## A novel phase-field monolithic scheme for brittle crack propagation based on the limited-memory BFGS method with adaptive mesh refinement

Jin, Tao; Li, Zhao; Chen, Kuiying



**Publisher's version / Version de l'éditeur:**
https://doi.org/10.1002/nme.7572
*International Journal for Numerical Methods in Engineering, 2024-07-29*

Received: 7 April 2024 | Revised: 7 June 2024 | Accepted: 8 July 2024

DOI: 10.1002/nme.7572

RESEARCH ARTICLE

WILEY

# A novel phase-field monolithic scheme for brittle crack propagation based on the limited-memory BFGS method with adaptive mesh refinement

**Tao Jin**[1] | **Zhao Li**[1] | **Kuiying Chen**[2]

[1]Department of Mechanical Engineering, University of Ottawa, Ottawa, Ontario, Canada

[2]Structures and Materials Performance Laboratory, Aerospace Research Centre, National Research Council Canada, Ottawa, Ontario, Canada

**Correspondence**
Tao Jin, Department of Mechanical Engineering, University of Ottawa, Ottawa, ON K1N 6N5, Canada.
Email: tao.jin@uOttawa.ca

**Funding information**
Natural Sciences and Engineering Research Council of Canada (NSERC), Grant/Award Number: RGPIN-2021-02561; National Research Council Canada, Grant/Award Number: A1-018177

## Abstract

The phase-field formulation for fracture propagation is widely adopted due to its capability of naturally treating complex crack geometries. The challenges of the phase-field crack simulation include the non-convexity of the underlying energy functional and the expensive computational cost associated with the fine mesh required to resolve the phase-field length-scale around the crack region. We present a novel phase-field monolithic scheme based on the limited-memory Broyden–Fletcher–Goldfarb–Shanno (BFGS) method, or the L-BFGS method, to address the convergence difficulties usually encountered by a Newton-based approach because of the non-convex energy functional. Comparing with the conventional BFGS method, the L-BFGS monolithic scheme avoids to store the fully dense Hessian approximation matrix. This feature is critical in the context of finite element simulations. To alleviate the expensive computational cost, we integrate the proposed L-BFGS monolithic scheme with an adaptive mesh refinement (AMR) technique. We provide the algorithmic details about the proposed L-BFGS monolithic scheme, especially about how to handle the hanging-node constraints generated during the AMR process as extra linear constraints. Several two-dimensional (2D) and three-dimensional (3D) numerical examples are provided to demonstrate the capabilities of the proposed monolithic scheme, including the accuracy, the robustness, and the computational efficiency regarding the memory consumption and the wall-clock time. Particularly, we emphasize the importance of the appropriately chosen convergence criteria for brute crack propagation. The proposed L-BFGS phase-field monolithic scheme combined with the AMR technique offers an accurate, robust, and efficient approach to model brittle crack propagation in both 2D and 3D problems.

**KEYWORDS**

adaptive mesh refinement, crack propagation, deal.II, limited-memory BFGS, monolithic scheme, phase-field method

## 1 | INTRODUCTION

One of the challenges in modeling crack propagation in solids is the handling of complex crack geometry. The level-set based on tracking strategy,[1–3] for example, attempts to use the iso-surface of an auxiliary scalar field to represent the sharp crack geometry. This type of tracking strategy can be combined with the embedded discontinuity technique[4–6] to model discontinuity propagation, such as crack and shear band, in both brittle and ductile materials.[7,8] Although the above numerical technique can successfully resolve the sharp crack for simple crack geometries, it becomes ineffective to handle complex crack propagation paths such as branching and merging. Instead of directly resolving the sharp crack surface, the phase-field method based on a variational approach[9–11] regularizes the sharp crack geometry in a diffusive manner. Due to the underlying variational structure and the unique capability to naturally handle complex crack geometries in both two-dimensional (2D) and three-dimensional (3D) problems, the phase-field approach becomes widely adopted to model crack propagation in solids during the last decade.[12–19]

The phase-field modeling of brittle crack propagation faces three major challenges, including the non-convexity of the underlying energy functional, the expensive computational cost, and the enforcement of the phase-field (damage) irreversibility. It is well known that the energy functional of the brittle crack phase-field formulation is non-convex.[10,11] Consequently, the Newton-based monolithic approach encounters convergence difficulties. To overcome the limitation of the Newton-based approach, several monolithic schemes based on the modified Newton method are proposed.[17,20,21] The central idea of the modified Newton method is to modify the Jacobian matrix (the Hessian of the energy functional after finite element discretization) in order to restore its positive definiteness. However, this process remains heuristic and involves several parameters that need to be selected during the modification of the Jacobian matrix. Other efforts to develop the monolithic scheme for the phase-field problem include a fracture energy based arc-length method,[22] a multilevel trust region method,[23] and a novel line-search assisted procedure.[24] Particularly, the Broyden–Fletcher–Goldfarb–Shanno method, or the BFGS method, which is a type of quasi-Newton method, provides a systematic approach to solve the phase-field crack propagation problem in a robust and efficient way.[18] However, in the classical BFGS procedure, each BFGS iteration involves an update of the approximate Hessian or its inverse, which requires the storage of a fully dense matrix. In the context of a finite element simulation, the classical BFGS procedure fails to take advantage of the finite element sparsity pattern, and the memory required to store the fully dense approximate Hessian becomes impractical even for a mid-size finite element problem. Therefore, special considerations have to be given during the implementation of the BFGS method for the phase-field simulation or any finite element problems to avoid the storage of a fully dense matrix.[25] Alternatively, the numerical difficulties arising from the non-convexity of the energy functional can be circumvented by adopting the so-called alternate minimization scheme based on a staggered approach.[13,26–28] The staggered approach decouples the tightly coupled problem into a displacement sub-problem and a phase-field sub-problem. The two sub-problems are both convex and can be efficiently solved. Despite of the robustness, the staggered approach might be computationally inefficient due to the required large number of iterations between the two sub-problems. Moreover, an inappropriate choice of convergence criteria may under-predict the crack propagation, especially when the crack propagates in a brutal manner.

The phase-field formulation for crack propagation simulation is computationally expensive. In order to model a sharp crack in a diffusive way, the phase-field length-scale needs to be resolved around the crack region. This requirement demands that the mesh size $h$ around the crack region needs to be smaller than the phase-field length-scale parameter $l$ during the spatial discretization. For a crack whose propagation path is known a priori, the mesh could simply be pre-refined along the expected propagation path to resolve the phase-field length-scale $l$. However, for scenarios where the crack propagation path is unknown, a large part of the computational domain needs to be discretized with a mesh size $h$ that is smaller than $l$. This level of spatial discretization makes the computational cost prohibitively expensive even for 2D problems, and is completely unpractical for 3D problems. Therefore, an adaptive mesh refinement (AMR) is essential to reduce the computational cost associated with the phase-field method for crack propagation simulations. Recently, Heister et al.[15] proposed a predictor-corrector local mesh adaptivity scheme, which does not require the knowledge of the crack path a priori. As a result, the number of degrees of freedom (DoFs) grows as the crack propagates. This AMR technique can ensure the adequate mesh resolution around the crack region and apply the highly refined mesh only when it is needed, therefore, reducing the computational cost.

To be consistent with thermodynamics, cracks do not self-heal after they are formed. In the phase-field formulation of crack propagation, this thermodynamic requirement is included as the phase-field irreversibility condition. In fact, this irreversibility condition should not only be enforced in the fully cracked region but also in any partially damaged region. In

the literature, there are different approaches to enforce the phase-field irreversibility condition. These approaches vary in the aspect of implementation complexity, mathematical rigor, and numerical performance. Bourdin[29] proposed to directly set Dirichlet boundary conditions on the fully developed cracks to prevent crack healing. However, this type of approach fails to enforce the irreversibility condition in partially damaged region. Miehe et al.[13] introduced a history variable of the maximum positive strain energy to drive the evolution of the phase-field. Despite of the convenience, the enforcement of the irreversibility condition based on this history variable approach makes the weak form inconsistent with the underlying energy functional. Wheeler et al.[30] proposed an augmented Lagrangian approach to enforce the irreversibility condition. This approach can avoid solving the variational inequality introduced by the phase-field irreversibility. However, it is difficult to identify the parameters in order to effectively enforce the irreversibility constraint while at the same time avoid the ill-conditioning of the system. Heister et al.[15] proposed a primal-dual active set method to enforce the phase-field irreversibility. This method can be considered as a type of semi-smooth Newton method[31] and is effective to enforce the phase-field irreversibility. However, this work relies on an heuristic phase-field extrapolation approach to circumvent the numerical difficulties associated with the non-convexity of the energy functional, which may not be valid. Wambacq et al.[20] proposed an interior-point method to directly treat the phase-field irreversibility as a variational inequality. In the authors' opinion, the interior-point method is the most rigorous way since it neither modifies the governing equations nor uses any penalty parameters. However, this work still requires the modification of the Hessian matrix through a stabilization factor to restore the positive definiteness.

As a type of quasi-Newton method, the BFGS method, which is originally based on the idea of Davidon,[32,33] has been widely used in nonlinear optimization. Comparing with the modified Newton method, the BFGS method does not attempt to restore the positive definiteness of the Hessian through matrix modification. Rather, it constructs a series of approximate Hessian (or its inverse) that are positive definite through a rank-two update using curvature information. The effectiveness of the BFGS method for the phase-field crack formulation is demonstrated by Wu et al.[18] However, the implementation of the BFGS method in their work[18] is through a commercial finite element software. Except mentioning about using the work by Matthies and Strang[25] to avoid the storage of a fully dense matrix during the BFGS update, the software manual does not provide more implementation details. Therefore, it is hard to extend the BFGS implementation to include other advanced features such as AMR.

In this article, we propose a phase-field monolithic scheme based on the limited-memory BFGS method,[34–36] or the L-BFGS method, combined with an AMR technique[15] to model brittle crack propagation. Comparing with the classical BFGS update procedure that stores a fully dense matrix, the L-BFGS method only requires to store a few vector-pairs in the memory during the calculation of the search direction. The L-BFGS method offers an effective and systematic approach to address the challenges associated with the non-convexity of the energy functional. To alleviate the expensive computational cost during the phase-field formulation, the proposed L-BFGS monolithic scheme is further integrated with the AMR technique originally proposed by Heister et al.[15] Several modifications are introduced into the original refinement technique to improve the convergence behavior of the L-BFGS monolithic scheme. During the mesh refinement process, a series of hanging nodes are generated at element interfaces. These hanging-node constraints along with the prescribed essential boundary conditions are treated as algebraic nodal constraints and incorporated into the proposed L-BFGS monolithic scheme. For the enforcement of the irreversibility condition, we adopt the phase-field formulation based on the history variable of the maximum positive strain energy.[13] Instead of involving more advanced numerical optimization techniques, such as interior point method, active set method and so forth to solve a constrained optimization problem that enforces the phase-field irreversibility as inequality constraints, the history variable approach only requires to store the maximum value of the positive strain energy as an extra scalar at each quadrature point. Additionally, this approach is shown to be motivated by the local damage mechanics.[13] Due to the relative simplicity of the history variable based approach, we can therefore focus on presenting the proposed L-BFGS monolithic scheme combined with the AMR technique to address the challenges of the non-convex energy functional and the expensive computational cost in the phase-field crack simulation. In order to consider the tension-compression asymmetry of brittle fracture propagation, a common approach is to split the total strain energy and only degrade the part related to the tensile deformation. There are multiple approaches in the literature to split the strain energy function, for instance, the volumetric-deviatoric decomposition of the elastic strain energy,[37] the positive-negative split based on the spectrum decomposition of the strain tensor,[12,13] the positive-negative split based on the effective stress tensor,[38,39] and the directional stress split based on the crack orientation,[40] to name a few. The review papers by Wu et al.[41] and De Lorenzis et al.[14] provide thorough comparisons of various splitting techniques. To the authors' best knowledge, each splitting technique has its own advantages and shortcomings depending on the specific loading conditions. In this work, we adopt the split based on the spectrum decomposition of the strain tensor popularized by Miehe et al.[12,13] It is worth emphasizing that the choice of the splitting technique is part

of the material constitutive formulation, and therefore, does not change the proposed L-BFGS monolithic approach. In case a different splitting technique needs to be adopted, we can readily replace the constitutive calculation at element quadrature points. The proposed L-BFGS approach and the AMR technique remain unchanged.

The remaining article is organized as follows. In Section 2, we briefly review the phase-field formulation for brittle fracture simulations originally proposed by Miehe et al.[13] and the corresponding finite element discretization. In Section 3, we propose a monolithic scheme based on the limited-memory BFGS method, or the L-BFGS method, to solve the weak form of the phase-field formulation. Furthermore, we incorporate a modified AMR technique[15] into the L-BFGS monolithic scheme. In particular, we present the algorithmic details about how to consider the hanging-node constraints generated during the AMR process and the prescribed essential boundary conditions in the L-BFGS procedure. In Section 4, we provide several 2D and 3D numerical examples widely adopted in the phase-field literature to evaluate the performance of the proposed L-BFGS monolithic scheme, including its accuracy, robustness, computational efficiency, and dimension-independence. In Section 5, we summarize the features of the proposed monolithic scheme and discuss about several avenues to improve the current treatment of the phase-field irreversibility condition, such as the primal-dual active set method[15] and the interior-point method.[20] The proposed L-BFGS monolithic scheme is implemented in deal.II,[42] which is an open-source C++ finite element library. In support of the efforts of open-source software and reproducible research, all the source codes and input files developed in this article are hosted on GitHub.*

## 2 | PHASE-FIELD FORMULATION

The phase-field formulation adopted in this article is mainly based on the work by Miehe et al.[13] For the sake of completeness, we briefly review the major components in this formulation, including the specific form of the phase-field degradation function, the additive decomposition of the strain energy, the enforcement technique of the phase-field irreversibility using the history variable of the maximum positive strain energy, and the viscous regularization. After the finite element discretization, we provide the discretized forms of the total energy of the cracked system and the corresponding residuals, which are needed for the subsequent development of the L-BFGS monolithic scheme and the line search method based on the strong Wolfe conditions.

### 2.1 | Governing equations

Let $\boldsymbol{u}(\boldsymbol{x})$ represent the unknown displacement field vector and $d(\boldsymbol{x})$ represent the unknown phase-field scalar. The energy functional of the fractured solid system is expressed as

$$\Pi(\boldsymbol{u}, d) = \int_{\Omega} \psi(\boldsymbol{\epsilon}(\boldsymbol{u}), d)\mathrm{d}\Omega + g_c \Gamma_l(d) - \int_{\Omega} \boldsymbol{b} \cdot \boldsymbol{u}\ \mathrm{d}\Omega - \int_{\partial\Omega} \boldsymbol{t} \cdot \boldsymbol{u}\ \mathrm{d}\Gamma, \tag{1}$$

where $\boldsymbol{b}$ is the body force, $\boldsymbol{t}$ is the traction load, $\boldsymbol{\epsilon} = \nabla^{(s)}\boldsymbol{u}$ is the small deformation linear strain tensor, $\psi$ represents the strain energy density function, $g_c$ is the critical energy release rate, and $\Gamma_l$ is an approximation of the crack surface area. Particularly, the approximated crack surface $\Gamma_l$ is defined as

$$\Gamma_l(d) = \int_{\Omega} \gamma(d, \nabla d)\mathrm{d}\Omega = \int_{\Omega} \frac{1}{2l}\left(d^2 + l^2 \nabla d \cdot \nabla d\right)\mathrm{d}\Omega, \tag{2}$$

where $\gamma(d, \nabla d)$ is considered as the crack surface density function, and $l$ is the phase-field length-scale parameter. The strain energy density function is based on the isotropic linear elasticity and the additive decomposition,

$$\psi(\boldsymbol{\epsilon}, d) = [g(d) + k]\psi^{+}(\boldsymbol{\epsilon}) + \psi^{-}(\boldsymbol{\epsilon}), \tag{3}$$

where $\psi^{+}$ is the positive part of the strain energy, $\psi^{-}$ is the negative part of the strain energy, $k$ is a small non-negative number, and $g(d)$ represents the phase-field degradation function and adopts the following form,

$$g(d) = (1 - d)^2. \tag{4}$$

For the additive decomposition of the strain energy, we introduce the following operators,

$$\langle x\rangle_+ = \frac{1}{2}(x+|x|), \quad \langle x\rangle_- = \frac{1}{2}(x-|x|), \quad H(x) = \begin{cases} 1 & x \geq 0, \\ 0 & x < 0. \end{cases}$$

The spectrum decomposition of the strain tensor $\boldsymbol{\epsilon}$ is expressed as

$$\boldsymbol{\epsilon} = \sum_\alpha \epsilon_\alpha \mathbf{M}_\alpha, \quad \mathbf{M}_\alpha = \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha,$$

where $\epsilon_\alpha$ and $\boldsymbol{n}_\alpha$ represent a pair of eigenvalue and eigenvector. The positive and negative parts of the strain tensor are defined as,

$$\boldsymbol{\epsilon}^+ = \sum_\alpha \langle \epsilon_\alpha\rangle_+ \mathbf{M}_\alpha, \quad \boldsymbol{\epsilon}^- = \sum_\alpha \langle \epsilon_\alpha\rangle_- \mathbf{M}_\alpha.$$

Using the above definitions, the positive and negative parts of the strain energy are expressed as

$$\begin{aligned} \psi^+(\boldsymbol{\epsilon}) &= \frac{1}{2}\lambda\langle \mathrm{tr}\boldsymbol{\epsilon}\rangle_+^2 + \mu\boldsymbol{\epsilon}^+ : \boldsymbol{\epsilon}^+, \\ \psi^-(\boldsymbol{\epsilon}) &= \frac{1}{2}\lambda\langle \mathrm{tr}\boldsymbol{\epsilon}\rangle_-^2 + \mu\boldsymbol{\epsilon}^- : \boldsymbol{\epsilon}^-, \end{aligned} \tag{5}$$

where $\lambda$ and $\mu$ are the Lamé parameters, and $\mathrm{tr}\boldsymbol{\epsilon}$ is the trace of the strain tensor.

The primary unknown fields $\boldsymbol{u}(\boldsymbol{x})$ and $d(\boldsymbol{x})$ can be obtained from the minimization of the energy functional

$$(\boldsymbol{u}(\boldsymbol{x}), d(\boldsymbol{x})) = \mathrm{argmin}\{\Pi(\boldsymbol{u}, d)\}$$

subject to the constraints

$$\dot{d}(\boldsymbol{x}) \geq 0 \text{ and } 0 \leq d(\boldsymbol{x}) \leq 1.$$

The first inequality constraint represents the phase-field (damage) irreversibility condition, and the second set of inequality constraints indicate that the phase-field is between $d = 0$ (the undamaged state) and $d = 1$ (the fully damaged state). Using the directional derivatives, the first variation of the energy functional is written as

$$\begin{aligned} \delta\Pi(\boldsymbol{u}, d) = \mathcal{D}_{(\delta\boldsymbol{u},\delta d)}\Pi(\boldsymbol{u}, d) &= \left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\right|_{\epsilon=0} \Pi(\boldsymbol{u} + \epsilon\ \delta\boldsymbol{u}, d + \epsilon\ \delta d) \\ &= \int_\Omega \left( \frac{\partial\psi(\boldsymbol{\epsilon}(\boldsymbol{u}), d)}{\partial\boldsymbol{\epsilon}} : \boldsymbol{\epsilon}(\delta\boldsymbol{u}) + \frac{\partial\psi(\boldsymbol{\epsilon}(\boldsymbol{u}), d)}{\partial d}\ \delta d \right)\mathrm{d}\Omega + \int_\Omega \frac{\mathrm{g}_c}{l}\left(d\ \delta d + l^2\nabla d\cdot\nabla\delta d\right)\mathrm{d}\Omega \\ &\quad - \int_\Omega \boldsymbol{b}\cdot\delta\boldsymbol{u}\ \mathrm{d}\Omega - \int_{\partial\Omega} \boldsymbol{t}\cdot\delta\boldsymbol{u}\ \mathrm{d}\Gamma \\ &= (\nabla^{(s)}\delta\boldsymbol{u},\ \boldsymbol{\sigma}) - (\delta\boldsymbol{u},\ \boldsymbol{b}) - (\delta\boldsymbol{u},\ \boldsymbol{t})_{\Gamma_t} + \left(\delta d,\ \frac{\mathrm{g}_c}{l}d\right) + (\nabla\delta d,\ \mathrm{g}_c l\nabla d) + (\delta d,\ g'(d)\psi^+(\boldsymbol{\epsilon})). \end{aligned}$$

The stress tensor $\boldsymbol{\sigma}$ is defined as

$$\boldsymbol{\sigma} = \frac{\partial\psi(\boldsymbol{\epsilon}(\boldsymbol{u}), d)}{\partial\boldsymbol{\epsilon}} = [g(d)+k]\frac{\partial\psi^+(\boldsymbol{\epsilon})}{\partial\boldsymbol{\epsilon}} + \frac{\partial\psi^-(\boldsymbol{\epsilon})}{\partial\boldsymbol{\epsilon}} = [g(d)+k]\boldsymbol{\sigma}^+ + \boldsymbol{\sigma}^-,$$

where

$$\boldsymbol{\sigma}^+ = \lambda\langle \mathrm{tr}\boldsymbol{\epsilon}\rangle_+\mathbf{I} + 2\mu\boldsymbol{\epsilon}^+, \quad \boldsymbol{\sigma}^- = \lambda\langle \mathrm{tr}\boldsymbol{\epsilon}\rangle_-\mathbf{I} + 2\mu\boldsymbol{\epsilon}^-.$$

Using the divergence theorem and the technique of integration by parts, the corresponding Euler–Lagrange equations are written as

$$\begin{cases} \text{div}\boldsymbol{\sigma} + \boldsymbol{b} = \boldsymbol{0} \\ \frac{g_c}{l}(d - l^2\Delta d) = -g'(d)\psi^+ \end{cases},$$

in addition to the Neumann boundary conditions

$$\begin{cases} \boldsymbol{\sigma}\boldsymbol{n} = \boldsymbol{t} & \text{on} \quad \Gamma_{\boldsymbol{t}} \\ \nabla d \cdot \boldsymbol{n} = 0 & \text{on} \quad \Gamma_{\nabla d} \end{cases}.$$

Among various approaches to enforce the phase-field irreversibility, we adopt the approach based on the history variable of the positive strain energy[13] due to its relative simplicity,

$$\mathcal{H}(\boldsymbol{x}, t) = \max_{s\in[0,t]} \psi^+(\boldsymbol{\epsilon}(\boldsymbol{x}, s)). \tag{6}$$

Moreover, a viscosity regularization term $\eta\dot{d}$ can be introduced to stabilize the numerical treatment. Even though this regularization term might not be necessary, we still include it in the formulation for flexibility. This term can always be turned off by setting $\eta = 0$. Including the above considerations, the governing equations of the cracked solid system can be expressed as

$$\begin{cases} \text{div}\boldsymbol{\sigma} + \boldsymbol{b} = \boldsymbol{0} \\ \frac{g_c}{l}(d - l^2\Delta d) + \eta\dot{d} = -g'(d)\mathcal{H} \end{cases}, \tag{7}$$

which is originally proposed by Miehe et al.[13]

We introduce a pseudo-time variable $t$ to represent load steps. Inside a time step $t \in [t_n, t_{n+1}]$ with $\Delta t = t_{n+1} - t_n$, the weak form of the governing PDEs shown in Equation (7) can be expressed as

$$r_{\boldsymbol{u}}(\boldsymbol{u}, d) = (\nabla^{(s)}\delta\boldsymbol{u},\ \boldsymbol{\sigma}) - (\delta\boldsymbol{u},\ \boldsymbol{b}) - (\delta\boldsymbol{u},\ \boldsymbol{t})_{\Gamma_t} = 0, \tag{8}$$

and

$$\begin{aligned} r_d(\boldsymbol{u}, d) &= \left(\delta d,\ \frac{g_c}{l}d\right) + (\nabla\delta d,\ g_c l\nabla d) + (\delta d,\ \eta\dot{d}) + (\delta d,\ g'(d)\mathcal{H}) \\ &= \left(\delta d,\ \frac{g_c}{l}d\right) + (\nabla\delta d,\ g_c l\nabla d) + \left(\delta d,\ \eta\frac{d - d_n}{\Delta t}\right) + (\delta d,\ g'(d)\mathcal{H}) = 0, \end{aligned} \tag{9}$$

where $\delta\boldsymbol{u}$ and $\delta d$ represent the test functions, and $d_n$ is the known phase-field at the beginning of the time step. The history variable of the positive strain energy is written as

$$\mathcal{H} = \begin{cases} \psi^+(\boldsymbol{\epsilon}(\boldsymbol{u}^{(i)})) & \text{if} \quad \psi^+(\boldsymbol{\epsilon}(\boldsymbol{u}^{(i)})) > \mathcal{H}_n, \\ \mathcal{H}_n & \text{if} \quad \psi^+(\boldsymbol{\epsilon}(\boldsymbol{u}^{(i)})) \le \mathcal{H}_n. \end{cases}$$

Note that the above expressions of the weak form are nonlinear. Assume $i$ represents the $i$th iteration, we can write the Newton–Raphson solution process for the coupled system as

$$\begin{cases} \mathcal{D}_{\Delta\boldsymbol{u}} r_{\boldsymbol{u}}(\boldsymbol{u}^{(i)}, d^{(i)}) + \mathcal{D}_{\Delta d} r_{\boldsymbol{u}}(\boldsymbol{u}^{(i)}, d^{(i)}) = -r_{\boldsymbol{u}}(\boldsymbol{u}^{(i)}, d^{(i)}) \\ \mathcal{D}_{\Delta\boldsymbol{u}} r_d(\boldsymbol{u}^{(i)}, d^{(i)}) + \mathcal{D}_{\Delta d} r_d(\boldsymbol{u}^{(i)}, d^{(i)}) = -r_d(\boldsymbol{u}^{(i)}, d^{(i)}), \end{cases} \tag{10}$$

where

$$\begin{aligned}
\mathcal{D}_{\Delta \boldsymbol{u}} r_{\boldsymbol{u}}(\boldsymbol{u}, d) &= \left(\nabla^{(s)} \delta \boldsymbol{u},\ \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\epsilon}} : \boldsymbol{\epsilon}(\Delta \boldsymbol{u})\right), \\
\mathcal{D}_{\Delta d} r_{\boldsymbol{u}}(\boldsymbol{u}, d) &= \left(\nabla^{(s)} \delta \boldsymbol{u},\ \frac{\partial \boldsymbol{\sigma}}{\partial d} \Delta d\right), \\
\mathcal{D}_{\Delta \boldsymbol{u}} r_d(\boldsymbol{u}, d) &= \left(\delta d,\ g'(d) \frac{\partial \mathcal{H}}{\partial \boldsymbol{\epsilon}} : \boldsymbol{\epsilon}(\Delta \boldsymbol{u})\right), \\
\mathcal{D}_{\Delta d} r_d(\boldsymbol{u}, d) &= \left(\delta d,\ \left(\frac{g_c}{l} + \frac{\eta}{\Delta t} + g''(d)\mathcal{H}\right)\Delta d\right) + (\nabla \delta d,\ g_c l \nabla(\Delta d)).
\end{aligned}$$

Particularly, since the phase-field degradation function is only applied to the positive part of the strain energy, the stress-strain relationship is nonlinear, and the material tangent modulus can be expressed as

$$\begin{aligned}
\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\epsilon}} &= [g(d) + k] \frac{\partial \boldsymbol{\sigma}^+}{\partial \boldsymbol{\epsilon}} + \frac{\partial \boldsymbol{\sigma}^-}{\partial \boldsymbol{\epsilon}} \\
&= [g(d) + k]\left[\lambda H(\mathrm{tr}\boldsymbol{\epsilon}) \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbb{P}^+\right] + \left[\lambda H(-\mathrm{tr}\boldsymbol{\epsilon}) \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbb{P}^-\right].
\end{aligned}$$

The two fourth-order projection tensors $\mathbb{P}^+$ and $\mathbb{P}^-$ are defined as

$$\mathbb{P}^+ = \frac{\partial \boldsymbol{\epsilon}^+}{\partial \boldsymbol{\epsilon}}, \quad \mathbb{P}^- = \frac{\partial \boldsymbol{\epsilon}^-}{\partial \boldsymbol{\epsilon}},$$

the specific expressions of which are given in Appendix A. The other derivatives needed for the linearization of the weak form are provided as below:

$$\frac{\partial \boldsymbol{\sigma}}{\partial d} = g'(d) \boldsymbol{\sigma}^+, \quad \frac{\partial \mathcal{H}}{\partial \boldsymbol{\epsilon}} = \begin{cases} \boldsymbol{\sigma}^+(\boldsymbol{u}^{(i)}) & \psi^+(\boldsymbol{\epsilon}(\boldsymbol{u}^{(i)})) > \mathcal{H}_n, \\ \mathbf{0} & \psi^+(\boldsymbol{\epsilon}(\boldsymbol{u}^{(i)})) \le \mathcal{H}_n. \end{cases}$$

## 2.2 | Finite element discretization

Let $\delta \boldsymbol{u}_A = \boldsymbol{N}_{\boldsymbol{u}_A}$ represent the vector-valued shape function for the displacement field $\boldsymbol{u}$ at node $A$, and $\delta d_A = N_{d_A}$ represent the scalar-valued shape function for the phase-field $d$ at node $A$. During the $i$th iteration in the Newton–Raphson method, the linear equations of the coupled system shown in Equation (10) can be discretized using the shape functions as

$$\begin{bmatrix} \mathbf{K}_{\boldsymbol{u}\boldsymbol{u}}^{(i)} & \mathbf{K}_{\boldsymbol{u}d}^{(i)} \\ \mathbf{K}_{d\boldsymbol{u}}^{(i)} & \mathbf{K}_{dd}^{(i)} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{u}^{(i)} \\ \Delta d^{(i)} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{r}_{\boldsymbol{u}}^{(i)} \\ -\boldsymbol{r}_d^{(i)} \end{bmatrix}, \tag{11}$$

where

$$\begin{aligned}
\mathbf{K}_{\boldsymbol{u}_A \boldsymbol{u}_B} &= \left(\nabla^{(s)} \boldsymbol{N}_{\boldsymbol{u}_A},\ \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\epsilon}} : \nabla^{(s)} \boldsymbol{N}_{\boldsymbol{u}_B}\right), \\
\mathbf{K}_{\boldsymbol{u}_A d_B} &= \left(\nabla^{(s)} \boldsymbol{N}_{\boldsymbol{u}_A},\ \frac{\partial \boldsymbol{\sigma}}{\partial d} N_{d_B}\right), \\
\mathbf{K}_{d_A \boldsymbol{u}_B} &= \left(N_{d_A},\ g'(d) \frac{\partial \mathcal{H}}{\partial \boldsymbol{\epsilon}} : \nabla^{(s)} \boldsymbol{N}_{\boldsymbol{u}_B}\right), \\
\mathbf{K}_{d_A d_B} &= \left(N_{d_A},\ \left(\frac{g_c}{l} + \frac{\eta}{\Delta t} + g''(d)\mathcal{H}\right) N_{d_B}\right) + \left(\nabla N_{d_A},\ g_c l \nabla N_{d_B}\right), \\
\boldsymbol{r}_{\boldsymbol{u}_A} &= \left(\nabla^{(s)} \boldsymbol{N}_{\boldsymbol{u}_A},\ \boldsymbol{\sigma}\right) - \left(\boldsymbol{N}_{\boldsymbol{u}_A},\ \boldsymbol{b}\right) - \left(\boldsymbol{N}_{\boldsymbol{u}_A},\ \boldsymbol{t}\right)_{\Gamma_t}, \\
\boldsymbol{r}_{d_A} &= \left(N_{d_A},\ \frac{g_c}{l} d + \eta \frac{d - d_n}{\Delta t} + g'(d)\mathcal{H}\right) + (\nabla N_{d_A},\ g_c l \nabla d).
\end{aligned} \tag{12}$$

Particularly, we define the following diagonal block matrix $\hat{\mathbf{K}}$ for the development of the L-BFGS monolithic scheme,

$$\hat{\mathbf{K}} = \begin{bmatrix} \mathbf{K}_{uu} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{dd} \end{bmatrix}. \tag{13}$$

Based on the shape functions $\{\boldsymbol{N}_{\boldsymbol{u}}, N_d\}$ and the nodal finite element solutions $\{\boldsymbol{u}, d\}$, the displacement field and the phase-field can be expressed as

$$\boldsymbol{u} = \boldsymbol{N}_{\boldsymbol{u}_A} \boldsymbol{u}_A, \quad \text{and} \quad d = N_{d_A} d_A,$$

where the Einstein summation is used. Plug the above expressions into the total energy defined in Equation (1), we have

$$\begin{aligned} \Pi(\boldsymbol{u}_A, d_A) = & \int_\Omega \psi\big(\boldsymbol{\epsilon}(\boldsymbol{N}_{\boldsymbol{u}_A}\boldsymbol{u}_A), N_{d_A} d_A\big) \mathrm{d}\Omega + \int_\Omega \frac{g_c}{2l}\big((N_{d_A} d_A)^2 + l^2(\nabla N_{d_A} d_A)\cdot(\nabla N_{d_A} d_A)\big)\mathrm{d}\Omega \\ & - \int_\Omega \boldsymbol{b}\cdot(\boldsymbol{N}_{\boldsymbol{u}_A}\boldsymbol{u}_A)\,\mathrm{d}\Omega - \int_{\partial\Omega} \boldsymbol{t}\cdot(\boldsymbol{N}_{\boldsymbol{u}_A}\boldsymbol{u}_A)\,\mathrm{d}\Gamma + \int_\Omega \frac{\eta}{2\Delta t}(N_{d_A} d_A - N_{d_A}(d_A)_n)^2\,\mathrm{d}\Omega, \end{aligned} \tag{14}$$

where $(d_A)_n$ is the phase-field value at node A in the beginning of the time step. The last term in the above expression represents the contribution from the viscosity term in Equation (7). For the development of the L-BFGS monolithic scheme in the following sections of this article, the residuals of the solid mechanics equation $\boldsymbol{r}_{\boldsymbol{u}}$ and the phase-field equation $\boldsymbol{r}_d$ shown in Equation (12), the diagonal block matrix $\hat{\mathbf{K}}$ defined in Equation (13), and the total energy $\Pi(\boldsymbol{u}, d)$ expressed in Equation (14) are required.

## 3 | L-BFGS MONOLITHIC SCHEME

It is well known that the energy functional in Equation (1) is non-convex.[10,11] As a result, the Newton–Raphson based monolithic approach, shown in Equation (10), may encounter convergence issues during the phase-field simulation. In this section, we propose a monolithic scheme based on the limited-memory BFGS (L-BFGS) method to solve the phase-field problem. First, we briefly review the essential steps of the classical BFGS method and discuss about its limitations in the context of the finite element simulation of the phase-field fracture propagation. Then, we present the L-BFGS monolithic approach for the phase-field simulation and the line search method based on the strong Wolfe conditions. The L-BFGS approach needs to be modified to consider the essential boundary conditions applied at the boundary nodes and the hanging-node constraints introduced during the AMR, which is necessary to alleviate the heavy computational cost associated with the phase-field method. Lastly, we combine the proposed L-BFGS monolithic scheme with an AMR technique in the Reference 15. We introduce some small modifications of the mesh refinement technique to further improve its computational efficiency during the L-BFGS iterations.

### 3.1 | BFGS method

Assume $f(\boldsymbol{x})$ is the objective function that needs to be minimized. The goal of the BFGS method is to find a solution $\boldsymbol{x}^*$ such that the residual $\boldsymbol{r} = \nabla f(\boldsymbol{x}^*) = \mathbf{0}$. Let $k$ represent the $k$th BFGS iteration, $\mathbf{B}_k$ and $\mathbf{H}_k = \mathbf{B}_k^{-1}$ represent the approximation of the Hessian $\nabla^2 f(\boldsymbol{x})$ and its inverse. A standard iteration in the BFGS method includes the following major steps[43]:

- Compute the search direction $\boldsymbol{p}_k$:

$$\mathbf{B}_k \boldsymbol{p}_k = -\boldsymbol{r}_k \quad \text{or} \quad \boldsymbol{p}_k = -\mathbf{H}_k \boldsymbol{r}_k.$$

- Update the solution vector $\boldsymbol{x}_{k+1}$:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k,$$

where $\alpha_k$ is obtained from a line search procedure.

- Define $\boldsymbol{s}_k$ and $\boldsymbol{y}_k$:

$$\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k, \quad \boldsymbol{y}_k = \boldsymbol{r}_{k+1} - \boldsymbol{r}_k.$$

- Update the approximate Hessian $\mathbf{B}_{k+1}$ or its inverse $\mathbf{H}_{k+1}$:

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k \boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}} \mathbf{B}_k}{\boldsymbol{s}_k^{\mathrm{T}} \mathbf{B}_k \boldsymbol{s}_k} + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^{\mathrm{T}}}{\boldsymbol{y}_k^{\mathrm{T}} \boldsymbol{s}_k},$$

or

$$\mathbf{H}_{k+1} = \left(\mathbf{I} - \rho_k \boldsymbol{s}_k \boldsymbol{y}_k^{\mathrm{T}}\right) \mathbf{H}_k \left(\mathbf{I} - \rho_k \boldsymbol{y}_k \boldsymbol{s}_k^{\mathrm{T}}\right) + \rho_k \boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}}, \quad \rho_k = \frac{1}{\boldsymbol{y}_k^{\mathrm{T}} \boldsymbol{s}_k}.$$

- Repeat the above steps until

$$\|\boldsymbol{r}_k\|_2 < \mathrm{tol}.$$

In the above BFGS iteration, the approximate Hessian $\mathbf{B}_k$ should be symmetric and positive definite, and the vectors $\boldsymbol{s}_k$ and $\boldsymbol{y}_k$ should satisfy the curvature condition:

$$\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{y}_k > 0. \tag{15}$$

The above curvature inequality condition is always satisfied when the objective function $f(\boldsymbol{x})$ is strongly convex. For a non-convex function such as the energy functional of the phase-field problem, this curvature condition does not always hold. Therefore, a line search procedure based on the Wolfe conditions or the strong Wolfe conditions is required to choose the step length $\alpha_k$ and guarantees that the curvature condition is satisfied. The Wolfe conditions are expressed as

$$f(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) \leq f(\boldsymbol{x}_k) + c_1 \alpha_k \boldsymbol{r}_k^{\mathrm{T}} \boldsymbol{p}_k, \quad \boldsymbol{r}(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k)^{\mathrm{T}} \boldsymbol{p}_k \geq c_2 \boldsymbol{r}_k^{\mathrm{T}} \boldsymbol{p}_k, \tag{16}$$

with $0 < c_1 < c_2 < 1$. The strong Wolfe conditions are expressed as

$$f(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) \leq f(\boldsymbol{x}_k) + c_1 \alpha_k \boldsymbol{r}_k^{\mathrm{T}} \boldsymbol{p}_k, \quad \left|\boldsymbol{r}(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k)^{\mathrm{T}} \boldsymbol{p}_k\right| \leq c_2 \left|\boldsymbol{r}_k^{\mathrm{T}} \boldsymbol{p}_k\right|, \tag{17}$$

with $0 < c_1 < c_2 < 1$.

As a type of quasi-Newton method, despite of its robustness, computational efficiency, and the superlinear rate of convergence for nonlinear optimization, the BFGS method has an obvious shortcoming when used in the finite element method. During the update of the approximate Hessian matrix or its inverse, the sparsity nature of the finite element formulation is not preserved and a fully dense matrix of $n$ by $n$ needs to be stored, where $n$ represents the number of DoFs. To see this, when $k = 0$, the initial BFGS matrix $\mathbf{B}_0$ can be assembled from the finite element procedure, for instance, using the diagonal blocks of the true Hessian as shown in Equation (13), which is indeed sparse. However, the challenges come from the two subsequent rank-one updates associated with the term $\mathbf{B}_k \boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}} \mathbf{B}_k$ and $\boldsymbol{y}_k \boldsymbol{y}_k^{\mathrm{T}}$. Recall that both $\boldsymbol{y}_k$ and $\boldsymbol{s}_k$ are column vectors with $n$ components. As a result, both $\mathbf{B}_k \boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}} \mathbf{B}_k = \mathbf{B}_k \boldsymbol{s}_k (\mathbf{B}_k \boldsymbol{s}_k)^{\mathrm{T}}$ and $\boldsymbol{y}_k \boldsymbol{y}_k^{\mathrm{T}}$ represent a rank-one update and result in a fully dense $n \times n$ matrix. The addition of a sparse matrix and two fully dense matrices results in another fully dense matrix. This limitation is too restrictive even for a mid-size finite element problem due to the required memory for the storage of the fully dense matrix. Therefore, special considerations need to be given to avoid the storage of the fully dense matrix.[25] This motivates us to introduce the *limited-memory* feature for the phase-field fracture simulations.

> *Comment* 1. To put the notations used in the BFGS update under the context of the phase-field formulation, the solution vector $\boldsymbol{x} = \{\boldsymbol{u}, d\}$ includes the displacement field $\boldsymbol{u}$ and the phase-field $d$ as the primary unknowns in the discretized finite element system. The residual $\boldsymbol{r} = \{\boldsymbol{r}_{\boldsymbol{u}}, \boldsymbol{r}_d\}$ is the gradient of the total energy functional $\Pi(\boldsymbol{u}, d)$. The discretized form of the residual is provided in Equation (12). The approximate Hessian $\mathbf{B}$ and its inverse $\mathbf{H}$ are used to replace the true Hessian (Jacobian) $\mathbf{K}$ and its inverse $\mathbf{K}^{-1}$in the linear system Equation (11).

### 3.2 | Limited-memory BFGS method with hanging-node constraints

In order to overcome the limitation of storing a fully dense matrix in the BFGS method, the limited-memory BFGS method, or the L-BFGS method, was developed.[34–36] "Limited-memory" suggests that instead of storing the fully dense $n$ by $n$ approximate Hessian matrix $\mathbf{B}$ or its inverse $\mathbf{H}$, only a few vectors of length $n$ are stored during the computation and used to calculate the search direction $\boldsymbol{p}$. According to Nocedal and Wright,[43] "the main idea of this method (L-BFGS) is to use curvature information from only the most recent iterations to construct the Hessian approximation. Curvature information from earlier iterations, which is less likely to be relevant to the actual behavior of the Hessian at the current iteration, is discarded in the interest of saving storage."

In this article, we adopt the L-BFGS procedure[43] to develop a monolithic solver for the phase-field formulation presented in Section 2. The objective function is the total energy discretized by the finite element method $\Pi(\boldsymbol{u}_A, d_A)$ shown in Equation (14). The unknowns are the displacement vector and the phase-field scalar $\boldsymbol{x} = (\boldsymbol{u}_A, d_A)^{\mathrm{T}}$ at the finite element nodes. The residuals are the residuals of the solid mechanics equation and the phase-field equation $\boldsymbol{r}(\boldsymbol{u}_A, d_A) = (\boldsymbol{r}_{\boldsymbol{u}}(\boldsymbol{u}_A, d_A), \boldsymbol{r}_d(\boldsymbol{u}_A, d_A))^{\mathrm{T}}$ given in Equation (12). At the $k$th L-BFGS iteration ($k \geq 0$), similar to the BFGS method, we define a pair of vectors $\{\boldsymbol{s}_k,\ \boldsymbol{y}_k\}$ as

$$\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k, \quad \boldsymbol{y}_k = \boldsymbol{r}_{k+1} - \boldsymbol{r}_k.$$

Let $m$ represent the maximum number of vector-pairs $\{\boldsymbol{s}_i,\ \boldsymbol{y}_i\}$ (the level of memory) allowed to be stored during the computation. We define an integer $\hat{m} = \min\{m,\ k\}$. At the beginning of the $k$th iteration, there are $\hat{m}$ sets of vector-pairs $\{\boldsymbol{s}_i,\ \boldsymbol{y}_i\},\ i = k - \hat{m},\ \ldots,\ k-1$ stored in the memory. The L-BFGS update adopts a two-loop recursive process[43] and is detailed in Algorithm 1.

In Algorithm 1, the initial inverse Hessian approximation $\mathbf{H}_k^0$ needs to be chosen for the calculation of the search direction $\boldsymbol{p}_k$. In this work, we choose the initial Hessian approximation as $\mathbf{B}_k^0 = \hat{\mathbf{K}}$, where $\hat{\mathbf{K}}$ is the diagonal block matrix defined in Equation (13). As a result

$$\mathbf{H}_k^0 = (\mathbf{B}_k^0)^{-1} = \hat{\mathbf{K}}^{-1} = \begin{bmatrix} \mathbf{K}_{\boldsymbol{uu}}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{dd}^{-1} \end{bmatrix}. \tag{18}$$

In fact, since $\hat{\mathbf{K}}$ is symmetric and positive definite, we can use the conjugate-gradient method to solve for $\boldsymbol{p}_k$ from $\mathbf{B}_k^0\ \boldsymbol{p}_k = \boldsymbol{q}$ during the intermediate step.

Once the search direction $\boldsymbol{p}_k$ is calculated from Algorithm 1 at the $k$th iteration, we still need to determine the step length along this direction, that is, we need to find a scalar $\alpha_k > 0$ such that

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k.$$

**Algorithm 1.** A two-loop recursive process to compute the search direction $\boldsymbol{p}_k = -\mathbf{H}_k \boldsymbol{r}_k$ at the $k$th L-BFGS iteration

$\boldsymbol{q} = \boldsymbol{r}_k$ and $\hat{m} = \min\{m, k\}$
**for** $i = k-1, \ldots, k-\hat{m}$ **do**
  $\rho_i = 1/(\boldsymbol{y}_i^{\mathrm{T}} \boldsymbol{s}_i)$
  $\alpha_i = \rho_i \boldsymbol{s}_i^{\mathrm{T}} \boldsymbol{q}$
  $\boldsymbol{q} \leftarrow \boldsymbol{q} - \alpha_i \boldsymbol{y}_i$
**end for**
$\boldsymbol{p}_k = \mathbf{H}_k^0 \boldsymbol{q}$ ▷ $\mathbf{H}_k^0$ is the initial inverse Hessian approximation
**for** $i = k-\hat{m}, \ldots, k-1$ **do**
  $\beta = \rho_i \boldsymbol{y}_i^{\mathrm{T}} \boldsymbol{p}_k$
  $\boldsymbol{p}_k \leftarrow \boldsymbol{p}_k + \boldsymbol{s}_i(\alpha_i - \beta)$
**end for**
$\boldsymbol{p}_k \overset{*}{=} -1$
return $\boldsymbol{p}_k$ ▷ $\boldsymbol{p}_k$ is the search direction at the current iteration

In this work, we use the strong Wolfe conditions expressed in Equation (17) to determine the step length $\alpha_k$. We can define a scalar function of $\alpha_k$ as

$$\phi(\alpha_k) = \Pi(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k).$$

Therefore, we also have

$$\phi'(\alpha_k) = \boldsymbol{r}(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k)^{\mathrm{T}} \boldsymbol{p}_k.$$

The two inequality conditions, namely the sufficient decrease condition and the curvature condition in the strong Wolfe conditions, can be recast as

$$\phi(\alpha_k) \leq \phi(0) + c_1 \alpha_k \phi'(0), \quad \left|\phi'(\alpha_k)\right| \leq c_2 \left|\phi'(0)\right|. \tag{19}$$

Note that most line search methods, including the one based on the strong Wolfe conditions, require that the search direction $\boldsymbol{p}_k$ is a descent direction, that is,

$$\phi'(0) = \boldsymbol{r}_k^{\mathrm{T}} \boldsymbol{p}_k < 0,$$

such that the step length $\alpha_k$ is positive. Since the L-BFGS update indicates that $\boldsymbol{p}_k = -\mathbf{H}_k \boldsymbol{r}_k$, where $\mathbf{H}_k$ is positive definite, the above condition holds. To find a step length $\alpha_k$ satisfying the strong Wolfe conditions, we use the implementation proposed by Moré and Thuente[44] with parameters $c_1 = 10^{-4}$ and $c_2 = 0.9$. The implementation details can also be found in Chapter 3 of the textbook by Nocedal and Wright.[43]

In the context of the finite element method with AMR, the above standard L-BFGS procedure needs to be modified to consider the applied essential boundary conditions and the so-called "hanging-node" constraints introduced during mesh refinement. Figure 1A shows an example of an adaptively refined mesh. Figure 1B shows the hanging nodes at the edges/faces shared by elements of different refinement level during mesh refinement. The DoFs associated with the hanging nodes and the nodes that are prescribed with essential boundary conditions are treated as algebraic constraints in the assembled linear system. For instance, in Figure 1B, the right element is refined by one more level than the left element, resulting in a hanging node (node 5) at the interface. Assume that both elements before the refinement use the bilinear shape functions (Q1). Then, the degree of freedom (DoF) of $x_5$ associated with the hanging node (node 5)

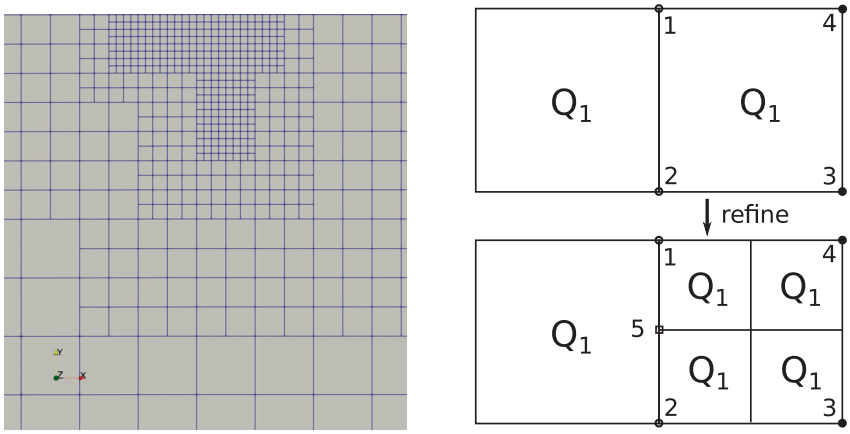


**FIGURE 1** During (A) the adaptive mesh refinement, (B) hanging nodes are generated at the edges/faces shared by elements of different refinement level. These hanging nodes, along with the nodes that are prescribed with essential boundary conditions, are treated as constrained nodes.

can be expressed by the DoFs of $x_1$ and $x_2$ associated with the neighboring nodes (nodes 1 and 2) in the following linear relationship,

$$x_5 = \frac{1}{2}x_1 + \frac{1}{2}x_2.$$

Moreover, if node 4 is prescribed with an essential boundary condition, then the DoF of $x_4$ associated with this node can be written as

$$x_4 = k_4,$$

where $k_4$ represents the prescribed value. In general, the DoF of a constrained node can be expressed in the following linear relationship

$$x_i = c_{ij}x_j + k_i,$$

where $c_{ij}$ represents the coefficients due to the hanging-node constraints and $k_i$ represents inhomogeneous constraints such as the essential boundary conditions. We can write the entire constraints in a linear system as below:

$$\boldsymbol{x} = \mathbf{C}\boldsymbol{x} + \boldsymbol{k}.$$

Particularly, the coefficient matrix $\mathbf{C}$ is idempotent, meaning that $\mathbf{C}^2 = \mathbf{C}$. The proof of the above property is shown in Appendix B. For a linear system $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ with the above set of constraints, we can instead solve the following modified linear system[45]

$$\left(\mathbf{C}^{\mathrm{T}}\mathbf{A}\mathbf{C} + \mathbf{I}_{d_c}\right)\hat{\boldsymbol{x}} = \mathbf{C}^{\mathrm{T}}(\boldsymbol{b} - \mathbf{A}\boldsymbol{k}), \tag{20}$$

and then recover the true solution $\boldsymbol{x}$ as

$$\boldsymbol{x} = \mathbf{C}\hat{\boldsymbol{x}} + \boldsymbol{k}. \tag{21}$$

In the modified linear system, the matrix $\mathbf{I}_{d_c}$ is defined as

$$(\mathbf{I}_{d_c})_{ij} = \begin{cases} 0 & \text{if} \quad i \neq j, \\ 1 & \text{if} \quad i = j \in \mathcal{T}, \\ 0 & \text{if} \quad i = j \notin \mathcal{T}, \end{cases} \tag{22}$$

where $\mathcal{T}$ represents the set of the DoFs of the constrained nodes, including the hanging nodes and the nodes prescribed with essential boundary conditions. Obviously, for a linear system without any nodal constraints, we have $\mathbf{C} = \mathbf{I}$, $\mathbf{I}_{d_c} = \mathbf{0}$, and $\boldsymbol{k} = \mathbf{0}$.

Inside a time (load) step $[t_n, t_{n+1}]$, before we enter the L-BFGS iterations, we first take care of the inhomogeneous essential boundary conditions by setting

$$\boldsymbol{x}_0 = \boldsymbol{x}_{t_n} + \Delta\boldsymbol{x}_{t_n},$$

where $\boldsymbol{x}_{t_n} = (\boldsymbol{u}_{t_n},\ d_{t_n})^{\mathrm{T}}$ represents the solution at the start of the current time step and $\Delta\boldsymbol{x}_{t_n} = (\Delta\boldsymbol{u}_{t_n},\ \Delta d_{t_n})^{\mathrm{T}}$ represents the increment of the prescribed essential boundary values, such as the displacement increment under a displacement-controlled external load. Since the inhomogeneous nodal constraints are included in the initial solution $\boldsymbol{x}_0$, we can eliminate the inhomogeneity by setting $\boldsymbol{k} = \mathbf{0}$ during the L-BFGS iterations. Recall that $m$ represents the maximum number of vector-pairs $\{\boldsymbol{s}_i,\ \boldsymbol{y}_i\}$ (the level of memory) allowed to be stored during the computation. The L-BFGS procedure at the $k$th iteration is described in Algorithm 2.

In Algorithm 2, the coefficient matrix $\mathbf{C}$ due to the nodal constraints caused by the applied essential boundary conditions and the hanging nodes from the AMR is used in two places. First, after the residual $\hat{\boldsymbol{r}}_k = (\boldsymbol{r}_{\boldsymbol{u}}^{(k)}, \boldsymbol{r}_d^{(k)})^{\mathrm{T}}$ is assembled

**Algorithm 2.** Major steps during the $k$th L-BFGS iteration

The solution $\boldsymbol{x}_k = (\boldsymbol{u}^{(k)}, d^{(k)})^{\mathrm{T}}$ at the beginning of the iteration is known.
Assemble the residual $\hat{\boldsymbol{r}}_k = (\boldsymbol{r}_{\boldsymbol{u}}^{(k)}, \boldsymbol{r}_d^{(k)})^{\mathrm{T}}$ according to Equation (12).
Incorporate the nodal constraints in the residual $\boldsymbol{r}_k = \mathbf{C}^{\mathrm{T}}\hat{\boldsymbol{r}}_k$.
Compute the search direction $\hat{\boldsymbol{p}}_k = -\mathbf{H}_k\boldsymbol{r}_k$ according to Algorithm 1.
Incorporate the nodal constraints in the search direction $\boldsymbol{p}_k = \mathbf{C}\hat{\boldsymbol{p}}_k$.
Compute $\phi(0) = \Pi(\boldsymbol{x}_k)$, $\phi'(0) = \boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{p}_k$, and a line search parameter $\alpha_k$ satisfying the strong Wolfe conditions.
Update $\Delta\boldsymbol{x}_k = \alpha_k\boldsymbol{p}_k = (\Delta\boldsymbol{u}^{(k)}, \Delta d^{(k)})^{\mathrm{T}}$ and $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta\boldsymbol{x}_k = (\boldsymbol{u}^{(k+1)}, d^{(k+1)})^{\mathrm{T}}$.
**if** $k > m$ **then**
  Remove the vector-pair $\{\boldsymbol{s}_{k-m}, \boldsymbol{y}_{k-m}\}$ from storage.
**end if**
Compute $\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$ and $\boldsymbol{y}_k = \boldsymbol{r}_{k+1} - \boldsymbol{r}_k$.
Add the vector-pair $\{\boldsymbol{s}_k, \boldsymbol{y}_k\}$ into the storage.

according to Equation (12), it is multiplied with $\mathbf{C}^{\mathrm{T}}$, that is, $\boldsymbol{r}_k = \mathbf{C}^{\mathrm{T}}\hat{\boldsymbol{r}}_k$, to remove the residuals associated with the constrained nodes. Second, after the search direction $\hat{\boldsymbol{p}}_k = -\mathbf{H}_k\boldsymbol{r}_k$ is computed, it is multiplied with $\mathbf{C}$, that is, $\boldsymbol{p}_k = \mathbf{C}\hat{\boldsymbol{p}}_k$, to ensure that the solution increments at the constrained nodes still satisfy the imposed nodal constraints. Therefore, after the $k$th iteration, the updated solution $\boldsymbol{x}_{k+1}$ still satisfies the imposed hanging node constraints and the applied essential boundary conditions. Note that in Algorithm 2, $\boldsymbol{p}_k$ is still a descent direction after considering the nodal constraints, which can be shown as the following

$$\begin{aligned}\phi'(0) = \boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{p}_k &= (\mathbf{C}^{\mathrm{T}}\hat{\boldsymbol{r}}_k)^{\mathrm{T}}(\mathbf{C}\hat{\boldsymbol{p}}_k) = (\mathbf{C}^{\mathrm{T}}\hat{\boldsymbol{r}}_k)^{\mathrm{T}}(-\mathbf{C}\mathbf{H}_k\boldsymbol{r}_k) = (\mathbf{C}^{\mathrm{T}}\hat{\boldsymbol{r}}_k)^{\mathrm{T}}(-\mathbf{C}\mathbf{H}_k\mathbf{C}^{\mathrm{T}}\hat{\boldsymbol{r}}_k)\\ &= -\hat{\boldsymbol{r}}_k^{\mathrm{T}}\mathbf{C}\mathbf{C}\mathbf{H}_k\mathbf{C}^{\mathrm{T}}\hat{\boldsymbol{r}}_k = -\hat{\boldsymbol{r}}_k^{\mathrm{T}}\mathbf{C}\mathbf{H}_k\mathbf{C}^{\mathrm{T}}\hat{\boldsymbol{r}}_k = -(\mathbf{C}^{\mathrm{T}}\hat{\boldsymbol{r}}_k)^{\mathrm{T}}\mathbf{H}_k(\mathbf{C}^{\mathrm{T}}\hat{\boldsymbol{r}}_k) < 0.\end{aligned} \tag{23}$$

In the above inequality, we use the fact that $\mathbf{C}$ is idempotent ($\mathbf{C}^2 = \mathbf{C}$) and $\mathbf{H}_k$ is positive definite. The steps listed in Algorithm 2 are repeatedly executed until certain convergence criteria are met. As demonstrated in the subsequent numerical examples, inappropriate convergence criteria may terminate the monolithic scheme prematurely and cause the under-prediction of the crack propagation during a time/load step. In this work, we adopt a set of strict convergence criteria based on the relative residuals and the relative increments of the solutions. During the first L-BFGS iteration ($k = 0$), we store the following $l_2$-norms, including $\|\boldsymbol{r}_{\boldsymbol{u}}^{(0)}\|_2$, $\|\boldsymbol{r}_d^{(0)}\|_2$, $\|\Delta\boldsymbol{u}^{(0)}\|_2$, and $\|\Delta d^{(0)}\|_2$. The convergence criteria are set as

$$\begin{aligned}&\|\boldsymbol{r}_{\boldsymbol{u}}^{(k)}\|_2 \,/\, \|\boldsymbol{r}_{\boldsymbol{u}}^{(0)}\|_2 < \mathrm{tol}_{\boldsymbol{u}}, \qquad && \|\boldsymbol{r}_d^{(k)}\|_2 \,/\, \|\boldsymbol{r}_d^{(0)}\|_2 < \mathrm{tol}_d,\\ &\|\Delta\boldsymbol{u}^{(k)}\|_2 \,/\, \|\Delta\boldsymbol{u}^{(0)}\|_2 < \mathrm{tol}_{\Delta\boldsymbol{u}}, && \|\Delta d^{(k)}\|_2 \,/\, \|\Delta d^{(0)}\|_2 < \mathrm{tol}_{\Delta d}.\end{aligned}$$

Using the above relative residuals in the convergence criteria are more strict than using the absolute residuals, since the initial residuals at the first iteration $\|(\text{»})^{(0)}\|_2$ are usually smaller than one.

*Comment* 2. The "limited-memory" feature is absolutely essential and has to be incorporated into the BFGS algorithm one way or another for any practical finite element simulations. The BFGS method used in the work by Wu et al.[18] is based on the commercial software Abaqus. In that implementation, the BFGS method also has the "limited-memory" feature. According to the Abaqus Theory Manual, to avoid the storage of the fully dense BFGS matrix, a so-called "kernel" matrix is stored and has a much smaller dimension (a 5 by 5 to 10 by 10 matrix) than the size of the discretized system. To our understanding, this "kernel" matrix has the same purpose as the vector-pairs stored in our L-BFGS method. However, there are no further details provided in the manual. This motivated us to develop the current L-BFGS monolithic scheme.

*Comment* 3. The nodal constraints due to the applied essential boundary conditions and the hanging nodes generated during the AMR are expressed as $\boldsymbol{x} = \mathbf{C}\boldsymbol{x} + \boldsymbol{k}$. These constraints can be easily rewritten into the standard form of linear constraints $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$. As a result, even though we assume that the phase-field irreversibility is incorporated through the history variable approach and there are no phase-field inequality constraints, the

minimization of the energy functional is still a constrained optimization problem due to these nodal constraints. However, since these nodal constraints are linear, we can eliminate the constrained variables using these linear equality constraints and convert the constrained problem into an unconstrained one. The multiplications of the coefficient matrix $\mathbf{C}$ on the residual and the search direction in Algorithm 2 are essentially for this purpose. For a detailed discussion about simple elimination using linear constraints, interested readers are referred to Chapter 15 in the textbook by Nocedal and Wright.[43]

*Comment* 4. During the implementation of Algorithm 2, vector-pairs $\{\boldsymbol{s}_i, \boldsymbol{y}_i\}$ need to be frequently added into or removed from memory during the L-BFGS iterations. Therefore, for better performance, a data structure such as "linked list" should be used to store these vector-pairs so that the complexity of adding or removing an element into/from the memory is constant ($\mathcal{O}(1)$).

## 3.3 | Adaptive mesh refinement

The phase-field method for crack propagation simulation is intrinsically expensive, because the phase-field length-scale $l$ needs to be resolved within the crack region using a mesh size $h < l$. When the crack path is unknown, an AMR technique is required to confine the usage of the highly refined mesh only within the crack region and avoid the spatial discretization of the full computational domain with such a small mesh size $h < l$. In this work, we adopt a so-called "predictor-corrector" AMR technique in the Reference 15 and integrate it into our L-BFGS monolithic scheme. Comparing with the original technique, we introduce several small modifications in the AMR procedure to improve the convergence behavior of the L-BFGS iterations. Also, from the AMR technique, the coefficients of hanging-node constraints $\mathbf{C}$ are generated and provided to the L-BFGS iterations shown in Algorithm 2.

In order to better illustrate the AMR technique, we introduce the following notations. Let $(h/l)_{\max} < 1.0$ be the maximum allowed ratio between the mesh size $h$ and the phase-field length-scale $l$. Let $d_{\text{threshold}}$ represent a phase-field threshold value. These two parameters are provided as input parameters. An element is refined, for example, see Figure 1, if the following two conditions are both true:

- The element contains a node that has the phase-field nodal value $d > d_{\text{threshold}}$;
- The ratio between the element mesh size and the phase-field length-scale $h/l$ is larger than the prescribed ratio $(h/l)_{\max}$.

This set of refinement conditions ensure that the phase-field length-scale within the crack region defined by $d > d_{\text{threshold}}$ is always resolved during the spatial discretization. For a finite element mesh, the element mesh size $h$ is calculated as

$$h = \begin{cases} \sqrt{A_e} & \text{for 2D,} \\ \sqrt[3]{V_e} & \text{for 3D,} \end{cases}$$

where $A_e$ represents the element area in a 2D problem, and $V_e$ represents the element volume in a 3D problem.

During a time (load) step $[t_n, t_{n+1}]$, we adopt a two-level mesh refinement strategy based on the above set of refinement conditions. Let $\mathcal{T}^{(i,\,j)}$ represent a mesh discretization, where the superscript $i$ represents the outer-level mesh refinement iteration, and the superscript $j$ represents the inner-level mesh refinement iteration. For examples of $\mathcal{T}^{(i,\,j)}$, see Figure 2. On a specific mesh discretization $\mathcal{T}^{(i,\,j)}$, the coefficients of hanging-node constraints $\mathbf{C}^{(i,j)}$ can be extracted for the proposed L-BFGS monolithic scheme presented in Section 3.2. Let $\boldsymbol{x}_n^{(i,\,j)}$ represent the solution at the beginning of the time step on the mesh $\mathcal{T}^{(i,\,j)}$, and let $\boldsymbol{x}_{n+1}^{(i,\,j)}$ represent the solution at the end of the time step on the mesh $\mathcal{T}^{(i,\,j)}$. During the two-level mesh refinement strategy, the following two operations are frequently performed:

- On the mesh $\mathcal{T}^{(i,\,0)}$, the solution $\boldsymbol{x}_{n+1}^{(i,\,0)}$ can be solved from $\boldsymbol{x}_n^{(i,\,0)}$ using the proposed L-BFGS monolithic scheme;
- During the inner-level mesh refinement iteration, the solutions $\boldsymbol{x}_n^{(i,\,j+1)}$ and $\boldsymbol{x}_{n+1}^{(i,\,j+1)}$ on the mesh $\mathcal{T}^{(i,\,j+1)}$ can be obtained through interpolation of the solutions $\boldsymbol{x}_n^{(i,\,j)}$ and $\boldsymbol{x}_{n+1}^{(i,\,j)}$ on the mesh $\mathcal{T}^{(i,\,j)}$.

The $i$th outer-level mesh refinement iteration involves multiple inner-level mesh refinement iterations. And the inner-level mesh refinement steps are summarized in Algorithm 3. Through the above steps inside the inner-level mesh

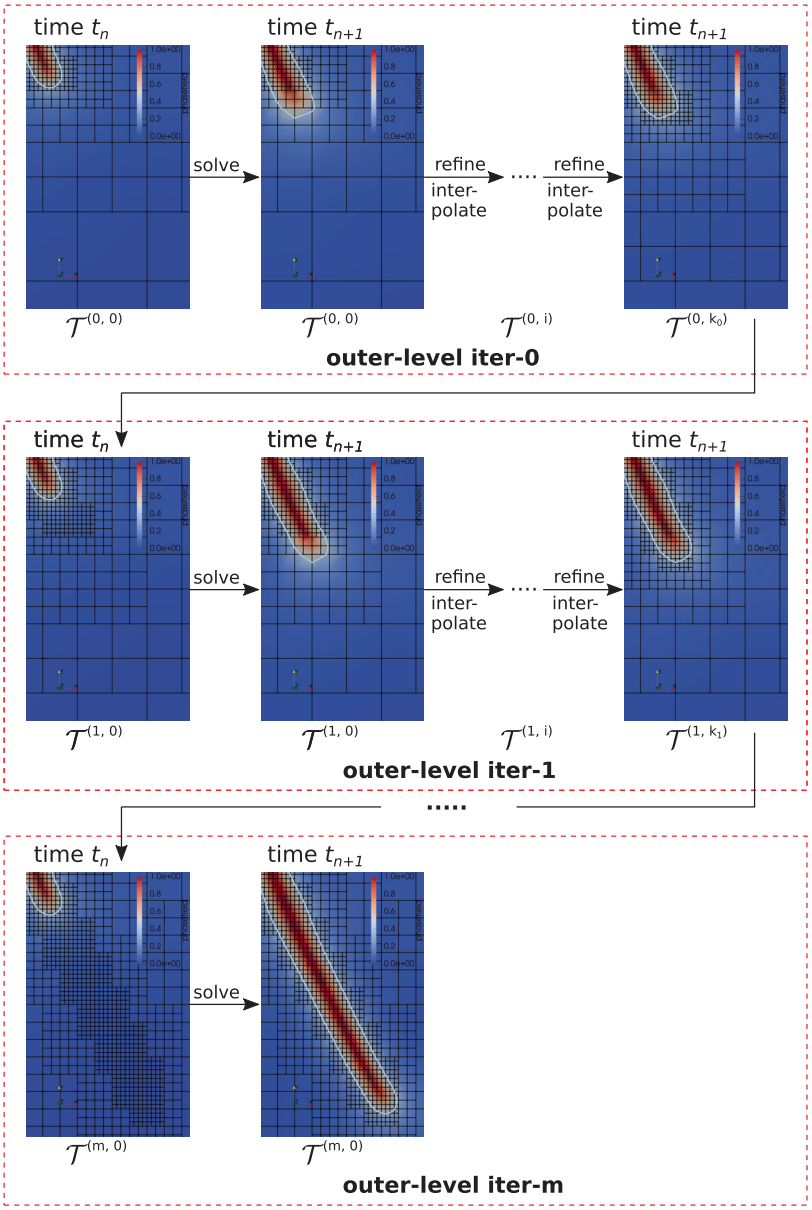


**FIGURE 2** Two-level mesh refinement strategy based on the work by Heister et al.[15] Inside the $i$th outer-level mesh refinement iteration, starting from the mesh $\mathcal{T}^{(i,\,0)}$, a series of meshes $\left\{\mathcal{T}^{(i,\,1)},\ \mathcal{T}^{(i,\,2)},\ \ldots,\mathcal{T}^{(i,\,k_i)}\right\}$ are generated during the inner-level mesh refinements as shown in a red dashed line box. The last refined mesh $\mathcal{T}^{(i,\,k_i)}$ provides the initial mesh for the next outer-level calculation. The region inside the contour line represents the area satisfying the phase-field threshold $d > d_{\text{threshold}}$. In the mesh $\mathcal{T}^{(i,\,k_i)}$, no element satisfies the refinement conditions. The outer-level iteration stops when no inner-loop mesh refinement is required, that is, in the mesh $\mathcal{T}^{(m,\,0)}$, no element satisfies the refinement conditions.

**Algorithm 3.** Inner-level mesh refinement steps at the $i$th outer-level.

Solve for $\boldsymbol{x}_{n+1}^{(i,0)}$ from $\boldsymbol{x}_n^{(i,0)}$ using the L-BFGS monolithic scheme on mesh $\mathcal{T}^{(i,0)}$.
Set $j=0$.
**while** Any element in $\mathcal{T}^{(i,j)}$ based on $\boldsymbol{x}_{n+1}^{(i,j)}$ satisfies refinement conditions **do**
  Refine $\mathcal{T}^{(i,j)}$ to obtain $\mathcal{T}^{(i,j+1)}$.
  Interpolate $\boldsymbol{x}_n^{(i,j)}$ and $\boldsymbol{x}_{n+1}^{(i,j)}$ on mesh $\mathcal{T}^{(i,j)}$ to obtain $\boldsymbol{x}_n^{(i,j+1)}$ and $\boldsymbol{x}_{n+1}^{(i,j+1)}$ on mesh $\mathcal{T}^{(i,j+1)}$.
  Set $j=j+1$.
**end while**

**Algorithm 4.** Outer-level mesh refinement steps.

```
set i = 0.
Obtain {T^(i,1), T^(i,2), ..., T^(i,k_i)} using the inner-level refinement Algorithm 3.
while k_i > 0 do                ▷ iteration stops when no refinement needed in inner-level
   Set T^(i+1,0) = T^(i,k_i) and x_n^(i+1,0) = x_n^(i,k_i).
   Obtain {T^(i+1,1), T^(i+1,2), ..., T^(i+1,k_{i+1})} using the inner-level refinement Algorithm 3.
   Set i = i + 1.
end while
```

refinements, starting from the mesh $\mathcal{T}^{(i,\,0)}$, a series of meshes $\left\{\mathcal{T}^{(i,\,1)},\ \mathcal{T}^{(i,\,2)},\ \ldots,\mathcal{T}^{(i,\,k_i)}\right\}$ are generated, and the mesh $\mathcal{T}^{(i,\,j+1)}$ is one-more level refined than the mesh $\mathcal{T}^{(i,\,j)}$. Also, the last mesh $\mathcal{T}^{(i,\,k_i)}$ inside the inner-level mesh refinements provides the initial mesh $\mathcal{T}^{(i+1,\,0)}$ for the next outer-level calculation. In the original AMR strategy,[15] $k_i \equiv 1$, that is, only one mesh refinement is performed to generate $\mathcal{T}^{(i,\,k_i)}$ during the $i$th outer-level iteration. Here, we ensure that in the last mesh $\mathcal{T}^{(i,\,k_i)}$, based on the solution $\boldsymbol{x}_{n+1}^{(i,\,k_i)}$, all the elements within the crack region ($d > d_{\text{threshold}}$) are sufficiently refined such that $h/l < (h/l)_{\max}$. When the phase-field solution does not change significantly during the time (load) step, one mesh refinement within the $i$th outer-level iteration would be sufficient to ensure that $\mathcal{T}^{(i,\,1)}$ does not contain elements satisfying the refinement conditions. In this case, the modified AMR strategy recovers the original one. When the change of the phase-field solution is large during the time step, multiple mesh refinements ($k_i > 1$) might be needed to ensure that no elements satisfy the refinement conditions in $\mathcal{T}^{(i,\,k_i)}$.

The outer-level mesh refinement iterations are summarized in Algorithm 4. The number of L-BFGS iterations required to achieve convergence during a time (load) step is impacted by the quality of the initial guess. To facilitate the L-BFGS convergence, when we solve for $\boldsymbol{x}_{n+1}^{(i+1,\,0)}$ from $\boldsymbol{x}_{n}^{(i+1,\,0)}$ on the mesh $\mathcal{T}^{(i+1,\,0)}$ using the L-BFGS monolithic scheme, instead of setting the initial solution increment as $\Delta\boldsymbol{x}_{n}^{(0),\,(i+1,\,0)} = \boldsymbol{0}$, we use the interpolated solution $\boldsymbol{x}_{n+1}^{(i,\,k_i)}$ on the mesh $\mathcal{T}^{(i,\,k_i)} = \mathcal{T}^{(i+1,\,0)}$ and set $\Delta\boldsymbol{x}_{n}^{(0),\,(i+1,\,0)} = \boldsymbol{x}_{n+1}^{(i,\,k_i)} - \boldsymbol{x}_{n}^{(i,\,k_i)}$ as the initial solution increment. The whole AMR strategy is illustrated in Figure 2 and works for both 2D and 3D problems.

## 4 | NUMERICAL EXAMPLES

In this section, we provide three 2D numerical examples and one 3D numerical example to demonstrate the capabilities of the proposed L-BFGS monolithic scheme combined with the AMR technique. The 2D examples, including the simple tension test, the simple shear test, and the three-point bending test, are widely adopted in the literature related to the phase-field formulation of fracture propagation.[12,13,15] Moreover, the fully 3D numerical example simulates the crack propagation inside a cubic box containing a rigid sphere inclusion. In these numerical examples, we aim to demonstrate the following aspects of the proposed monolithic scheme, including the accuracy in the sense of equation residuals and convergence criteria, the robustness during a critical load step in which crack propagates rapidly in a brutal manner, the efficiency about both the memory consumption and the wall-clock time, and the dimension-independence such that it works for both the 2D and 3D cases. The proposed L-BFGS monolithic scheme is implemented in deal.II,[42] which is an open-source C++ finite element library. This library has several built-in functions that could facilitate the implementation of the AMR technique, the solution transfer between meshes before and after refinement, and the enforcement of the hanging-node constraints during the L-BFGS process. All the simulations are performed on a workstation laptop with 16 CPUs (8 cores, 2 threads per core) of 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30 GHz. In each simulation, the Threading Building Blocks (TBB) library is used during the global stiffness matrix and the right-hand side calculations.

### 4.1 | Simple tension test

In the first example, we use the proposed phase-field monolithic scheme to simulate the fracture propagation inside a unit square sample (1 mm edge length) that has a preexisting crack under a tensile load. The boundary conditions are applied such that the bottom left corner of the domain is fixed in both $x$- and $y$-directions. The bottom edge is fixed in the

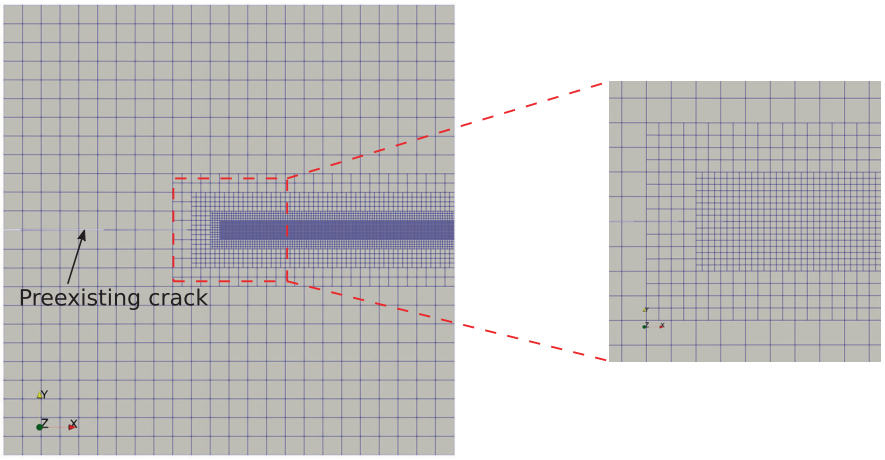


**FIGURE 3** Crack propagation under simple tension inside a unit square (1.0 mm edge length) with a preexisting crack of 0.5 mm long. The preexisting crack is from the center of the left edge to the center of the domain. The boundary conditions are applied such that the bottom left corner of the domain is fixed in both $x$- and $y$-directions. The bottom edge is fixed in the $y$-direction, and the top edge is applied with a displacement-controlled load $u_y$ in the $y$-direction. The mesh is pre-refined along the anticipated path that is known a priori such that the mesh size $h = 0.00265$ mm ($h/l \approx 1/3$, $l = 0.0075$ mm).

$y$-direction, and the top edge is applied with a displacement-controlled load $u_y$ in the $y$-direction. The material parameters use the same values as the ones reported in the Reference 12, including the Lamé parameters $\lambda = 121.15$ kN/mm$^2$ and $\mu = 80.77$ kN/mm$^2$, the critical energy release rate $g_c = 2.7 \times 10^{-3}$ kN/mm, the viscosity parameter $\eta = 0.0$ kN s/mm$^2$, and the small positive parameter $k = 0.0$ in the degradation function $g(d)$. The phase-field length-scale parameter is chosen as $l = 0.0075$ mm. Since the crack propagation path is known a priori, which extends the preexisting crack until it reaches the right edge of the sample, the mesh is pre-refined around the anticipated propagation path as shown in Figure 3 such that the mesh size is $h = 0.00265$ mm and the ratio between the mesh size and the phase-field length-scale is $h/l \approx 1/3$. When the displacement-controlled load applied at the top edge is smaller than 0.005 mm, that is, $u_y \in [0.0, 0.005]$, the phase-field inside the domain is small and the material exhibits a linear elastic response. Therefore, a large load increment $\Delta u_y = 10^{-3}$ is applied inside each load step. When the displacement load $u_y > 0.005$, the load increment is reduced to $\Delta u_y = 10^{-4}$ to investigate the crack propagation process. A set of strict convergence criteria are adopted such that

$$\begin{aligned} &\|\boldsymbol{r}_{\boldsymbol{u}}^{(i)}\|_2/\|\boldsymbol{r}_{\boldsymbol{u}}^{(0)}\|_2 < \mathrm{tol}_{\boldsymbol{u}} = 10^{-9}, \qquad &&\|\boldsymbol{r}_d^{(i)}\|_2/\|\boldsymbol{r}_d^{(0)}\|_2 < \mathrm{tol}_d = 10^{-9}, \\ &\|\Delta\boldsymbol{u}^{(i)}\|_2/\|\Delta\boldsymbol{u}^{(0)}\|_2 < \mathrm{tol}_{\Delta\boldsymbol{u}} = 10^{-9}, \qquad &&\|\Delta d^{(i)}\|_2/\|\Delta d^{(0)}\|_2 < \mathrm{tol}_{\Delta d} = 10^{-9}. \end{aligned} \tag{24}$$

Using the above relative residuals in the convergence criteria are more strict than using the absolute residuals, since the initial residuals at the first iteration $\|(\text{»})^{(0)}\|_2$ are smaller than one.

Figure 4 shows the phase-field distributions inside the domain when the top-edge displacement is $u_y = 5.9 \times 10^{-3}$ mm and $u_y = 6.0 \times 10^{-3}$ mm, respectively. Within this load step, the crack grows brutally from the tip of the preexisting crack (the center of the domain) to the right edge of the domain. The crack path follows the pre-refined mesh as expected. Figure 5 shows the L-BFGS convergence history of the total energy $E(\boldsymbol{u}, d)$ of the system and the $l_2$-norm of the residuals $\boldsymbol{r}_{\boldsymbol{u}}(\boldsymbol{u}, d)$ and $\boldsymbol{r}_d(\boldsymbol{u}, d)$ defined in Equation (12) at $u_y = 6.0 \times 10^{-3}$ mm. During this critical step, the L-BFGS monolithic scheme requires 777 iterations to meet the strict convergence criteria based on the relative residuals of the displacement and the phase-field equations as well as the solution increments. As shown in Figure 5A, the total energy of the system consistently decreases, which is expected due to the line search method based on the strong Wolfe conditions. The total energy reaches a plateau and barely changes after the 700th iteration. However, as shown in Figure 5B, both of the displacement equation residual $\boldsymbol{r}_{\boldsymbol{u}}(\boldsymbol{u}, d)$ and the phase-field equation residual $\boldsymbol{r}_d(\boldsymbol{u}, d)$ are still relatively large ($>10^{-6}$). Therefore, more iterations are required to drive the residuals to the prescribed tolerance. Figure 5C,D show the phase-field distribution at the 600th and 700th iterations, respectively. Without properly setting the convergence criteria, the crack propagation would be under-predicted.

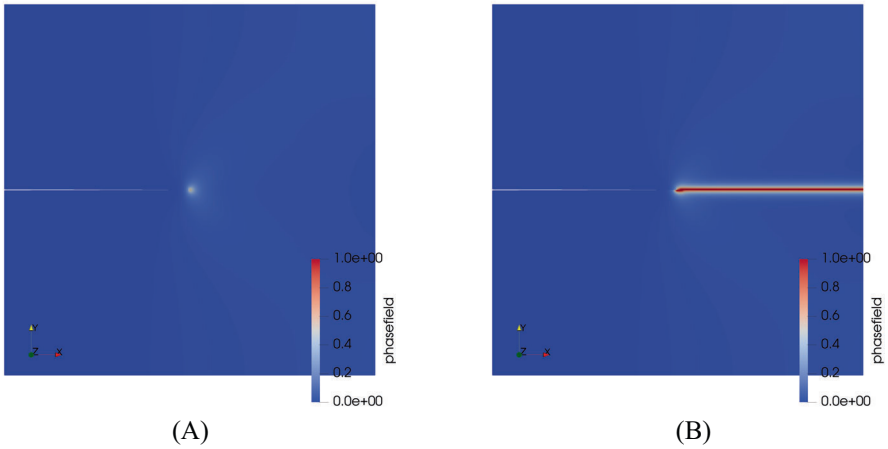


**FIGURE 4** Phase field distribution when the top edge displacement is (A) $u_y = 5.9 \times 10^{-3}$ mm and (B) $u_y = 6.0 \times 10^{-3}$ mm (critical step). In this case, the viscosity parameter is $\eta = 0.0$ and the displacement increment inside a time step is $\Delta u_y = 1.0 \times 10^{-4}$ mm. Once initiated at the preexisting crack tip, the crack brutally propagates to the right edge of the domain along the pre-refined mesh.

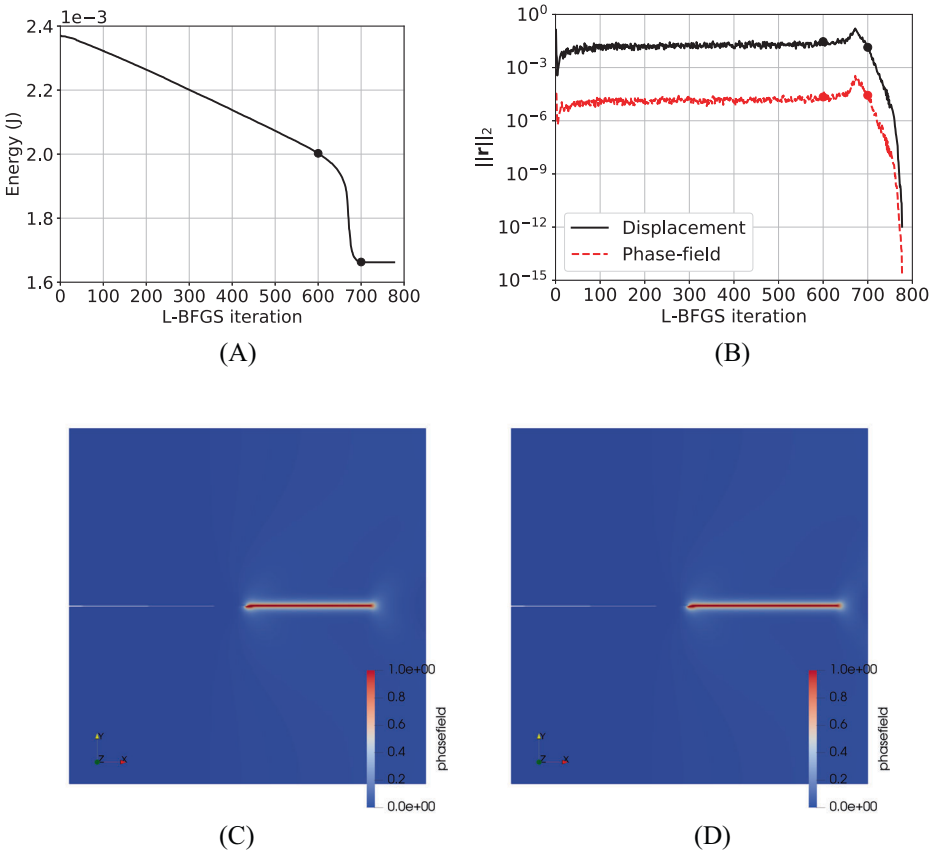


**FIGURE 5** L-BFGS convergence history at the critical step of the simple tension test: (A) The total energy of the system $E(\boldsymbol{u}, d)$, (B) the $l_2$-norm of the absolute residuals of the displacement equation $\boldsymbol{r_u}(\boldsymbol{u}, d)$ and the phase-field equation $\boldsymbol{r_d}(\boldsymbol{u}, d)$, the phase-field distribution at the (C) 600th iteration and (D) 700th iteration. Without setting the appropriate residual-based convergence criteria, the crack propagation would be under predicted.

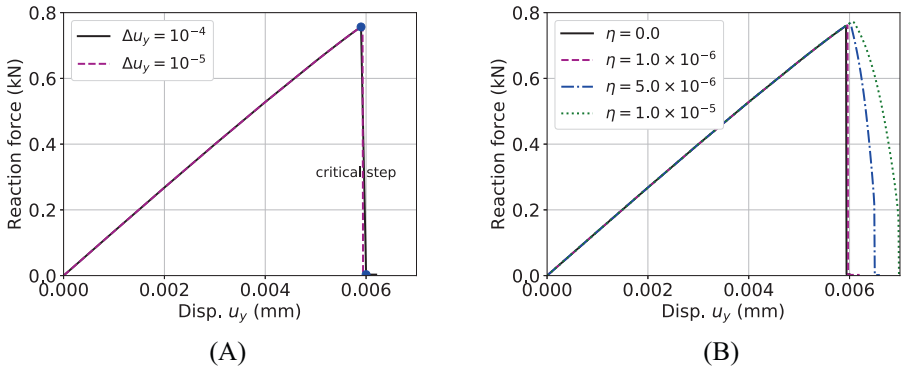


**FIGURE 6** Impact of the load step size and the viscosity parameter $\eta$ on the force-displacement relationship for the unit square sample under the tension test. (A) When $\eta = 0.0$, regardless of the load increment size, the sample loses load bearing capacity during the critical step immediately after the peak load. (B) When $\eta$ increases, the sample retains some load bearing capacity after the peak load.

Figure 6 further demonstrates the impact of the load increment size and the viscosity parameter on the crack propagation behavior. As shown in Figure 6A, when the viscosity parameter is $\eta = 0.0$, the crack propagates brutally regardless of the adopted load increment size ($\Delta u_y = 10^{-4}$ or $\Delta u_y = 10^{-5}$) and the sample losses the load bearing capacity immediately after the peak load. When a small viscosity parameter $\eta$ is introduced into the phase-field formulation, as shown in Figure 6B, the crack propagation is regularized and the sample retains some load bearing capacity after the peak load, depending on the magnitude of the viscosity parameter $\eta$.

## 4.2 | Simple shear test

In the second example, we use the proposed phase-field monolithic scheme to simulate the fracture propagation inside the same unit square sample as the first example that has a preexisting crack under a shear load. The boundary conditions are applied such that the bottom edge is fixed in both $x$- and $y$-directions. The top edge is fixed in the $y$-direction and is applied with a displacement-controlled load $u_x$ in the $x$-direction. The material parameters use the same values as the ones reported in the Reference 12, including the Lamé parameters $\lambda = 121.15$ kN/mm$^2$ and $\mu = 80.77$ kN/mm$^2$, the critical energy release rate $g_c = 2.7 \times 10^{-3}$ kN/mm, the viscosity parameter $\eta = 0.0$ kN s/mm$^2$, and the small positive parameter $k = 0.0$ in the degradation function $g(d)$. The phase-field length-scale parameter is chosen as $l = 0.0075$ mm. Unlike the previous simple tension test where the crack propagation path is known a priori, in this case the exact crack path is unknown beforehand. Therefore, two different meshing strategies are adopted, as shown in Figure 7. In the first strategy as shown in Figure 7A, the mesh is pre-refined in a quarter of the domain where crack is anticipated to grow. The mesh size is chosen as $h = 0.0026$ mm such that the ratio between the mesh size and the phase-field length-scale is $h/l \approx 1/3$. The total number of DoFs is 135,849. In the second meshing strategy as shown in Figure 7B, the mesh is only pre-refined such that $h/l \approx 1/3$ around the preexisting crack tip region where the crack is expected to initiate. The total number of DoFs is 3792 for this initial mesh. Subsequently, the AMR technique is adopted as the crack propagates. In the AMR technique, the phase-field refinement threshold is $d_{\text{threshold}} = 0.4$, and the maximum allowed ratio between the mesh size and the phase-field length-scale is $(h/l)_{\max} = 0.5$, which guarantees that there are at least three elements within the phase-field length-scale. The displacement-controlled load increment inside each step is adopted as $\Delta u_x = 10^{-3}$ mm, and the convergence of the L-BFGS iterations adopts the same strict criteria based on the relative residuals and the solution increments as shown in Equation (24).

Figure 8 compares the phase-field distributions at various loading stages obtained from the pre-refined mesh and the adaptively refined mesh. These two meshing strategies yield identical phase-field distribution results. Yet, the computational cost of the AMR-based approach is significantly lower, since the number of DoFs of the mesh in which the crack is fully developed is only 19,002 compared with the number of DoFs of 135,849 in the pre-refined meshing strategy. Figure 9A compares the impact of the load increment size $\Delta u_x$ on the force-displacement relationship of the sample under the shear test using the pre-refined mesh. Regardless of the adopted load increment size $\Delta u_x = 10^{-3}$ or $\Delta u_x = 10^{-4}$, the proposed monolithic scheme obtains similar results. Figure 9B demonstrates the impact of the viscosity parameter $\eta$

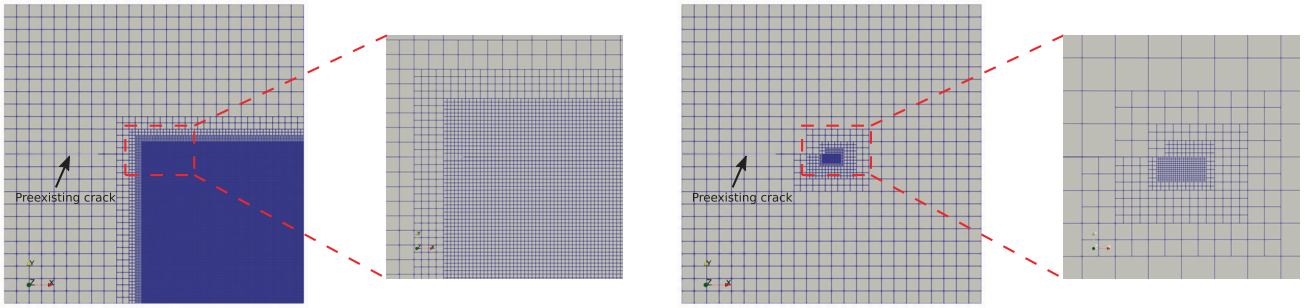


**FIGURE 7** Crack propagation under simple shear inside a unit square (1.0 mm edge length) with a preexisting crack of 0.5 mm long. The preexisting crack is from the center of the left edge to the center of the domain. The boundary conditions are applied such that the bottom edge is fixed in both $x$- and $y$-directions. The top edge is fixed in the $y$-direction and is applied with a displacement-controlled load $u_x$ in the $x$-direction. (A) The mesh is pre-refined in the region where crack propagation is anticipated such that the mesh size $h = 0.0026$ mm ($h/l \approx 1/3$, $l = 0.0075$ mm). The number of degrees of freedom (DoFs) is 135,849. (B) The mesh is only pre-refined near the crack tip such that the mesh size $h = 0.0026$ mm ($h/l \approx 1/3$, $l = 0.0075$ mm). The number of DoFs for this initial mesh is 3792. (A) Pre-refined mesh (135,849 DoFs); (B) initial mesh with adaptive refinement (3792 DoFs).

on the material response obtained from the adaptively refined mesh. As expected, the material sample exhibits a larger load bearing capacity when the phase-field equation is regularized with a larger magnitude of viscosity parameter.

Figure 10 reports the convergence history of the total energy $E(\boldsymbol{u}, d)$ of the material sample and the $l_2$-norm of the residuals $\boldsymbol{r}_{\boldsymbol{u}}(\boldsymbol{u}, d)$ and $\boldsymbol{r}_d(\boldsymbol{u}, d)$ in the simple shear test. Using the pre-refined mesh, the critical step (the loading step that immediately follows the peak load) at $u_x = 0.010$ mm requires 251 L-BFGS iterations. Similar to the observations from the previous simple test problem, the total energy consistently decreases due to the applied line search method based on the strong Wolfe conditions. After decreasing to a plateau at the 150th iteration, the total energy barely changes, as shown in Figure 10A. However, at the 150th iteration, the absolute residuals $\boldsymbol{r}_{\boldsymbol{u}}(\boldsymbol{u}, d)$ and $\boldsymbol{r}_d(\boldsymbol{u}, d)$ are still large ($\|\boldsymbol{r}_{\boldsymbol{u}}(\boldsymbol{u}, d)\|_2 \approx 10^{-3}$ and $\|\boldsymbol{r}_d(\boldsymbol{u}, d)\|_2 \approx 10^{-6}$). Therefore, more L-BFGS iterations are required to reduce the residuals to the prescribed tolerance shown in Equation (24). Figure 11 reports the growth of the DoF and the required L-BFGS iterations when the AMR strategy is adopted for the simple shear test. Since the mesh is gradually refined as the fracture propagates, the required L-BFGS iterations are significantly smaller than the ones required when the pre-refined mesh as shown in Figure 7A is used. The wall-clock time needed for simulations based on both the pre-refined mesh and the adaptively refined mesh is reported in Table 1 at the end of this section.

One of the parameters required in the L-BFGS algorithm is the level of memory $m$ that indicates the number of $\{\boldsymbol{s}, \boldsymbol{y}\}$-pairs used to update the approximation of the inverse Hessian $\mathbf{H}$. Figure 12 compares the impact of selected parameter $m$ on the required L-BFGS iterations of the simple shear test using the pre-refined mesh. When $m = 0$, it means that the update of the inverse Hessian approximation does not include any curvature information from the previous iterations, $\mathbf{H} = \mathbf{H}_0 = \hat{\mathbf{K}}^{-1}$. In this case, the L-BFGS iterations do not converge after several load steps. As the level of memory $m$ increases, the required number of L-BFGS iterations for convergence decreases. However, in this simple shear example, there are no significant differences regarding the required iterations for convergence when $m = 40$ or $m = 100$. On the other hand, since more $\{\boldsymbol{s}, \boldsymbol{y}\}$-pairs are included to update the inverse Hessian approximation for a larger level of memory, the computational cost slightly increases regarding both the memory consumption and the CPU time.

## 4.3 | Three-point bending test

In the third example, we use the proposed monolithic scheme to simulate the crack propagation inside a beam that has a V-notch under the three-point bending test. Figure 13A shows the dimensions of the beam and the location of the preexisting V-notch. The boundary conditions are applied such that the bottom left corner is fixed in both $x$- and $y$-directions, and the bottom right corner is fixed in $y$-direction. A vertical displacement-controlled load $u_y$ is applied at the center of the top edge. The material parameters use the same values as the ones in the Reference 13, including the Lamé parameters $\lambda = 12.0$ kN/mm$^2$ and $\mu = 8.0$ kN/mm$^2$, the critical energy release rate $g_c = 5.0 \times 10^{-4}$ kN/mm, the

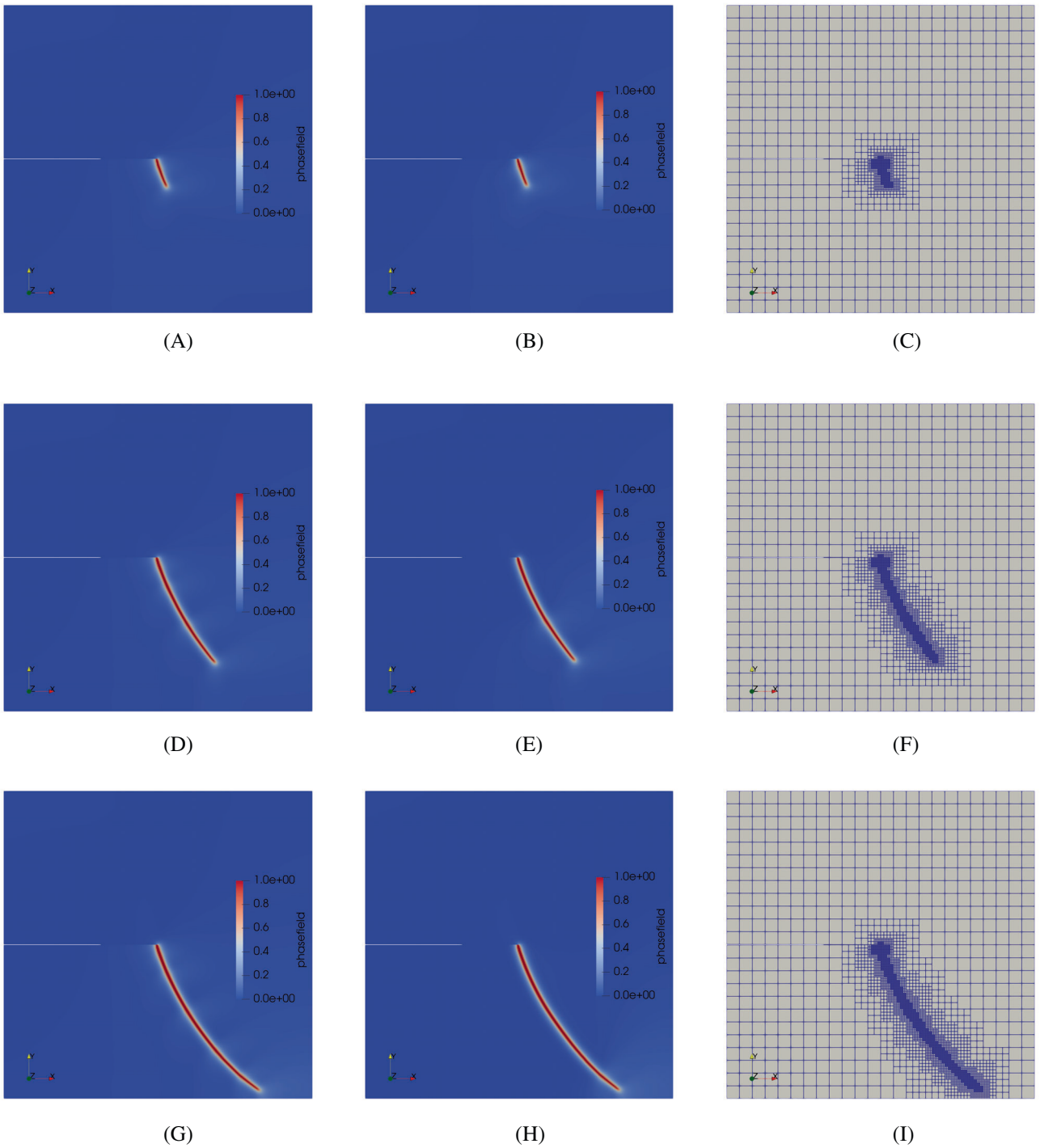


**FIGURE 8** Phase-field distributions of the simple shear test using the pre-refined mesh and the adaptively refined mesh at different displacement-controlled load (unit: mm). (A), (D), (G) Show the results obtained from the pre-refined mesh. (B), (E), (H) Show the results using the adaptive mesh refinement (AMR). (C), (F), (I) Show the corresponding adaptively refined meshes. (A) $u_x = 0.01$ (pre-refine); (B) $u_x = 0.01$ (AMR); (C) mesh (AMR); (D) $u_x = 0.012$ (pre-refine); (E) $u_x = 0.012$ (AMR); (F) mesh (AMR); (G) $u_x = 0.014$ (pre-refine); (H) $u_x = 0.014$ (AMR); (I) mesh (AMR).

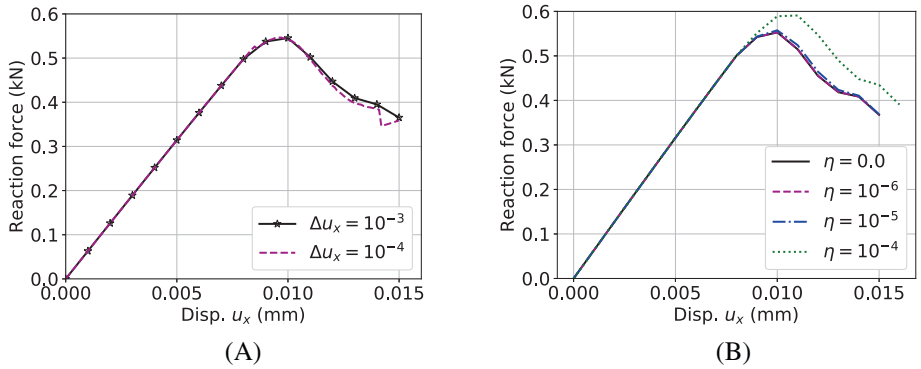


**FIGURE 9** The load-displacement relationships of the simple shear test. (A) The impact of the load step size $\Delta u_x$ using the pre-refined mesh. (B) The impact of the viscosity parameter $\eta$ using the adaptive mesh refinement (AMR).

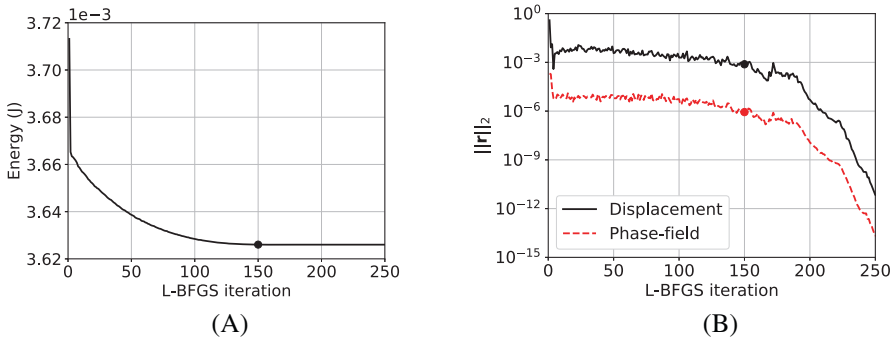


**FIGURE 10** L-BFGS convergence history (A) the total energy and (B) the residuals of the displacement and the phase-field equations of the simple shear test at $u_x = 0.01$ mm obtained from the pre-refined mesh.

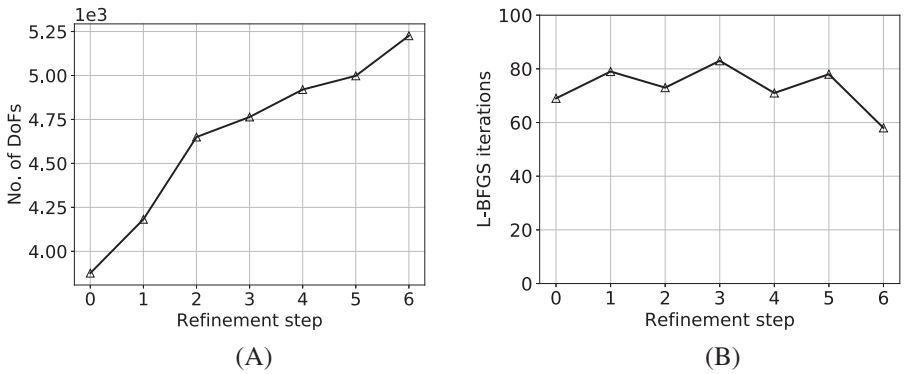


**FIGURE 11** Adaptive mesh refinement at the critical step ($u_x = 0.01$ mm) of the simple shear test: (A) The growth of the number of degrees of freedom (DoFs) and (B) the L-BFGS iterations required for convergence of each adaptive mesh refinement step.

viscosity parameter $\eta = 0.0$ kN s/mm$^2$, and the small positive parameter $k = 0.0$ in the degradation function $g(d)$. The phase-field length-scale parameter is chosen as $l = 0.0075$ mm. Two different meshing strategies are adopted. In the first strategy, as shown in Figure 13B, the mesh is pre-refined along the anticipated crack propagation path. The mesh size is chosen as $h = 0.0031$ mm such that the ratio between the mesh size and the phase-field length-scale is $h/l \approx 0.4$. The total number of DoFs is 54,957. In the second meshing strategy as shown in Figure 13C, the mesh is only pre-refined such that $h/l \approx 0.4$ around the tip region of the V-notch where the crack is expected to initiate. The total number of DoFs is 14,862 for this initial mesh. Subsequently, the AMR technique is adopted as the crack propagates. In the AMR

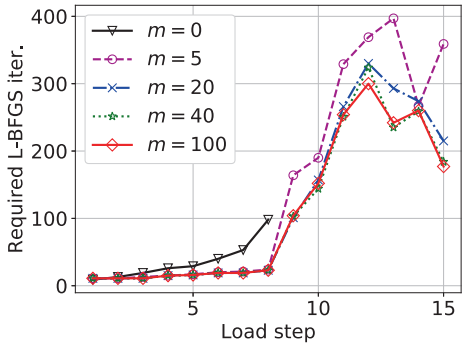


**FIGURE 12** Impact of levels of memory $m$ on the required number of L-BFGS iterations for each load step of the simple shear test using a load increment of $\Delta u_x = 10^{-3}$ mm and the pre-refined mesh.

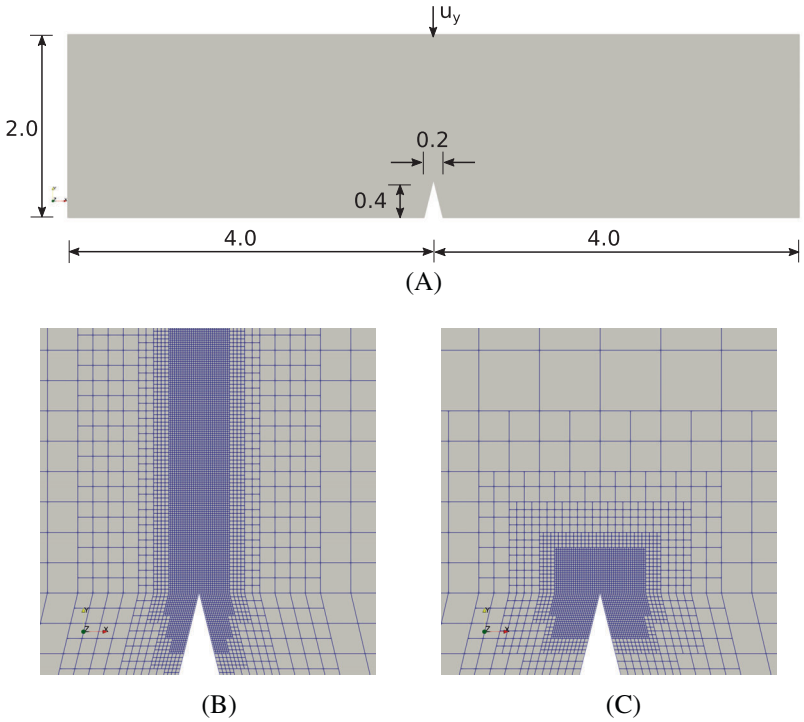


**FIGURE 13** Three-point bending test: (A) the sample geometry. Boundary conditions are applied such that the bottom left corner is fixed in both $x$- and $y$-directions, and the bottom right corner is fixed in $y$-direction. A vertical displacement-controlled load $u_y$ is applied at the center of the top edge. (B) The mesh is pre-refined along the anticipated crack propagation path (DoFs 54,957) and the mesh size is $h = 0.0031$ mm ($h/l \approx 0.4$, $l = 0.0075$ mm). (C) The initial mesh with adaptive mesh refinement (DoFs 14,862).

technique, the phase-field refinement threshold is $d_{\text{threshold}} = 0.2$, and the maximum allowed ratio between the mesh size and the phase-field length-scale is $(h/l)_{\max} = 0.5$, which guarantees that there are at least three elements within the phase-field length-scale. The displacement-controlled load increment inside each step is adopted as $\Delta u_y = 10^{-3}$ mm, and the convergence of the L-BFGS iterations adopts the following set of criteria based on the absolute residuals and the solution increments,

$$\begin{aligned} \|\boldsymbol{r}_{\boldsymbol{u}}^{(i)}\|_2 < \text{tol}_{\boldsymbol{u}} = 10^{-9}, \qquad & \|\boldsymbol{r}_{d}^{(i)}\|_2 < \text{tol}_{d} = 10^{-9}, \\ \|\Delta\boldsymbol{u}^{(i)}\|_2 < \text{tol}_{\Delta\boldsymbol{u}} = 10^{-9}, \qquad & \|\Delta d^{(i)}\|_2 < \text{tol}_{\Delta d} = 10^{-9}. \end{aligned} \tag{25}$$

Figure 14 compares the phase-field distributions at various load steps obtained from the pre-refined mesh and the adaptively refined mesh, respectively. These two meshing strategies based on the proposed L-BFGS monolithic scheme

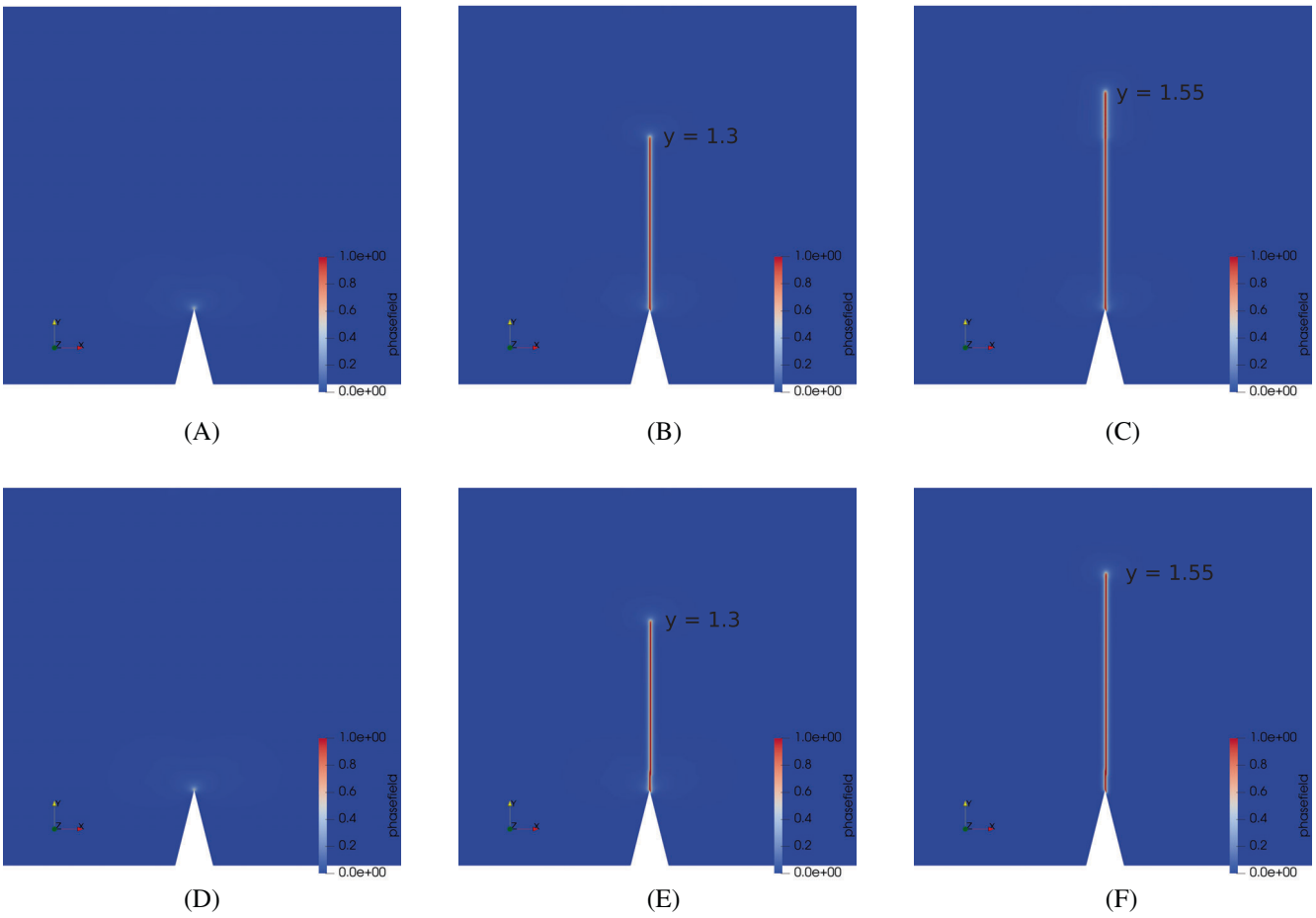


**FIGURE 14** Comparison of the phase-field distributions at different displacement-controlled loading magnitudes $u_y$ (unit: mm). (A)–(C) are obtained using the pre-refined mesh, and (D)–(F) are obtained using the adaptive mesh refinement. The viscosity parameter is $\eta = 0.0$, and the load increment per step is $\Delta u_y = 0.001$ mm. Only the domain of interest around the V-notch region is shown. (A) $u_y = 0.047$ (pre-refine); (B) $u_y = 0.048$ (pre-refine); (C) $u_y = 0.060$ (pre-refine); (D) $u_y = 0.047$ (AMR); (E) $u_y = 0.048$ (AMR); (F) $u_y = 0.060$ (AMR).

provide identical numerical results. Once initialized at the V-notch tip when the displacement-controlled load is $u_y = 0.048$ mm, the crack propagates abruptly and stops at around $y = 1.3$ mm inside the beam. Subsequently, the crack slowly grows inside the beam and reaches $y = 1.55$ mm when the displacement-controlled load is $u_y = 0.060$ mm. Figure 15 compares the force-displacement relationship obtained from the two meshing strategies. Since the crack abruptly propagates at $u_y = 0.048$ mm, the load bearing capacity of the beam is significantly reduced after the peak load during this critical step. Figure 16 reports the convergence history of the total energy $E(\boldsymbol{u}, d)$ of the beam and the $l_2$-norm of the residuals $\boldsymbol{r}_{\boldsymbol{u}}(\boldsymbol{u}, d)$ and $\boldsymbol{r}_d(\boldsymbol{u}, d)$ in the three-point bending test at $u_y = 0.048$ mm using the pre-refined mesh. During this critical load step, the L-BFGS monolithic scheme requires 2344 iterations to converge. As shown in Figure 16A, the total energy of the beam consistently decreases after each L-BFGS iteration due to the applied strong Wolfe conditions and reaches a plateau after 2000 iterations. However, at the 2000th iteration, the absolute residuals $\boldsymbol{r}_{\boldsymbol{u}}(\boldsymbol{u}, d)$ and $\boldsymbol{r}_d(\boldsymbol{u}, d)$ are still large ($\|\boldsymbol{r}_{\boldsymbol{u}}(\boldsymbol{u}, d)\|_2 \approx 10^{-3}$ and $\|\boldsymbol{r}_d(\boldsymbol{u}, d)\|_2 \approx 10^{-7}$), as shown in Figure 16B. Therefore, more iterations are required to reduce the residuals to the prescribed tolerance shown in Equation (25). When the pre-refined mesh is used, during the critical time (load) step, the fracture is initialized at the tip of the V-notch and propagates to the center of the beam within a single step. This process is also manifested by the sudden drop of the reaction force as shown in Figure 15, which is consistent with the nature of brittle fracture. Recall that no viscosity is introduced ($\eta = 0.0$) and the time (load) step does not need to be reduced for the critical load step. This example demonstrates the robustness of the proposed monolithic scheme. Figure 17 further reports the growth of the number of DoFs and the required L-BFGS iterations when the AMR is used for the three-point bending test. Since the mesh is gradually refined as the fracture propagates, the required L-BFGS iterations are significantly smaller than the ones required when the pre-refined mesh as shown in Figure 13B is used in the

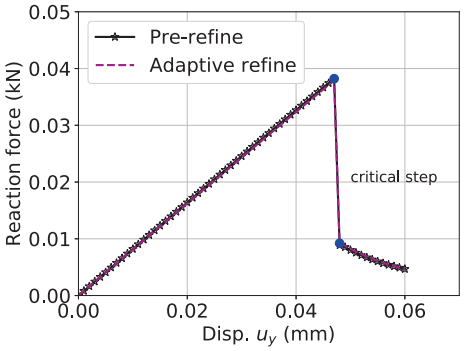


**FIGURE 15** The force-displacement relationships of the three-point bending test obtained using the pre-refined mesh and the adaptively refined mesh. The crack brutally propagates after the peak load.

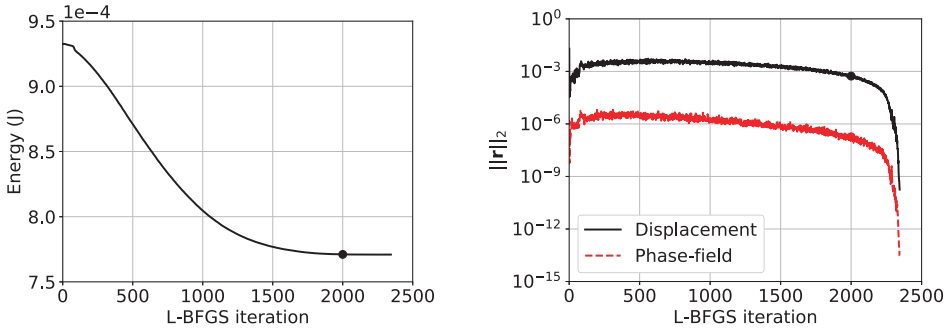


**FIGURE 16** L-BFGS convergence history at the critical step ($u_y = 0.048$ mm) of the three-point bending test: (A) Total energy and (B) the residuals of the displacement equation and the phase-field equation obtained using the pre-refined mesh.

simulation. The wall-clock time needed for simulations based on both the pre-refined mesh and the adaptively refined mesh is reported in Table 1 at the end of this section.

## 4.4 | 3D sphere inclusion test

In the last example, we use the proposed L-BFGS monolithic scheme to simulate the fracture propagation inside a 3D cube containing a rigid sphere inclusion in the center under tension. Since 3D phase-field problems are computationally expensive, we take advantage of the symmetries around the $x$-$y$ plane, $x$-$z$ plane, and $y$-$z$ plane, and reduce the computational domain to one-eighth of the original domain. Figure 18 shows the geometry of the problem, which contains two materials, the sphere inclusion and the surrounding matrix. The radius of the sphere inclusion (shown in red in Figure 18) is 0.49 mm, and the edge length of the cubic matrix domain is 1.0 mm. The boundary conditions are applied such that at the top surface ($z = 1.0$ mm), a displacement-controlled load $u_z$ is applied. At the surface $x = 0$, the displacement is fixed in the $x$-direction. At the surface $y = 0$, the displacement is fixed in the $y$-direction. At the bottom surface $z = 0.0$, the displacement is fixed in the $z$-direction.

The material parameters for the matrix include the Lamé parameters $\lambda = 12.0$ kN/mm$^2$ and $\mu = 8.0$ kN/mm$^2$, the critical energy release rate $g_c = 5.0 \times 10^{-4}$ kN/mm, and the small positive parameter $k = 0.0$ in the degradation function $g(d)$. A small viscosity parameter $\eta = 10^{-6}$ kN s/mm$^2$ is also adopted. The Lamé parameters $\lambda = 12.0 \times 10^5$ kN/mm$^2$ and $\mu = 8.0 \times 10^5$ kN/mm$^2$ are used for the sphere inclusion to emulate the rigidness. The phase-field length-scale parameter is chosen as $l = 0.01$ mm. Since the exact crack propagation path is unknown, it is prohibitively expensive to resolve the selected phase-field length-scale across the entire computational domain. Therefore, the AMR is the only valid option. Moreover, due to the drastically different material properties, the crack is expected to initialize near the interface between

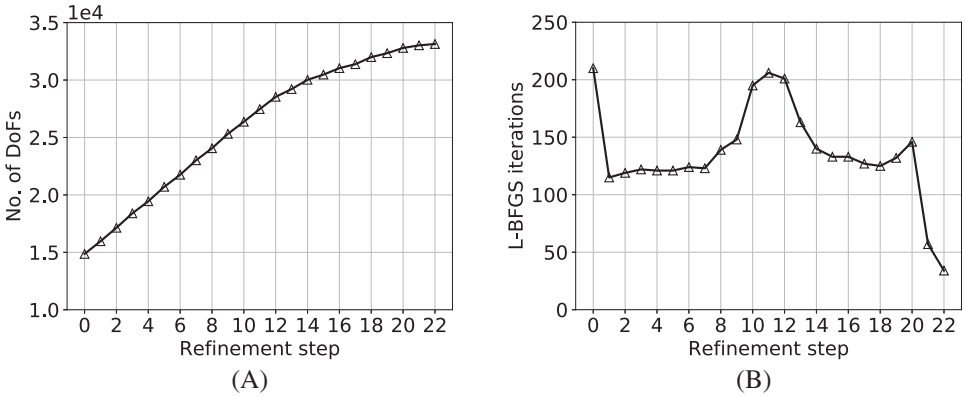


**FIGURE 17** Adaptive mesh refinement at the critical step ($u_y = 0.048$ mm) of the three-point bending test: (A) The growth of the number of degrees of freedom (DoFs) and (B) the L-BFGS iterations required for convergence of each adaptive mesh refinement step.

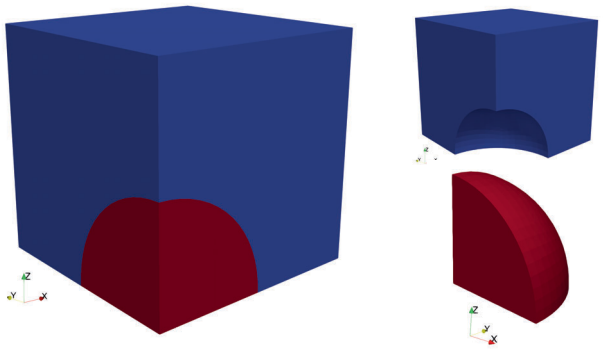

**FIGURE 18** Tensile test of a cubic matrix containing a rigid sphere inclusion in the center. By using the symmetry to reduce the computational cost, only one-eighth of the domain is modeled. The radius of the sphere inclusion (shown in red) is 0.49 mm, and the length of the cubic domain is 1.0 mm. The boundary conditions are applied such that at the top surface ($z = 1.0$ mm), a displacement-controlled load $u_z$ is applied. At the surface $x = 0$, the displacement is fixed in the $x$-direction. At the surface $y = 0$, the displacement is fixed in the $y$-direction. At the bottom surface $z = 0.0$, the displacement is fixed in the $z$-direction.

the sphere inclusion and the surrounding matrix. Based on this information, the mesh is initially refined in a small region near the inclusion-matrix interface so that the mesh size is $h = 0.007812$ mm. Figure 19A,B show the initial meshes of the matrix and the sphere inclusion. Notice that the mesh only contains eight-node brick trilinear elements. Also, the interface between the sphere inclusion and the surrounding matrix is smooth, so that no artificial stress concentration is introduced due to the spatial discretization. In the AMR technique, the phase-field refinement threshold is set as $d_{\text{threshold}} = 0.8$, and the maximum allowed ratio between the mesh size and the phase-field length-scale is $(h/l)_{\max} = 0.8$. The displacement-controlled load increment inside each step is adopted as $\Delta u_z = 10^{-3}$ mm when the total displacement $u_z$ is smaller than 0.011 mm, since the phase-field is small inside the domain and the sample exhibits a linear elastic behavior. Subsequently, a smaller load increment $\Delta u_z = 2.0 \times 10^{-4}$ mm is adopted to obtain the crack propagation process. The convergence of the L-BFGS iterations adopts the same set of residual-based criteria shown in Equation (25).

Figure 20 demonstrates the phase-field distributions at the exterior surfaces of the 3D domain, inside the 3D domain, and the corresponding adaptively refined mesh before and after the critical load step. The number of DoFs rapidly grows from 56,224 in the initial mesh, as shown in Figure 20C, to 908,504 when the displacement-controlled load reaches $u_z = 0.012$ mm, as shown in Figure 20F. This rapid growth of degree of freedom (DoF) number is expected since the mesh needs to be adaptively refined in all three directions to maintain the prescribed ratio $h/l$. Due to the nature of the brittle fracture, once the crack is initialized near the interface between the sphere inclusion and the surrounding matrix, it propagates brutally inside the matrix until reaching the exterior surfaces within the critical load step.

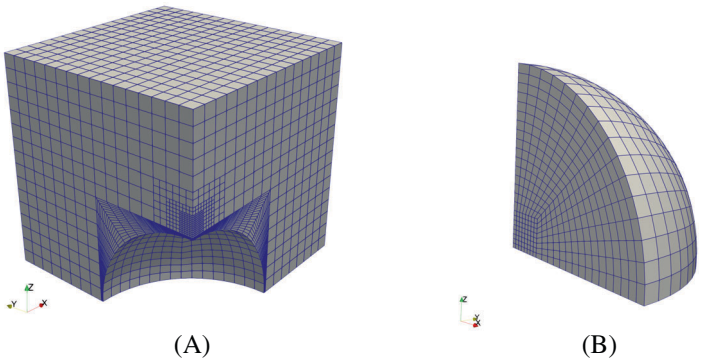


**FIGURE 19** The initial meshes of (A) the surrounding matrix and (B) the sphere inclusion. The mesh around the location where fracture is expected to initialize is pre-refined such that the mesh size is $h = 0.007812$ mm. The mesh is adaptively refined as the crack propagates.

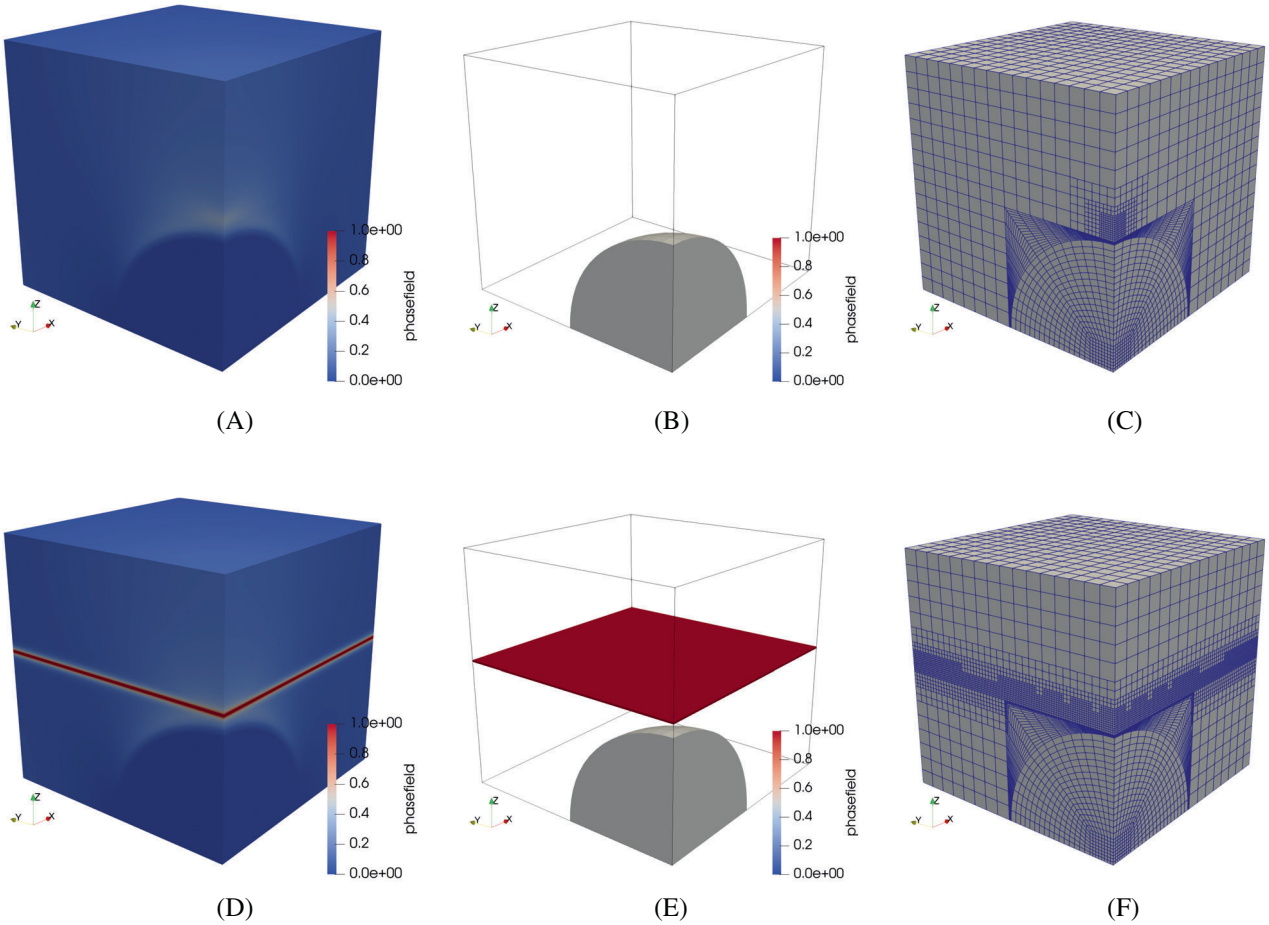


**FIGURE 20** Phase-field distributions at various displacement-controlled loading magnitudes $u_z$ (unit: mm) for the 3D sphere inclusion test. (A), (D) show the phase-field distribution across the computational domain. (B), (E) show the phase-field ($d > 0.8$) distribution inside the 3D sample (the sphere inclusion is shown in color gray). (C), (F) show the adaptively refined mesh. The viscosity parameter is $\eta = 10^{-6}$ kN s/mm$^2$, and the load increment per step is $\Delta u_z = 2.0 \times 10^{-4}$ mm. (A) Phase-field ($u_z = 0.0118$); (B) zoom-in ($d > 0.8$); (C) mesh (56,224 DoFs); (D) phase-field ($u_z = 0.0120$); (E) zoom-in ($d > 0.8$); (F) mesh (908,504 DoFs).

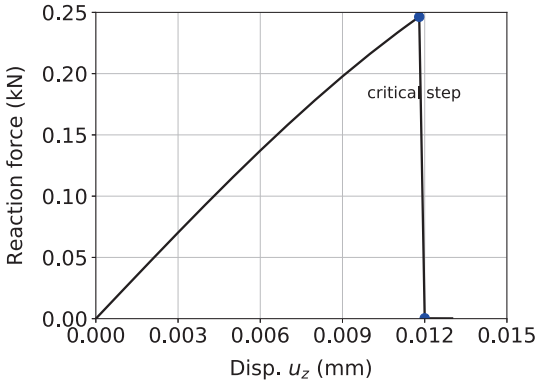


**FIGURE 21** The force-displacement relationship of the 3D sphere inclusion test. The viscosity parameter $\eta = 10^{-6}$ kN s/mm$^2$ is used to regularize the crack propagation.

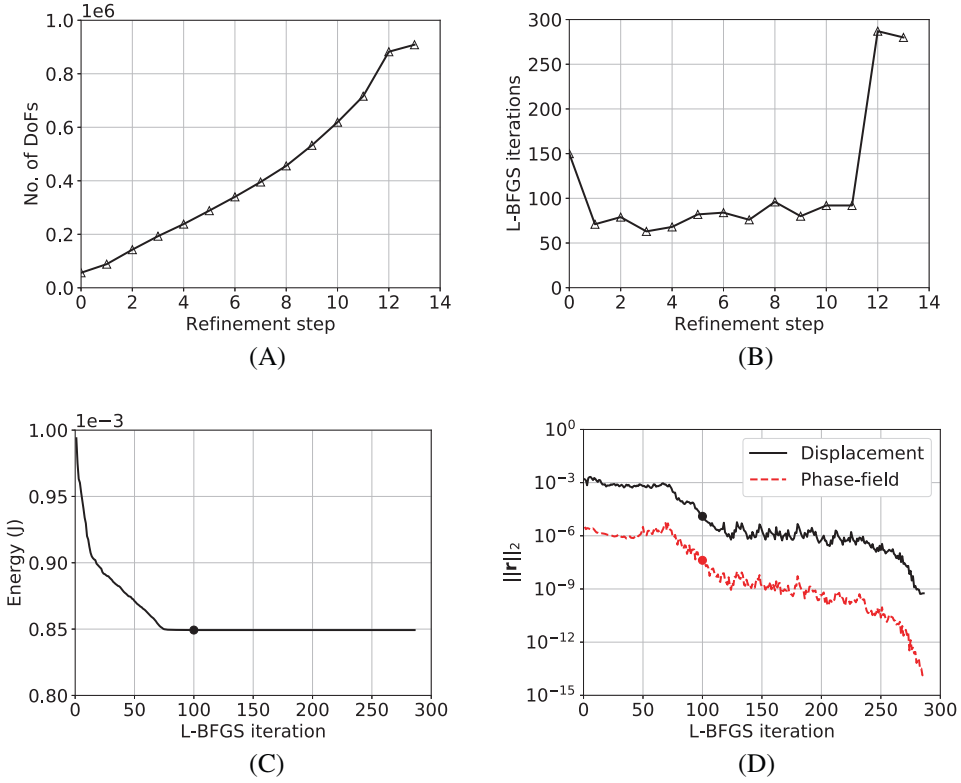


**FIGURE 22** For the 3D sphere inclusion test at $u_z = 0.0120$ mm (the critical load step), (A) the growth of the number of DoFs and (B) the L-BFGS iterations required for convergence for each adaptive mesh refinement step, the convergence history of (C) the total energy and (D) the residuals of the displacement and the phase-field equations for the 12th refinement step (the worst case).

Figure 21 reports the force-displacement relationship obtained from the 3D sphere inclusion test. Since the crack brutally propagates inside the matrix once it is initialized, the sample loses load bearing capacity during the critical step. During the critical load step, the crack propagates inside the matrix in a brutal way. Therefore, multiple AMR steps are required to resolve the phase-field length-scale as the crack propagates. Figure 22A shows the growth of the total number of DoFs during the AMR process inside the critical load step, which grows from 56,224 DoFs in the initial mesh to 908,504 DoFs in the final mesh. Figure 22B shows the number of L-BFGS iterations required for convergence during each AMR inside the critical load step. The highest number of L-BFGS iterations (287 iterations) happens at the 12th mesh refinement step. Figure 22C,D show the convergence history of this worst-case refinement step inside the critical load step. Due to the line search method based on the strong Wolfe conditions, the total energy consistently decreases until it reaches a

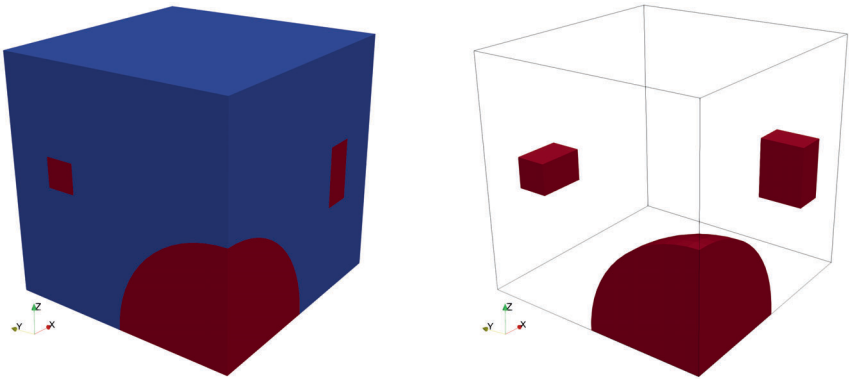

**FIGURE 23** Two rigid box-barriers are added to the 3D sphere inclusion problem. The barriers have the same material properties as the sphere inclusion.

plateau at around the 100th L-BFGS iteration. However, at this iteration, the absolute residuals $\boldsymbol{r}_{\boldsymbol{u}}(\boldsymbol{u}, d)$ and $\boldsymbol{r}_d(\boldsymbol{u}, d)$ are still large ($\|\boldsymbol{r}_{\boldsymbol{u}}(\boldsymbol{u}, d)\|_2 \approx 10^{-5}$ and $\|\boldsymbol{r}_d(\boldsymbol{u}, d)\|_2 \approx 10^{-8}$), as shown in Figure 22D. Therefore, more iterations are required to reduce the residuals to the prescribed tolerance shown in Equation (25).

In order to demonstrate the performance of the proposed monolithic scheme when the brittle crack propagates and forms a non-planar surface, two extra rigid box-barriers are added into the matrix, as shown in Figure 23. All the boundary conditions and material parameters are kept the same as before, and the barriers have the same material properties as the sphere inclusion. Figure 24 demonstrates the phase-field distributions at the exterior surfaces of the 3D domain, inside the 3D domain, and the corresponding adaptively refined mesh before and after the critical load step. The number of DoFs rapidly grows from 56,224 in the initial mesh, as shown in Figure 24C, to 975,136 when the displacement-controlled load reaches $u_z = 0.012$ mm, as shown in Figure 24F. Due to the presence of the two rigid box-barriers, the brutally propagated crack forms a non-planar surface inside the matrix during the critical load step, as shown in Figure 24E. This example highlights the necessity of the AMR technique for the phase-field formulation of crack propagation in 3D problems when the crack propagation path is unknown a priori and potentially non-planar.

Figure 25 reports the force-displacement relationship obtained from the 3D sphere inclusion test containing the two rigid box-barriers. Similar to the example without the extra barriers, the crack brutally propagates inside the matrix once it is initialized and the sample loses load bearing capacity during the critical step. Figure 26 shows the required number of AMR steps during the critical load step and the convergence history for the worse case. As shown in Figure 26A, the number of DoFs grows from 56,224 in the initial mesh to 975,136 in the final mesh after 20 AMRs. Also, after 15 AMRs, the number of DoFs only slightly increases, indicating that the mesh change is insignificant subsequently. However, in order to fulfill the mesh refinement criterion dictated by the phase-field threshold value $d_{\text{threshold}}$ and the maximum allowed $h/l$ ratio, the AMR is still performed after 15 refinement steps until the stop criterion is satisfied. Figure 26B reports the number of L-BFGS iterations required for each AMR step. Due to the presence of the two rigid box-barriers, comparing with the previous case, the monolithic scheme requires significantly more L-BFGS iterations for convergence when the mesh around the rigid box-barriers is adaptively refined. However, the high number of L-BFGS iterations required for convergence is also due to the prescribed strict convergence criteria based on both the residuals and the solution increments.

In Figure 26B, we can see that the required number of L-BFGS iterations oscillates. Recall that the Lamé parameters for the two box barriers are $10^5$ times of the counterparts for the surrounding matrix. The first spike of L-BFGS iteration number in Figure 26B happens at the 12th mesh refinement step requiring about 1200 iterations for convergence. This is when the fracture front meets the first box barrier during the AMR and tries to go around it. The second spike happens at the 15th mesh refinement step requiring about 1100 iterations for convergence. This is when the fracture front meets the second box barrier and tries to go around it. Due to the huge difference between the material properties of the barriers and the surrounding matrix, these spikes of required iterations are expected. In comparison, the required L-BFGS iterations are much more consistent and smaller when there is no barrier on the fracture propagation path, see Figure 22B for the case without box barriers. Figure 26C,D show the convergence history of the worst-case AMR step (the 12th refinement step) inside the critical load step. Due to the line search method based on the strong Wolfe conditions, the total energy consistently decreases until it reaches a plateau at around the 500th L-BFGS iteration. However, at this iteration,

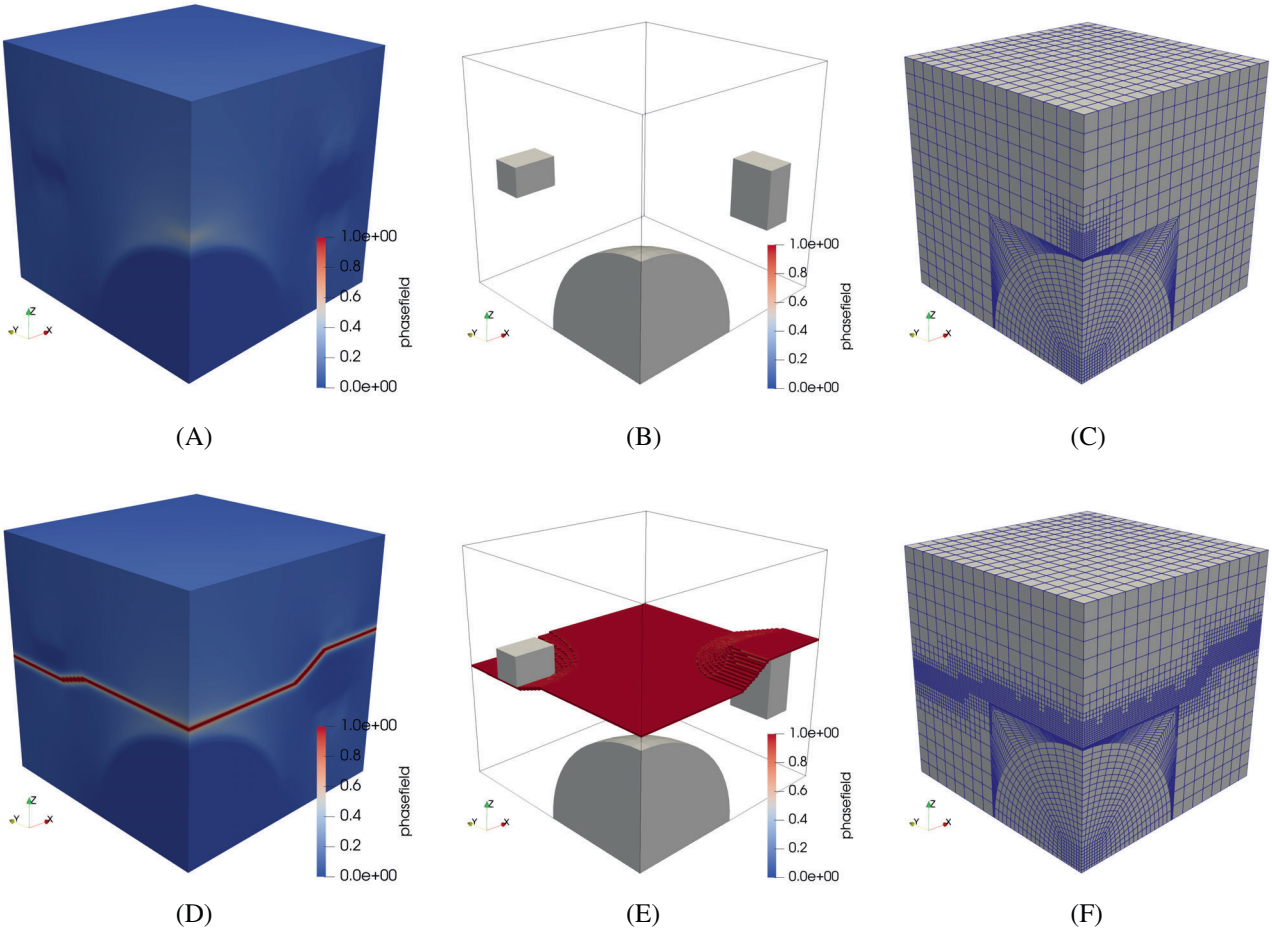


**FIGURE 24** Phase-field distributions at various displacement-controlled loading magnitudes $u_z$ (unit: mm) for the 3D sphere inclusion test containing two extra rigid box-barriers. (A), (D) show the phase-field distribution across the computational domain. (B), (E) show the phase-field ($d > 0.8$) distribution inside the 3D sample (the sphere inclusion and the two extra box-barriers are shown in color gray). (C), (F) show the adaptively refined mesh. The viscosity parameter is $\eta = 10^{-6}$ kN s/mm$^2$, and the load increment per step is $\Delta u_z = 2.0 \times 10^{-4}$ mm. (A) Phase-field ($u_z = 0.0118$); (B) zoom-in ($d > 0.8$); (C) mesh (56,224 DoFs); (D) phase-field ($u_z = 0.0120$); (E) zoom-in ($d > 0.8$); (F) mesh (975,136 DoFs).

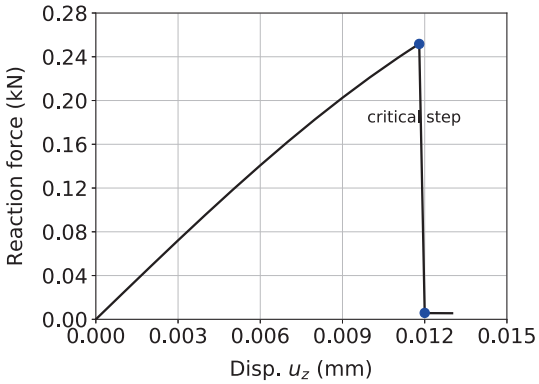


**FIGURE 25** The force-displacement relationship of the 3D sphere inclusion test containing two extra rigid box-barriers. The viscosity parameter $\eta = 10^{-6}$ kN s/mm$^2$ is used to regularize the crack propagation.

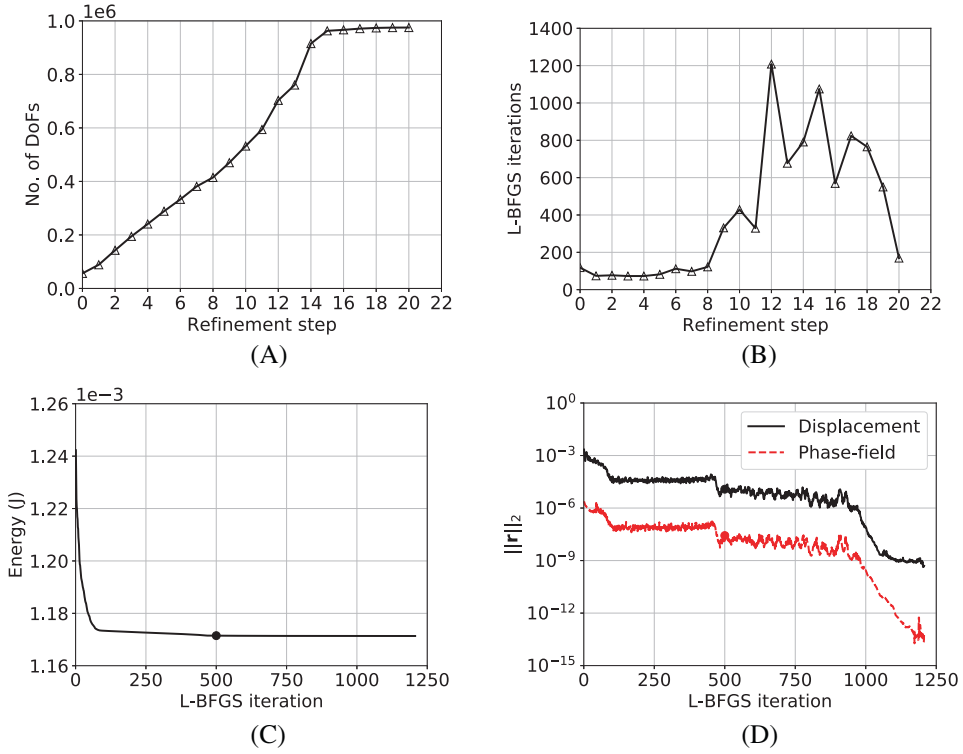


**FIGURE 26** For the 3D sphere inclusion test containing two extra rigid box-barriers at $u_z = 0.0120$ mm (the critical load step), (A) the growth of the number of degrees of freedom (DoFs) and (B) the L-BFGS iterations required for convergence for each adaptive mesh refinement step, the convergence history of (C) the total energy and (D) the residuals of the displacement and the phase-field equations for the 12th refinement step (the worst case).

the absolute residuals $\boldsymbol{r_u}(\boldsymbol{u}, d)$ and $\boldsymbol{r}_d(\boldsymbol{u}, d)$ are still large ($\|\boldsymbol{r_u}(\boldsymbol{u}, d)\|_2 \approx 10^{-5}$ and $\|\boldsymbol{r}_d(\boldsymbol{u}, d)\|_2 \approx 10^{-8}$). Therefore, more iterations are required to reduce the residuals to the prescribed tolerance shown in Equation (25). Since the AMR technique is used in this example, during the first L-BFGS iteration associated with a newly refined mesh, the initial solution is interpolated from the previous solve on the old mesh. As a result, the absolute residual for the first iteration is already quite small (around $10^{-6}$ for the phase-field residual). Therefore, using a small relative residual, for instance, $10^{-9}$, as the convergence criteria would make the absolute residual smaller than $10^{-15}$ needed for convergence. This extremely restrictive convergence tolerance would not improve the solution quality while unnecessarily increasing the number of required iterations. In fact, as shown in Figures 22D and 26D, the absolute residuals of the phase-field are still below $10^{-12}$ during convergence when the absolute residuals are adopted in the convergence criteria.

## 4.5 | Wall-clock time

In general, it is difficult to fairly compare the computational cost among various phase-field methods in the literature even for the same test problem. Because there are so many factors that could influence the number of iterations required for convergence and the simulation wall-clock time, such as the choice of the linear solver and preconditioner, the load step sizes, the type of the convergence criteria and the prescribed tolerance, the level of intermediate output, as well as the computer architecture used to perform the simulation and so forth. Therefore, we only show the computational cost of the proposed L-BFGS monolithic scheme. Table 1 reports the number of DoFs and the wall-clock time for all the numerical examples provided in this section. When the AMR technique is used in a simulation, both the DoFs for the initial mesh and the DoFs for the final mesh are reported. Recall that at each L-BFGS iteration, the initial update of the inverse Hessian approximation $\mathbf{H} = \mathbf{H}_0 = \hat{\mathbf{K}}^{-1}$ requires to solve a linear system containing the diagonal block matrix $\hat{\mathbf{K}}$. For 2D problems where the size of the system is relatively small (less than 50,000 DoFs), the sparse direct solver UMFPACK[†] is adopted. For the 3D sphere inclusion test, the conjugate-gradient (CG) iterative solver combined with a simple Jacobi preconditioner is

**TABLE 1** Computational cost of all the numerical examples.

| Example | Mesh strategy | No. DoFs | Wall time (s) |
|---|---|---|---|
| Simple tension | Pre-refine | 16,401 | $1.693 \times 10^2$ |
| | AMR | – | – |
| Simple shear | Pre-refine | 135,849 | $4.128 \times 10^3$ |
| | AMR | 3792 to 19,002 | $7.611 \times 10^2$ |
| Three-point bending | Pre-refine | 54,957 | $2.217 \times 10^4$ |
| | AMR | 14,862 to 39,477 | $8.240 \times 10^3$ |
| Sphere inclusion | Pre-refine | – | – |
| | AMR | 56,224 to 908,504 | $2.812 \times 10^4$ |
| Sphere inclusion with box-barriers | Pre-refine | – | – |
| | AMR | 56,224 to 975,136 | $4.763 \times 10^5$ |

used to solve the linear system. In all the numerical examples provided in this section, the prescribed convergence criteria include both the residuals of the coupled nonlinear system and the magnitude of the solution increment. Obviously, the choices of these tolerances will have an impact on the number of required iterations and wall-clock time. How to choose these tolerances depend on the specific engineering applications and the aims of the work. Moreover, in all the numerical examples provided in this article, we calculate many intermediate results during the nonlinear iterations and write them to a log file. For instance, after each L-BFGS iteration, the total energy of the system (numerical integration across all the elements) is calculated and written to the output log file. The wall-clock time could be further reduced by about 20% if the output of intermediate results are turned off.

## 5 | CONCLUSIONS AND FUTURE WORK

In this article, we propose a monolithic scheme based on the limited-memory BFGS method, or the L-BFGS method, to solve the coupled system of the phase-field formulation for crack propagation simulations. The Newton-based monolithic scheme, such as the Newton–Raphson method, encounters convergence difficulties due to the non-convexity of the underlying total energy functional. The BFGS-based monolithic scheme, which is a type of quasi-Newton methods, can overcome the convergence difficulties and is ideal for phase-field crack simulations. Comparing with the classical BFGS method that updates the fully dense approximate Hessian matrix or its inverse, the L-BFGS method only requires to store a few vector-pairs of length $n$, where $n$ represents the number of DoFs in the finite element discretization. This reduced memory requirement makes the L-BFGS method practical for not only phase-field crack propagation problems but also for general finite element simulations.

The proposed L-BFGS monolithic scheme is further combined with an AMR technique.[15] Several modifications are introduced into the original refinement technique to improve the convergence behavior of the proposed L-BFGS monolithic scheme. More importantly, a series of hanging nodes are generated at element interfaces during the AMR process. These hanging-node constraints, together with the essential boundary conditions prescribed on the finite element mesh boundaries, are treated as algebraic nodal constraints and can be easily incorporated into the L-BFGS approach through two extract matrix-vector multiplications as shown in Algorithm 2.

The proposed L-BFGS monolithic scheme is extremely robust. There are two crucial ingredients for the L-BFGS scheme to work. First, the line search method based on the strong Wolfe condition is applied at each iteration. This line search procedure is crucial to ensure that the curvature condition Equation (15) is valid. Due to this curvature condition, each BFGS update results in a positive-definite approximate Hessian $\mathbf{B}_{k+1}$ (and its inverse $\mathbf{H}_{k+1}$), as long as the initial approximate Hessian $\mathbf{B}_0$ is positive definite. Indeed, the initial approximate Hessian used in this article is the diagonal block matrix shown in Equation (13) and is indeed positive definite. Second, the search direction $\boldsymbol{p}$ is guaranteed to be a descent direction even considering the hanging node constraints, see Equation (23) for the proof. This property ensures the monotonically decrease of the total energy as demonstrated in all the numerical examples. We want to emphasize the

importance of the strong Wolfe condition for the L-BFGS scheme. A line search based on the simple Armijo backtracking technique does not ensure the positive-definiteness of the updated approximate Hessian.

Through several 2D and 3D numerical examples, we demonstrate the following features of the proposed L-BFGS monolithic scheme. First, the scheme is accurate in the sense that the residuals of the displacement (force-balance) equation and the phase-field equation can both satisfy a set of stringent convergence criteria simultaneously and ensure that the crack propagation obtained from the fully coupled system is not delayed, which is a common problem in the staggered approach. Second, the scheme is robust in the sense that it converges without the necessity of adjusting the magnitude of the load increment (time step size) during the critical load step, when the crack propagates rapidly in a brutal way. Third, the scheme is efficient regarding both the memory consumption and the wall-clock time due to the limited-memory feature and the AMR technique. Lastly, the scheme and its implementation are dimension-independent and can be used for both 2D and 3D crack propagation simulations.

In the current work, the proposed L-BFGS monolithic scheme is based on the phase-field formulation that enforces the phase-field irreversibility through the history variable of the maximum positive strain energy $\mathcal{H}$.[13] Despite of the convenience due to the introduction of this history variable, there is an inconsistency between the underlying energy functional and the weak form, in which the positive strain energy $\psi^+$ is replaced by the history variable $\mathcal{H}$. Our future work will focus on combining the proposed L-BFGS monolithic scheme with a more rigorous mathematical treatment of the phase-field irreversibility condition, such as the primal-dual active set method[15] or the interior-point method.[20] We envision that this improved phase-field formulation will overcome the numerical difficulties arising from the non-convexity of the energy functional, improve the computational efficiency regarding both the required memory and wall-clock time through the limited-memory feature as well as the AMR, and rigorously impose the damage irreversibility condition.

## ACKNOWLEDGMENTS

Tao Jin is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under the Discovery Grants Program (funding reference number: RGPIN-2021-02561). Kuiying Chen is supported by the National Research Council Canada DTS-SCS program A1-018177 project. Their supports are greatly appreciated.

## CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

## DATA AVAILABILITY STATEMENT

All the source code and input files developed in this article are hosted on GitHub https://github.com/taojinllnl/Phasefield_L-BFGS_monolithic_solver.

## ENDNOTES

*https://github.com/taojinllnl/Phasefield_L-BFGS_monolithic_solver.
†https://people.engr.tamu.edu/davis/suitesparse.html.

## ORCID

*Tao Jin* https://orcid.org/0000-0001-6658-8941
*Zhao Li* https://orcid.org/0009-0007-7296-8810

## REFERENCES

1. Oliver J, Huespe AE, Samaniego E, Chaves EWV. Continuum approach to the numerical simulation of material failure in concrete. *Int J Numer Anal Methods Geomech*. 2004;28(7-8):609-632. doi:10.1002/nag.365
2. Armero F, Kim J. Three-dimensional finite elements with embedded strong discontinuities to model material failure in the infinitesimal range. *Int J Numer Methods Eng*. 2012;91(12):1291-1330. doi:10.1002/nme.4314
3. Jin T, Mourad HM, Bronkhorst CA. A comparative study of shear band tracking strategies in three-dimensional finite elements with embedded weak discontinuities. *Finite Elem Anal Des*. 2019;155:11-31. doi:10.1016/j.finel.2018.11.001
4. Simo JC, Oliver J, Armero F. An analysis of strong discontinuities induced by strain-softening in rate-independent inelastic solids. *Comput Mech*. 1993;12:277-296. doi:10.1007/BF00372173
5. Linder C, Armero F. Finite elements with embedded strong discontinuities for the modeling of failure in solids. *Int J Numer Methods Eng*. 2007;72(12):1391-1433. doi:10.1002/nme.2042
6. Armero F, Linder C. New finite elements with embedded strong discontinuities in the finite deformation range. *Comput Methods Appl Mech Eng*. 2008;197(33):3138-3170. doi:10.1016/j.cma.2008.02.021

7. Linder C, Zhang X. A marching cubes based failure surface propagation concept for three-dimensional finite elements with non-planar embedded strong discontinuities of higher-order kinematics. *Int J Numer Methods Eng*. 2013;96(6):339-372. doi:10.1002/nme.4546
8. Jin T, Mourad HM, Bronkhorst CA, et al. Three-dimensional explicit finite element formulation for shear localization with global tracking of embedded weak discontinuities. *Comput Methods Appl Mech Eng*. 2019;353:416-447. doi:10.1016/j.cma.2019.05.011
9. Francfort G, Marigo JJ. Revisiting brittle fracture as an energy minimization problem. *J Mech Phys Solids*. 1998;46(8):1319-1342. doi:10.1016/S0022-5096(98)00034-9
10. Bourdin B, Francfort G, Marigo JJ. Numerical experiments in revisited brittle fracture. *J Mech Phys Solids*. 2000;48(4):797-826. doi:10.1016/S0022-5096(99)00028-9
11. Bourdin B, Francfort GA, Marigo JJ. The variational approach to fracture. *J Elast*. 2008;91(1):5-148. doi:10.1007/s10659-007-9107-3
12. Miehe C, Welschinger F, Hofacker M. Thermodynamically consistent phase-field models of fracture: variational principles and multi-field FE implementations. *Int J Numer Methods Eng*. 2010;83(10):1273-1311. doi:10.1002/nme.2861
13. Miehe C, Hofacker M, Welschinger F. A phase field model for rate-independent crack propagation: robust algorithmic implementation based on operator splits. *Comput Methods Appl Mech Eng*. 2010;199(45):2765-2778. doi:10.1016/j.cma.2010.04.011
14. Ambati M, Gerasimov T, De Lorenzis L. A review on phase-field models of brittle fracture and a new fast hybrid formulation. *Comput Mech*. 2015;55(2):383-405. doi:10.1007/s00466-014-1109-y
15. Heister T, Wheeler MF, Wick T. A primal-dual active set method and predictor-corrector mesh adaptivity for computing fracture propagation using a phase-field approach. *Comput Methods Appl Mech Eng*. 2015;290:466-495. doi:10.1016/j.cma.2015.03.009
16. Wu JY. A unified phase-field theory for the mechanics of damage and quasi-brittle failure. *J Mech Phys Solids*. 2017;103:72-99. doi:10.1016/j.jmps.2017.03.015
17. Wick T. Modified Newton methods for solving fully monolithic phase-field quasi-static brittle fracture propagation. *Comput Methods Appl Mech Eng*. 2017;325:577-611. doi:10.1016/j.cma.2017.07.026
18. Wu JY, Huang Y, Nguyen VP. On the BFGS monolithic algorithm for the unified phase field damage theory. *Comput Methods Appl Mech Eng*. 2020;360:112704. doi:10.1016/j.cma.2019.112704
19. Costa A, Cusini M, Jin T, Settgast R, Dolbow JE. A multi-resolution approach to hydraulic fracture simulation. *Int J Fract*. 2022;237(1):165-188. doi:10.1007/s10704-022-00662-y
20. Wambacq J, Ulloa J, Lombaert G, François S. Interior-point methods for the phase-field approach to brittle and ductile fracture. *Comput Methods Appl Mech Eng*. 2021;375:113612. doi:10.1016/j.cma.2020.113612
21. Lampron O, Therriault D, Lévesque M. An efficient and robust monolithic approach to phase-field quasi-static brittle fracture using a modified Newton method. *Comput Methods Appl Mech Eng*. 2021;386:114091. doi:10.1016/j.cma.2021.114091
22. Bharali R, Goswami S, Anitescu C, Rabczuk T. A robust monolithic solver for phase-field fracture integrated with fracture energy based arc-length method and under-relaxation. *Comput Methods Appl Mech Eng*. 2022;394:114927. doi:10.1016/j.cma.2022.114927
23. Kopaničáková A, Krause R. A recursive multilevel trust region method with application to fully monolithic phase-field models of brittle fracture. *Comput Methods Appl Mech Eng*. 2020;360:112720. doi:10.1016/j.cma.2019.112720
24. Gerasimov T, De Lorenzis L. A line search assisted monolithic approach for phase-field computing of brittle fracture. *Comput Methods Appl Mech Eng*. 2016;312:276-303. doi:10.1016/j.cma.2015.12.017
25. Matthies H, Strang G. The solution of nonlinear finite element equations. *Int J Numer Methods Eng*. 1979;14(11):1613-1626. doi:10.1002/nme.1620141104
26. Brun M, Wick T, Berre I, Nordbotten JM, Radu FA. An iterative staggered scheme for phase field brittle fracture propagation with stabilizing parameters. *Comput Methods Appl Mech Eng*. 2020;361:112752. doi:10.1016/j.cma.2019.112752
27. Storvik E, Both JW, Sargado JM, Nordbotten JM, Radu FA. An accelerated staggered scheme for variational phase-field models of brittle fracture. *Comput Methods Appl Mech Eng*. 2021;381:113822. doi:10.1016/j.cma.2021.113822
28. Luo C. Fast staggered schemes for the phase-field model of brittle fracture based on the fixed-stress concept. *Comput Methods Appl Mech Eng*. 2023;404:115787. doi:10.1016/j.cma.2022.115787
29. Bourdin B. Numerical implementation of the variational formulation for quasi-static brittle fracture. *Interfaces Free Bound*. 2007;9(3):411-430. doi:10.4171/IFB/171
30. Wheeler M, Wick T, Wollner W. An augmented-Lagrangian method for the phase-field approach for pressurized fractures. *Comput Methods Appl Mech Eng*. 2014;271:69-85. doi:10.1016/j.cma.2013.12.005
31. Hintermüller M, Ito K, Kunisch K. The primal-dual active set strategy as a semismooth Newton method. *SIAM J Optim*. 2002;13(3):865-888. doi:10.1137/S1052623401383558
32. Davidon WC. Variable metric method for minimization; 1959. doi:10.2172/4252678
33. Davidon WC. Variable metric method for minimization. *SIAM J Optim*. 1991;1(1):1-17. doi:10.1137/0801001
34. Nocedal J. Updating quasi-Newton matrices with limited storage. *Math Comput*. 1980;35(151):773-782. doi:10.2307/2006193
35. Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. *Math Program*. 1989;45(1):503-528. doi:10.1007/BF01589116
36. Byrd RH, Nocedal J, Schnabel RB. Representations of quasi-Newton matrices and their use in limited memory methods. *Math Program*. 1994;63(1):129-156. doi:10.1007/BF01582063
37. Amor H, Marigo JJ, Maurini C. Regularized formulation of the variational brittle fracture with unilateral contact: numerical experiments. *J Mech Phys Solids*. 2009;57(8):1209-1229. doi:10.1016/j.jmps.2009.04.011
38. Wu JY, Cervera M. A novel positive/negative projection in energy norm for the damage modeling of quasi-brittle solids. *Int J Solids Struct*. 2018;139–140:250-269. doi:10.1016/j.ijsolstr.2018.02.004

39. Wu JY, Nguyen VP, Zhou H, Huang Y. A variationally consistent phase-field anisotropic damage model for fracture. *Comput Methods Appl Mech Eng*. 2020;358:112629. doi:10.1016/j.cma.2019.112629
40. Steinke C, Kaliske M. A phase-field crack model based on directional stress decomposition. *Comput Mech*. 2019;63:1019-1046. doi:10.1007/s00466-018-1635-0
41. Wu JY, Nguyen VP, Nguyen CT, Sutula D, Sinaie S, Bordas SP. Chapter One—phase-field modeling of fracture. *Advances in Applied Mechanics*. Vol 53. Elsevier; 2020:1-183. doi:10.1016/bs.aams.2019.08.001
42. Arndt D, Bangerth W, Feder M, et al. The deal.II library, version 9.4. *J Numer Math*. 2022;30(3):231-246. doi:10.1515/jnma-2022-0054
43. Nocedal J, Wright SJ. *Numerical Optimization*. 2nd ed. Springer; 2006.
44. Moré JJ, Thuente DJ. Line search algorithms with guaranteed sufficient decrease. *ACM Trans Math Softw*. 1994;20:286-307. doi:10.1145/192115.192132
45. Shephard MS. Linear multipoint constraints applied via transformation as part of a direct stiffness assembly process. *Int J Numer Methods Eng*. 1984;20(11):2107-2112. doi:10.1002/nme.1620201112
46. Miehe C. Comparison of two algorithms for the computation of fourth-order isotropic tensor functions. *Comput Struct*. 1998;66(1):37-43. doi:10.1016/S0045-7949(97)00073-4
47. Miehe C, Lambrecht M. Algorithms for computation of stresses and elasticity moduli in terms of Seth–Hill's family of generalized strain tensors. *Commun Numer Methods Eng*. 2001;17(5):337-353. doi:10.1002/cnm.404

**How to cite this article:** Jin T, Li Z, Chen K. A novel phase-field monolithic scheme for brittle crack propagation based on the limited-memory BFGS method with adaptive mesh refinement. *Int J Numer Methods Eng*. 2024;e7572. doi: 10.1002/nme.7572

## APPENDIX A. PROJECTION TENSORS

Recall that the spectrum decomposition of the strain tensor $\boldsymbol{\epsilon}$ is expressed as

$$\boldsymbol{\epsilon} = \sum_{\alpha} \epsilon_{\alpha} \mathbf{M}_{\alpha}, \quad \mathbf{M}_{\alpha} = \boldsymbol{n}_{\alpha} \otimes \boldsymbol{n}_{\alpha},$$

where $\epsilon_{\alpha}$ and $\boldsymbol{n}_{\alpha}$ represent a pair of eigenvalue and eigenvector. The positive and negative parts of the strain tensor are defined as,

$$\boldsymbol{\epsilon}^{+} = \sum_{\alpha} \langle \epsilon_{\alpha} \rangle_{+} \mathbf{M}_{\alpha}, \quad \boldsymbol{\epsilon}^{-} = \sum_{\alpha} \langle \epsilon_{\alpha} \rangle_{-} \mathbf{M}_{\alpha}.$$

Furthermore, we introduce the following function definitions:

$$\langle x \rangle_{+} = \frac{1}{2}(x + |x|), \quad \langle x \rangle_{-} = \frac{1}{2}(x - |x|), \quad H(x) = \begin{cases} 1 & x \geq 0, \\ 0 & x < 0. \end{cases}$$

The expressions of the two projection tensors $\mathbb{P}^{+}$ and $\mathbb{P}^{-}$ are based on the work by Miehe and Lambrecht.[46,47] First, a fourth-order tensor is defined as

$$\mathbb{G}_{\alpha\beta} = \boldsymbol{n}_{\alpha} \otimes \boldsymbol{n}_{\beta} \otimes \boldsymbol{n}_{\alpha} \otimes \boldsymbol{n}_{\beta} + \boldsymbol{n}_{\alpha} \otimes \boldsymbol{n}_{\beta} \otimes \boldsymbol{n}_{\beta} \otimes \boldsymbol{n}_{\alpha}.$$

For the expression of $\mathbb{P}^{+}$, we introduce the scalar

$$\vartheta^{+}_{\alpha\beta} = \begin{cases} \frac{\langle \epsilon_{\alpha} \rangle_{+} - \langle \epsilon_{\beta} \rangle_{+}}{\epsilon_{\alpha} - \epsilon_{\beta}}, & \text{if } \epsilon_{\alpha} \neq \epsilon_{\beta}, \\ \frac{H(\epsilon_{\alpha}) + H(\epsilon_{\beta})}{2}, & \text{if } \epsilon_{\alpha} = \epsilon_{\beta}. \end{cases}$$

Then,

$$\mathbb{P}^{+} = \frac{\partial \boldsymbol{\epsilon}^{+}}{\partial \boldsymbol{\epsilon}} = \sum_{\alpha} H(\epsilon_{\alpha}) \mathbf{M}_{\alpha} \otimes \mathbf{M}_{\alpha} + \frac{1}{2} \sum_{\alpha} \sum_{\beta \neq \alpha} \vartheta^{+}_{\alpha\beta} \frac{\mathbb{G}_{\alpha\beta} + \mathbb{G}_{\beta\alpha}}{2}.$$

For the expression of $\mathbb{P}^-$, we introduce the scalar

$$\vartheta^-_{\alpha\beta} = \begin{cases} \frac{\langle \epsilon_\alpha \rangle_- - \langle \epsilon_\beta \rangle_-}{\epsilon_\alpha - \epsilon_\beta}, & \text{if } \epsilon_\alpha \neq \epsilon_\beta, \\ \frac{H(-\epsilon_\alpha)+H(-\epsilon_\beta)}{2}, & \text{if } \epsilon_\alpha = \epsilon_\beta. \end{cases}$$

Then,

$$\mathbb{P}^- = \frac{\partial \boldsymbol{\epsilon}^-}{\partial \boldsymbol{\epsilon}} = \sum_\alpha H(-\epsilon_\alpha)\mathbf{M}_\alpha \otimes \mathbf{M}_\alpha + \frac{1}{2}\sum_\alpha \sum_{\beta \neq \alpha} \vartheta^-_{\alpha\beta} \frac{\mathbb{G}_{\alpha\beta} + \mathbb{G}_{\beta\alpha}}{2}.$$

The fourth-order projection tensors $\mathbb{P}^+$ and $\mathbb{P}^-$ have the following properties:

$$\mathbb{P}^+ : \boldsymbol{\epsilon} = \boldsymbol{\epsilon}^+, \quad \mathbb{P}^+ : \boldsymbol{\epsilon}^+ = \boldsymbol{\epsilon}^+, \quad \mathbb{P}^- : \boldsymbol{\epsilon} = \boldsymbol{\epsilon}^-, \quad \mathbb{P}^- : \boldsymbol{\epsilon}^- = \boldsymbol{\epsilon}^-,$$

and

$$\mathbb{P}^+ : \boldsymbol{\epsilon}^- = \mathbf{0}, \quad \mathbb{P}^- : \boldsymbol{\epsilon}^+ = \mathbf{0}.$$

## APPENDIX B. PROOF OF IDEMPOTENT PROPERTY OF LINEAR CONSTRAINTS

In order to prove for the idempotent property of the linear constraint matrix $\mathbf{C}^2 = \mathbf{C}$, we assume that all the interdependence among the linear constraints is properly resolved. For instance, if

$$\begin{aligned} x_1 &= c_{12}x_2 + c_{13}x_3 + \cdots, \\ x_2 &= c_{23}x_3 + c_{24}x_4 + \cdots, \end{aligned}$$

we can simply eliminate the second constraint by plugging the second equation into the first one and obtain

$$x_1 = c_{12}(c_{23}x_3 + c_{24}x_4 + \cdots) + c_{13}x_3 + \cdots.$$

The linear constraint matrix $\mathbf{C}$ can be expressed as

$$\mathbf{C} = \mathbf{I} - \mathbf{I}_{d_c} + \hat{\mathbf{C}},$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{I}_{d_c}$ is defined in Equation (22), and $\hat{\mathbf{C}}$ contains all the non-zero constraint coefficients $\{c_{ij}\}$. After all the interdependence among the linear constraints is properly resolved, it can be easily shown that for the $j$th coefficient $c_{kj}$ associated with the linear constraint on the $k$th degree of freedom,

$$c_{kj} \neq 0 \quad \Rightarrow \quad c_{jl} = 0 \quad \forall\, l = 1, \ldots, n \quad \text{and} \quad \left(\mathbf{I}_{d_c}\right)_{jj} = 0. \tag{B1}$$

That is, the $j$th row in $\hat{\mathbf{C}}$ are all zeros, and the $j$th diagonal entry of $\mathbf{I}_{d_c}$ is zero. On the other hand, we have

$$\begin{aligned} \mathbf{C}^2 &= \left(\mathbf{I} - \mathbf{I}_{d_c} + \hat{\mathbf{C}}\right) \cdot \left(\mathbf{I} - \mathbf{I}_{d_c} + \hat{\mathbf{C}}\right) \\ &= \mathbf{I} - \mathbf{I}_{d_c} + \hat{\mathbf{C}} - \mathbf{I}_{d_c} + \mathbf{I}_{d_c} \cdot \mathbf{I}_{d_c} - \mathbf{I}_{d_c} \cdot \hat{\mathbf{C}} + \hat{\mathbf{C}} - \hat{\mathbf{C}} \cdot \mathbf{I}_{d_c} + \hat{\mathbf{C}} \cdot \hat{\mathbf{C}}. \end{aligned}$$

We can easily verify that

$$\mathbf{I}_{d_c} \cdot \mathbf{I}_{d_c} = \mathbf{I}_{d_c}, \quad \mathbf{I}_{d_c} \cdot \hat{\mathbf{C}} = \hat{\mathbf{C}}.$$

Furthermore,

$$\left(\hat{\mathbf{C}} \cdot \mathbf{I}_{d_c}\right)_{ij} = \sum_k c_{ik}\left(\mathbf{I}_{d_c}\right)_{kj} = c_{ij}\left(\mathbf{I}_{d_c}\right)_{jj} = 0,$$

and

$$(\hat{\mathbf{C}} \cdot \hat{\mathbf{C}})_{ij} = \sum_k c_{ik}c_{kj} = 0.$$

The above two equations are based on the property shown in Equation (B1). As a result,

$$\begin{aligned}
\mathbf{C}^2 &= \left(\mathbf{I} - \mathbf{I}_{d_c} + \hat{\mathbf{C}}\right) \cdot \left(\mathbf{I} - \mathbf{I}_{d_c} + \hat{\mathbf{C}}\right) \\
&= \mathbf{I} - \mathbf{I}_{d_c} + \hat{\mathbf{C}} - \mathbf{I}_{d_c} + \mathbf{I}_{d_c} \cdot \mathbf{I}_{d_c} - \mathbf{I}_{d_c} \cdot \hat{\mathbf{C}} + \hat{\mathbf{C}} - \hat{\mathbf{C}} \cdot \mathbf{I}_{d_c} + \hat{\mathbf{C}} \cdot \hat{\mathbf{C}} \\
&= \mathbf{I} - \mathbf{I}_{d_c} + \hat{\mathbf{C}} - \mathbf{I}_{d_c} + \mathbf{I}_{d_c} - \hat{\mathbf{C}} + \hat{\mathbf{C}} - \mathbf{0} + \mathbf{0} \\
&= \mathbf{I} - \mathbf{I}_{d_c} + \hat{\mathbf{C}} \\
&= \mathbf{C}.
\end{aligned}$$

The idempotent property of the linear constraint matrix $\mathbf{C}$ is proven.