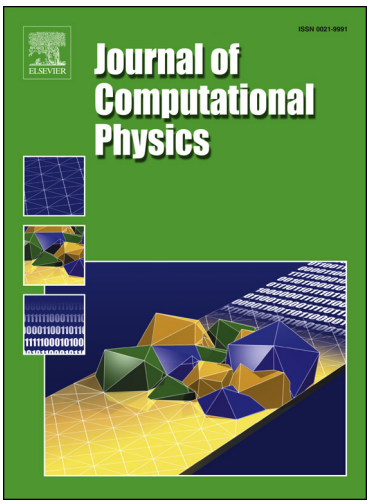
# Journal Pre-proof




A pore-level multiscale method for the elastic deformation of fractured porous media

Kangan Li and Yashar Mehmani

PII: S0021-9991(23)00169-9
DOI: https://doi.org/10.1016/j.jcp.2023.112074
Reference: YJCPH 112074

To appear in: *Journal of Computational Physics*

Received date: 13 October 2022
Revised date: 21 January 2023
Accepted date: 14 March 2023

Please cite this article as: K. Li and Y. Mehmani, A pore-level multiscale method for the elastic deformation of fractured porous media, *Journal of Computational Physics*, 112074, doi: https://doi.org/10.1016/j.jcp.2023.112074.

## Highlights

- Pore-level multiscale method developed for deformation of fractured porous media.
- The method decomposes a porous solid into subdomains to decouple computations.
- The decomposition need not conform to cracks, removing a key previous barrier.
- Implications for fixed-crack versus evolving-crack problems are analyzed.
- Approximation errors can be estimated and controlled iteratively.

# A Pore-Level Multiscale Method for the Elastic Deformation of Fractured Porous Media

Kangan Li[a], Yashar Mehmani[a]

[a]*Energy and Mineral Engineering Department, The Pennsylvania State University, University Park, Pennsylvania 16801*

**Abstract**

Pore-scale models are useful tools for understanding and upscaling the mechanical deformation of fractured porous media. The highest fidelity solutions are obtained by direct numerical simulation (DNS), in which governing equations are discretized and solved with a fine-grid solver (e.g., finite elements) on a 3D image (e.g., X-ray $\mu$CT) of a porous sample. However, the downside of DNS is its high computational cost. We present a pore-level multiscale method (PLMM) that approximates DNS efficiently and with controllable accuracy in modeling the linear elastic response of porous solids with arbitrary microstructure and crack pattern. PLMM decomposes the solid into subdomains, over which local basis and correction functions are built. These functions are then coupled with a global problem to yield an approximate solution, whose errors can be iteratively corrected. A key feature of PLMM is that the decomposition need not conform to the cracks, unlike a previous variant developed by the authors. This paves the way towards solving crack nucleation and growth problems in the future without the need to dynamically update the decomposition. We represent cracks diffusely using a phase-field variable and explore three strategies for capturing them through either the basis or correction functions. The implications of each on convergence rate and computational cost are analyzed.

*Keywords:* Porous media, Pore-scale modeling, Multiscale method, Fracture mechanics, Phase-field method

## 1. Introduction

The deformation and failure of porous media is relevant to a wide range of applications from geologic $CO_2$ sequestration, $H_2$ storage, geothermal energy extraction to the design of lightweight, high-strength biomimetic materials for airplanes [1], armored vehicles [2], and thermo-acoustically insulated buildings [3]. In the subsurface, which motivates the authors, fractures are induced by either human activity (e.g., fluid injection) or tectonic stresses. In either case, they significantly impact the hydro-mechanical properties of a porous rock. To quantify these alterations, pore-scale models are useful, because they can simulate deformation and failure at the scale of individual grains and pores comprising the rock. The output from pore-scale models can be used to parameterize macroscopic models, which would otherwise contain phenomenological constants [4]. Here, we focus on the elastic deformation of fractured porous media at the pore scale, wherein cracks are assumed to be static.

With recent advances in X-ray $\mu$CT imaging [5, 6], combined with energy-dispersive X-ray spectroscopy [7–9], the 3D microstructure and mineralogy of geologic porous samples can be captured in great detail. Such images serve as inputs to pore-scale models, which divide into two categories: direct numerical simulation (DNS) and discrete-element methods (DEM). In DNS, the governing equations are discretized on a Cartesian mesh that often coincides with the image (voxels). Neither the geometry nor the equations are simplified [10]. Examples include extended finite element (XFEM) [11, 12], shifted boundary method [13, 14], immersed boundary finite volume [15, 16], phase-field models [17–19], and peri-dynamics [20, 21]. The disadvantage of DNS is that it is computationally prohibitive for analyzing large or many samples. To render computations tractable, in DEM, both the governing equations and the microstructure of a porous sample are simplified [22–24]. The solid is replaced by an assembly of "particles"

*Corresponding author: Yashar Mehmani. Email: yzm5192@psu.edu
*Email addresses:* kbl5610@psu.edu (Kangan Li), yzm5192@psu.edu (Yashar Mehmani)

that have elementary shapes (e.g., spheres, polyhedral, clusters of spheres) [24–26], often assumed rigid. Failure is modeled by breaking fictitious "bonds" between particles when forces exceed the material's strength [23, 27]. Recent extensions like FEM-DEM have removed the particle rigidity assumption of DEM [28, 29]. While useful in many problems, DEM and its variants have two drawbacks [30, 31]: (1) They are ill-suited for the non-particulate and highly heterogeneous porous rocks encountered in the deep subsurface; And (2) they lack a built-in mechanism to estimate and control errors. The latter is because *convergence* to a fine-scale solution is undefined.

Recently, a pore-level multiscale method (PLMM) was proposed by the authors to approximate DNS efficiently and with controllable accuracy [16] for the linear-elastic response of a fractured porous solid with arbitrary microstructure. The idea was to decompose the solid into subdomains, over which local fine-grid problems are solved to construct basis and correction functions. The decomposition was done in such a way that subdomains corresponded to physical grains in a granular medium (as in DEM), and interfaces between the subdomains to physical contacts between the grains. The basis and correction functions were then coupled to each other with a small global problem in terms of coarse-grid displacement unknowns defined at subdomain interfaces. PLMM was shown to yield an accurate initial approximation to DNS that could be corrected up to any desired accuracy using an iterative strategy. Despite these benefits, the method has three major drawbacks: (1) The decomposition must *conform* to cracks, which forbids subdomains from containing (or sustaining through later loading) any interior cracks; (2) The conformity could severely degrade the quality of the multiscale approximation in cases where cracks do not coincide with physical contacts; And (3) the iterative strategy does not correct errors near cracks, only away from them.

In this work, we reformulate PLMM [16] to remove the above limitations. The resulting algorithm approximates DNS efficiently for arbitrary cracks patterns, which need not conform to the solid's decomposition. Moreover, PLMM errors are controllable throughout the domain. We represent cracks diffusely using a phase-field variable [32, 33], but emphasize that a sharp-crack representation is equally applicable in PLMM [11, 15]. To solve evolving-crack problems, a momentum equation must be coupled to a crack (or phase-field) evolution equation and solved repeatedly, often in a staggered fashion, at each loading step. Here, we focus only on the momentum equation, since it is more expensive to solve. However, anticipating the needs of solving evolving-crack problems in the future, we propose three strategies for capturing cracks via basis and correction functions, which have important implications on convergence and computational cost: (1) All cracks are captured by the bases (Strategy A); (2) No cracks are captured by the bases (Strategy N); and (3) only some cracks, namely those that cross subdomain interfaces, are captured by the bases (Strategy U). If cracks are static, Strategy A is the the fastest to converge. But if cracks are evolving, Strategy U is more efficient. This is because basis (not correction) functions are the most accurate way of capturing cracks, but their repeated update during crack evolution is costly. Strategy U strikes a delicate balance.

Multiscale methods for porous media are extensively developed at the continuum (or Darcy) scale. They include the multiscale finite element (MsFE) [12, 34–41], multiscale finite volume (MsFV) [42–47]journal, and multiscale mortar finite element (MoMsFE) [48–51] methods to solve elliptic/parabolic equations of flow and mechanics. At the pore scale, multiscale methods are much less developed and have focused primarily on fluid flow [52–61]. The PLMM herein builds on ideas originally developed for single-phase flow [59], two-phase flow [60], and compressible gas flow [61]. A recent review of existing multiscale methods at the pore and Darcy scales can be found in [62].

The article is organized as follows: the governing equations approximated by PLMM are described in Section 2. Section 3 briefly overviews the fine-scale DNS solver employed. Section 4 discusses the algorithmic details of PLMM. Specifically, local basis and correction problems are defined in Section 4.3, the global problem that couples these local problems is formulated in Section 4.4, and the iterative strategy that corrects the PLMM errors is outlined in Section 4.5. In Sections 5–6 we validate PLMM against DNS for various 2D and 3D microstructures and a wide range of crack patterns. The results are discussed in Section 7 and the findings summarized in Section 8.

## 2. Problem description

Consider the porous domain $\Omega \subset \mathbb{R}^D$ with Lipschitz boundary $\partial\Omega$, where $D$ is the number of spatial dimensions; as shown in Fig. 1a. The open set $\Omega$ is partitioned into a solid phase $\Omega_s$ and a void space $\Omega_v$ such that $\Omega = \Omega_s \cup \Omega_v \cup \Gamma^w$ and $\Omega_s \cap \Omega_v = \emptyset$, where $\Gamma^w = \overline{\Omega}_v \cap \overline{\Omega}_s$ is the interface between $\Omega_s$ and $\Omega_v$. The black lines shown in Fig. 1a are possible cracks within $\Omega_s$. We consider the following linear elastic momentum equations on $\Omega_s$:

$$\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}) = \boldsymbol{f} \tag{1}$$

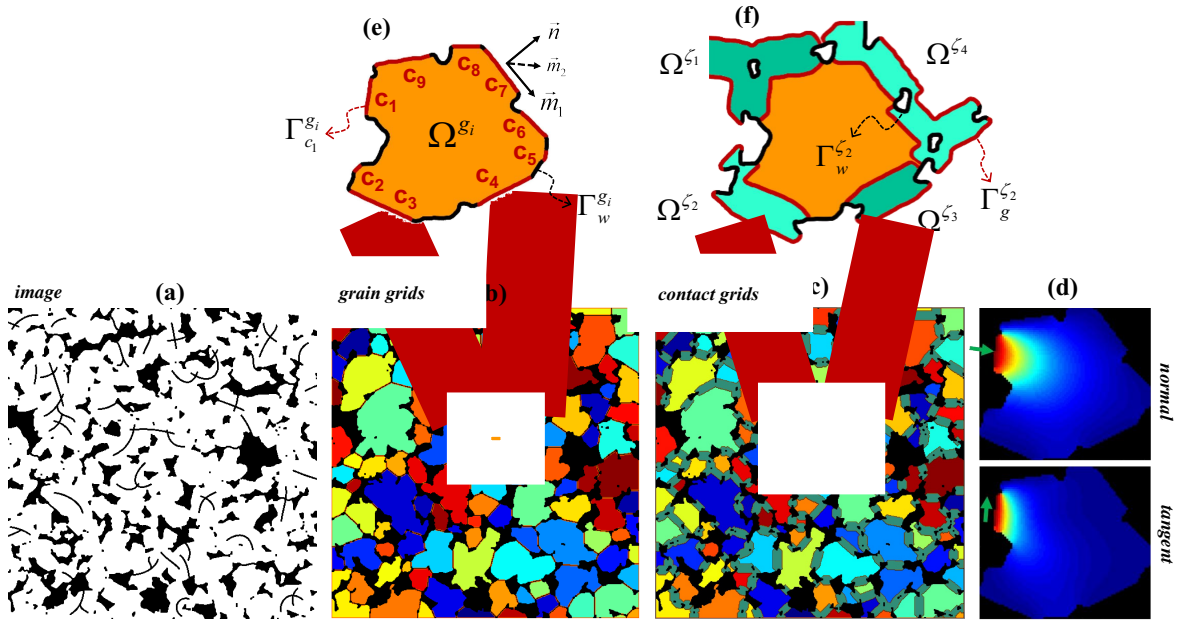



Figure 1: Schematic of a pore-scale image, its decomposition into grain grids, and contact grids. (a) The image $\Omega$ consists of a solid phase $\Omega_s$ (white), a void space $\Omega_v$ (black), and cracks $\Gamma^F$ (black lines). (b) The solid $\Omega_s$ is decomposed into grain grids $\Omega^{g_i}$ (randomly colored). (c) Contact grids $\Omega^{\zeta_k}$ (cyan) cover interfaces, $\Gamma^{c_j}$, between the grain grids. (d) The displacement fields of two basis functions on $\Omega^{g_i}$ are shown, corresponding to normal (compressive) and tangential (shear) BCs imposed (green arrows). (e) $\partial\Omega^{g_i}$ consists of $\Gamma^{g_i}_w$ (black) and $\Gamma^{g_i}_{c_{1-9}}$ (red) boundaries. The unit normal, $\boldsymbol{n}$, and unit tangents, $\{\boldsymbol{m}_1, \boldsymbol{m}_2\}$, on $\Gamma^{g_i}_{c_7}$ are shown. (f) The contact grids $\Omega^{\zeta_1}$, $\Omega^{\zeta_2}$, $\Omega^{\zeta_3}$, $\Omega^{\zeta_4}$ (shades of cyan) cover $\Gamma^{g_i}_{c_{1,c_9}}$, $\Gamma^{g_i}_{c_{2-3}}$, $\Gamma^{g_i}_{c_4}$, $\Gamma^{g_i}_{c_{5-8}}$, respectively. The boundary $\partial\Omega^{\zeta_k}$ consists of $\Gamma^{\zeta_k}_w$ (black) and $\Gamma^{\zeta_k}_g$ (red).

where $\boldsymbol{f}$ is the body force, $\nabla\cdot$ the divergence operator, $\boldsymbol{\sigma}$ the stress tensor, and $\boldsymbol{\varepsilon}$ the strain tensor. The constitutive stress-strain relation for an elastic material is:

$$\boldsymbol{\sigma}(\boldsymbol{u}) = \mathbf{C} : \boldsymbol{\varepsilon}(\boldsymbol{u}) \tag{2}$$

where $\boldsymbol{\varepsilon}(\boldsymbol{u}) = \nabla^s \boldsymbol{u} = 1/2\,(\nabla \boldsymbol{u} + \nabla^\top \boldsymbol{u})$ and $\mathbf{C} = [\mathsf{C}_{ijkl}]$ is the fourth-order stiffness tensor. Note $\nabla^s$ denotes the symmetric gradient operator, $\boldsymbol{u}$ the displacement field, and the superscript $\top$ transposition. For isotropic materials, $\mathbf{C}$ equals:

$$\mathsf{C}_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) \tag{3}$$

where $\lambda$ and $\mu$ are called Lamé parameters. Substituting Eq.3 into Eq.2, we obtain:

$$\boldsymbol{\sigma}(\boldsymbol{u}) = \lambda\,\mathrm{tr}(\boldsymbol{\varepsilon}(\boldsymbol{u}))\boldsymbol{I} + 2\mu\,\boldsymbol{\varepsilon}(\boldsymbol{u}) \tag{4}$$

in which $\mathrm{tr}(\boldsymbol{\varepsilon})$ is the trace of $\boldsymbol{\varepsilon}$.

Let $\partial\Omega_s$ be partitioned into a Dirichlet $\Gamma^D$ and a Neumann $\Gamma^N$ subset, such that $\Omega_s = \overline{\Gamma^D \cup \Gamma^N}$ and $\Gamma^D \cap \Gamma^N = \emptyset$. We impose displacement boundary conditions (BCs) on $\Gamma^D$ and traction BCs on $\Gamma^N$ as follows:

$$\boldsymbol{u}_{|\Gamma^D} = \boldsymbol{u}_D(\boldsymbol{x}) \tag{5a}$$

$$\boldsymbol{\sigma}(\boldsymbol{u}) \cdot \boldsymbol{n}_{|\Gamma^N} = \boldsymbol{t}_N(\boldsymbol{x}) \tag{5b}$$

where $\boldsymbol{u}_D$ is the displacement on $\Gamma^D$, and $\boldsymbol{t}_N$ and $\boldsymbol{n}$ are the traction and outward-pointing normal on $\Gamma^N$, respectively. With reference to Fig.1a, we assume $\Gamma^w \subset \Gamma^N$ and set $\boldsymbol{t}_N = \boldsymbol{0}$ on $\Gamma^w$, which implies the pore pressure is zero.

To represent cracks $\Gamma^F$ ($\subset \Omega_s$), we use a continuous damage (or phase-field) variable $\alpha$ satisfying [33]:

$$\alpha(\boldsymbol{x}) = \begin{cases} \frac{BG_c}{2l}\left(1 - \frac{\text{dist}(\boldsymbol{x},\Gamma^F)}{l/2}\right), & \text{dist}(\boldsymbol{x},\Gamma^F) < l/2 \\ 0, & \text{dist}(\boldsymbol{x},\Gamma^F) \geq l/2 \end{cases} \tag{6}$$

where $G_c$ is the fracture toughness, $l$ is a length-scale parameter that controls the width of the diffuse crack, $\text{dist}(\boldsymbol{x},\Gamma^F)$ is the distance from an arbitrary point $\boldsymbol{x}$ to $\Gamma^F$, and $B$ is a constant ($=10^3$ here). If $\alpha(\boldsymbol{x})=0$, the solid is intact at point $\boldsymbol{x}$, and if $\alpha(\boldsymbol{x})=1$, it is fully damaged. In phase-field models, the variable $\alpha$ impacts the stress-strain Eq.2 by degrading the stress tensor in certain directions (e.g., tension/shear but not compression) [63, 64]. Here, we choose an isotropic degradation model for simplicity [32], because our goal is to develop a multiscale approximation of DNS:

$$\boldsymbol{\sigma}(\boldsymbol{u}) = g(\alpha)\,\mathbf{C} : \boldsymbol{\varepsilon}(\boldsymbol{u}) \tag{7}$$

In Eq.7, $g(\alpha)=(1-\alpha)^2$ is the degradation function. If the solid is fully damaged, $g(\alpha)=0$ and a traction-free condition is imposed on $\Gamma^F$. If the solid is intact, $g(\alpha)=1$ and Eq.7 reduces to Eq.2. We note that more complex degradation functions also exist and can be used in PLMM without loss of generality [65]. We also note that if an anisotropic degradation model were used, PLMM would apply to the linearized form of the governing equations.

## 3. Single-scale modeling

The porous solid $\Omega_s$ is often represented by a digitized image (e.g., X-ray $\mu$CT) at the pore scale, as shown by Fig.1a. The pixels of this image can be used to construct a Cartesian *fine grid*, over which Eqs.1 and 7 are discretized and solved with a *fine-scale solver*. The fine grid need not have the same size as the image pixels, and is often refined along the coordinate directions. We choose the finite element method as our fine-scale solver, but note that others apply equally throughout this work (e.g., finite volume [66]). We define *single-scale modeling* as the act of using the fine-scale solver over $\Omega_s$ without any approximation or domain decomposition. This yields the DNS reference solution against which PLMM is later compared.

## 4. Multiscale modeling

We propose a pore-level multiscale method (PLMM) to approximate, and thereby accelerate, DNS. In PLMM, the following steps are executed: (1) $\Omega_s$ is decomposed into subdomains, or coarse grids, in a way that is non-conforming to $\Gamma^F$; (2) Basis and correction functions are built on each coarse grid by numerically solving local fine-scale problems; (3) The basis and correction functions are coupled by solving a global coarse-scale problem; And (4) a global fine-scale solution to Eqs.1 and 7 on $\Omega_s$ is obtained by assembling the coupled basis and correction functions. These steps are detailed in Sections 4.1-4.4, which yield an initial approximation to DNS. In Section 4.5, we present an iterative approach to correct the errors of this approximation to any desired level. We explore different strategies for incorporating cracks into the basis and correction functions in Section 4.3.

### 4.1. Domain decomposition

To decompose $\Omega_s$, we modify the algorithm proposed by Mehmani *et al.*[16] slightly. The approach involves applying watershed segmentation, a well-known morphological operation in image analysis [67], to $\Omega_s$. This partitions $\Omega_s$ into $N^g$ non-overlapping subdomains, or *grain grids*, $\Omega^{g_i}$ such that $\Omega_s = \cup_i \Omega^{g_i}$, as shown in Fig.1b. Because of the unique properties of watershed segmentation, the subdomains correspond to local enlargements (or grains) in $\Omega_s$, and interfaces between subdomains correspond to local constrictions (or contacts) in $\Omega_s$. We refer to a shared interface between two adjacent grain grids as a *contact interface*, $\Gamma^{c_j}$. Notice from Fig.1b-c that $\Gamma^F$ is allowed to intersect $\Gamma^{c_j}$. In other words, the decomposition need not conform to the cracks, which is very different from, and much more flexible than, the conforming decomposition proposed in [16]. The interested reader is referred to [5, 59, 67] for further details on the watershed segmentation and domain decomposition.

We define another set of coarse grids, $\Omega^{\zeta_k}$, called *contact grids* that overlap and cover a thin neighborhood around each contact interface, $\Gamma^{c_j}$. As seen from Figs.1c and 1f, contact grids (cyan) do not cover all of $\Omega_s$ (i.e., $\Omega_s \neq \cup_k \Omega^{\zeta_k}$).

We create $\Omega^{\zeta_k}$ by morphologically "dilating", an operation in image analysis [68], the image pixels overlaying $\Gamma^{c_j}$. The thickness of $\Omega^{\zeta_k}$ can be adjusted by the user, although ~16 pixels has been shown to be sufficient in practice (8 per contact side) [16, 59]. More details on how $\Omega^{\zeta_k}$ are constructed can be found in [59]. We refer to $\Omega^{g_i}$ and $\Omega^{\zeta_k}$ as *coarse grids* and note that they are aggregates of many fine grids, which, in this work, conform at $\Omega^{g_i} \cap \Omega^{\zeta_k}$. Such conformity is not required by PLMM and different fine grids (e.g., unstructured) may, in principle, be used on $\Omega^{g_i}$ and $\Omega^{\zeta_k}$ as long as appropriate mappings between the two can be defined.

### 4.2. Mathematical notation

In the following sections, we adopt a notation similar to [16] to formulate PLMM. Let the indices $g_i$, $c_j$, $\zeta_k$ enumerate entities associated with $\Omega^{g_i}$, $\Gamma^{c_j}$, $\Omega^{\zeta_k}$, respectively. The boundary $\partial\Omega^{g_i}$ is partitioned as $\partial\Omega^{g_i} = \Gamma_w^{g_i} \cup \Gamma_{c_j}^{g_i}$, where $\Gamma_w^{g_i} = \partial\Omega^{g_i} \cap \Gamma^w$ is the void-solid interface (black in Fig.1e) and $\Gamma_{c_j}^{g_i} = \partial\Omega^{g_i} \cap \Gamma^{c_j}$ is a contact interface (red in Fig.1e). Similarly, $\partial\Omega^{\zeta_k}$ consists of $\Gamma_w^{\zeta_k} = \partial\Omega^{\zeta_k} \cap \Gamma^w$ (black in Fig.1f) and $\Gamma_{g_i}^{\zeta_k} = \partial\Omega^{\zeta_k} \cap \Omega^{g_i}$, $\forall g_i$ (red in Fig.1f), where the latter corresponds to the portion of $\partial\Omega^{\zeta_k}$ that intersects the interior of grain grids (see also Fig.2). If $\partial\Omega^{g_i}$ or $\partial\Omega^{\zeta_k}$ intersect the bounding box of $\Omega_s$ ($= \partial\Omega_s \backslash \Gamma^w$, i.e., four sides of Fig.1a), then the global BCs from Eq.5 are inherited.

Because two grain grids may share more than one contact interface (see Fig.2), and one contact grid may cover multiple interfaces (see Figs.1-2), we define the following mappings between $g_i$, $c_j$ and $\zeta_k$: (1) $C^{g_i} = \{c_j \,|\, \Gamma_{c_j}^{g_i} \neq \emptyset\}$ is the index set of all $\Gamma^{c_j}$ that are subsets of $\partial\Omega^{g_i}$; (2) $G^{c_j} = \{g_i \,|\, \Gamma_{c_j}^{g_i} \neq \emptyset\}$ is the index set of all grain grids sharing $\Gamma^{c_j}$. Since $\Gamma^{c_j}$ is shared by exactly two grain grids, $G^{c_j}$ has only two members ($= \{g_{i_1}, g_{i_2}\}$); (3) $Z^{c_j} = \{\zeta_k \,|\, \Gamma^{c_j} \subseteq \Omega^{\zeta_k}\}$ is the index set of all contact grids covering $\Gamma^{c_j}$. Notice $Z^{c_j}$ has only one member, which we denote by $\zeta^{c_j}$; And (4) $C^{\zeta_k} = \{c_j \,|\, \Gamma^{c_j} \subseteq \Omega^{\zeta_k}\}$ is the index set of all contact interfaces covered by $\Omega^{\zeta_k}$. We use the symbol # is to denote the number of members in a set. For example, $\#C^{g_i} = 9$ in Fig.1e, $\#C^{\zeta_4} = 4$ in Fig.1f, $\#G^{c_j} = 2$, and $\#Z^{c_j} = 1$. Finally, we use $N^g$, $N^c$, $N^\zeta$ to denote the total number of grain grids, contact interfaces, and contact grids in $\Omega_s$.

The terms *fine-scale* and *coarse-scale* refer to entities associated with the fine grid (Section 3) and coarse grid ($\Omega^{g_i}$ or $\Omega^{\zeta_k}$), respectively. Fine-scale variables are specified by superscript $f$, while coarse-scale variable by superscript $o$. An entity is *local* if it is defined on only one coarse grid, and *global* if it is defined over the whole domain $\Omega_s$. To simplify the exposition, we use a 2D notation to present all PLMM equations. For example, the symbol $\boldsymbol{m}$ is used to denote the tangent vector on a boundary or interface (Fig.1e) . In 2D, $\boldsymbol{m}$ is a single vector, but in 3D, it consists of two orthonormal vectors $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$. We use the former compact notation to express tangential BCs.

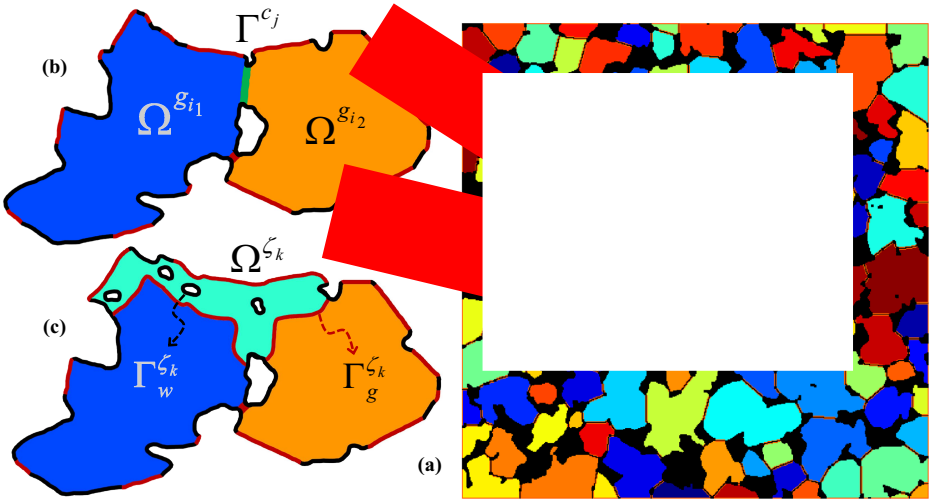



Figure 2: Schematic of a contact interface $\Gamma^{c_j}$ (green) between two grain grids $\Omega^{g_{i_1}}$ and $\Omega^{g_{i_2}}$. Notice $\Gamma^{c_j}$ is one of two shared contacts. $\Gamma_{c_j}^{g_{i_1}}$ corresponds to the $-$ side of $\Gamma^{c_j}$ and $\Gamma_{c_j}^{g_{i_2}}$ to its $+$ side. The global problem imposes continuity of traction and displacement on $\Gamma^{c_j}$. (c) The contact grid $\Omega^{\zeta_k}$ (cyan) covers $\Gamma^{c_j}$, among other contacts (red in b).

### 4.3. Local problems

The goal of PLMM is to solve Eqs.1 and 7 restricted to each grain grid $\Omega^{g_i}$:

$$\nabla \cdot \left( g(\alpha)\, \mathbf{C} : \nabla^s \boldsymbol{u}^f_{g_i} \right) = \boldsymbol{f} \qquad s.t. \qquad \begin{cases} \boldsymbol{u}^f_{g_i} \cdot \boldsymbol{n} \,|_{\Gamma^{g_i}_{c_j}} = h_{g_i c_j 1}(\boldsymbol{x}), & \forall c_j \in C^{g_i} \\ \boldsymbol{u}^f_{g_i} \cdot \boldsymbol{m} \,|_{\Gamma^{g_i}_{c_j}} = h_{g_i c_j 2}(\boldsymbol{x}), & \forall c_j \in C^{g_i} \end{cases} \tag{8}$$

where $h_{g_i c_j 1}(\boldsymbol{x})$ and $h_{g_i c_j 2}(\boldsymbol{x})$ are normal and tangential displacements on $\Gamma^{g_i}_{c_j}$, respectively, and $\boldsymbol{x}$ is the position vector on $\Gamma^{g_i}_{c_j}$. Given the BCs on $\Gamma^{g_i}_{w}$ are stress-free (i.e., $\boldsymbol{\sigma}^f_{g_i}(\boldsymbol{u}) \cdot \boldsymbol{n} = \boldsymbol{0}$), we omit them from all subsequent local problems for brevity. Note that in 3D, the tangential displacement BC in Eq.8 consists of two separate equations corresponding to two orthonormal tangents on $\Gamma^{g_i}_{c_j}$. If Eq.8 is solved, the global fine-scale solution can be assembled from the local solutions $\boldsymbol{u}^f_{g_i}$ on each grain grid $\Omega^{g_i}$. However, because $h_{g_i c_j 1}(\boldsymbol{x})$ and $h_{g_i c_j 2}(\boldsymbol{x})$ are unknown, Eq.8 cannot be solved. To make progress, we introduce the following *localization assumption*:

$$h_{g_i c_j d}(\boldsymbol{x}) \approx u^o_{g_i c_j d}, \quad d = 1, \ldots, D \tag{9}$$

in which we have replaced the functions $h_{g_i c_j 1}(\boldsymbol{x})$ and $h_{g_i c_j 2}(\boldsymbol{x})$ with unknown scalars $u^o_{g_i c_j 1}$ and $u^o_{g_i c_j 2}$. The latter are referred to as *coarse-scale unknowns*. The localization assumption is justified by the fact that $\Gamma^{g_i}_{c_j}$ corresponds to a local constriction of $\Omega_s$ (i.e., $\Gamma^{g_i}_{c_j}$ has a small measure), which is a unique feature of the decomposition discussed in Section 4.1. It is, therefore, reasonable to neglect the dependence of the BCs in Eq.8 on the position vector $\boldsymbol{x}$.

Substituting Eq.9 into Eq.8, we obtain:

$$\nabla \cdot \left( g(\alpha)\, \mathbf{C} : \nabla^s \boldsymbol{u}^f_{g_i} \right) = \boldsymbol{f} \qquad s.t. \qquad \begin{cases} \boldsymbol{u}^f_{g_i} \cdot \boldsymbol{n} \,|_{\Gamma^{g_i}_{c_j}} = u^o_{g_i c_j 1}, & \forall c_j \in C^{g_i} \\ \boldsymbol{u}^f_{g_i} \cdot \boldsymbol{m} \,|_{\Gamma^{g_i}_{c_j}} = u^o_{g_i c_j 2}, & \forall c_j \in C^{g_i} \end{cases} \tag{10}$$

The main advantage of Eq.10 over Eq.8 is that the solution of the former can be written as the superposition of a set of numerically-constructed *basis functions*, $\boldsymbol{\varphi}^f_{g_i c_k d}$, and one *correction function*, $\widetilde{\boldsymbol{\varphi}}^f_{g_i}$, on $\Omega^{g_i}$:

$$\boldsymbol{u}^f_{g_i} = \widetilde{\boldsymbol{\varphi}}^f_{g_i} + \sum_{\forall c_k \in C^{g_i}} \sum_{d=1}^{D} u^o_{g_i c_k d} \boldsymbol{\varphi}^f_{g_i c_k d} \tag{11}$$

for arbitrary $u^o_{g_i c_j d}$. In Sections 4.3.1–4.3.3, we present three approaches, referred to as Strategies A, N, and U, for computing $\boldsymbol{\varphi}^f_{g_i c_k d}$ and $\widetilde{\boldsymbol{\varphi}}^f_{g_i}$. They differ in the way the solution around cracks is captured. Section 4.4 outlines how $u^o_{g_i c_j d}$ is computed. Given $\boldsymbol{u}^f_{g_i}$ from Eq.11, PLMM produces a global approximate fine-scale solution via:

$$\boldsymbol{u}^f \,|_{\Omega_s} = \sum_{\forall g_i} \boldsymbol{u}^f_{g_i} \tag{12}$$

where each $\boldsymbol{u}^f_{g_i}$ is extended by zero outside of $\Omega^{g_i}$.

#### 4.3.1. Strategy A: include cracks in all basis functions

The simplest approach for computing $\boldsymbol{\varphi}^f_{g_i c_k d}$ and $\widetilde{\boldsymbol{\varphi}}^f_{g_i}$ is to solve the following *basis problem*:

$$\nabla \cdot \left( g(\alpha)\, \mathbf{C} : \nabla^s \boldsymbol{\varphi}^f_{g_i c_k d} \right) = \boldsymbol{0} \qquad s.t. \qquad \begin{cases} \boldsymbol{\varphi}^f_{g_i c_k d} \cdot \boldsymbol{n} \,|_{\Gamma^{g_i}_{c_j}} = \delta_{kj}\delta_{1d}, & \forall c_j \in C^{g_i} \\ \boldsymbol{\varphi}^f_{g_i c_k d} \cdot \boldsymbol{m} \,|_{\Gamma^{g_i}_{c_j}} = \delta_{kj}\delta_{2d}, & \forall c_j \in C^{g_i} \end{cases} \tag{13}$$

and the following *correction problem*:

$$\nabla \cdot \left( g(\alpha)\, \mathbf{C} : \nabla^s \widetilde{\boldsymbol{\varphi}}^f_{g_i} \right) = \boldsymbol{f} \qquad s.t. \qquad \begin{cases} \widetilde{\boldsymbol{\varphi}}^f_{g_i} \cdot \boldsymbol{n} \,|_{\Gamma^{g_i}_{c_j}} = 0, & \forall c_j \in C^{g_i} \\ \widetilde{\boldsymbol{\varphi}}^f_{g_i} \cdot \boldsymbol{m} \,|_{\Gamma^{g_i}_{c_j}} = 0, & \forall c_j \in C^{g_i} \end{cases} \tag{14}$$

on $\Omega^{g_i}$. In Eq.13, $\delta_{kj}$, $\delta_{1d}$, and $\delta_{2d}$ are the Kronecker delta. If $d=1$, $\boldsymbol{\varphi}^f_{g_i c_k d}$ corresponds to setting the normal displacement on $\Gamma^{g_i}_{c_k}$ to one, and the tangential displacement on $\Gamma^{g_i}_{c_k}$ and the normal/tangential displacements on $\Gamma^{g_i}_{c_j}, \forall\, j \neq k \in C^{g_i}$ to zero. Similarly, if $d=2$, only the tangential displacement on $\Gamma^{g_i}_{c_k}$ is set to one, while all other displacement BCs on all contacts are set to zero (Fig.1d). The basis problem accounts for inhomogeneities due to the BCs of Eq.10, and the correction problem for the inhomogeneity caused by the body force $\boldsymbol{f}$. If parts of $\partial\Omega^{g_i}$ intersect the external boundary of $\Omega_s$ $(=\partial\Omega_s \setminus \Gamma^w)$, then Eqs.13 and 14 inherit the global BCs in Eq.5. However, the right-hand side (RHS) of these BCs is set to zero for Eq.13 (homogeneous) but not Eq.14 (inhomogeneous). Given $\boldsymbol{\varphi}^f_{g_i c_k d}$ and $\widetilde{\boldsymbol{\varphi}}^f_{g_i}$ from Eqs.13 and 14, it is easy to verify that Eq.11 satisfies Eq.10. We call this approach for computing $\boldsymbol{\varphi}^f_{g_i c_k d}$ and $\widetilde{\boldsymbol{\varphi}}^f_{g_i}$ Strategy A, a key feature of which is that the degradation function, $g(\alpha)$, is included in the basis problems of *all* grain grids.

**Remark 1.** Eq.13 entails solving $\#C^{g_i} \times D$ linear systems on $\Omega^{g_i}$, one per contact interface per normal/tangential BC, whereas Eq.14 entails solving only one system. If cracks evolve, so does $g(\alpha)$, and the basis functions must be updated at every loading step, which can become computationally expensive. To reap maximum speedup, basis functions must be updated sparingly, which makes Strategy A best suited for fixed-crack, not evolving-crack, problems. Note the basis and correction problems are fully decoupled across all grain grids and can be solved in parallel.

*4.3.2. Strategy N: exclude cracks from all basis functions*

Since updating basis functions in evolving-crack problems is expensive, a naïve strategy is to never update them at all. In other words, we may altogether neglect the presence of cracks in formulating the basis problem on $\Omega^{g_i}$:

$$\nabla \cdot \left( \mathbf{C} : \nabla^s \boldsymbol{\varphi}^f_{g_i c_k d} \right) = \mathbf{0} \qquad s.t. \qquad \begin{cases} \boldsymbol{\varphi}^f_{g_i c_k d} \cdot \boldsymbol{n}\,|_{\Gamma^{g_i}_{c_j}} = \delta_{kj}\delta_{1d}\,, & \forall c_j \in C^{g_i} \\ \boldsymbol{\varphi}^f_{g_i c_k d} \cdot \boldsymbol{m}\,|_{\Gamma^{g_i}_{c_j}} = \delta_{kj}\delta_{2d}\,, & \forall c_j \in C^{g_i} \end{cases} \tag{15}$$

We call this Strategy N. Notice the degradation function, $g(\alpha)$, does not appear in Eq.15, and $\boldsymbol{\varphi}^f_{g_i c_k d}$ corresponds to an intact grain grid. Hence, Eq.15 need only be solved once. To account for cracks, a modified correction problem is needed because if we substitute $\widetilde{\boldsymbol{\varphi}}^f_{g_i}$ from Eq.14 into Eq.11, we obtain $\boldsymbol{u}^f_{g_i}$ that does not satisfy Eq.10, unless $\Omega^{g_i}$ is intact. In other words, the following *source residual*:

$$\Re_s(\boldsymbol{u}^f_{g_i}) = f_s - \nabla \cdot \left( g(\alpha)\, \mathbf{C} : \nabla^s \boldsymbol{u}^f_{g_i} \right) \tag{16}$$

will be non-zero. To drive $\Re_s$ to zero, we propose an iterative scheme in which the modified correction problem:

$$\nabla \cdot \left( g(\alpha)\, \mathbf{C} : \nabla^s \delta\widetilde{\boldsymbol{\varphi}}^{f,\eta}_{g_i} \right) = \Re_s(\boldsymbol{u}^{f,\eta-1}_{g_i}) \qquad s.t. \qquad \begin{cases} \widetilde{\boldsymbol{\varphi}}^f_{g_i} \cdot \boldsymbol{n}\,|_{\Gamma^{g_i}_{c_j}} = 0\,, & \forall c_j \in C^{g_i} \\ \widetilde{\boldsymbol{\varphi}}^f_{g_i} \cdot \boldsymbol{m}\,|_{\Gamma^{g_i}_{c_j}} = 0\,, & \forall c_j \in C^{g_i} \end{cases} \tag{17}$$

is solved to obtain the *incremental correction function*, $\delta\widetilde{\boldsymbol{\varphi}}^{f,\eta}_{g_i}$ on $\Omega^{g_i}$ at iteration $\eta$. Notice the RHS of Eq.17 depends on the reconstructed solution $\boldsymbol{u}^{f,\eta-1}_{g_i}$ from the previous iteration. To compute it, we use a modified form of Eq.11:

$$\boldsymbol{u}^{f,\eta}_{g_i} = \widetilde{\boldsymbol{\varphi}}^{f,\eta}_{g_i} + \sum_{\forall c_k \in C^{g_i}} \sum_{d=1}^{D} u^{o,\eta}_{g_i c_k d} \boldsymbol{\varphi}^f_{g_i c_k d} \tag{18a}$$

$$\widetilde{\boldsymbol{\varphi}}^{f,\eta}_{g_i} = \gamma\, \delta\widetilde{\boldsymbol{\varphi}}^{f,\eta}_{g_i} + \widetilde{\boldsymbol{\varphi}}^{f,\eta-1}_{g_i} \tag{18b}$$

where $\gamma$ is a relaxation parameter. We refer to iterations in $\eta$ as *source iterations*, which we declare "converged" if $\|\Re_s(\boldsymbol{u}^{f,\eta}_{g_i})\|_{L_2} < 10^{-2}$. We denote the total number of source iterations performed by $n_s$. To start iterating, we set $\delta\widetilde{\boldsymbol{\varphi}}^{f,0}$ to zero, set $\widetilde{\boldsymbol{\varphi}}^{f,0}$ to the solution of Eq.14, and compute $\boldsymbol{u}^{f,0}_{g_i}$ from Eq.11 with $\widetilde{\boldsymbol{\varphi}}^{f,0}$ substituting the correction function. The only remaining unknown in Eq.18 is $u^{o,\eta}_{g_i c_k d}$, whose computation we detail in Section 4.4.

**Remark 2.** In Section 6, we show that setting $\gamma = 1$ can cause source iterations to diverge if $\Gamma^F$ intersects any of the contact interfaces of $\Omega^{g_i}$. The remedy is to use a smaller $\gamma \in (0, 1]$. However, this slows convergence and the optimal value of $\gamma$ is not known a priori. Therefore, Strategy N is not attractive for arbitrary crack patterns.

#### 4.3.3. Strategy U: adaptively include cracks in some basis functions

An ideal formulation of basis and correction problems is one where basis functions are updated infrequently and $\gamma = 1$ can be set in Eq.18 without source iterations diverging. As stated in Remark 2, Strategy N diverges if $\gamma = 1$ and $\Gamma^F \cap \Gamma^{g_i}_{c_j} \neq \emptyset$ for some $c_j \in C^{g_i}$. In such cases, we empirically found that including $\Gamma^F$ in the basis problem of $\Omega^{g_i}$ ensures convergence (see Section 6). Concretely, we formulate the basis problem on $\Omega^{g_i}$ as follows:

$$\begin{cases} \nabla \cdot \left( g(\alpha)\, \mathbf{C} : \nabla^s \boldsymbol{\varphi}^f_{g_i c_k d} \right) = \mathbf{0}\,, & \Gamma^F \cap \Gamma^{g_i}_{c_j} \neq \emptyset\,, \quad \exists c_j \in C^{g_i} \\ \nabla \cdot \left( \mathbf{C} : \nabla^s \boldsymbol{\varphi}^f_{g_i c_k d} \right) = \mathbf{0}\,, & \Gamma^F \cap \Gamma^{g_i}_{c_j} = \emptyset\,, \quad \forall c_j \in C^{g_i} \end{cases} \quad s.t. \quad \begin{cases} \boldsymbol{\varphi}^f_{g_i c_k d} \cdot \boldsymbol{n}\,|_{\Gamma^{g_i}_{c_j}} = \delta_{kj}\delta_{1d}\,, & \forall c_j \in C^{g_i} \\ \boldsymbol{\varphi}^f_{g_i c_k d} \cdot \boldsymbol{m}\,|_{\Gamma^{g_i}_{c_j}} = \delta_{kj}\delta_{2d}\,, & \forall c_j \in C^{g_i} \end{cases} \tag{19}$$

where the degradation function, $g(\alpha)$, is included if $\Gamma^F$ intersects at least one of the interfaces of $\Omega^{g_i}$. The corresponding correction problem then is Eq.14 (similar to Strategy A). If $\Gamma^F \cap \Gamma^{g_i}_{c_j} = \emptyset$ for all $c_j \in C^{g_i}$, then Eq.17 is used to compute the correction function (similar to Strategy N). Note that if $\Omega^{g_i}$ does not contain any cracks (i.e. $\Omega^{g_i} \cap \Gamma^F = \emptyset$), Eq.17 reduces to Eq.14 and converges in only one source iteration. We call this approach for computing basis and correction functions Strategy U. In evolving-crack problems, the basis functions of $\Omega^{g_i}$ are updated only if $\Gamma^F$ crosses one of its interfaces with adjacent grain grids. The rationale for the update is that the localization assumption in Eq.9 thus the PLMM solution, degrades in accuracy. Having computed all basis and correction functions on $\Omega$ via one of Strategies A, N, and U, we proceed next to compute $u^o_{g_i c_j d}$ by solving a global problem. We note that global problem is also needed as part of internal source iterations in Strategies N and U.

### 4.4. Global problem

Recall the coarse-scale unknown $u^o_{g_i c_j d}$ corresponds to the normal displacement if $d = 1$, or the tangential displacement if $d = 2$ (or 3 in 3D) on $\Gamma^{g_i}_{c_j}$, respectively. For compactness, define $\boldsymbol{u}^o_{g_i c_j} = [u^o_{g_i c_j 1}, ..., u^o_{g_i c_j D}]$. To compute $\boldsymbol{u}^o_{g_i c_j}$, we formulate a global problem by imposing *momentum balance* and *kinematic constraints* on all contact interfaces. Let $\Gamma^{c_j}$ be shared between $\Omega^{g_{i_1}}$ and $\Omega^{g_{i_2}}$ with the two corresponding sides $\Gamma^{g_{i_1}}_{c_j}$ and $\Gamma^{g_{i_2}}_{c_j}$ (Fig.2b). The coarse-scale displacements associated with each side are $\boldsymbol{u}^o_{g_{i_1} c_j}$ and $\boldsymbol{u}^o_{g_{i_2} c_j}$, respectively totalling $2{\times}D$ unknowns. An equal number of equations on $\Gamma^{c_j}$ are required to close the system. The first equation is a momentum (or force) balance:

$$\boldsymbol{t}^o_{g_{i_1} c_j} = \boldsymbol{t}^o_{g_{i_2} c_j} \tag{20}$$

where $\boldsymbol{t}^o_{g_{i_1} c_j}$ and $\boldsymbol{t}^o_{g_{i_2} c_j}$ are integrated tractions on $\Gamma^{g_{i_1}}_{c_j}$ and $\Gamma^{g_{i_2}}_{c_j}$, respectively. The second is displacment continuity:

$$\boldsymbol{u}^o_{g_{i_1} c_j} = \boldsymbol{u}^o_{g_{i_2} c_j} \tag{21}$$

across $\Gamma^{c_j}$, justified by the diffuse-crack representation adopted in Section 2.

To tranform Eqs.20 and 21 into an algebraic system in terms of $u^o_{g_i c_j d}$, we must express $\boldsymbol{t}^o_{g_{i_1} c_j}$ and $\boldsymbol{t}^o_{g_{i_2} c_j}$ in terms of $u^o_{g_i c_j d}$. Following the approach of [11], we first derive an expression for $\boldsymbol{t}^o_{g_i c_j}$ defined on $\Gamma^{g_i}_{c_j}$ and drop the superscripts 1 and 2 momentarily for convenience. We begin by upscaling the basis functions on $\Omega^{g_i}$ via:

$$\boldsymbol{\psi}^f_{g_i c_k d} = g(\alpha)\, \mathbf{C} : \nabla^s \boldsymbol{\varphi}^f_{g_i c_k d} \tag{22a}$$

$$\boldsymbol{t}^o_{g_i c_j d} = \frac{1}{|\Gamma^{c_j}|} \int_{\Gamma^{g_i}_{c_j}} \boldsymbol{\psi}^f_{g_i c_k d} \cdot \mathrm{d}\boldsymbol{s} \tag{22b}$$

$$T^o_{g_i c_k d} = \left[ \boldsymbol{t}^o_{g_i c_k c_{j_1} d},\ \boldsymbol{t}^o_{g_i c_k c_{j_2} d},\ ... \right]^\top_{(\#C^{g_i} D)\times 1} \tag{22c}$$

$$B^o_{g_i} = \left[ T^o_{g_i c_{k_1} 1} \,|\, ... \,|\, T^o_{g_i c_{k_1} D} \,|\, T^o_{g_i c_{k_2} 1} \,|\, ... \,|\, T^o_{g_i c_{k_2} D} \,|\, ... \right]_{(\#C^{g_i} D)\times(\#C^{g_i} D)} \tag{22d}$$

Eq.22a yields the fine-scale stress $\boldsymbol{\psi}^f_{g_i c_k d}$ associated with $\boldsymbol{\varphi}^f_{g_i c_k d}$. Eq.22b computes the spatial average of $\boldsymbol{\psi}^f_{g_i c_k d}$ along $\Gamma^{g_i}_{c_j}$, where $|\Gamma^{c_j}|$ is the measure of $\Gamma^{c_j}$. Eq.22c concatenates the $1{\times}D$ coarse-scale tractions $\boldsymbol{t}^o_{g_i c_k c_{j_l} d}, \forall c_{j_l} \in C^{g_i}$, into a $(\#C^{g_i} D){\times}1$ vector $T^o_{g_i c_k d}$. Finally, Eq.22d assembles $T^o_{g_i c_{k_l} d}, \forall c_{k_l} \in C^{g_i}$ into a matrix $B^o_{g_i}$ with size $(\#C^{g_i} D){\times}(\#C^{g_i} D)$. In effect, Eq.22 upscales the fine-scale basis functions $\boldsymbol{\varphi}^f_{g_i c_k d}$ on $\Omega^{g_i}$ into the small matrix $B^o_{g_i}$.

Analogously, we upscale the correction function on $\Omega^{g_i}$ via:

$$\widetilde{\boldsymbol{\psi}}^f_{g_i} = g(\alpha)\,\mathbf{C} : \nabla^s \widetilde{\boldsymbol{\varphi}}^f_{g_i} \tag{23a}$$

$$\widetilde{\boldsymbol{t}}^o_{g_i c_j} = \frac{1}{|\Gamma^{c_j}|}\int_{\Gamma^{g_i}_{c_j}} \widetilde{\boldsymbol{\psi}}^f_{g_i} \cdot \mathrm{d}\boldsymbol{s} \tag{23b}$$

$$\widetilde{T}^o_{g_i} = \left[\widetilde{\boldsymbol{t}}^o_{g_i c_{j_1}}, \widetilde{\boldsymbol{t}}^o_{g_i c_{j_2}}, \ldots\right]^\top_{(\#C^{g_i}D)\times 1} \tag{23c}$$

Eq. 23a computes the fine-scale stress $\widetilde{\boldsymbol{\psi}}^f_{g_i}$ associated with $\widetilde{\boldsymbol{\varphi}}^f_{g_i}$. Eq. 23b computes a spatial average of this stress on $\Gamma^{g_i}_{c_j}$ to yield the $1\times D$ traction vector $\widetilde{\boldsymbol{t}}^o_{g_i c_j}$. Finally, Eq. 23c concatenates these tractions into the $(\#C^{g_i}D)\times 1$ vector $\widetilde{T}^o_{g_i}$. In effect, Eq. 23 upscales the fine-scale correction function $\widetilde{\boldsymbol{\varphi}}^f_{g_i}$ on $\Omega^{g_i}$ into the small vector $\widetilde{T}^o_{g_i}$.

**Remark 3.** Regradless of whether Strategy A, N, or U is used to compute the basis and correction functions, the degradation function, $g(\alpha)$, must be included in the upscaling of $\boldsymbol{\varphi}^f_{g_i c_k d}$ and $\widetilde{\boldsymbol{\varphi}}^f_{g_i}$ via Eqs. 22 and 23.

With $B^o_{g_i}$ and $\widetilde{T}^o_{g_i}$ computed for $\Omega^{g_i}$, we can now express the coarse-scale traction $\boldsymbol{t}^o_{g_i c_j}$ on $\Gamma^{g_i}_{c_j}$ as a function of the coarse-scale displacements $\boldsymbol{u}^o_{g_i c_{j_l}}, \forall c_{j_l} \in C^{g_i}$ via Eq. 24a:

$$T^o_{g_i} = B^o_{g_i}\, U^o_{g_i} + \widetilde{T}^o_{g_i} \tag{24a}$$

$$U^o_{g_i} = \left[\boldsymbol{u}^o_{g_i c_{j_1}}, \boldsymbol{u}^o_{g_i c_{j_2}}, \ldots\right]^\top_{(\#C^{g_i}D)\times 1} \tag{24b}$$

$$T^o_{g_i} = \left[\boldsymbol{t}^o_{g_i c_{j_1}}, \boldsymbol{t}^o_{g_i c_{j_2}}, \ldots\right]^\top_{(\#C^{g_i}D)\times 1} \tag{24c}$$

Eq. 24a superposes the integrated tractions on $\Gamma^{g_i}_{c_j}$ computed from the basis and correction functions on $\Omega^{g_i}$. Notice $U^o_{g_i}$ is a $(\#C^{g_i}D)\times 1$ vector of the coarse-scale displacements $\boldsymbol{u}^o_{g_i c_{j_l}}, \forall c_{j_l} \in C^{g_i}$. Similarly, $T^o_{g_i}$ is a $(\#C^{g_i}D)\times 1$ vector of the coarse-scale tractions $\boldsymbol{t}^o_{g_i c_{j_1}}, \forall c_{j_l} \in C^{g_i}$. Intuitively, Eq. 24a is an upscaled representation of the mechanical response of $\Omega^{g_i}$, which relates the coarse-scale tractions on the grain grid's interfaces to the displacements imposed on them. In the literature, the matrix $B^o_{g_i}$ has been referred to as a *flu[illegible]trix* [32].

Given Eq. 24a, the momentum balance Eq. 20 on $\Gamma^{c_j}$ can now be expressed as follows:

$$\left(B^o_{g_{i_1}}\, U^o_{g_{i_1}} + \widetilde{T}^o_{g_{i_1}}\right)_{c_j} = \left(B^o_{g_{i_2}}\, U^o_{g_{i_2}} + \widetilde{T}^o_{g_{i_2}}\right)_{c_j} \tag{25}$$

where we have used the notation $\boldsymbol{t}^o_{g_{i_l} c_j} = (T^o_{g_{i_l}})_{c_j}$ for $l \in \{1, 2\}$. Specifically, $(T^o_{g_{i_l}})_{c_j}$ is the entry in $T^o_{g_{i_l}}$ that corresponds to $\Gamma^{c_j}$. Note that $\boldsymbol{t}^o_{g_{i_l} c_j}$ not only depends on $\boldsymbol{u}^o_{g_{i_l} c_j}$, but also on the coarse-scale displacements on all other contacts of $\Omega^{g_{i_l}}$. Concretely, Eq. 25 depends on $\boldsymbol{u}^o_{g_{i_1} c_n}, \forall c_n \in C^{g_{i_1}}$, and $\boldsymbol{u}^o_{g_{i_2} c_m}, \forall c_m \in C^{g_{i_2}}$. The *global problem* is obtained by assembling Eqs. 25 and 21, written for all $\Gamma^{c_j}$, into a single algebraic system:

$$\Re\left(\mathbf{U}\right) = \Re\left([u^o_{g_i c_j d}]_{(2N^c D)\times 1}\right) = 0 \tag{26}$$

The residual vector $\Re(\cdot)$ consists of $(2N^cD)\times 1$ entries, which depend on the coarse-scale unknowns $u^o_{g_i c_j d}$ arranged into a $(2N^cD)\times 1$ vector $\mathbf{U}$. Recall $N^c$ is the total number of contacts interfaces. Since Eq. 21 imposes continuity of displacement across all $\Gamma^{c_j}$, the number of coarse-scale unknowns can be halved from $2N^cD$ to $N^cD$.

We now summarize the steps in PLMM thusfar: (a) Solve local fine-scale basis and correction problems on all $\Omega^{g_i}$ using Eqs. 13 and 17 for Strategy A, Eqs. 15 and 17 for Strategy N, or Eqs. 19 and 17 for Strategy U, respectively; (b) Solve the global problem, Eq. 26, to obtain coarse-scale displacements $u^o_{g_i c_j d}$ on all $\Gamma^{c_j}$; (c) Reconstruct the fine-scale solution $\boldsymbol{u}^f_{g_i}$ on each $\Omega^{g_i}$ via Eq. 11 for Strategy A, and Eq. 18 for Strategies N and U; And (d) assemble the global fine-scale solution $\boldsymbol{u}^f$ on $\Omega_s$ via Eq. 12. In Strategies N and U, source iterations between Steps (a) and (c) are required to compute $\widetilde{\boldsymbol{\varphi}}^f_{g_i}$, and before proceeding to Step (d). The above steps for PLMM are detailed in Algorithm 1 and yield an initial approximation to single-scale DNS, which we denote by $\mathrm{A}_0$, $\mathrm{N}_0$, or $\mathrm{U}_0$ depending on the strategy used to compute basis and correction functions. We next discuss how to iteratively reduce the errors of this approximation.

**Algorithm 1:** PLMM without boundary iterations (output: $A_0$, $N_0$, or $U_0$)

Decompose $\Omega_s$ into non-overlapping grain grids $\Omega^{g_i}$
**Do** $g_i \in \{1, ..., N^g\}$
  **Do** $c_j \in C^{g_i}$
    **Do** $d \in \{1, ..., D\}$
      **If** (Strategy A), Solve Eq.13 to obtain basis functions $\boldsymbol{\varphi}^f_{g_i c_k d}$, **End if**
      **If** (Strategy N), Solve Eq.15 to obtain basis functions $\boldsymbol{\varphi}^f_{g_i c_k d}$, **End if**
      **If** (Strategy U), Solve Eq.19 to obtain basis functions $\boldsymbol{\varphi}^f_{g_i c_k d}$, **End if**
    **End do**
  **End do**
  Solve Eq.14 to obtain correction function $\widetilde{\boldsymbol{\varphi}}^f_{g_i}$ ($= \widetilde{\boldsymbol{\varphi}}^{f,0}_{g_i}$ in source iterations)
  Upscale basis and correction functions via Eqs.22-23
**End do**
Solve global problem in Eq.26 to obtain $u^o_{g_i c_j d}, \forall g_i, c_j, d$
Reconstruct the local fine-scale solution $\boldsymbol{u}^f_{g_i}$ via Eq.11
Compute initial source residual $\Re^0_s$ via Eq.16
**While** ($\eta < n_s$ and $\Re^\eta_s > tol_s$) $\rightarrow$ *source iterations*
  Solve Eq.17 to obtain incremental correction function $\delta\widetilde{\boldsymbol{\varphi}}^{f,\eta}_{g_i}$
  Update correction function $\widetilde{\boldsymbol{\varphi}}^{f,\eta}_{g_i}$ via Eq.18b (set $\gamma = 1$ for Strategy A or U)
  Solve modified global problem in Eq.26 to obtain $u^{o,\eta}_{g_i c_j d}, \forall g_i, c_j, d$
  Reconstruct local fine-scale solution $\boldsymbol{u}^{f,\eta}_{g_i}$ via Eq.18a
  Update the source residual $\Re^\eta_s$ via Eq.16
**End while**
Assemble global fine-scale solution $\boldsymbol{u}^{f,0}$ via Eq.12

### 4.5. Iterative correction

Errors in PLMM, measured against the single-scale solution, are due to the localization assumption, $h_{g_i c_j d}(\boldsymbol{x}) \approx u^o_{g_i c_j d}$ in Eq.10. To correct them, Eq.14 must be modified as follows:

$$\nabla \cdot \left( g(\alpha)\, \mathbf{C} : \nabla^s \widetilde{\boldsymbol{\varphi}}^f_{g_i} \right) = \boldsymbol{f} \qquad s.t. \quad \begin{cases} \widetilde{\boldsymbol{\varphi}}^f_{g_i} \cdot \boldsymbol{n}\,|_{\Gamma^{g_i}_{c_j}} = h_{g_i c_j 1}(\boldsymbol{x}) - u^o_{g_i c_j 1}\,, & \forall c_j \in C^{g_i} \\ \widetilde{\boldsymbol{\varphi}}^f_{g_i} \cdot \boldsymbol{m}\,|_{\Gamma^{g_i}_{c_j}} = h_{g_i c_j 2}(\boldsymbol{x}) - u^o_{g_i c_j 2}\,, & \forall c_j \in C^{g_i} \end{cases} \tag{27}$$

where the BCs include information that is missing from Eq.14. If Eq.27 is solved instead of Eq.14, all in PLMM errors would be removed and $\boldsymbol{u}^f_{g_i}$ obtained from Eq.11 would satisfy Eq.8 exactly. However, Eq.27 cannot be solved because $h_{g_i c_j 1}(\boldsymbol{x})$ and $h_{g_i c_j 2}(\boldsymbol{x})$ are not known a priori. We therefore proceed to approximate them instead by adopting an iterative scheme, similar to ideas in [16, 45, 59]. We refer to such corrective iterations as *boundary* iterations, to differentiate them from the *source* iterations discussed in Section 4.3.

**Remark 4.** In Strategy A, replacing Eq.14 with Eq.27 would immediately yield an exact correction function. But in Strategies N and U, the replacement would merely provide a more accurate initial guess to source iterations (Eq.18), which ultimately converge to the exact correction function. In other words, Eq.17 would remain unaltered, implying the BCs of Eq.27 may fixed across source iterations. Because Eq.27 in unsolvable in practice, these statements apply equally to the approximated form of Eq.27 presented below (i.e., Eq.29).

Let $\omega$ be the boundary iteration index. Starting from $\omega - 1$, at which the global fine-scale solution $\boldsymbol{u}^{f,\omega-1}$ is known, we proceed to compute $\boldsymbol{u}^{f,\omega}$. The initial guess $\boldsymbol{u}^{f,0}$ is taken to be the initial appximation produced by PLMM (Algorithm 1). To estimate the BCs of Eq.27, we first solve a *contact problem* on each contact grid $\Omega^{\zeta_k}$ (Fig.1c):

$$\nabla \cdot \left( g(\alpha)\, \mathbf{C} : \nabla^s \boldsymbol{u}^{f,\omega}_{\zeta_k} \right) = \boldsymbol{f} \qquad s.t. \quad \begin{cases} \boldsymbol{u}^{f,\omega}_{\zeta_k} \cdot \boldsymbol{n}\,|_{\Gamma^{\zeta_k}_g} = R_{\Gamma^{\zeta_k}_g}\left[\boldsymbol{u}^{f,\omega-1}\right] \cdot \boldsymbol{n} \\ \boldsymbol{u}^{f,\omega}_{\zeta_k} \cdot \boldsymbol{m}\,|_{\Gamma^{\zeta_k}_g} = R_{\Gamma^{\zeta_k}_g}\left[\boldsymbol{u}^{f,\omega-1}\right] \cdot \boldsymbol{m} \end{cases} \tag{28}$$

where the operator $R_{\Gamma_g^{\zeta_k}}[\cdot]$ restricts $\boldsymbol{u}^{f,\omega-1}$ onto $\Gamma_g^{\zeta_k}$. The BCs on $\Gamma_w^{\zeta_k}$ are stress-free (black in Fig. 2c), while the Dirichlet BCs on $\Gamma_g^{\zeta_k}$ (red in Fig. 2c) are inherited from the fine-scale solution $\boldsymbol{u}^{f,\omega-1}$ at the previous iteration. If $\partial\Omega^{\zeta_k}$ intersects the external boundary of $\Omega_s$, that is $\partial\Omega^{\zeta_k} \cap (\partial\Omega_s \backslash \Gamma^w) \neq 0$, then the global BCs from Eq.5 are inherited.

After $\boldsymbol{u}_{\zeta_k}^{f,\omega}, \forall \zeta_k$ are computed, the correction problem Eq.27 is replaced by its approximated form:

$$\nabla \cdot \left(g(\alpha)\,\mathbf{C} : \nabla^s \widetilde{\boldsymbol{\varphi}}_{g_i}^{f,\omega}\right) = \boldsymbol{f} \qquad s.t. \qquad \begin{cases} \widetilde{\boldsymbol{\varphi}}_{g_i}^{f,\omega} \cdot \boldsymbol{n}\,|_{\Gamma_{c_j}^{g_i}} = \left(R_{\Gamma_{c_j}^{g_i}}\left[\boldsymbol{u}_{\zeta^{c_j}}^{f,\omega}\right] - \langle R_{\Gamma_{c_j}^{g_i}}\left[\boldsymbol{u}_{\zeta^{c_j}}^{f,\omega}\right]\rangle\right)\cdot \boldsymbol{n} \\ \widetilde{\boldsymbol{\varphi}}_{g_i}^{f,\omega} \cdot \boldsymbol{m}\,|_{\Gamma_{c_j}^{g_i}} = \left(R_{\Gamma_{c_j}^{g_i}}\left[\boldsymbol{u}_{\zeta^{c_j}}^{f,\omega}\right] - \langle R_{\Gamma_{c_j}^{g_i}}\left[\boldsymbol{u}_{\zeta^{c_j}}^{f,\omega}\right]\rangle\right)\cdot \boldsymbol{m} \end{cases} \tag{29}$$

where $R_{\Gamma_{c_j}^{g_i}}[\cdot]$ restricts $\boldsymbol{u}_{\zeta^{c_j}}^{f,\omega}$ onto $\Gamma_{c_j}^{g_i}$. Recall $\zeta^{c_j}$ is the index of the contact grid covering $\Gamma_{c_j}^{g_i}$ ($\subset \Gamma^{c_j}$). The operator $\langle\cdot\rangle$ spatially averages its argument over $\Gamma_{c_j}^{g_i}$. Notice $R_{\Gamma_{c_j}^{g_i}}[\boldsymbol{u}_{\zeta^{c_j}}^{f,\omega}]\cdot\boldsymbol{n}$ and $\langle R_{\Gamma_{c_j}^{g_i}}[\boldsymbol{u}_{\zeta^{c_j}}^{f,\omega}]\rangle\cdot\boldsymbol{n}$ approximate $h_{g_ic_j1}(\boldsymbol{x})$ and $u_{g_ic_j1}^{o}$ in Eq.27, respectively. The same holds for the tangential BCs in Eq.29. Once $\widetilde{\boldsymbol{\varphi}}_{g_i}^{f,\omega}$ is obtained from Eq.29, we can compute $\boldsymbol{u}^{f,\omega}$ using equations similar to Eqs.22-24. For brevity, we only list those that are updated by $\omega$:

$$\widetilde{\boldsymbol{t}}_{g_ic_j}^{o,\omega} = \frac{1}{|\Gamma^{c_j}|}\int_{\Gamma_{c_j}^{g_i}} \widetilde{\boldsymbol{\psi}}_{g_i}^{f,\omega}\cdot \mathrm{d}\boldsymbol{s} \tag{30a}$$

$$\widetilde{T}_{g_i}^{o,\omega} = \left[\widetilde{\boldsymbol{t}}_{g_ic_{j_1}}^{o,\omega}, \widetilde{\boldsymbol{t}}_{g_ic_{j_2}}^{o,\omega}, ...\right]^{\top}_{(\#C^{g_i}D)\times 1} \tag{30b}$$

$$T_{g_i}^{o,\omega} = B_{g_i}^{o}\, U_{g_i}^{o,\omega} + \widetilde{T}_{g_i}^{o,\omega} \tag{30c}$$

$$\Re\left(\mathbf{U}^{\omega}\right) = \Re\left([u_{g_ic_jd}^{o,\omega}]_{(2N^cD)\times 1}\right) = 0 \tag{30d}$$

$$\boldsymbol{u}^{f,\omega}\,|_{\Omega_s} = \sum_{\forall g_i}\left(\widetilde{\boldsymbol{\varphi}}_{g_i}^{f,\omega} + \sum_{\forall c_k \in C^{g_i}}\sum_{\forall d} u_{g_ic_kd}^{o,\omega}\boldsymbol{\varphi}_{g_ic_kd}^{f}\right) \tag{30e}$$

Notice $B_{g_i}^{o}$ and $\boldsymbol{\varphi}_{g_ic_kd}^{f}$, associated with the basis functions, remain unaltered by $\omega$. The global fine-scale solution $\boldsymbol{u}^{f,\omega}$ obtained from Eq.30e converges to the single-scale solution as $\omega \to \infty$. Eqs.28 and 29 are fully decoupled across all contact grids and grain grids and can be easily parallelized. Algorithm 2 summarizes the overall workflow of PLMM, in which $n_b$ is the total number of boundary iterations performed. We denote the global fine-scale solution $\boldsymbol{u}^{f,\omega}$ obtained from Algorithm 2 by $\mathrm{A}_\omega$, $\mathrm{N}_\omega$, or $\mathrm{U}_\omega$ depending on the strategy used to compute basis/correction functions.

**Algorithm 2:** PLMM with boundary iterations (output: $\mathrm{A}_\omega$, $\mathrm{N}_\omega$, or $\mathrm{U}_\omega$ where $\omega > 0$)

Compute initial approximation $\boldsymbol{u}^{f,0}$ by Algorithm 1 (i.e., $\mathrm{A}_0$, $\mathrm{N}_0$, or $\mathrm{U}_0$)
**Do** $\omega \in \{1, ..., n_b\}$
  **Do** $\zeta_k \in \{1, ..., N^{\zeta}\}$
    Solve Eq.28 to obtain contact-grid solution $\boldsymbol{u}_{\zeta_k}^{f,\omega}$ (inherit BCs from $\boldsymbol{u}^{f,\omega-1}$)
  **End do**
  **Do** $g_i \in \{1, ..., N^{g}\}$
    Solve Eq.29 to obtain correction function $\widetilde{\boldsymbol{\varphi}}_{g_i}^{f,\omega}$ (inhert BCs from $\bigcup_k \boldsymbol{u}_{\zeta_k}^{f,\omega}$)
    [*Note: require source iterations for Strategies N and U; see Algorithm 1*]
  **End do**
  Solve global problem in Eq.30d to obtain $u_{g_ic_jd}^{o,\omega}, \forall g_i, c_j, d$
  Reconstruct and assemble global fine-scale solution $\boldsymbol{u}^{f,\omega}$ via Eq.30e
**End do**

## 5. Validation set

We compare the proposed PLMM against single-scale DNS for different porous geometries shown in Fig.3. They include a 2D disk pack (P2D) [16], a 2D sandstone (S2D) [69], and a 3D sandstone (S3D) [69]. All have Lamé parameters $\lambda = 8.3$ GPa and $\mu = 44.3$ GPa corresponding to $\alpha$-quartz [16, 70]. In both P2D and S2D, the bottom

boundary of the domain ($y = 0$) is fixed and a vertical-upward displacement of $u_D = 0.004$ mm is applied to the top boundary ($y = L_y$). In S3D, the domain is clamped on the right-back surface ($x = 0$) and a displacement of $u_D = 0.008$ mm along the positive $x$-direction is imposed on the left-front surface ($x = L_x$). All other boundaries are stress-free.

The fine grids in Fig 3 coincide with the image pixels, which are 716 × 576 for P2D, 541 × 546 for S2D, and 134 × 136 × 142 for S3D. The phase-field length-scale parameter is set to $l = 2h = 0.02$ mm, where $h$ is the fine-grid size, and the fracture toughness to $G_c = 2.7 \times 10^{-3}$ kN/mm. Fig 3 also depicts the grain grids and contact grids of each domain obtained from the decomposition algorithm of Section 4.1. The number of grain grids, $N^g$, and contact grids, $N^\zeta$, are 76 and 127 for P2D, 121 and 145 for S2D and 43 and 52 for S3D, respectively. Contact grids are 16 pixels wide in P2D and S2D, and 12 pixels wide in S3D.

Figs 4 and 5 depict the crack patterns considered in each domain. For P2D and S2D, they include: Pattern 1, in which all cracks are confined to the interior of the grain grids, i.e., none crosses a contact interface; Pattern 2, in which all cracks cross contact interfaces, i.e., none is confined solely to the interior of a grain grid; and Pattern 3, which is a superposition of cracks in patterns 1 and 2. We use P2D-$c_1$ and S2D-$c_1$ to refer to pattern 1, P2D-$c_2$ and S2D-$c_2$ to refer to pattern 2, and P2D-$c_3$ and S2D-$c_3$ to refer to pattern 3. For S3D, we only consider one crack pattern, which consists of 20 elliptical cracks with different sizes and orientations placed randomly in the domain. Most cracks intersect contact interfaces. This pattern is denoted by S3D-$c_{20}$ and is most comparable to P2D-$c_2$ and S2D-$c_2$.

Crack pattern 2 in Fig 4 consists of two subgroups: (1) cracks parallel to contact interfaces, which sever adjacent subdomains; and (2) cracks that strike interfaces at some angle. The latter presents a more significant challenge to PLMM as the displacement *along* each interface is discontinous, invalidating the localization assumption in Eq 9. Cracks in subgroup 1 may still preserve continuity. Hence, Fig 4 considers more cracks from subgroup 2.

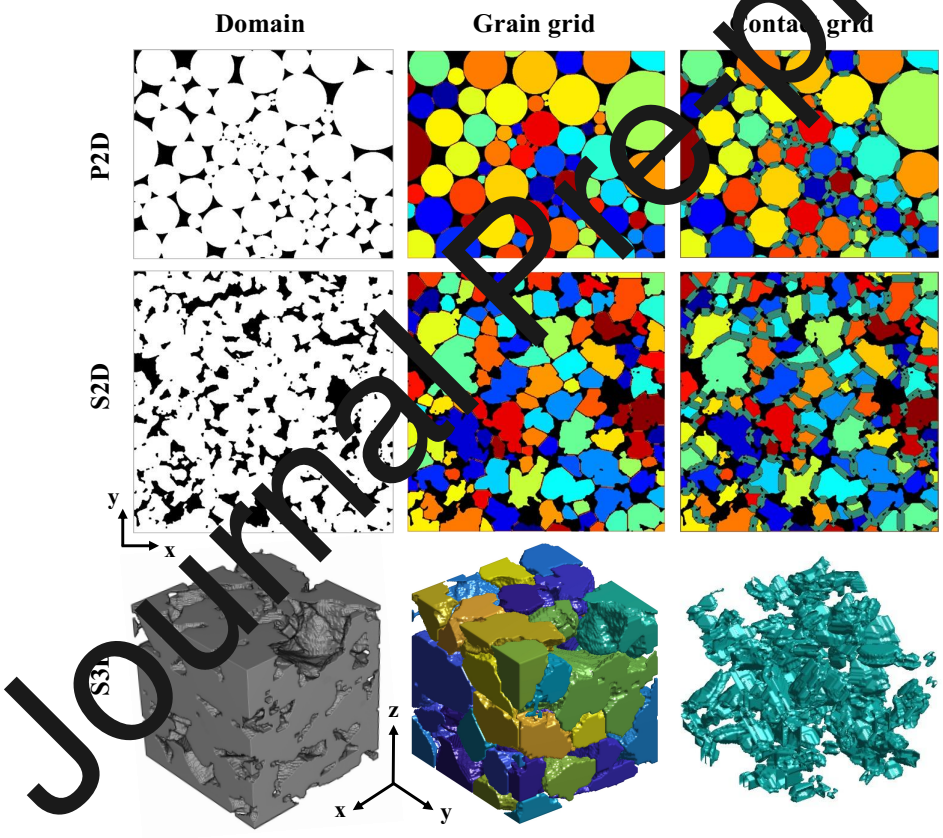



Figure 3: Domains used to validate PLMM. From top to bottom: 2D disk pack (P2D), 2D sandstone (S2D), and 3D sandstone (S3D). From left to right: domain image, grain grids (randomly colored), and contact grids (cyan).

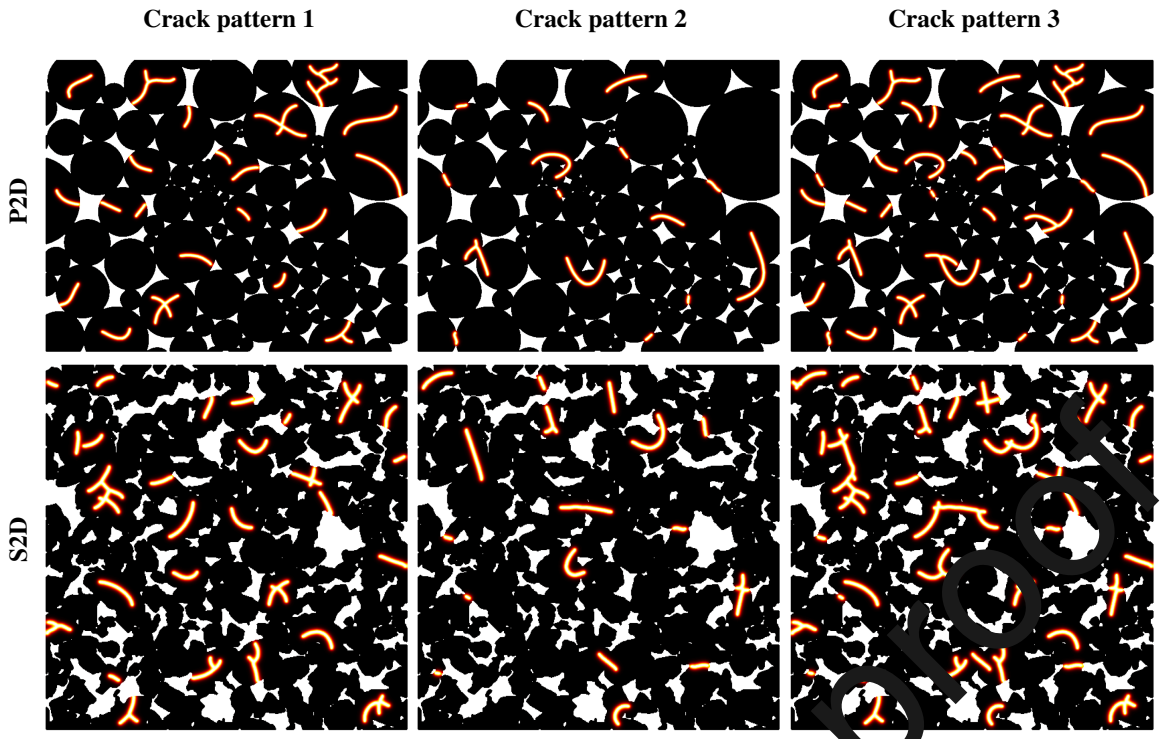



Figure 4: Crack patterns considered for the P2D and S2D domains in Fig. 3. Pattern 1: all cracks are confined to the interior of grain grids, i.e., none crosses a contact interface. Pattern 2: all cracks cross contact interfaces, i.e., none is solely confined to the interior of a grain grid. Pattern 3: superposes cracks from patterns 1 and 2.

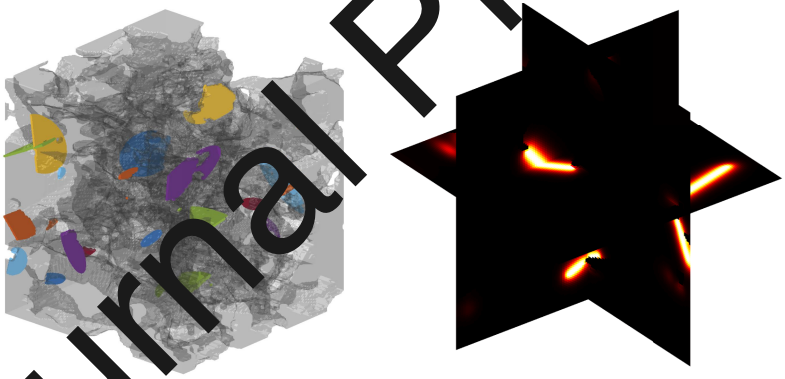


Figure 5: Crack pattern considered for the S3D domain in Fig. 3. (Left) Twenty elliptical cracks with different sizes and orientations placed randomly inside the domain. (Right) A slice plot of the phase-field variable corresponding to the cracks. The length scale parameter is set to $l = 0.02$ mm.

## 6. Results

We solve Eqs.1 and 7 on the pore-scale domains in Fig.3 via PLMM and compare the solutions against single-scale DNS. To quantify the PLMM errors, we define the following error metrics similar to [16, 59]:

$$E_p^{\chi} = \frac{\|\chi_M - \chi_S\|}{\sup_{\Omega_s}\|\chi_S\|} \times 100\,, \qquad E_2^{\chi} = \left(\frac{1}{|\Omega_s|}\int_{\Omega_s}(E_P^{\chi})^2 \mathrm{d}\Omega\right)^{1/2} \tag{31}$$

In Eq.31, $\chi$ is a placeholder for either displacement $\boldsymbol{u}$, volumetric strain $\nabla\cdot\boldsymbol{u}$, or maximum shear stress $\boldsymbol{\sigma_t}$. The subscripts $M$ and $S$ refer to variables associated with PLMM and DNS, respectively. In Eq.31, $E_p^{\chi}$ measures the relative pointwise error between PLMM and DNS, and $E_2^{\chi}$ the relative $L_2$-error, both expressed as percentages. In the following sections, PLMM solutions are computed using Strategies A, N, and U (Section 4.3) for obtaining the basis and correction functions, and different numbers of source and boundary iterations are considered (Section 4).

Before discussing the results, we note the PLMM errors ($E_p^{\chi}$ and $E_2^{\chi}$) herein are the result of three contributions:

$$E = E_b + E_s + E_f \tag{32}$$

where we have dropped the superscript $\chi$ and subscripts $p$ and 2 for clarity. $E_b$ represents errors that are corrected by boundary iterations, $E_s$ errors corrected by source iterations, and $E_f$ errors reduced by only refining the fine grid. Specifically, holding the fine-grid size, $h$, fixed, if the number of boundary, $n_b$, and source, $n_s$, iterations $\to\infty$, then $E_b$ and $E_s \to 0$, respectively. The remaining error, $E_f$, is due to fine-scale discretization differences between PLMM and DNS near contact interfaces. This is caused by our specific implementation of PLMM and can, in principle, be avoided [16, 59]. In Appendix A, we present a mesh convergence analysis that shows $E_f \sim O(h)$ for displacement.

### 6.1. Initial multiscale approximation vs. single-scale solution

Figs.6-7 compare the initial PLMM approximation $U_0$ to single-scale DNS for P2D-c$_3$ and S2D-c$_3$. Results for $N_0$ and $A_0$ are similar and thus omitted. Since crack pattern 3 is more challenging than patterns 1 and 2, the latter results are omitted too. Recall that $U_0$ implies no boundary iterations are performed ($n_b = 0$). Here, we set the number of source iterations, $n_s$, to be sufficiently large to ensure convergence ($tol_s = 10^{-12}$). Figs.6-7 compare the displacement magnitude $|\boldsymbol{u}|$, volumetric strain $|\nabla\cdot\boldsymbol{u}|$, and maximum shear stress $\boldsymbol{\sigma_t}$ from PLMM and DNS. In all cases, the agreement is very good. Fig.8 shows a similarly good agreement between $U_0$ and DNS for the S3D-c$_{20}$ domain.

### 6.2. Multiscale errors vs. boundary iterations

We next quantify the PLMM errors as a function of boundary iterations. Similar to Section 6.1, we set $n_s$ to be sufficiently large to ensure source iterations converge ($tol_s = 10^{-12}$). We continue to focus on Strategy U because the results for Strategies A and N are similar. Fig.9 depicts pointwise errors for $U_0$, $U_4$, and $U_8$ in P2D-c$_3$. The errors correspond to the displacement, $\boldsymbol{u}$, volumetric strain, $\nabla\cdot\boldsymbol{u}$, and maximum shear stress, $\boldsymbol{\sigma_t}$. Focusing on $U_0$, we see that $E_p^{\boldsymbol{u}} < 8\%$, $E_p^{\nabla\cdot\boldsymbol{u}} < 1\%$, and $E_p^{\boldsymbol{\sigma_t}} < 1\%$ in most parts of the domain except near cracks and contact interfaces. The errors near contacts are due to the localization assumption and are most obvious for $\nabla\cdot\boldsymbol{u}$ and $\boldsymbol{\sigma_t}$. We note that the $U_0$ errors correspond to the PLMM solution depicted in Fig.6. Focusing next on $U_4$ and $U_8$, we see that after 4 and 8 boundary iterations, respectively, errors decrease throughout the domain. The relative $L_2$-error for displacement, $E_2^{\boldsymbol{u}}$, is 3.28% for $U_0$, 2.20% for $U_4$, and 2.00% for $U_8$. The apparent stagnation in errors after $U_4$ is due to the fine-grid discretization error, $E_f$ in Eq.32, which is $O(h)$ and can be reduced only by refining the fine grid (see Appendix A).

Almost identical observations are made for S2D-c$_3$ and S3D-c$_{20}$ based on pointwise errors shown in Figs.10 and 11, respectively. Focusing on S2D-c$_3$, the $U_0$ errors are $E_p^{\boldsymbol{u}} < 10\%$, $E_p^{\nabla\cdot\boldsymbol{u}} < 0.8\%$, and $E_p^{\boldsymbol{\sigma_t}} < 10\%$ in most parts of the domain except near cracks and contact interfaces. However, after 4 and 8 boundary iterations, errors decrease throughout the domain and $E_2^{\boldsymbol{u}}$ drops from 8.08% for $U_0$, to 4.87% for $U_4$, to 3.60% for $U_8$. The same trend is seen for S3D-c$_{20}$, where $E_2^{\boldsymbol{u}}$ decreases from 2.80% for $U_0$, to 1.66% for $U_4$, to 1.57% for $U_8$.

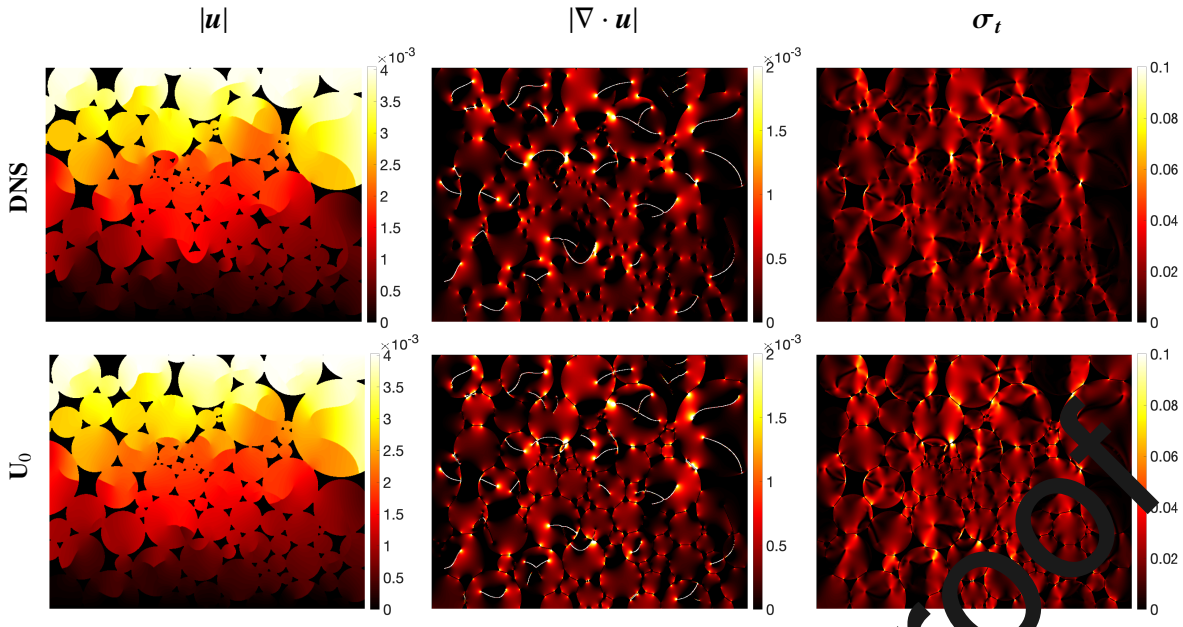



Figure 6: Comparison between the multiscale approximation $U_0$ and single-scale DNS for P2D-$c_3$. From left to right, the displacement magnitudes $|\boldsymbol{u}|$, volumetric strains $|\nabla \cdot \boldsymbol{u}|$, and maximum shear stresses $\boldsymbol{\sigma}_t$ are shown.

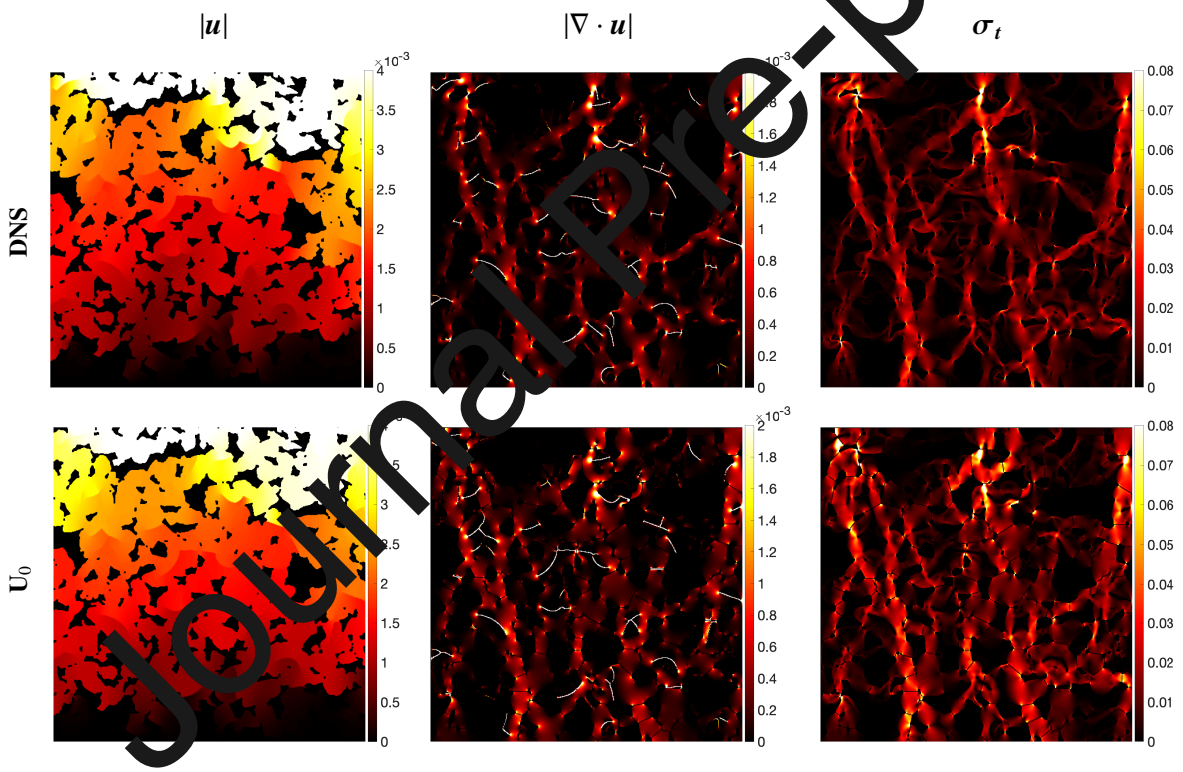



Figure 7: Comparison between the multiscale approximation $U_0$ and single-scale DNS for S2D-$c_3$. From left to right, the displacement magnitudes $|\boldsymbol{u}|$, volumetric strains $|\nabla \cdot \boldsymbol{u}|$, and maximum shear stresses $\boldsymbol{\sigma}_t$ are shown.

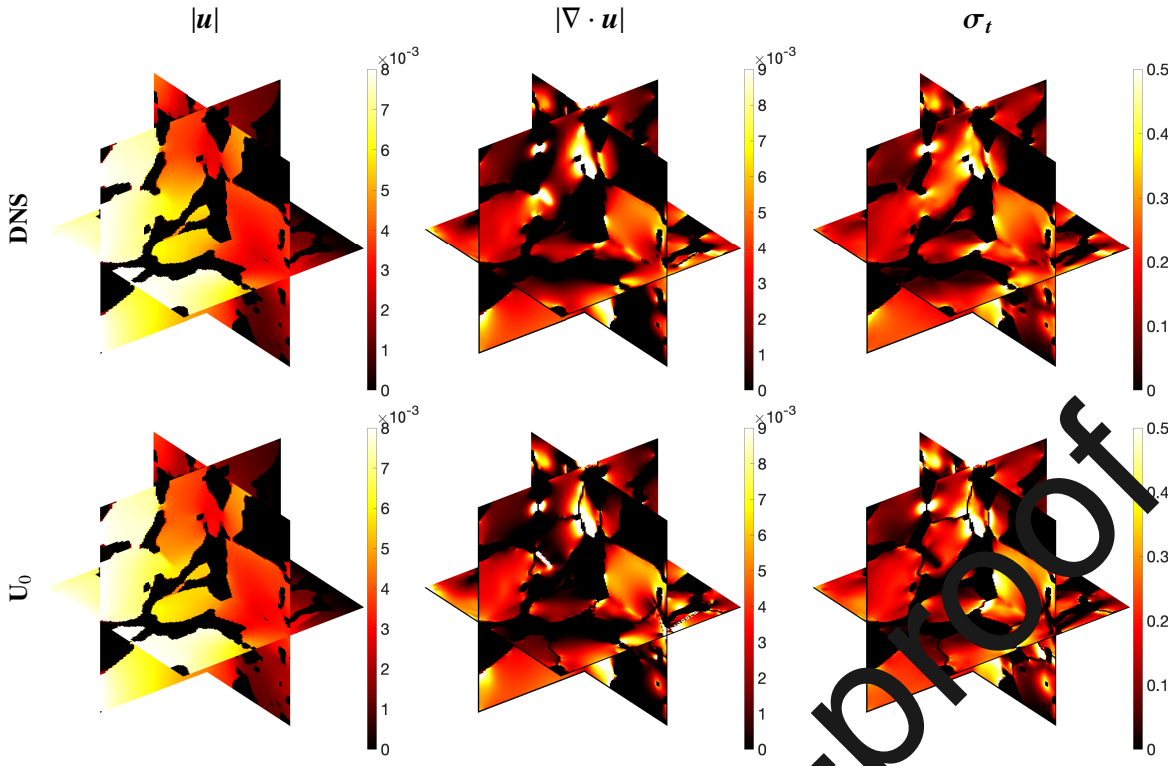



Figure 8: Comparison between the multiscale approximation $U_0$ and single-scale DNS for S3D-$c_{20}$. From left to right, the displacement magnitudes $|\boldsymbol{u}|$, volumetric strains $|\nabla \cdot \boldsymbol{u}|$, and maximum shear stresses $\boldsymbol{\sigma}_t$ are shown.

### *6.3. Multiscale errors vs. source iterations*

In Section 6.2, we discussed PLMM errors for when $n_s$ is sufficiently large for source iterations to converge at *each* boundary iteration. This is computationally wasteful, as smaller $n_s$ can still ensure convergence to the DNS solution. Here, we analyze Strategies A, N, and U for different combinations of $n_s$ and $n_b$ to understand their convergence rates. We focus only on S2D and P2D with crack patterns 1 and 2 due to the large number of simulations required.

Fig. 12 depicts contour plots of $E_2^{\boldsymbol{u}}$ versus $n_s$ and $n_b$ for Strategies A, N, and U applied to P2D-$c_1$ and P2D-$c_2$. The following key observations can be made: (1) Regardless of the crack pattern, Strategy A does not require any source iterations to converge, as expected from Section 4.3.1; (2) When all cracks are confined to the interior of grain grids (pattern 1), Strategy U reduces to Strategy N, consistent with Eqs. 19 and 15; (3) When all cracks intersect contact interfaces, Strategy U reduces to Strategy A, consistent with Eqs. 19 and 13; (4) In Strategies N and U, source iterations need not fully converge for PLMM to converge to the DNS solution; And (5) convergence to the DNS solution is slower in all strategies when cracks intersect contact interfaces (pattern 2), because the localization assumption (Eq. 9) becomes degraded. Similar observations can be made for S2D-$c_1$ and S2D-$c_2$ from the contour plots of $E_2^{\boldsymbol{u}}$ shown in Fig. 13. We note that Strategy A converges the fastest, in ~8 boundary iterations, regardless of the crack pattern, which is comparable to the results reported in [16] for intact domains.

In Figs. 12 and 13, to ensure that source iterations in Strategy N converge in crack pattern 2, a relaxation parameter of $\gamma = 0.3$ was used in Eq. 18b. In fact, larger values of $\gamma$ caused Strategy N to *diverge*, as shown by Fig. 14 for $\gamma = 0.6$ and 0.9. We note that relaxing source iterations was deemed necessary only in Strategy N and in the presence of cracks intersecting contact interfaces (patterns 2 and 3). In all other cases, $\gamma = 1$ was found sufficient and used. This renders Strategies A and U, which do not require relaxation, superior to Strategy N, because not only does $\gamma < 1$ slow down convergence, but the exact value of $\gamma$ cannot be determined *a priori* (at least based on this work).

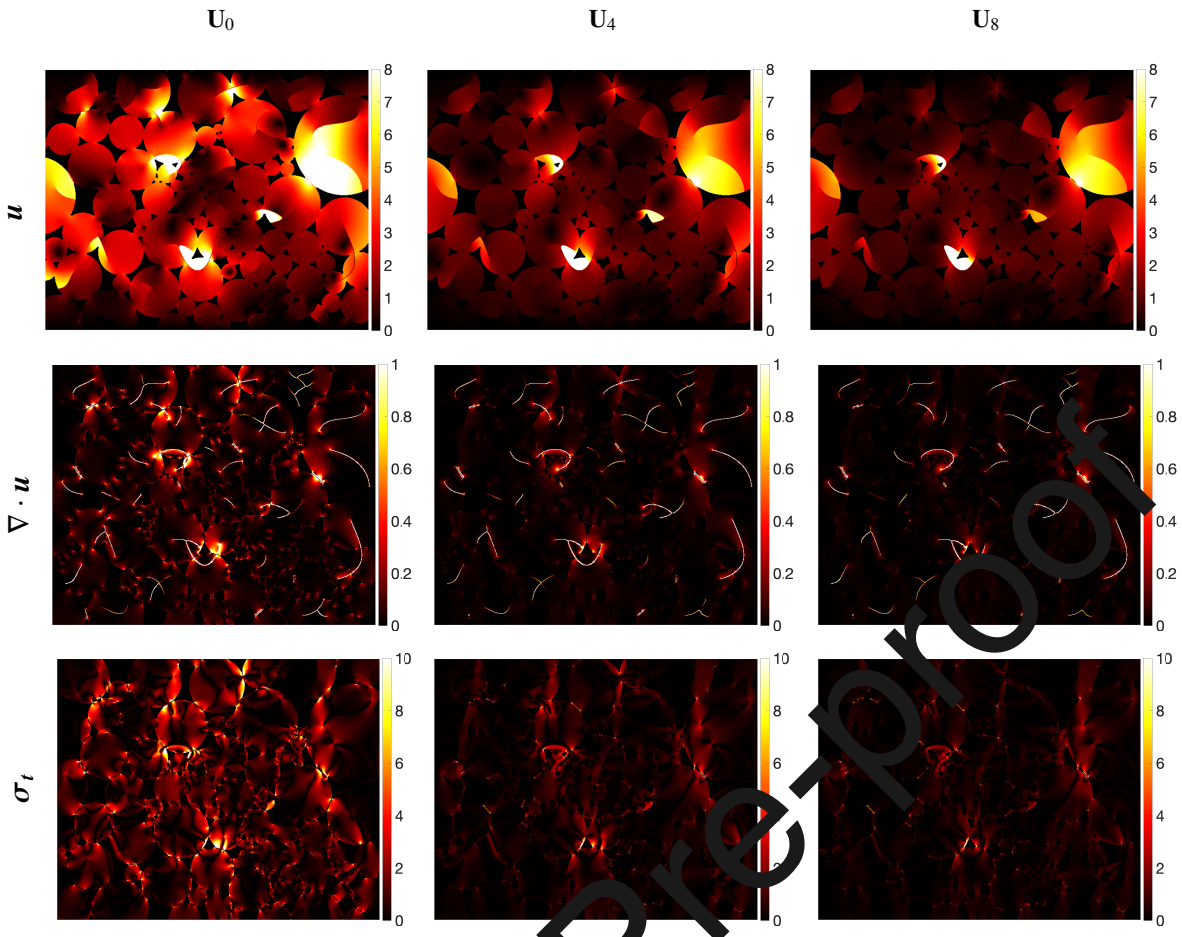



Figure 9: Pointwise errors (%) in displacement ($E_p^{\boldsymbol{u}}$), volumetric strain ($E_p^{\nabla\cdot\boldsymbol{u}}$), and maximum shear stress ($E_p^{\sigma_t}$) for the PLMM solutions $U_0$, $U_4$, and $U_8$ in the domain P2D-$c_3$. The $U_0$ plots correspond to Fig.6.

## 7. Discussion

### *7.1. Convergence rate for different strategies*

Figs.12-14 show that Strategy A converges the fastest, followed closely by Strategy U. Strategy N is the slowest to converge, and may even diverge if not relaxed in crack patterns 2 and 3. The reason for divergence is that when cracks intersect contact interfaces, the localization assumption (Eq.9) becomes degraded and boundary iterations are required to correct it. In addition, since cracks are excluded from the basis functions of Strategy N, source iterations are needed to capture displacement discontinuities within grain grids. Hence, the correction problem (Eq.17) attempts to correct too much, leading to an overshoot in the correction increment $\delta\widetilde{\boldsymbol{\varphi}}_{g_i}^{f,\eta}$. Relaxation tempers this increment, thereby encouraging convergence. Strategies A and U, by contrast, include contact-intersecting cracks in their basis functions, requiring only boundary iterations to correct for inaccuracies in the localization assumption. Therefore, the correction problem has less to correct, obviating relaxation. If cracks are fully confined to the interior of grain grids (pattern 1), the localization assumption is not degraded, and Strategies N and U require only source iterations to yield an accurate approximate solution; because neither strategy includes interior-cracks in their basis problems. Strategy A also includes interior cracks in its bases, and thus, does not even require source iterations.

### *7.2. Fixed-crack vs. evolving-crack problems*

Given Strategy A converges the fastest, then why bother with Strategies N and U? If cracks are fixed, the answer is: we should not. Strategy A is the most efficient computationally. But in problems where cracks evolve over many

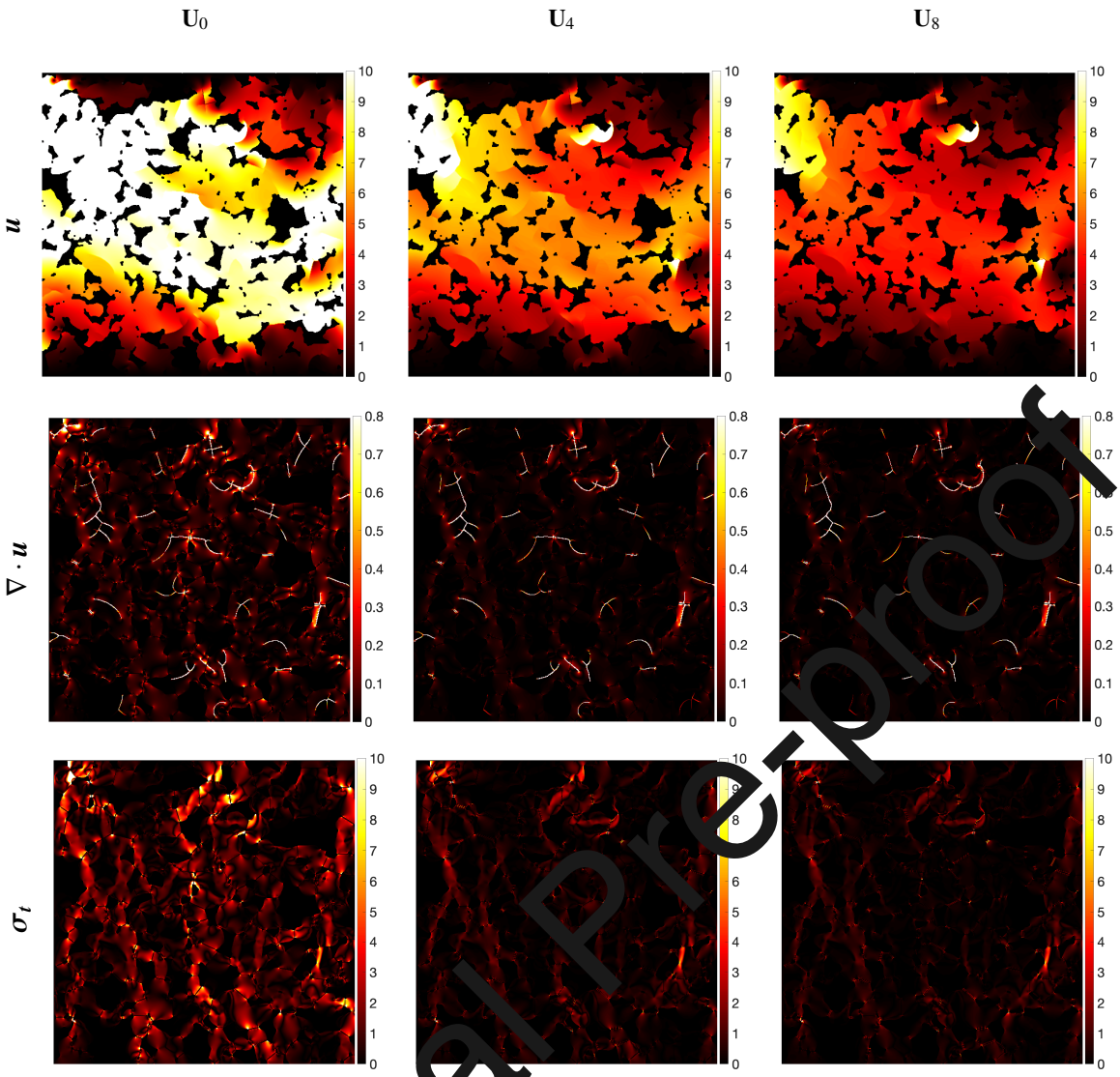



Figure 10: Pointwise errors (%) in displacement ($E_p^{\boldsymbol{u}}$), volumetric strain ($E_p^{\nabla\cdot\boldsymbol{u}}$), and maximum shear stress ($E_p^{\boldsymbol{\sigma}_t}$) for the PLMM solutions $U_0$, $U_4$, and $U_8$ in the domain S2D-c$_3$. The $U_0$ plots correspond to Fig. 7.

loading steps (or staggered iterations in phase-field models), Strategy A becomes prohibitively slow. This is because as cracks nucleate and grow within grain grids, basis functions must be recomputed, which in turn entails solving multiple local systems on the grain grids. Notice the Kronecker delta in the BCs of Eq. 13. Put differently, *frequent recomputing of basis functions in an evolving-crack problem is a bad idea*. Strategy N would have been the ideal solution, because no basis updates are necessary. However, as we observed in Section 6.3, it performed poorly. This leaves Strategy U as the preferred method for solving evolving-crack problems, because basis functions are updated only when cracks cross contact interfaces, which is a far less frequent occurance.

### *7.3. Computational cost*

Here, we analyze the algorithmic complexity of PLMM compared to DNS, deferring a complete numerical demonstration to future work. Assume $\Omega_s$ consists of $N_f$ fine grids (FEM nodes here), partitioned into $N^g$ grain grids, $N^c$ contact interfaces, and $N^\zeta$ contact grids via the decomposition algorithm in Section 4.2. For simplicity, assume also that the decomposition is *balanced*, meaning all grain grids, and separately all contact grids, consist of roughly the

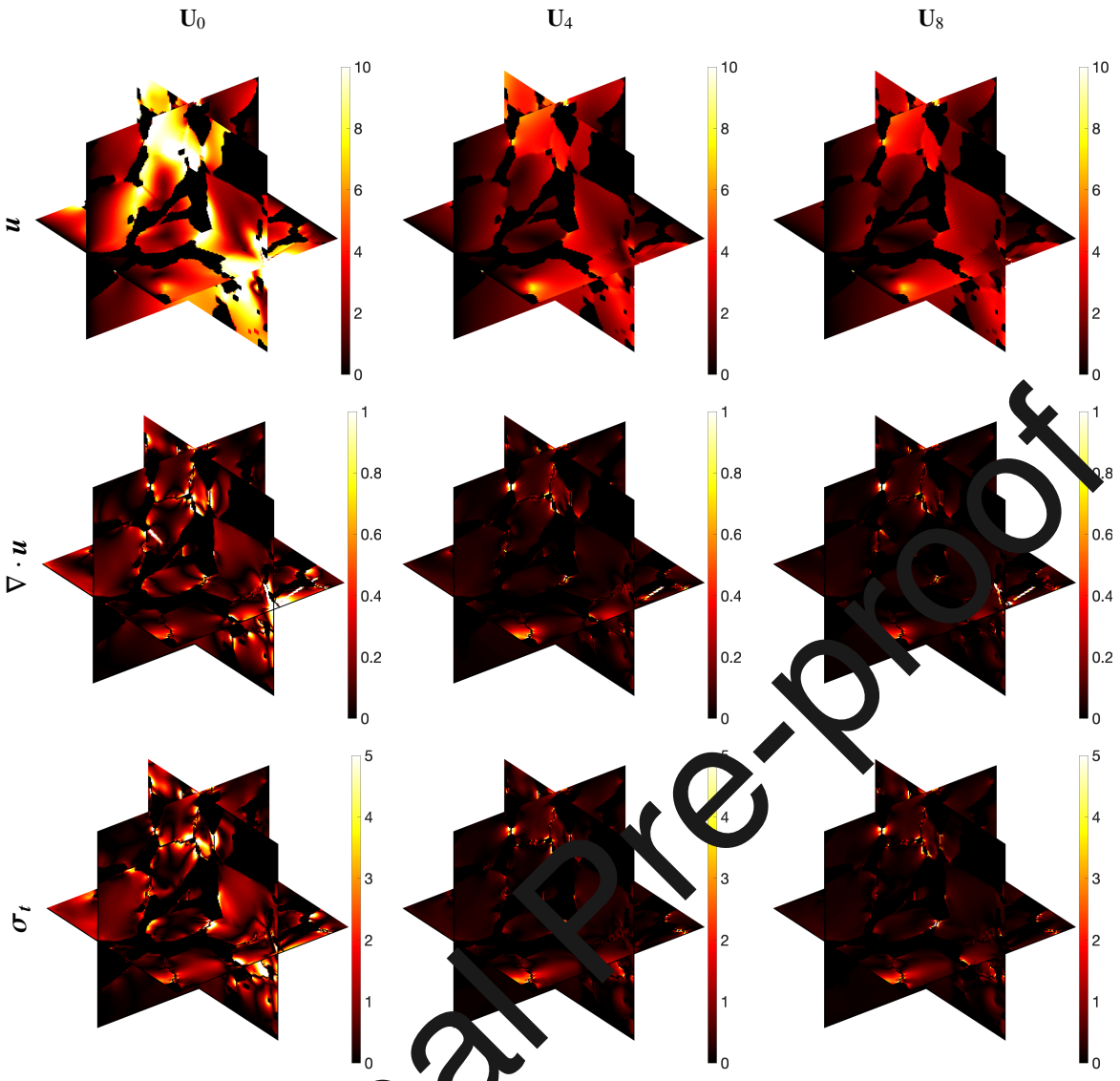



Figure 11: Pointwise errors (%) in displacement ($E_p^{\boldsymbol{u}}$), volumetric strain ($E_p^{\nabla\cdot\boldsymbol{u}}$), and maximum shear stress ($E_p^{\sigma_t}$) for the PLMM solutions $U_0$, $U_4$, and $U_8$ in the domain S3D-c$_{20}$. The $U_0$ plots correspond to Fig. 8.

same number of fine grids. Let $\overline{n_{g_i}}$ denote the average number of contacts per grain grid, and $f^{\zeta}$ be the fraction of $\Omega_s$ covered by contact grids. As seen from Fig. 3, $f^{\zeta}$ is generally small (= 0.28 for P2D and S2D). Assume we solve all linear systems with the same solver, whose wall-clock time $\mathscr{T}$ scales as $O(n^{\beta})$, where $n$ is the number of degrees of freedom in the system and $\beta \in (1, 3)$ is the computational complexity of the solver [71]. The wall-clock times of DNS ($\mathscr{T}_S$) and PLMM ($\mathscr{T}_{M_{n_b}}$), where $n_b$ is the number of boundary iterations, can be estimated as:

$$\mathscr{T}_S = O(N^f D)^{\beta} \tag{33a}$$

$$\mathscr{T}_{M_0} = \mathscr{T}_{basis} + \mathscr{T}_{corr} + \mathscr{T}_{glob} + n_s\,(\mathscr{T}_{corr} f^F + \mathscr{T}_{glob}) \tag{33b}$$

$$\mathscr{T}_{M_{n_b}} = \mathscr{T}_{M_0} + n_b\,(\mathscr{T}_{cont} + \mathscr{T}_{corr} + \mathscr{T}_{glob} + n_s\,(\mathscr{T}_{corr} f^F + \mathscr{T}_{glob})) \tag{33c}$$

where

$$\mathscr{T}_{basis} = O(N^f D/N^g)^\beta \times (N^g \overline{\#C^{g_i}} D/P_{basis}) \tag{33d}$$

$$\mathscr{T}_{corr} = O(N^f D/N^g)^\beta \times (N^g/P_{corr}) \tag{33e}$$

$$\mathscr{T}_{cont} = O(N^f D f^\zeta/N^\zeta)^\beta \times (N^\zeta/P_{cont}) \tag{33f}$$

$$\mathscr{T}_{glob} = O(N^c D)^\beta \tag{33g}$$

The wall-clock times $\mathscr{T}_{basis}$, $\mathscr{T}_{corr}$, $\mathscr{T}_{cont}$ and $\mathscr{T}_{glob}$ correspond to solving the basis problem (Eq.13, 15, or 19), correction problem (Eq.14, 17, or 29), contact problem (Eq.28), and global problem (Eq.26 or 30d) in PLMM, respectively. $P_{basis}$, $P_{corr}$ and $P_{cont}$ denote the number of processors used to solve the basis, correction, and contact problems (= 1 if serial computing). Because these local problems are decoupled, the maximum number of processors that can be used in each case is $P_{basis} = N^g \overline{\#C^{g_i}} D$, $P_{corr} = N^g$, and $P_{cont} = N^\zeta$. In Eq.33b-c, $f^F$ is the fraction of grain grids that contain cracks *not* captured by the basis functions; thus requiring source corrections. In DNS, the total number of unknonws is $N^f D$, whereas in PLMM, it is $(N^f + N^c)D$. The extra $N^c D$ unknowns are due to coarse-scale displacements defined at the contact interfaces, which are continuous due to Eq.21 (i.e., factor of 2 not required). The increase in unknowns is negligible because $N^c \ll N^f$ (e.g., $N^c/N^f = 10^{-3}$ in S2D).

If Strategy A is used to compute basis/correction functions, $n_s = 0$ because no source iterations are needed, and $f^F = 0$ because cracks are included in all basis functions. Since $\mathscr{T}_{glob}$ and $\mathscr{T}_{cont}$ are negligible compared to $\mathscr{T}_{basis}$ and $\mathscr{T}_{corr}$, we can approximate $\mathscr{T}_{M_0} \approx \mathscr{T}_{basis} + \mathscr{T}_{corr}$ and $\mathscr{T}_{M_{n_b}} \approx \mathscr{T}_{basis} + (n_b+1)\mathscr{T}_{corr}$. The blue contour lines in Fig.13 of

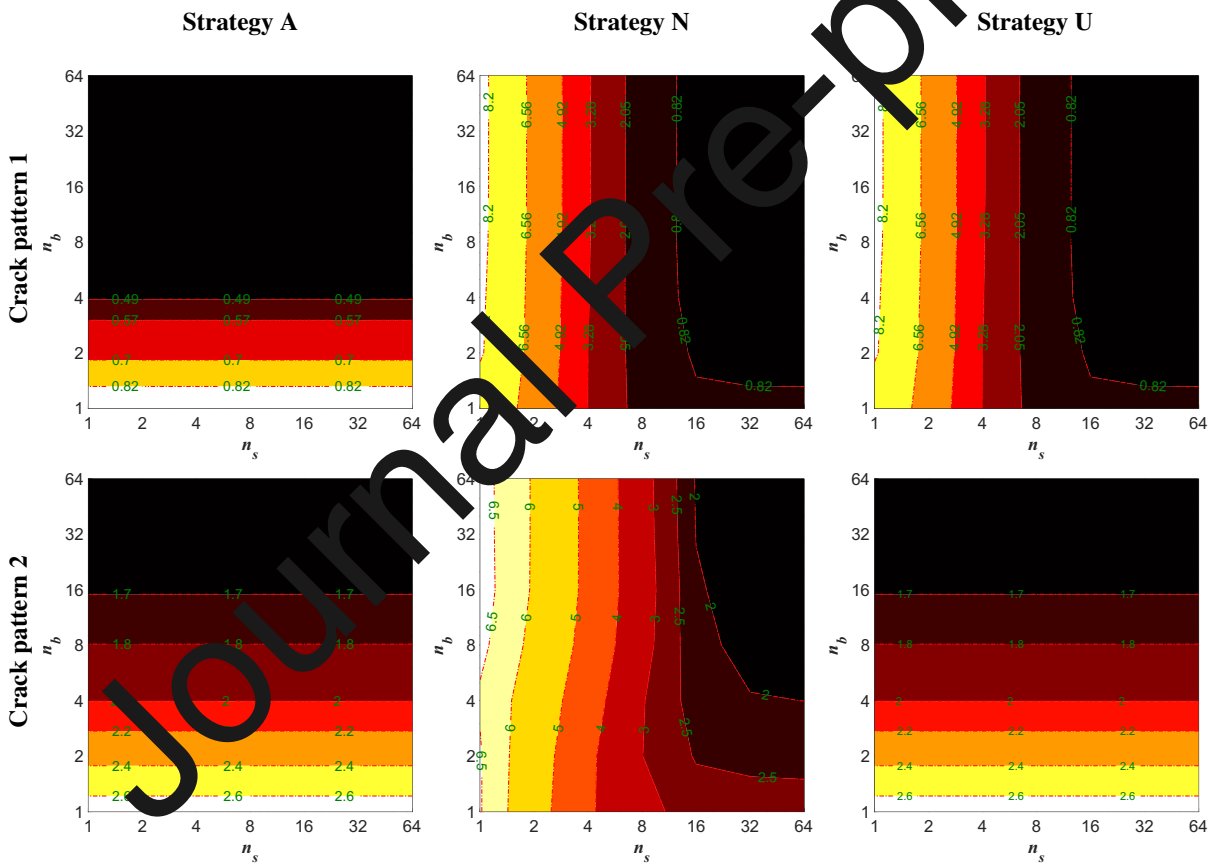



Figure 12: Relative $L_2$-errors (%) in displacement ($E_2^{\boldsymbol{u}}$) plotted for different combinations of source ($n_s$) and boundary ($n_b$) iterations in the P2D-c$_1$ and P2D-c$_2$ domains using Strategies A, N, and U.

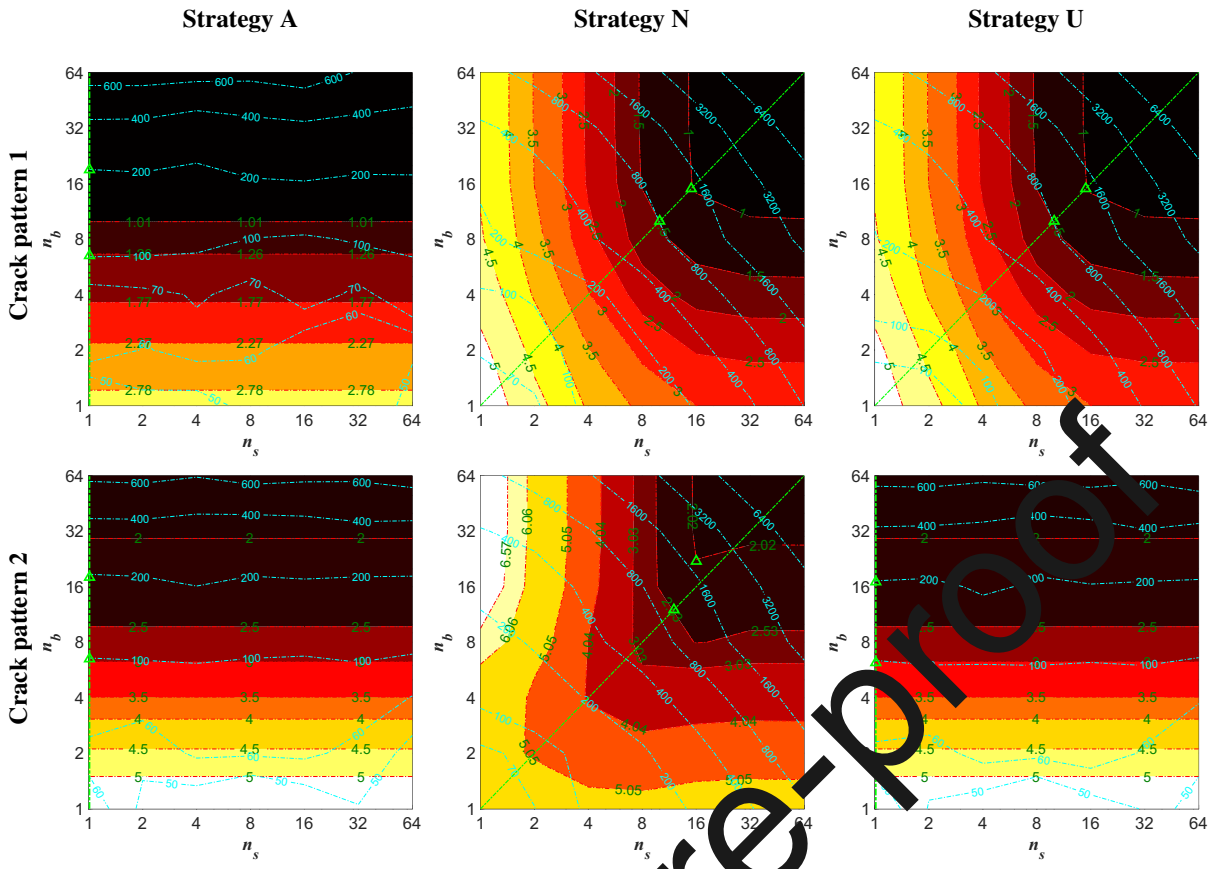



Figure 13: Relative $L_2$-errors (%; red contours) in displacement ($L^2$) plotted for different combinations of source ($n_s$) and boundary ($n_b$) iterations in the S2D-c$_1$ and S2D-c$_2$ domains using Strategies A, N, and U. Blue contours correspond to measured wall-clock times in seconds. Optimal $(n_s, n_b)$-pairings are marked by green triangles. Diagonal and vertical green lines correspond to $n_s = n_b$ and $n_s = 0$, respectively; the approximate loci of optimal pairings.

measured wall-clock times for S2D-c$_1$ and S2D-c$_2$ confirm these estimates. If Strategy N or U is used, $n_s \neq 0$ and $f^F \neq 0$ and we can approximate $\mathcal{T}_{M_0} \approx (1 + n_b f^F)\mathcal{T}_{corr}$ and $\mathcal{T}_{M_{n_b}} \approx (1 + n_b)(1 + n_s f^F)\mathcal{T}_{corr}$, where we omitted $\mathcal{T}_{basis}$ because it is a one-time cost that becomes negligible as $n_s$ and $n_b$ grow. The difference between Strategies N and U is that both $n_s$ and $f^F$ are larger for Strategy N with the possibility $n_s = \infty$ if source iterations diverge. The blue contours in Fig. 13 of measured wall-clock times once again confirm these estimates.

Can $n_s$ and $n_b$ be chosen optimally in PLMM to satisfy a desired error tolerance at minimal computational cost? The answer is *yes*, and Fig. 13 shows how. Focusing on Strategy N and crack pattern 1, as an example, assume our tolerance for error is $E_2^u = 1.5\%$. The corresponding contour line in Fig. 13, colored red and annotated green, is the locus of all $(n_s, n_b)$ pairings that satisfy this tolerance. If we now consider the superposed contour lines of wall-clock time, colored and annotated blue, we can determine the optimal pairing: *the point at which the blue and red contours become tangent*, marked by a green triangle in Fig. 13. Similar optimal pairings can be found for different tolerances (e.g., second triangle in Fig. 13 for $E_2^u = 1\%$). By connecting these optimal pairings we find that they lie roughly on the line $n_s = n_b$, which is a key takeaway. In Strategy A, optimal pairings lie on the line $n_s = 0$, shown by the vertical green lines in Fig. 13. In Strategy U, optimal pairings lie on $n_s = n_b$ for crack pattern 1 (similar to Strategy N), but on $n_s = 0$ for pattern 2 (similar to Strategy A), for reasons discussed in Section 6.3.

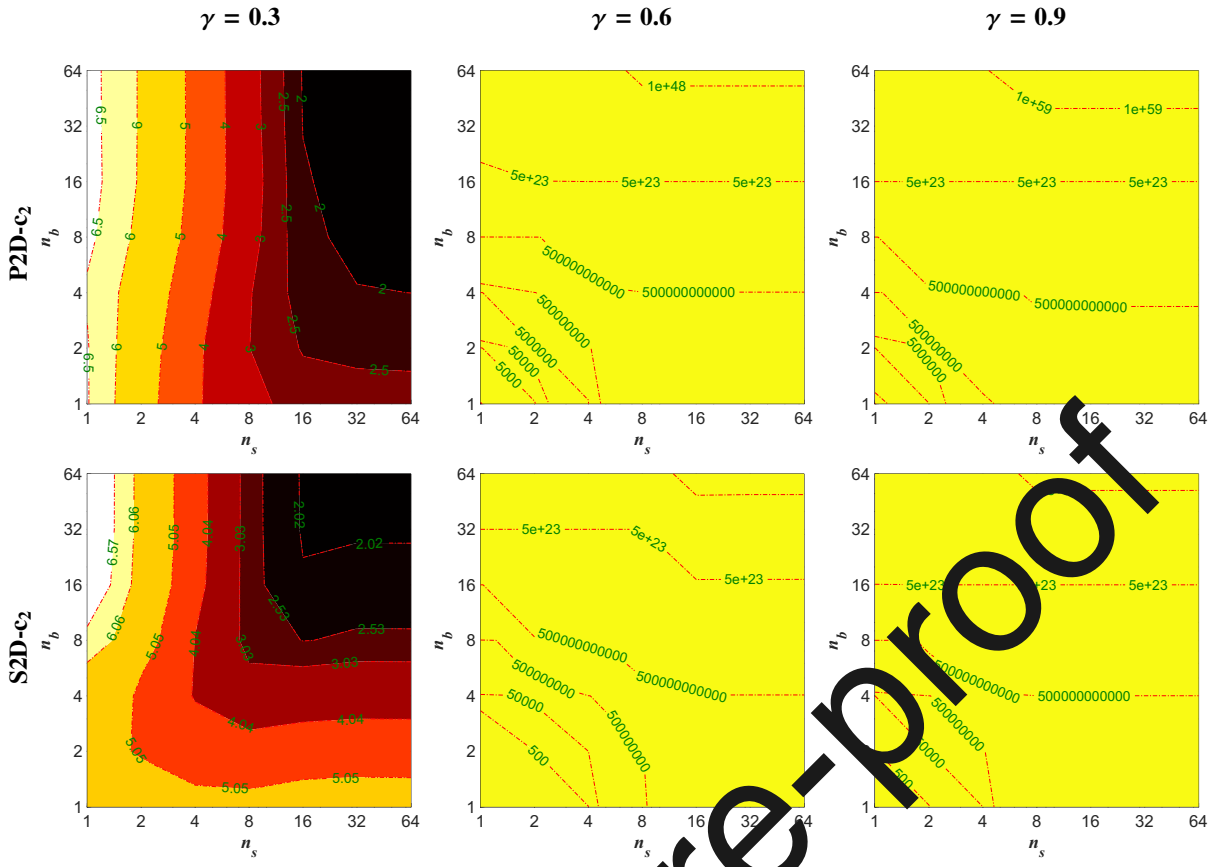



Figure 14: Relative $L_2$-errors (%) in displacement ($E_2^{\boldsymbol{u}}$) plotted for different combinations of source ($n_s$) and boundary ($n_b$) iterations in the P2D-c$_2$ and S2D-c$_2$ domains using Strategy N with relaxation parameters $\gamma = 0.3$, 0.6 and 0.9.

## 8. Conclusion

We presented a pore-level multiscale method (PLMM) to model elastic deformation of fractured porous materials with arbitrary microstructure and crack patterns. The key feature of the proposed PLMM is that the domain decomposition need not conform to crack positions, unlike a previous formulation that had been proposed by the authors [16]. The added flexbility leads to more accurate solutions, better error control, and the absence of a need to dynamically update the decomposition when cracks evolve. We validated PLMM against single-scale direct numerical simulations (DNS) and showed it produces an accurate initial approximation (e.g., $E_s^{\boldsymbol{u}} < 10\%$) on complex 2D and 3D geometries. We also proposed an iterative strategy to reduce the errors of this approximate solution to any desired level.

Anticipating application to evolving-crack problems in the future, three strategies for computing basis functions within PLMM were explored: (1) Strategy A, where *all* cracks are included in the basis functions; (2) Strategy N, where *no* cracks are included in the basis functions; and (3) Strategy U, where only those cracks are included that intersect at least one contact interface. If cracks are fixed, Strategy A is preferred, as it converges the fastest. If cracks evolve, Strategy U is preferred because it requires infrequent updates of the basis functions (unlike Strategy A), which is computationally more efficient. Strategy N either diverges, if not relaxed, or converges slowly. It is therefore not recommended. Future work will focus on applying PLMM to crack nucleation and growth problems. This work also opens the possiblity to adapt PLMM as the linear solver for nonlinear problems involving finite strains or plasticity.

**Acknowledgments**

This material is based upon work supported by the National Science Foundation under Grant No. CMMI-2145222. The authors also thank the Institute for Computational and Data Sciences (ICDS) at Penn State University for providing access to computational resources.

## Appendix A. Multiscale error versus fine-grid refinement

Consider a rectangular domain $\Omega = [0, 0.96] \times [0, 0.98]$ with different crack patterns, as shown in Fig.A.1. The domain is fixed at the bottom boundary and a vertical-upward displacement of $u_D = 0.001$ mm is applied to the top boundary. All material properties are the same as those in Section 6. For each domain, Eqs.1 and 7 are solved using Strategy U of PLMM and the displacement error, $E_2^{\boldsymbol{u}}$, is calculated against the single-scale DNS solution. We set the number of boundary, $n_b$, and source, $n_s$, iterations to be sufficiently large so that the PLMM solution no longer improves in accuracy. This ensures $E_b = E_s = 0$ in Eq.32. We repeat the above process for different fine-grids, refined according to $h = h_0/N_{ref}$, where $h_0$ is the initial fine-grid size and $N_{ref}$ is the refinement factor. We set the phase-field length scale parameter to $l = 2h$. Fig.A.2 shows that the plot of $E_2^{\boldsymbol{u}}$ versus $N_{ref}$ follows a first-order convergence behavior, regardless of the crack pattern. Together with Eq.32, this implies $E_f \sim O(h)$.

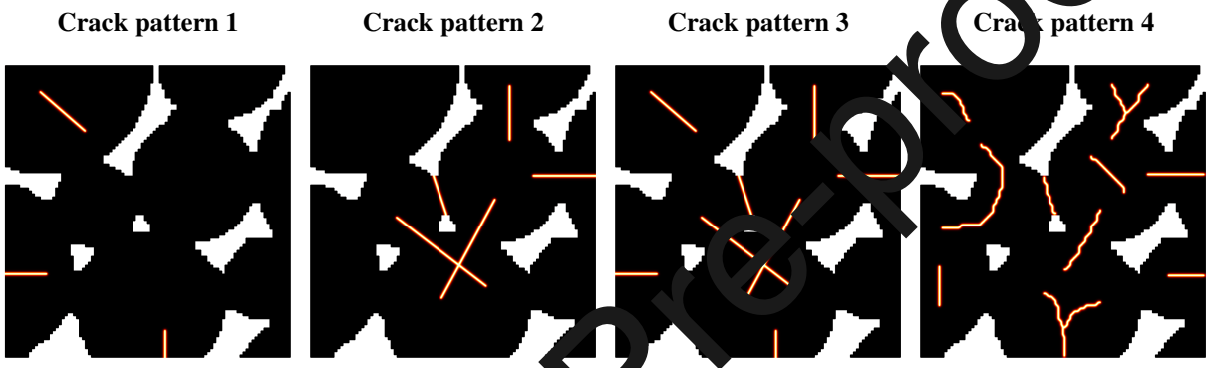



Figure A.1: Crack patterns considerd in the fine-grid convergence analysis. Pattern 1: all cracks are confined to the interior of grain grids. Pattern 2: all cracks intersect at least one contact interface. Pattern 3: a combination of crack patterns 1 and 2. Pattern 4: a complex crack pattern with curved and branching cracks.

## References

[1] Jacob C Marx, Samuel J Robbins, Lane A Grady, Frank L Palmieri, Christopher J Wohl, and Afsaneh Rabiei. Polymer infused composite metal foam as a potential aircraft leading edge material. Applied Surface Science, 505:144114, 2020.

[2] Jacob Marx, Marc Portanova, and Afsaneh Rabiei. Performance of composite metal foam armors against various threat sizes. Journal of Composites Science, 4(4):176, 2020.

[3] Hyeong-Ki Kim, JH Jeon, and Hae-Ki Lee. Workability, and mechanical, acoustic and thermal properties of lightweight aggregate concrete with a high volume of entrained air. Construction and Building Materials, 29:193–200, 2012.

[4] Salvatore Torquato and IV Haslach Jr. Random heterogeneous materials: microstructure and macroscopic properties. Appl. Mech. Rev., 55 (4):B62–B63, 2002.

[5] Dorthe Wildenschild and Adrian P Sheppard. X-ray imaging and analysis techniques for quantifying pore-scale structure and processes in subsurface porous medium systems. Advances in Water Resources, 51:217–246, 2013.

[6] Veerle Cnudde and Matthieu Nicolaas Boone. High-resolution x-ray computed tomography in geosciences: A review of the current technology and applications. Earth-Science Reviews, 123:1–17, 2013.

[7] Kenneth Pye. Forensic examination of rocks, sediments, soils and dusts using scanning electron microscopy and x-ray chemical microanalysis. Geological Society, London, Special Publications, 232(1):103–121, 2004.

[8] Min Liu, Mehdi Shabaninejad, and Peyman Mostaghimi. Predictions of permeability, surface area and average dissolution rate during reactive transport in multi-mineral rocks. Journal of Petroleum Science and Engineering, 170:130–138, 2018.

[9] Simona Hapca, Philippe C Baveye, Clare Wilson, Richard Murray Lark, and Wilfred Otten. Three-dimensional mapping of soil chemical characteristics at micrometric scale by combining 2d sem-edx data and 3d x-ray ct images. PloS one, 10(9):e0137205, 2015.

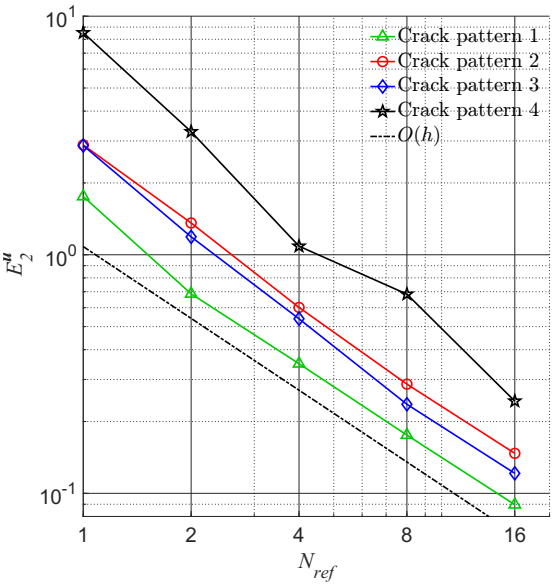



Figure A.2: Relative $L_2$-errors in displacement ($E_2^{\boldsymbol{u}}$) versus fine-grid refinement factors, $N_{ref}$, for different crack patterns shown in Fig A.1 solved with Strategy U of PLMM. At each $N_{ref}$, boundary, $n_b$, and source, $n_s$, iterations are sufficiently large so that $E_2^{\boldsymbol{u}}$ no longer improves. Plotted errors represent $E_f$ in Eq. 32 and are $O(h)$.


[10] Heiko Andrä, Nicolas Combaret, Jack Dvorkin, Erik Glatt, Junehee Han, Matthias Kabel, Youngseuk Keehm, Fabian Krzikalla, Minhui Lee, Claudio Madonna, et al. Digital rock physics benchmarks part ii: Computing effective properties. Computers & Geosciences, 50:33–43, 2013.

[11] Nicolas Moës, John Dolbow, and Ted Belytschko. A finite element method for crack growth without remeshing. International journal for numerical methods in engineering, 46(1):131–150, 1999.

[12] Fushen Liu and Ronaldo I Borja. A contact algorithm for frictional crack propagation with the extended finite element method. International Journal for Numerical methods in engineering, 76(10):1489–1512, 2008.

[13] Kangan Li, Nabil M Atallah, Antonio Rodríguez-Ferran, Daniel M Valiveti, and Guglielmo Scovazzi. The shifted fracture method. International Journal for Numerical Methods in Engineering, 122(23):6641–6679, 2021.

[14] Kangan Li, Antonio Rodríguez-Ferran, and Guglielmo Scovazzi. An extended shifted-fracture/phase-field framework for sharp/diffuse crack modeling. International Journal for Numerical Methods in Engineering, 2022.

[15] Yashar Mehmani, Nicola Castelletto, and Hamdi A Tchelepi. Nonlinear convergence in contact mechanics: Immersed boundary finite volume. Computer Methods in Applied Mechanics and Engineering, 383:113929, 2021.

[16] Yashar Mehmani, Nicola Castelletto, and Hamdi A Tchelepi. Multiscale formulation of frictional contact mechanics at the pore scale. Journal of Computational Physics, 430:110092, 2021.

[17] Mary F Wheeler, Thomas Wick, and W Wollner. An augmented-lagrangian method for the phase-field approach for pressurized fractures. Computer Methods in Applied Mechanics and Engineering, 271:69–85, 2014.

[18] Igor Shovkun and D Nicolas Espinoza. Fracture propagation in heterogeneous porous media: pore-scale implications of mineral dissolution. Rock Mechanics and Rock Engineering, 52(8):3197–3211, 2019.

[19] Thanh Tung Nguyen, Julien Yvonnet, Q-Z Zhu, Michel Bornert, and Camille Chateau. A phase field method to simulate crack nucleation and propagation in strongly heterogeneous materials from direct imaging of their microstructure. Engineering Fracture Mechanics, 139:18–39, 2015.

[20] Ziguang Chen, Si[illegible] and Florin Bobaru. A peridynamic model for brittle damage and fracture in porous materials. International Journal of Rock Mechanics and Mining Sciences, 122:104059, 2019.

[21] Hisanao Ouchi, Abhijit Katiyar, Jason York, John T Foster, and Mukul M Sharma. A fully coupled porous flow and geomechanics model for fluid driven cracks: a peridynamics approach. Computational Mechanics, 55(3):561–576, 2015.

[22] Peter A Cundall and Otto DL Strack. A discrete numerical model for granular assemblies. geotechnique, 29(1):47–65, 1979.

[23] David O Potyondy and PA Cundall. A bonded-particle model for rock. International journal of rock mechanics and mining sciences, 41(8): 1329–1364, 2004.

[24] Catherine O'Sullivan. Particulate discrete element modelling: a geomechanics perspective. CRC Press, 2011.

[25] Erfan G Nezami, Youssef MA Hashash, Dawei Zhao, and Jamshid Ghaboussi. Simulation of front end loader bucket–soil interaction using discrete element method. International journal for numerical and analytical methods in geomechanics, 31(9):1147–1162, 2007.

[26] Nivedita Das, Patrick Giordano, Daniel Barrot, Shreekanth Mandayam, Beena Sukumaran, Alaa K Ashmawy, et al. Discrete element modeling and shape characterization of realistic granular shapes. In The Eighteenth International Offshore and Polar Engineering Conference. International Society of Offshore and Polar Engineers, 2008.

[27] Kimiro Meguro, Kazuyoshi Iwashita, and Motohiko Hakuno. Fracture analysis of media composed of irregularly shaped regions by the extended distinct element method. Structural Eng./Earthquake Eng, 8(3):131–142, 1991.

[28] A Munjiza, DRJ Owen, and N Bicanic. A combined finite-discrete element method in transient dynamics of fracturing solids. Engineering computations, 12(2):145–174, 1995.

[29] Antonio A Munjiza. The combined finite-discrete element method. John Wiley & Sons, 2004.

[30] Chuanqi Liu and WaiChing Sun. Ils-mpm: An implicit level-set-based material point method for frictional particulate contact mechanics of deformable particles. Computer Methods in Applied Mechanics and Engineering, 369:113168, 2020.

[31] Mengsu Hu and Jonny Rutqvist. Microscale mechanical modeling of deformable geomaterials with dynamic contacts based on the numerical manifold method. Computational Geosciences, 24(5):1783–1797, 2020.

[32] Marreddy Ambati, Tymofiy Gerasimov, and Laura De Lorenzis. A review on phase-field models of brittle fracture and a new fast hybrid formulation. Computational Mechanics, 55(2):383–405, 2015.

[33] Michael J Borden, Clemens V Verhoosel, Michael A Scott, Thomas JR Hughes, and Chad M Landis. A phase-field description of dynamic brittle fracture. Computer Methods in Applied Mechanics and Engineering, 217:77–95, 2012.

[34] Ivo Babuška and John E Osborn. Generalized finite element methods: their performance and their relation to mixed methods. SIAM Journal on Numerical Analysis, 20(3):510–536, 1983.

[35] Thomas Y Hou and Xiao-Hui Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. Journal of computational physics, 134(1):169–189, 1997.

[36] Zhiming Chen and Thomas Hou. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. Mathematics of Computation, 72(242):541–576, 2003.

[37] Jorg E Aarnes. On the use of a mixed multiscale finite element method for greaterflexibility and increased speed or improved accuracy in reservoir simulation. Multiscale Modeling & Simulation, 2(3):421–439, 2004.

[38] Yalchin Efendiev and Thomas Y Hou. Multiscale finite element methods: theory and applications, volume 4. Springer Science & Business Media, 2009.

[39] Nicola Castelletto, Hadi Hajibeygi, and Hamdi A Tchelepi. Multiscale finite-element method for linear elastic geomechanics. Journal of Computational Physics, 331:337–356, 2017.

[40] Eric T Chung and Chak Shing Lee. A mixed generalized multiscale finite element method for planar linear elasticity. Journal of Computational and Applied Mathematics, 348:298–313, 2019.

[41] Fanxiang Xu, Hadi Hajibeygi, and Lambertus J Sluys. Multiscale extended finite element method for deformable fractured porous media. Journal of Computational Physics, 436:110287, 2021.

[42] Patrick Jenny, SH Lee, and Hamdi A Tchelepi. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. Journal of Computational Physics, 187(1):47–67, 2003.

[43] Ivan Lunati and Patrick Jenny. The multiscale finite volume method: A flexible tool to model physically complex flow in porous media. In Proceedings of European Conference of Mathematics of Oil Recovery X, Amsterdam, The Netherlands, 2006.

[44] Hui Zhou, Hamdi A Tchelepi, et al. Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models. SPE Journal, 17 (02):523–539, 2012.

[45] Hadi Hajibeygi, Giuseppe Bonfigli, Marc Andre Hesse, and Patrick Jenny. Iterative multiscale finite-volume method. Journal of Computational Physics, 227(19):8604–8621, 2008.

[46] Irina Sokolova, Muhammad Gusti Bastisya, and Hadi Hajibeygi. Multiscale finite volume method for finite-volume-based simulation of poroelasticity. Journal of Computational Physics, 379:309–324, 2019.

[47] Matei Ţene, Mohammed Saad Al Kobaisi, and Hadi Hajibeygi. Algebraic multiscale method for flow in heterogeneous porous media with embedded discrete fractures (f-ams). Journal of Computational Physics, 321:819–845, 2016.

[48] Christine Bernardi, Y Maday, and A T Patera. A new nonconforming approach to domain decomposition: the mortar element method. Nonlinear partial equations and their applications, 1994.

[49] Todd Arbogast, Lawrence C Cowsar, Mary F Wheeler, and Ivan Yotov. Mixed finite element methods on nonmatching multiblock grids. SIAM Journal on Numerical Analysis, 37(4):1295–1315, 2000.

[50] Todd Arbogast, Gergina Pencheva, Mary F Wheeler, and Ivan Yotov. A multiscale mortar mixed finite element method. Multiscale Modeling & Simulation, 6(1):319–346, 2007.

[51] Benjamin Ganis and Ivan Yotov. Implementation of a mortar mixed finite element method using a multiscale flux basis. Computer Methods in Applied Mechanics and Engineering, 198(49-52):3989–3998, 2009.

[52] Matthew T Balhoff, Sunil G Thomas, and Mary F Wheeler. Mortar coupling and upscaling of pore-scale models. Computational Geosciences, 12(1):15–27, 2008.

[53] Yashar Mehmani, Tie Sun, MT Balhoff, P Eichhubl, and S Bryant. Multiblock pore-scale modeling and upscaling of reactive transport: application to carbon sequestration. Transport in porous media, 95(2):305–326, 2012.

[54] Yashar Mehmani and Matthew T Balhoff. Bridging from pore to continuum: A hybrid mortar domain decomposition framework for subsurface flow and transport. Multiscale Modeling & Simulation, 12(2):667–693, 2014.

[55] Pavel Tomin and Ivan Lunati. Hybrid multiscale finite volume method for two-phase flow in porous media. Journal of Computational Physics, 250:293–307, 2013.

[56] Pavel Tomin and Ivan Lunati. Local–global splitting for spatiotemporal-adaptive multiscale methods. Journal of Computational Physics, 280: 214–231, 2015.

[57] Karim Khayrat and Patrick Jenny. A multi-scale network method for two-phase flow in porous media. Journal of Computational Physics, 342:194–210, 2017.

[58] Karim Khayrat, Robert Epp, and Patrick Jenny. Approximate multiscale flow solver for unstructured pore networks. Journal of Computational Physics, 372:62–79, 2018.

[59] Yashar Mehmani and Hamdi A Tchelepi. Multiscale computation of pore-scale fluid dynamics: Single-phase flow. Journal of Computational Physics, 375:1469–1487, 2018.

[60] Yashar Mehmani and Hamdi A Tchelepi. Multiscale formulation of two-phase flow at the pore scale. Journal of Computational Physics, 389: 164–188, 2019.
[61] Bo Guo, Yashar Mehmani, and Hamdi A Tchelepi. Multiscale formulation of pore-scale compressible darcy-stokes flow. Journal of Computational Physics, 397:108849, 2019.
[62] Yashar Mehmani, Timothy Anderson, Yuhang Wang, Saman A Aryana, Ilenia Battiato, Hamdi A Tchelepi, and Anthony R Kovscek. Striving to translate shale physics across ten orders of magnitude: What have we learned? Earth-Science Reviews, 223:103848, 2021.
[63] Christian Miehe, Martina Hofacker, and Fabian Welschinger. A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits. Computer Methods in Applied Mechanics and Engineering, 199(45-48):2765–2778, 2010.
[64] Hanen Amor, Jean-Jacques Marigo, and Corrado Maurini. Regularized formulation of the variational brittle fracture with unilateral contact: Numerical experiments. Journal of the Mechanics and Physics of Solids, 57(8):1209–1229, 2009.
[65] Juan Michael Sargado, Eirik Keilegavlen, Inga Berre, and Jan Martin Nordbotten. High-accuracy phase-field models for brittle fracture based on a new family of degradation functions. Journal of the Mechanics and Physics of Solids, 111:458–489, 2018.
[66] Eirik Keilegavlen and Jan Martin Nordbotten. Finite volume methods for elasticity with weak symmetry. International Journal for Numerical Methods in Engineering, 112(8):939–962, 2017.
[67] Serge Beucher and Christian Lantuéjoul. Use of watersheds in contour detection. In International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation, Rennes, France, 1979.
[68] Jean Serra. Introduction to mathematical morphology. Computer vision, graphics, and image processing, 35(3):283–305, 1986.
[69] Steffen Berg, Ryan Armstrong, and Andreas Wiegmann. Gildehauser sandstone. http://www.digitalrocksportal.org/projects/134, 2018.
[70] Ronaldo I Borja. Plasticity, volume 2. Springer, 2013.
[71] Yousef Saad. Iterative methods for sparse linear systems, volume 82. siam, 2003.

**Declaration of interests**

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

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

**Kangan Li:** Methodology, Software Development, Formal analysis, Writing - Original Draft

**Yashar Mehmani**: Conceptualization, Methodology, Software development, Writing - Review & Editing, Funding acquisition, Supervision