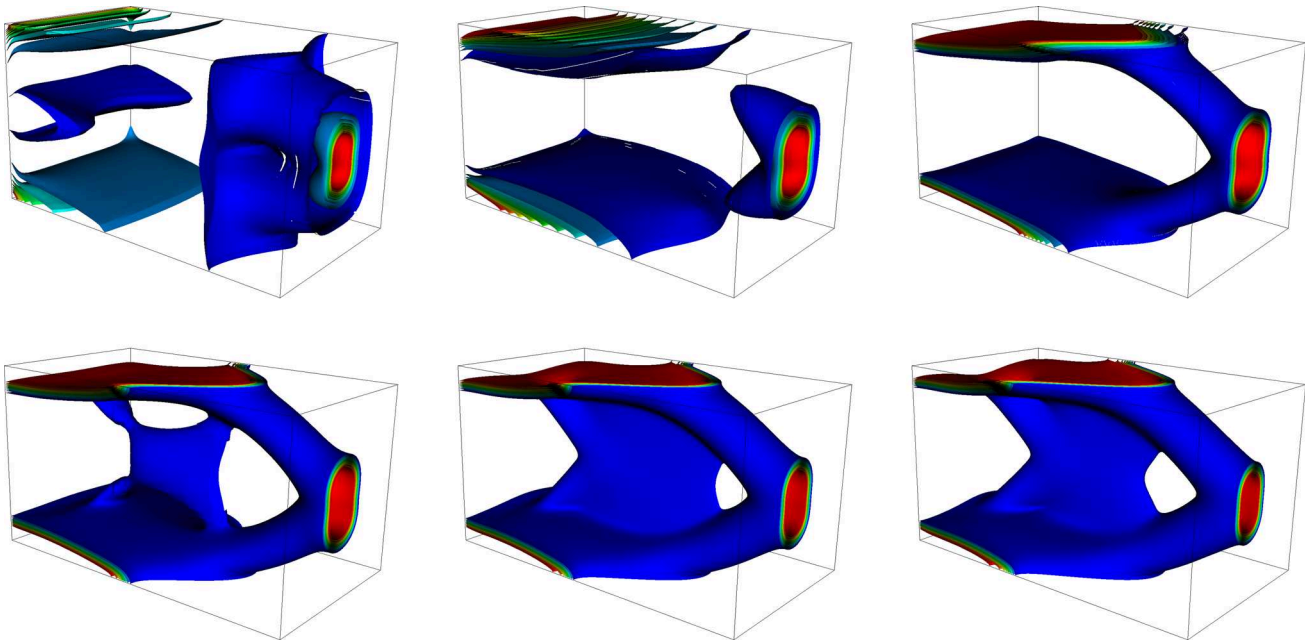
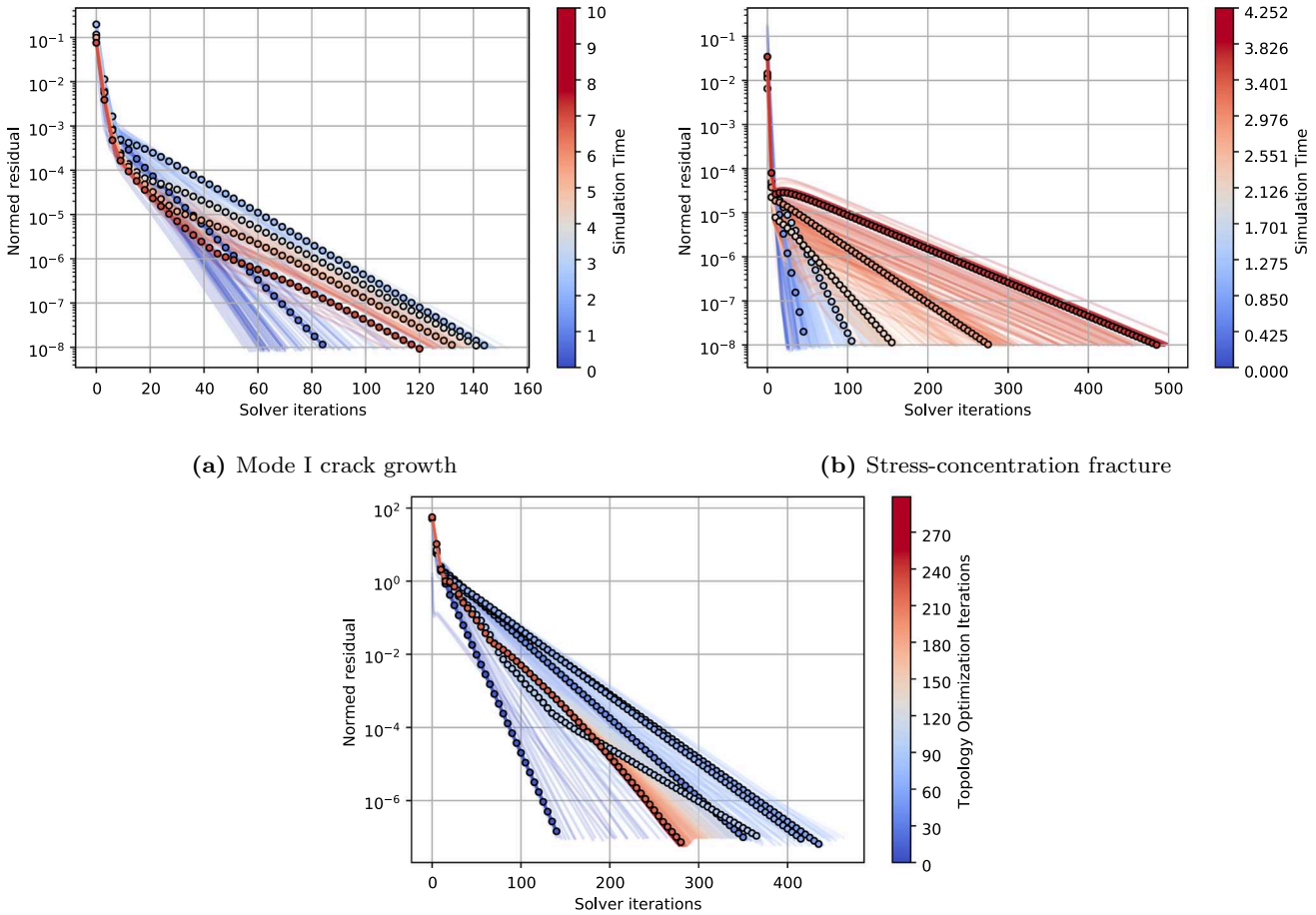


Fig. 9 Three dimensional topology optimization results for a cantilever load. Contours correspond to increments of $\Delta\eta = 0.1$, snapshots are shown every 10 timesteps. The full simulation took 4 h to run on 128 processors; the approximate result was reached after about 1 h of computing time. Two levels of refinement are used for a total of 3 BSAMR levels



(a) Mode I crack growth

(b) Stress-concentration fracture

(c) Topology optimization

Fig. 10 Convergence of the linear elastic solver for mode I fracture, stress-concentration fracture, and topology optimization. In each plot, each line represents the residual versus iteration number for a single elastic solve. The colors of the lines correspond to the time at which the solve occurred in the larger simulation, as well as the color in the corresponding convergence plots. The highlighted lines (with markers) correspond to the specific simulation points as highlighted in Figs. 5, 6, and 8

the resulting structure is similar to the 1.5×1 case except with a secondary truss structure in addition to the first. The top and bottom support beams also decrease in width so as to prioritize support near the wall where the bending moment is the highest. As the domain continues to increase in length, the double support structure is elongated but does not generate the third structure. This is due to the width of the diffuse interface, as the thickness of the third support structure would be thinner than the diffuseness of the boundary. Therefore, it is not possible to nucleate the third structure. We emphasize that we did not account for buckling in this model. The slenderness of the beams would, in reality, result in buckling that renders the structure unstable. While it is possible to modify the model to avoid this effect by differentiating between tension and compression, it is outside the scope of the present work.

Of particular interest is the performance of the solver during the solution of the phase field topology optimization problem. We present the history of the design evolution for a representative structure in Fig. 8a. The colored regions represent contributions of the energy functional corresponding to the chemical potential, boundary energy, and elastic energy. (We note that the regions are stacked and are plotted on a log scale in the x and y axes.) The initial iterations are dominated by the chemical potential, which drives the segregation and, as a result, a sharp increase in the elastic energy. Eventually, support structures are spontaneously nucleated, rapidly relieving the elastic stress and moving the solution toward its equilibrium state. We superimpose the performance of the multigrid solver (gray points) and plot it with respect to the right axes, also on a log scale. In general, the solver always converged linearly within a couple hundred iterations and generally no more than 400. (See 1 for more details on

solver convergence.) We notice a decrease in the solver's performance when topological changes occurred, which can be connected to the presence of problematic "islands" and "peninsulas" in the solution. This can be seen by observing the design evolution (Fig. 8b) at the indicated points (black circles) on the performance curve.

All of the simulations took less than an hour to complete on a single node with 32 cores. We used the UCCS INCLINE cluster to perform the simulations, but only due to the large number of simulations that were considered. We note that all the 2D results can be quickly reproduced on a desktop computer in a matter of minutes to hours (depending on domain size and computing power). We emphasize that the objective of this work is not to compete with commercially available topology optimization codes, but to demonstrate the verstility of the method and its ability to easily adapt to a diverse range of problems..

Finally, we tested the method in three dimensions (Fig. 9). We chose a configuration that was generally similar to the 2D case except for the following differences. We chose the domain as $1.8 \times 1.0 \times 1.0$, with a base grid of $64 \times 32 \times 32$. We used a larger value of $\beta = 0.1$. We applied the same point load at the right end, except that it was applied at the center over an area of 0.1×0.1. The figure shows the evolution of η during the solution, with isosurfaces plotted at increments of $\Delta\eta = 0.1$. We ran the simulation on a single node (128 processors) of the INCLINE cluster for four hours, although we observed that the result converged well before the conclusion of the run time (about an hour). As with the two-dimensional case, the result is a truss-like structure with a central support mechanism. The result differs from the 2D case in that the central truss is replaced with a webbed structure. (Once again, we note that our model did not account for buckling, which would significantly change the final result.) In general, we observed results as expected and satisfactory performance from the model in 3D.

4 Conclusion

In this work, we presented a comprehensive computational approach for solving near-singular problems with the smoothed boundary method and block-structured adaptive mesh refinement. Problems in solid mechanics are nearly universally solved using weak form methods (finite elements), and this work aims to formalize and elucidate the theory and best practices associated with solving problems of interest (specifically, near-singular problems) using the near-singular, smoothed boundary, strong-form approach. In particular, a key insight of this work is the essential role that is played by placing certain quantities at node or cell centers in finite difference elasticity. This has been well-known in fluid mechanics and has generally been enforced implicitly in solid mechanics due to the prevalence of strong-form methods. This, combined with the model vector space construction and the reflux-free method for elasticity on BSAMR grids, results in a powerful technique for quickly solving many problems of interest in solid mechanics. We demonstrated this approach's versatility and the solver's performance by studying three problems: plasticity, fracture, and topology optimization.

We conclude by discussing the limitations of this work. As with all diffuse boundary methods, a higher resolution is needed than most discrete boundary approaches. BSAMR is able to alleviate this computational cost significantly, but the number of points will still scale with the amount of surface area, even for surfaces with low stress. (Discrete boundary methods, by contrast, can afford low resolution at uninteresting boundaries.) While this may be unavoidable, we believe that the proposed approach offers increased versatility and ease of implementation. We also note that while the solver performs well on near-singular problems, it can still struggle on problems that exhibit extreme irregularity or high stress concentrations. In this regard, it is comparable to equivalent solvers for alternative approaches. Finally, we emphasize that the examples considered here do not reflect the state-of-the-art in the fields of plasticity, fracture, and topology optimization; rather, we have used tested and well-understood models in order to showcase the performance of the elastic solver. Implementing more sophisticated methods using this solver shall be left to future work.

Acknowledgements VA acknowledges the Auburn University Easley Cluster for support of this work. BR acknowledges support from Lawrence Berkeley National Laboratory, subcontract #7645776, and from the Office of Naval Research, Grant #N00014-21-1-2113. This work used the INCLINE cluster at the University of Colorado Colorado Springs. INCLINE is supported by the National Science Foundation, Grant #2017917. The authors wish to thank Dr. Scott Runnels at the University of Colorado Boulder, whose insights on cell and node-based fields led to key breakthroughs in solver development.

Appendix A: Convergence data

The linear elastic, strong-form near-singular multigrid solver exhibited nearly universally linear convergence in the example problems presented in this work. Here, we present a more detailed exposition of the solver behavior in time for the fracture cases and the topology optimization example (Fig. 10). We note that in all cases, the results from the initial solve are not included because it is for the initial, non-regularized version of the problem. In all examples, we observe a rapid convergence during the first 3–4 iterations. This rapid convergence lasts until the error is reduced to 10^{-3}, 10^{-4}, and 10^{-1} for mode I, stress-concentration, and topology optimization, respectively. A sharp reduction follows this in the conver-

gence rate, which is almost always non-decreasing during the remaining solve.

In mode-I fracture and topology optimization, interestingly, the worst convergence occurs during the middle of the simulation; subsequently, convergence improves and approaches a constant rate. Both exhibit a couple of solves in which the convergence rate turned sharply from a higher to a lower value; in both cases, the convergence might be described as "piecewise linear."

In the stress-concentration fracture case, linear convergence is constantly observed, but the convergence rate decreases steadily as the simulation progresses. We attribute this to the increasing irregularity of the problem, resulting from large chunks of material that have been degraded, and regions of the boundary that are under-regularized. In other words, it appears to be the fracture model, not the solver, that is responsible for the degrading convergence.

References

1. Runnels B, Agrawal V, Zhang W, Almgren A (2021) Massively parallel finite difference elasticity using block-structured adaptive mesh refinement with a geometric multigrid solver. J Comput Phys 427:110065
2. Celestine A-DN, Agrawal V, Runnels B (2020) Experimental and numerical investigation into mechanical degradation of polymers. Compos Part B Eng 201:108369
3. Runnels B, Agrawal V (2020) Phase field disconnections: a continuum method for disconnection-mediated grain boundary motion. Scr Mater 186:6–10
4. Gokuli M, Runnels B (2021) Multiphase field modeling of grain boundary migration mediated by emergent disconnections. Acta Mater 217:117149
5. Strutton JW, Moser NH, Garboczi EJ, Jennings AR, Runnels B, McCollum JM (2022) Interface history on strain field evolution in epoxy resins. ACS Appl Polym Mater 4:1535–1542
6. Agrawal V, Runnels B (2021) Block structured adaptive mesh refinement and strong form elasticity approach to phase field fracture with applications to delamination, crack branching and crack deflection. Comput Methods Appl Mech Eng 385:114011
7. Chadwick AF, Stewart JA, Enrique RA, Du S, Thornton K (2018) Numerical modeling of localized corrosion using phase-field and smoothed boundary methods. J Electrochem Soc 165(10):C633
8. Schmidt EM, Quinlan JM, Runnels B (2022) Self-similar diffuse boundary method for phase boundary driven flow. Phys Fluids 34:117108
9. Yu H-C, Chen H-Y, Thornton K (2012) Extended smoothed boundary method for solving partial differential equations with general boundary conditions on complex boundaries. Model Simul Mater Sci Eng 20(7):075008
10. Li X, Lowengrub J, Rätz A, Voigt A (2009) Solving PDEs in complex geometries: a diffuse domain approach. Commun Math Sci 7(1):81
11. Zhang W, Almgren A, Beckner V, Bell J, Blaschke J, Chan C, Day M, Friesen B, Gott K, Graves D et al (2019) AMReX: a framework for block-structured adaptive mesh refinement. J Open Source Softw 4(37):1370–1370
12. Hittinger JA, Banks JW (2013) Block-structured adaptive mesh refinement algorithms for Vlasov simulation. J Comput Phys 241:118–140
13. Schornbaum F, Rüde U (2018) Extreme-scale block-structured adaptive mesh refinement. SIAM J Sci Comput 40(3):C358–C387
14. Dubey A, Almgren A, Bell J, Berzins M, Brandt S, Bryan G, Colella P, Graves D, Lijewski M, Löffler F et al (2014) A survey of high level frameworks in block-structured adaptive mesh refinement packages. J Parallel Distrib Comput 74(12):3217–3227
15. Berger M, Rigoutsos I (1991) An algorithm for point clustering and grid generation. IEEE Trans Syst Man Cybern 21(5):1278–1286
16. Almgren AS, Bell JB, Colella P, Howell LH, Welcome ML (1998) A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations. J Comput Phys 142(1):1–46
17. Piller M, Stalio E (2004) Finite-volume compact schemes on staggered grids. J Comput Phys 197(1):299–340
18. Alves M, Oliveira P, Pinho F (2021) Numerical methods for viscoelastic fluid flows. Annu Rev Fluid Mech 53:509–541
19. Wesseling P, Segal A, Vankan J, Oosterlee C, Kassels C (1991) Finite volume discretization of the incompressible Navier-Stokes equations in general coordinates on staggered grids. In: Presented at the 4th international symposium on computational fluid dynamics
20. Anderson JD, Wendt J (1995) Computational fluid dynamics, vol 206. Springer, Berlin
21. Li B, Habbal F, Ortiz M (2010) Optimal transportation meshfree approximation schemes for fluid and plastic flows. Int J Numer Methods Eng 83(12):1541–1579
22. Monaghan JJ (1992) Smoothed particle hydrodynamics. Annu Rev Astron Astrophys 30:543–574
23. Sulsky D, Chen Z, Schreyer HL (1994) A particle method for history-dependent materials. Comput Methods Appl Mech Eng 118(1–2):179–196
24. Sulsky D, Zhou S-J, Schreyer HL (1995) Application of a particle-in-cell method to solid mechanics. Comput Phys Commun 87(1–2):236–252
25. Liang Y, Zhang X, Liu Y (2019) An efficient staggered grid material point method. Comput Methods Appl Mech Eng 352:85–109
26. Cardiff P, Demirdžić I (2021) Thirty years of the finite volume method for solid mechanics. Arch Comput Methods Eng 28(5):3721–3780
27. Wesseling P (1995) Introduction to multigrid methods. Tech. rep
28. Kanso E, Arroyo M, Tong Y, Yavari A, Marsden JG, Desbrun M (2007) On the geometric character of stress in continuum mechanics. Z Angew Math Phys 58(5):843–856
29. Yavari A (2008) On geometric discretization of elasticity. J Math Phys 49(2):022901
30. Yavari A (2010) A geometric theory of growth mechanics. J Nonlinear Sci 20(6):781–830
31. Desbrun M, Kanso E, Tong Y (2006) Discrete differential forms for computational modeling. In: ACM SIGGRAPH 2006 courses, pp 39–54
32. Zhu C, Lee CT, Rangamani P (2022) Mem3dg: modeling membrane mechanochemical dynamics in 3d using discrete differential geometry. Biophys J 121(3):71a
33. Ruocco E, Reddy J (2021) A discrete differential geometry-based approach to buckling and vibration analyses of inhomogeneous Reddy plates. Appl Math Model 100:342–364
34. Frankel T (2011) The geometry of physics: an introduction. Cambridge University Press, Cambridge
35. Aigner G, Hölzle U (1996) Eliminating virtual function calls in c++ programs. In: ECOOP'96-object-oriented programming: 10th European conference Linz, Austria, July 8–12, 1996 Proceedings 10. Springer, pp 142–166
36. Eijkhout V, Chow E, van de Geijn R (2022) The science of computing
37. Abrahams D, Gurtovoy A (2004) C++ template metaprogramming: concepts, tools, and techniques from Boost and beyond. Pearson Education, London

38. Bower AF (2009) Applied mechanics of solids. CRC Press, Boca Raton
39. Simo JC, Hughes TJ (2006) Computational inelasticity, vol 7. Springer, Berlin
40. Moës N, Dolbow J, Belytschko T (1999) A finite element method for crack growth without remeshing. Int J Numer Methods Eng 46(1):131–150
41. Sukumar N, Moës N, Moran B, Belytschko T (2000) Extended finite element method for three-dimensional crack modelling. Int J Numer Methods Eng 48(11):1549–1570
42. Li H, Li J, Yuan H (2018) A review of the extended finite element method on macrocrack and microcrack growth simulations. Theor Appl Fract Mech 97:236–249
43. Song C, Wolf JP (1997) The scaled boundary finite-element method-alias consistent infinitesimal finite-element cell method-for elastodynamics. Comput Methods Appl Mech Eng 147(3–4):329–355
44. Song C, Ooi ET, Natarajan S (2018) A review of the scaled boundary finite element method for two-dimensional linear elastic fracture mechanics. Eng Fract Mech 187:45–73
45. Wolf JP, Song C (2000) The scaled boundary finite-element method-a primer: derivations. Comput Struct 78(1–3):191–210
46. Song C, Wolf JP (2000) The scaled boundary finite-element method-a primer: solution procedures. Comput Struct 78(1–3):211–225
47. Egger A, Pillai U, Agathos K, Kakouris E, Chatzi E, Aschroft IA, Triantafyllou SP (2019) Discrete and phase field methods for linear elastic fracture mechanics: a comparative study and state-of-the-art review. Appl Sci 9(12):2436
48. Sedmak A (2018) Computational fracture mechanics: an overview from early efforts to recent achievements. Fatigue Fract Eng Mater Struct 41(12):2438–2474
49. Ambati M, Gerasimov T, De Lorenzis L (2015) A review on phase-field models of brittle fracture and a new fast hybrid formulation. Comput Mech 55(2):383–405
50. Francfort GA, Marigo J-J (1998) Revisiting brittle fracture as an energy minimization problem. J Mech Phys Solids 46(8):1319–1342
51. Nguyen T-T, Yvonnet J, Zhu Q-Z, Bornert M, Chateau C (2016) A phase-field method for computational modeling of interfacial damage interacting with crack propagation in realistic microstructures obtained by microtomography. Comput Methods Appl Mech Eng 312:567–595
52. Hansen-Dörr AC, Dammaß F, de Borst R, Kästner M (2020) Phase-field modeling of crack branching and deflection in heterogeneous media. Eng Fract Mech 232:107004
53. Teichtmeister S, Kienle D, Aldakheel F, Keip M-A (2017) Phase field modeling of fracture in anisotropic brittle solids. Int J Non-Linear Mech 97:1–21
54. Li B, Peco C, Millán D, Arias I, Arroyo M (2015) Phase-field modeling and simulation of fracture in brittle materials with strongly anisotropic surface energy. Int J Numer Methods Eng 102(3–4):711–727
55. Doan DH, Bui TQ, Duc ND, Fushinobu K (2016) Hybrid phase field simulation of dynamic crack propagation in functionally graded glass-filled epoxy. Compos Part B Eng 99:266–276
56. Dinachandra M, Alankar A (2020) A phase-field study of crack propagation and branching in functionally graded materials using explicit dynamics. Theor Appl Fract Mech 109:102681
57. Kumar PAV, Dean A, Reinoso J, Lenarda P, Paggi M (2021) Phase field modeling of fracture in functionally graded materials: γ-convergence and mechanical insight on the effect of grading. Thin-Walled Struct 159:107234
58. Karma A, Kessler DA, Levine H (2001) Phase-field model of mode III dynamic fracture. Phys Rev Lett 87(4):045501
59. Borden MJ, Verhoosel CV, Scott MA, Hughes TJ, Landis CM (2012) A phase-field description of dynamic brittle fracture. Comput Methods Appl Mech Eng 217:77–95
60. Hofacker M, Miehe C (2012) Continuum phase field modeling of dynamic fracture: variational principles and staggered FE implementation. Int J Fract 178(1):113–129
61. Ambati M, Gerasimov T, De Lorenzis L (2015) Phase-field modeling of ductile fracture. Comput Mech 55(5):1017–1040
62. Miehe C, Hofacker M, Schänzel L-M, Aldakheel F (2015) Phase field modeling of fracture in multi-physics problems. Part II. Coupled brittle-to-ductile failure criteria and crack propagation in thermo-elastic-plastic solids. Comput Methods Appl Mech Eng 294:486–522
63. Kuhn C, Noll T, Müller R (2016) On phase field modeling of ductile fracture. GAMM-Mitteilungen 39(1):35–54
64. Mesgarnejad A, Imanian A, Karma A (2019) Phase-field models for fatigue crack growth. Theor Appl Fract Mech 103:102282
65. Carollo V, Reinoso J, Paggi M (2018) Modeling complex crack paths in ceramic laminates: a novel variational framework combining the phase field method of fracture and the cohesive zone model. J Eur Ceram Soc 38(8):2994–3003
66. Tarafder P, Dan S, Ghosh S (2020) Finite deformation cohesive zone phase field model for crack propagation in multi-phase microstructures. Comput Mech 66(3):723–743
67. Quintanas-Corominas A, Turon A, Reinoso J, Casoni E, Paggi M, Mayugo J (2020) A phase field approach enhanced with a cohesive zone model for modeling delamination induced by matrix cracking. Comput Methods Appl Mech Eng 358:112618
68. Pham K, Marigo J-J, Maurini C (2011) The issues of the uniqueness and the stability of the homogeneous response in uniaxial tests with gradient damage models. J Mech Phys Solids 59(6):1163–1190
69. Kuhn C, Schlüter A, Müller R (2015) On degradation functions in phase field fracture models. Comput Mater Sci 108:374–384
70. Chen Y, Vasiukov D, Gélébart L, Park CH (2019) A FFT solver for variational phase-field modeling of brittle fracture. Comput Methods Appl Mech Eng 349:167–190
71. Ernesti F, Schneider M, Böhlke T (2020) Fast implicit solvers for phase-field fracture problems on heterogeneous microstructures. Comput Methods Appl Mech Eng 363:112793
72. Muixí A, Rodríguez-Ferran A, Fernández-Méndez S (2020) A hybridizable discontinuous Galerkin phase-field model for brittle fracture with adaptive refinement. Int J Numer Methods Eng 121(6):1147–1169
73. Nagaraja S, Elhaddad M, Ambati M, Kollmannsberger S, De Lorenzis L, Rank E (2019) Phase-field modeling of brittle fracture with multi-level hp-fem and the finite cell method. Comput Mech 63(6):1283–1300
74. Giovanardi B, Scotti A, Formaggia L (2017) A hybrid xfem-phase field (xfield) method for crack propagation in brittle elastic materials. Comput Methods Appl Mech Eng 320:396–420
75. Lo Y-S, Borden MJ, Ravi-Chandar K, Landis CM (2019) A phase-field model for fatigue crack growth. J Mech Phys Solids 132:103684
76. Sun X, Duddu R et al (2021) A poro-damage phase field model for hydrofracturing of glacier crevasses. Extreme Mech Lett 45:101277
77. Clayton T, Duddu R, Siegert M, Martínez-Pañeda E (2022) A stress-based poro-damage phase field model for hydrofracturing of creeping glaciers and ice shelves. Eng Fract Mech 272:108693
78. Mo X, Zhi H, Xiao Y, Hua H, He L (2021) Topology optimization of cooling plates for battery thermal management. Int J Heat Mass Transf 178:121612
79. Andreasen CS, Sigmund O (2013) Topology optimization of fluid-structure-interaction problems in poroelasticity. Comput Methods Appl Mech Eng 258:55–62

80. Lógó J, Ismail H et al (2020) Milestones in the 150-year history of topology optimization: a review. Comput Assist Methods Eng Sci 27(2–3):97–132
81. Bendsøe MP (1989) Optimal shape design as a material distribution problem. Struct Optim 1(4):193–202
82. Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Arch Appl Mech 69(9):635–654
83. Sokolowski J, Zochowski A (1999) On the topological derivative in shape optimization. SIAM J Control Optim 37(4):1251–1272
84. Allaire G, Jouve F, Toader A-M (2002) A level-set method for shape optimization. C R Math 334(12):1125–1130
85. Allaire G, Jouve F, Toader A-M (2004) Structural optimization using sensitivity analysis and a level-set method. J Comput Phys 194(1):363–393
86. Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. Comput Methods Appl Mech Eng 192(1–2):227–246
87. Xie YM, Steven GP (1993) A simple evolutionary procedure for structural optimization. Comput Struct 49(5):885–896
88. Li B, Huang C, Li X, Zheng S, Hong J (2019) Non-iterative structural topology optimization using deep learning. Comput-Aided Des 115:172–180
89. Rade J, Balu A, Herron E, Pathak J, Ranade R, Sarkar S, Krishnamurthy A (2021) Algorithmically-consistent deep learning frameworks for structural topology optimization. Eng Appl Artif Intel 106:104483
90. Chi H, Zhang Y, Tang TLE, Mirabella L, Dalloro L, Song L, Paulino GH (2021) Universal machine learning for topology optimization. Comput Methods Appl Mech Eng 375:112739
91. Sigmund O, Maute K (2013) Topology optimization approaches. Struct Multidiscip Optim 48(6):1031–1055
92. Jihong Z, Han Z, Chuang W, Lu Z, Shangqin Y, Zhang W (2021) A review of topology optimization for additive manufacturing: status and challenges. Chin J Aeronaut 34(1):91–110
93. Bourdin B, Chambolle A (2003) Design-dependent loads in topology optimization. ESAIM Control Optim Calc Var 9:19–48
94. Wallin M, Ristinmaa M, Askfelt H (2012) Optimal topologies derived from a phase-field method. Struct Multidiscip Optim 45(2):171–183
95. Wang MY, Zhou S (2004) Phase field: a variational method for structural topology optimization. CMES-Comput Model Eng Sci 6(6):547
96. Burger M, Stainko R (2006) Phase-field relaxation of topology optimization with local stress constraints. SIAM J Control Optim 45(4):1447–1466
97. Jeong SH, Yoon GH, Takezawa A, Choi D-H (2014) Development of a novel phase-field method for local stress-based shape and topology optimization. Comput Struct 132:84–98
98. Salazar de Troya MA, Tortorelli DA (2018) Adaptive mesh refinement in stress-constrained topology optimization. Struct Multidiscip Optim 58(6):2369–2386
99. Jung M, Yoo J (2021) Phase field-based topology optimization of metallic structures for microwave applications using adaptive mesh refinement. Struct Multidiscip Optim 63(6):2685–2704