The multiscale finite element method for nonlinear continuum localization problems at full fine-scale fidelity, illustrated through phase-field fracture and plasticity

Lam H. Nguyen*, Dominik Schillinger

Department of Civil, Environmental, and Geo- Engineering, University of Minnesota, Twin Cities, USA

Abstract

The residual-driven iterative corrector scheme recently presented by the authors for linear problems has opened a pathway to achieve the best possible fine-mesh accuracy in the multiscale finite element method (MsFEM). In this article, we focus on a series of algorithmic and variational extensions that enable efficient residual-driven correction for nonlinear localization problems. These include a synergistic combination of Newton and corrector iterations to reduce the algorithmic complexity, the use of corrector degrees of freedom in the Galerkin projection to eliminate the repeated recomputation of multiscale basis functions during Newton iterations, and a natural residual-based strategy for fully automatic fine-mesh adaptivity. We illustrate through numerical examples from phase-field fracture and plasticity that the MsFEM with residual-driven adaptive correction achieves full fine-scale fidelity while also being computationally more efficient than the pristine MsFEM. We also show that for localization problems, it significantly increases accuracy and robustness over standard oversampling.

Keywords: Multiscale finite element method, nonlinear localization, residual-driven iterative correction, phase-field fracture, plasticity

*Corresponding author;
Lam H. Nguyen, Department of Civil, Environmental, and Geo- Engineering, University of Minnesota, 500 Pillsbury Drive S.E., Minneapolis, MN 55455; E-mail: nguy2451@umn.edu

Contents

1. Introduction

In the computational modeling of heterogeneous materials such as bone, composites or soils, the interaction of multiple scales needs to be taken into account. One can use assumptions such as scale separation and periodicity to derive continuum homogenization methods [1–8]. In situations where these assumptions do not hold, for instance in localization problems in plasticity or fracture, one possibility to obtain accurate results is to fully resolve all scales [9, 10]. The computational cost associated with full scale resolution, however, easily exceeds current computing resources, even with today's rapid development in supercomputing. Therefore, there is a demand for multi-resolution methods not based on scale separation or periodicity that are significantly less expensive and achieve comparable accuracy with respect to full resolution [11, 12]. Approaches that have been explored include the combination of strong discontinuities with the variational multiscale method [9, 13, 14] and with microscale regularized damage models [15–17], and variational scale interaction mechanisms in multi-resolution meshes [18–21].

The multiscale finite element method (MsFEM) originally introduced by Hou and Wu [22] directly transfers localized fine-scale kinematics to the macroscale via a so-called multiscale basis [23–27]. The solution of the macroscale system requires significantly less memory compared to the full-resolution mesh, while the computation of multiscale basis functions can be efficiently carried out in parallel in an offline step [28]. The MsFEM, however, is currently not associated with the efficient solution of localization problems, as its accuracy is adversely affected by local constraints imposed at the interfaces of coarse-scale elements. These constraints prevent accurate solutions when localized features are crossing element interfaces, such as propagating cracks or plastic bands [9, 13, 21]. To counteract this issue, oversampling methods have been proposed that improve accuracy at element interfaces [22, 29–32]. For localization problems, however, oversampling does not always guarantee accurate solutions, which we will illustrate as part of this work.

In a recent article [33], the authors introduced a local corrector scheme that restores the best possible fine-mesh accuracy, while preserving the core computational advantages of the MsFEM. It is based on a series of corrector problems, each associated with one multiscale basis function and defined locally on its support. The corrector problems are driven by the local residual of the previous multiscale solution [26, 28, 34]. When cast into an iterative scheme, the MsFEM with residual-driven correction converges with a few iterations to the same solution as obtained by full scale resolution, irrespective of the ratio between fine-scale and coarse-scale meshes. In addition, it is simple to implement and can be efficiently parallelized due to the locality of the corrector problems. What remains to be clarified, however, is whether and how residual-driven iterative correction can be extended efficiently to nonlinear problems, in particular since integrating nested corrector loops in the Newton-Raphson algorithm seems an undue computational burden.

In this article, we focus on a series of non-trivial algorithmic and variational extensions that enable efficient residual-driven correction for nonlinear localization problems. Our main idea is to synergistically combine Newton and corrector iterations in the sense of a predictor-corrector scheme [35]. In other words, only one corrector iteration is carried out per Newton iteration instead of a full corrector loop. Given n_{itr} Newton iterations and n_{cor} corrector iterations, this reduces the algorithmic complexity from $\mathcal{O}(n_{itr} \cdot n_{cor})$ to $\mathcal{O}(\max(n_{itr}, n_{cor}))$. This opens the door for fully accurate MsFEM analysis at practically the same cost as the pristine MsFEM.

We discuss a number of additional ideas that further reduce the computational cost. We show that multiscale basis functions need to be computed only once, when we update the multiscale solution throughout the Newton-Raphson procedure by corrector solutions and their degrees of freedom at coarse-scale nodes. To prevent errors due to the loss of orthogonality between the harmonic and bubble parts of the multiscale solution in certain situations, we re-introduce a few coupled bubble degrees of freedom in the macroscale Galerkin projection. In the scope of localization problems, we also describe a residual-based strategy for automatic mesh refinement that naturally reuses the local residual computed by the corrector scheme to automatically drive fine-mesh adaptivity.

We cast all of these developments into MsFEM algorithms whose accuracy and computational efficiency are illustrated through numerical examples from phase-field fracture [36–38] and plasticity [39]. We illustrate, first for the staggered multifield case and then for the fully nonlinear case, that the MsFEM with residual-driven adaptive correction not only achieves full fine-scale accuracy, but also is computationally less expensive than the pristine MsFEM. We also show that for localization problems, residual-driven correction significantly increases accuracy and flexibility over standard oversampling. We emphasize that our nonlinear MsFEM framework is general and can be applied to any other multiscale continuum localization model such as gradient damage [40–43], strain softening laws [44, 45], or cohesive zone models [46, 47].

Our article is organized as follows: In Section 2, we briefly review phase-field fracture and plasticity as two representative examples of continuum localization. In Section 3, we briefly outline the MsFEM, the classical correction based on oversampling, and discuss the residual-driven iterative corrector scheme introduced in [33]. Section 4 focuses in detail on the algorithmic and variational extensions that enable efficient residual-driven correction for nonlinear problems. We also briefly address what seems to be a novel oversampling variant that enables its application to phase-field fracture. Section 5 presents numerical examples that assess the accuracy of the MsFEM with different interelement correctors, demonstrate the efficiency gains due to synergistically combining Newton and corrector iterations and due to residual-based automatic fine-mesh adaptivity, and illustrate the significant memory and computing time savings with respect to full scale resolution. Finally, we summarize and draw conclusions in Section 6.

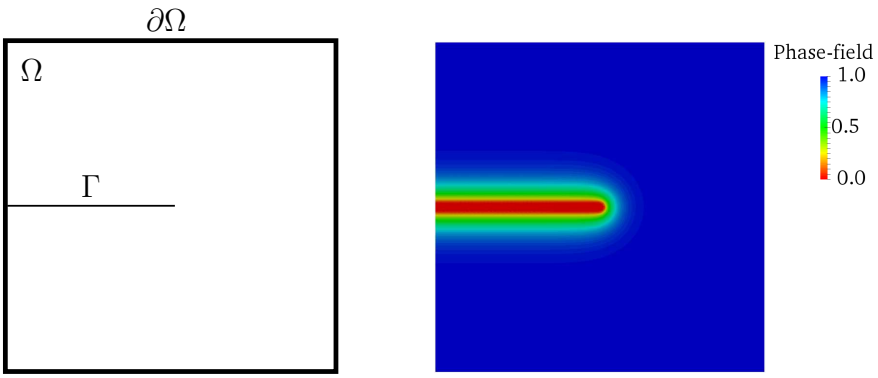
2. Phase-field fracture and plasticity: two examples of continuum localization

We start by reviewing the variational formulations of phase-field fracture and plasticity that we use throughout this work as representative examples of multifield and nonlinear continuum localization problems. Since our presentation of the multiscale finite element method will largely rely on these two examples, we review the material with sufficient depth. Readers with a firm background in these topics are therefore encouraged to skip this section.

2.1. Variational phase-field fracture

This subsection largely follows the presentations in [48, 49]. We consider the domain $\Omega \subset \mathbb{R}^d$ (with $d \in \{1; 2; 3\}$) with external boundary $\partial\Omega$ and fracture surface Γ (see Fig. 1a). We assume small strain isotropic linear elasticity, where the symmetric infinitesimal strain tensor $\boldsymbol{\varepsilon}(\boldsymbol{x}, t)$ is

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right) \tag{1}$$



(a) (b)

Figure 1: The concept of phase-field fracture as an example of a continuum localization problem: (a) a solid body Ω with internal discontinuity boundary Γ that represents a crack, (b) phase-field approximation of Γ.

where $u(\boldsymbol{x}, t)$ is the displacement at point $\boldsymbol{x} \in \Omega$ at pseudo-time t. The elastic energy density is

$$\psi_0(\boldsymbol{\varepsilon}) = \frac{1}{2}\lambda \, \mathrm{tr}(\boldsymbol{\varepsilon})^2 + \mu \, \boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} \tag{2}$$

in which λ and μ are the Lamé constants. The stresses are obtained as

$$\boldsymbol{\sigma} = \partial_{\boldsymbol{\varepsilon}} \psi_0(\boldsymbol{\varepsilon}) = \lambda \, \mathrm{tr}(\boldsymbol{\varepsilon}) \, \boldsymbol{I} + 2\mu \, \boldsymbol{\varepsilon} \tag{3}$$

where $\boldsymbol{I}$ is the unit tensor. In the scope of this work, we focus on quasi-static scenarios, in which we assume that loads are applied infinitely slowly. Following Griffith's theory of brittle fracture, the crack will propagate if the computed strain energy release rate $\mathcal{G}$ is higher than the critical rate $\mathcal{G}_c$. The total internal potential of the body is

$$\Psi(\boldsymbol{\varepsilon}, \Gamma) = \int_\Omega \psi_0(\boldsymbol{\varepsilon}) \, \mathrm{d}\Omega + \int_\Gamma \mathcal{G}_c \, \mathrm{d}\Gamma \tag{4}$$

The first term represents the elastic energy stored in the continuum part of the body and the second term expresses the work necessary to create the current fracture surface Γ.

2.1.1. Phase-field approximation and tensile strain energy degradation

To predict nucleation, propagation and interaction of cracks, Francfort and Marigo [50] proposed to find a global minimizer of (4) for a given load. Solving the associated variational problem, however, is not trivial, since the crack path Γ evolves with (pseudo-)time. To alleviate this issue, Bourdin et al. [36, 51] transferred the surface integral in (4) into a volumetric integral,

$$\int_\Gamma \mathcal{G}_c \, \mathrm{d}\Gamma \approx \int_\Omega \mathcal{G}_c \, \Gamma_c \, \mathrm{d}\Gamma \tag{5}$$

and defined the crack density function Γ_c as

$$\Gamma_c = \frac{1}{4l_0} \left[(c-1)^2 + 4l_0^2 \, |\nabla c|^2 \right] \tag{6}$$

with a length-scale parameter l_0. The scalar-valued phase-field, $c \in [0;1]$, represents the crack, assuming a value of one away from the crack and zero at the crack (see Fig. 1b). To account for the effect of cracks on the behavior of the elastic body, the energy density in (4) is redefined as

$$\tilde{\psi}(\boldsymbol{\varepsilon}, c) = \left(c^2 + \kappa\right) \psi_0(\boldsymbol{\varepsilon}) \tag{7}$$

When the phase-field is equal to zero, the elastic energy density in the localization zone is completely penalized. The numerical parameter $\kappa \ll 1$ prevents the full degradation of the stored energy at the fully-broken state $c = 0$.

To maintain resistance in compression during crack closure, the stress degradation should act in tension only. To this end, Miehe et al. [37, 52] redefined the elastic energy density as

$$\tilde{\psi}(\boldsymbol{\varepsilon}, c) = \left[(1-\kappa)c^2 + \kappa\right] \psi_0^+(\boldsymbol{\varepsilon}) + \psi_0^-(\boldsymbol{\varepsilon}) \tag{8}$$

where ψ_0^+ and ψ_0^- are the tensile and compressive parts of ψ_0, defined as

$$\psi_0^+(\boldsymbol{\varepsilon}) = \frac{1}{2}\lambda \left(\langle \mathrm{tr}(\boldsymbol{\varepsilon})\rangle^+\right)^2 + \mu\, \boldsymbol{\varepsilon}^+ : \boldsymbol{\varepsilon}^+ \quad \text{and} \quad \psi_0^-(\boldsymbol{\varepsilon}) = \frac{1}{2}\lambda \left(\langle \mathrm{tr}(\boldsymbol{\varepsilon})\rangle^-\right)^2 + \mu\, \boldsymbol{\varepsilon}^- : \boldsymbol{\varepsilon}^- \tag{9}$$

In (9), the tensile and compressive modes of the strain tensor, $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^+ + \boldsymbol{\varepsilon}^-$, can be computed based on the spectral decomposition

$$\boldsymbol{\varepsilon}^+ = \sum_{i=1}^{d} \langle \varepsilon_i \rangle^+ \boldsymbol{n}_i \otimes \boldsymbol{n}_i \quad \text{and} \quad \boldsymbol{\varepsilon}^- = \sum_{i=1}^{d} \langle \varepsilon_i \rangle^- \boldsymbol{n}_i \otimes \boldsymbol{n}_i \tag{10}$$

where ε_i and $\boldsymbol{n}_i$ are eigenvalues and eigenvectors of the strain tensor, respectively. The bracket operators are defined as follows

$$\langle x \rangle^+ = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \quad \text{and} \quad \langle x \rangle^- = \begin{cases} x & \text{if } x < 0 \\ 0 & \text{if } x \geq 0 \end{cases} \tag{11}$$

The derivatives in (10) with respect to the total strains

$$\mathbb{P}^+ = \partial_{\boldsymbol{\varepsilon}}\, \boldsymbol{\varepsilon}^+ \quad \text{and} \quad \mathbb{P}^- = \partial_{\boldsymbol{\varepsilon}}\, \boldsymbol{\varepsilon}^- = \mathbb{I} - \mathbb{P}^+ \tag{12}$$

define fourth-order tensors that project the total strains onto its positive and negative parts, i.e. $\boldsymbol{\varepsilon}^+ = \mathbb{P}^+ : \boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}^- = \mathbb{P}^- : \boldsymbol{\varepsilon}$.

2.1.2. Weak form of the coupled multifield problem

We derive the variational form of the phase-field fracture problem from an incremental variational principle that balances the rate of different energy terms. The balance that includes the rates of the stored energy, the dissipated energy due to the work done by fracture, and the energy caused by external forces can be expressed as

$$\dot{\mathcal{E}}_{\text{int}} + \dot{\mathcal{F}}_{\text{frac}} - \dot{\mathcal{P}}_{\text{ext}} = 0 \tag{13}$$

The rate of the stored energy is the time derivative of the integral of the energy density (8),

$$\dot{\mathcal{E}}_{\text{int}} = \frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \tilde{\psi}\,\mathrm{d}\Omega = \int_\Omega \left(\boldsymbol{\sigma}^+ + \boldsymbol{\sigma}^-\right) : \dot{\boldsymbol{\varepsilon}}\,\mathrm{d}\Omega + \int_\Omega 2\,(1-\kappa)\,\psi_0^+ c\,\dot{c}\,\mathrm{d}\Omega \tag{14}$$

where the tensile and compressive parts of the stress tensor are

$$\boldsymbol{\sigma}^+ = \mathbb{D}^+ : \boldsymbol{\varepsilon} = \left[(1-\kappa)c^2+\kappa\right]\partial_{\boldsymbol{\varepsilon}}\psi_0^+ = \left[(1-\kappa)c^2+\kappa\right]\left(\lambda\,\langle \mathrm{tr}(\boldsymbol{\varepsilon})\rangle^+\,\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}^+\right) \tag{15}$$

$$\boldsymbol{\sigma}^- = \mathbb{D}^- : \boldsymbol{\varepsilon} = \partial_{\boldsymbol{\varepsilon}}\psi_0^- = \lambda\,\langle \mathrm{tr}(\boldsymbol{\varepsilon})\rangle^-\,\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}^- \tag{16}$$

Using the two projectors in (12), the corresponding fourth-order material tensors are

$$\mathbb{D}^+ = \begin{cases} \left[(1-\kappa)c^2+\kappa\right](\mathbb{D} - 2\mu\mathbb{P}^-) \\ \left[(1-\kappa)c^2+\kappa\right]2\mu\mathbb{P}^+ \end{cases} \quad \text{and} \quad \mathbb{D}^- = \begin{cases} 2\mu\mathbb{P}^- & \text{if } \mathrm{tr}(\boldsymbol{\varepsilon}) \geq 0 \\ \mathbb{D} - 2\mu\mathbb{P}^+ & \text{if } \mathrm{tr}(\boldsymbol{\varepsilon}) < 0 \end{cases} \tag{17}$$

where $\mathbb{D}$ is the undamaged material tensor. With (5) and (6), the rate of the dissipated energy is

$$\dot{\mathcal{F}}_{\text{frac}} = \frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \mathcal{G}_c\Gamma_{c,2}\,\mathrm{d}\Omega = \int_\Omega \frac{\mathcal{G}_c}{2l_0}(c-1)\dot{c}\,\mathrm{d}\Omega + \int_\Omega 2\mathcal{G}_c l_0 \nabla c\cdot\nabla\dot{c}\,\mathrm{d}\Omega \tag{18}$$

The rate of energy due to the work done by external forces is

$$\dot{\mathcal{P}}_{\text{ext}} = \int_\Omega \boldsymbol{f}\cdot\dot{\boldsymbol{u}}\,\mathrm{d}\Omega + \int_{\Gamma_N}\hat{\boldsymbol{t}}\cdot\dot{\boldsymbol{u}}\,\mathrm{d}\partial\Omega \tag{19}$$

in which $\boldsymbol{f}$ and $\hat{\boldsymbol{t}}$ denote body forces and boundary tractions, respectively. We can now substitute (14), (15), (16), (18) and (19) into (13) and separate the rate of energy functional into a phase-field part and an elasticity part, based on the observation that the balance (13) must hold for arbitrary $\dot{\boldsymbol{u}}$ and $\dot{c}$. This results into the following coupled variational equations,

$$\int_\Omega \nabla\boldsymbol{u} : \left(\mathbb{D}^+ + \mathbb{D}^-\right) : \nabla\dot{\boldsymbol{u}}\,\mathrm{d}\Omega = \int_\Omega \boldsymbol{f}\cdot\dot{\boldsymbol{u}}\,\mathrm{d}\Omega + \int_{\Gamma_N}\hat{\boldsymbol{t}}\cdot\dot{\boldsymbol{u}}\,\mathrm{d}\partial\Omega \tag{20}$$

$$\int_\Omega \left(\frac{4l_0(1-\kappa)\psi_0^+}{\mathcal{G}_c} + 1\right) c\,\dot{c}\,\mathrm{d}\Omega + \int_\Omega 4l_0^2\,\nabla c\cdot\nabla\dot{c}\,\mathrm{d}\Omega = \int_\Omega \dot{c}\,\mathrm{d}\Omega \tag{21}$$

where we identify $\dot{\boldsymbol{u}}$ and $\dot{c}$ as test functions.

2.1.3. Irreversibility and staggered solution procedure

In (21), the tensile part of the strain energy, ψ_0^+, drives the crack evolution. To prevent cracks from healing, if loads are removed and ψ_0^+ decreases, the tensile strain energy ψ_0^+ in (21) is replaced by the following local history field [37],

$$\mathcal{H}(\boldsymbol{x},t) = \max_{s\in[0;t]}\psi_0^+\left(\boldsymbol{\varepsilon}(\boldsymbol{x},s)\right) \tag{22}$$

In the unloading case, the maximum value of $\mathcal{H}$ is preserved such that cracks cannot heal. From a computational viewpoint, this irreversibility concept facilitates the staggered approach for the iterative solution of the phase-field fracture problem [37], summarized in Algorithm 1.

Algorithm 1: Staggered scheme for phase-field fracture

Result: Phase-field c and displacements $\boldsymbol{u}$

if *initial crack* **then**
  Solve (21) with initial $\mathcal{H}$ for the initial phase-field c;
end
while *load* $\leq$ *maxload* **do**
  % Multifield staggered iterations;
  while c *and* $\boldsymbol{u}$ *not converged* **do**
    Solve (20) for displacements $\boldsymbol{u}$;
    Solve (21) for phase-field c;
    Check residuals derived from (20) and (21);
  end
  Update the history value $\mathcal{H}$;
  Load increment *load*++;
end

2.2. *Plasticity: weak form and Newton-Raphson procedure*

We consider the plasticity problem defined by the following boundary value problem

$$-\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{\alpha}, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega \tag{23}$$

$$\boldsymbol{u}|_{\Gamma_D} = \boldsymbol{g} \tag{24}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n}|_{\Gamma_N} = \hat{\boldsymbol{t}} \tag{25}$$

on the domain $\Omega \subset \mathbb{R}^d$ (with $d \in \{1; 2; 3\}$). The stress tensor,

$$\boldsymbol{\sigma} = \mathbb{D}(\boldsymbol{\alpha}, \boldsymbol{u}) : \nabla_s \boldsymbol{u}, \tag{26}$$

depends on the nonlinear material tensor $\mathbb{D}$, which is now a function of a general set of internal variables $\boldsymbol{\alpha}$ and the displacement vector $\boldsymbol{u}$. In this work, we focus on von Mises plasticity, based on the von Mises failure criterion

$$\sqrt{\frac{3}{2}\boldsymbol{s} \cdot \boldsymbol{s}} \leq \sigma_y \,, \tag{27}$$

where $\boldsymbol{s}$ and σ_y are the deviatoric stress tensor and the yield stress, respectively. We assume the standard associated flow rule and no hardening (perfect plasticity) to support localization.

Following the presentation in [39], based on the principle of virtual work, equation (23) can be cast into the following weak from,

$$\int_\Omega \delta\boldsymbol{\varepsilon} : \boldsymbol{\sigma} \, \mathrm{d}\Omega = \int_\Omega \delta\boldsymbol{u} \cdot \boldsymbol{f} \, \mathrm{d}\Omega + \int_{\Gamma_N} \delta\boldsymbol{u} \cdot \hat{\boldsymbol{t}} \, \mathrm{d}\partial\Omega \,, \tag{28}$$

where the virtual strain tensor $\delta\boldsymbol{\varepsilon}$ follows from the variation of (1). We assume that equation (28) is discretized by a finite element approach [39] with a set of basis functions that are contained in the

vector $\boldsymbol{N}$. The solution of the resulting set of equations that is nonlinear and path-dependent can be obtained with the Newton-Raphson scheme [39, 53]. In this context, the nonlinear system of equations is typically reformulated in residual form,

$$\boldsymbol{r} = \boldsymbol{f}^{ext} - \boldsymbol{f}^{int} \,. \tag{29}$$

The residual $\boldsymbol{r}$ is defined as the difference between the external force vector,

$$\boldsymbol{f}^{ext} = \int_{\Omega} \boldsymbol{N}^T \boldsymbol{f} \, \mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{N}^T \hat{\boldsymbol{t}} \, \mathrm{d}\partial\Omega \,, \tag{30}$$

and the internal force vector,

$$\boldsymbol{f}^{int} = \int_{\Omega} \boldsymbol{B}^T \boldsymbol{\sigma}(\boldsymbol{\alpha}, \boldsymbol{u}) \, \mathrm{d}\Omega \,, \tag{31}$$

where $\boldsymbol{B}$ denotes the standard strain-displacement matrix.

In the Newton-Raphson scheme, equation (29) is linearized at the current displacement solution $\boldsymbol{u}_k$, which gives rise to the following incremental format,

$$\boldsymbol{K}_t \Delta\tilde{\boldsymbol{u}} = \boldsymbol{r}(\boldsymbol{u}_k) \,, \tag{32}$$

where the vectors $\tilde{\boldsymbol{u}}_k$ and $\Delta\tilde{\boldsymbol{u}}$ contain the coefficients of the current displacement approximation and its increment, respectively. The tangent stiffness matrix $\boldsymbol{K}_t$ can be computed as

$$\boldsymbol{K}_t = \left. \frac{\partial \boldsymbol{R}}{\tilde{\boldsymbol{u}}} \right|_{\partial\tilde{\boldsymbol{u}}_k} = \int_{\Omega} \boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B} \, \mathrm{d}\Omega \,, \tag{33}$$

where $\boldsymbol{D}$ is the consistent tangent matrix that can be obtained as detailed in Appendix A. Solving the linearized system (32), we can find the coefficient vector of the updated displacement approximation, $\tilde{\boldsymbol{u}}_{k+1} = \tilde{\boldsymbol{u}}_k + \Delta\tilde{\boldsymbol{u}}$. This gives rise to the iterative procedure, summarized in Algorithm 2.

Algorithm 2: Newton-Raphson procedure for plasticity

```
Result: Displacements u
while load ≤ maxload do
    % Newton iterations;
    while u not converged do
        Compute r from (29), check convergence;
        Compute K_t from (33);
        Solve (32) for Δũ;
        Update displacement solution;
    end
    Update internal variables α;
    Load increment load++;
end
```

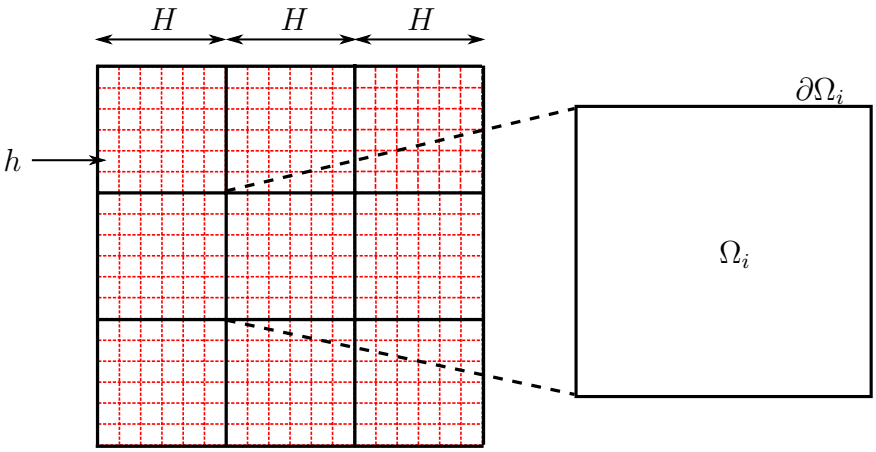


Figure 2: Multiscale discretization of the domain Ω.

3. The multiscale finite element method with interelement correction

In the next step, we review the multiscale finite element method (MsFEM) for the analysis of multiscale vector problems. The pristine MsFEM applies simple local constraints based on linear interpolation that precludes fine-scale accuracy across coarse-scale element interfaces. In this section, we therefore focus on correction procedures that improve interelement accuracy, which is decisive for the accurate analysis of localization phenomena. We first discuss the standard oversampling method and then present an alternative residual-driven corrector scheme that we recently introduced in [33] for linear problems.

3.1. Multiscale solution procedure

The fundamental ideas underlying the multiscale finite element method were firstly introduced by Hou and Wu [22] for scalar-field elliptic problems. In this work, we directly consider vector-field problems, where we follow the displacement-based MsFEM formulation for elasticity derived in [54]. To fix notation, we simplify the boundary value problem (23) to (25) to the following two-dimensional elasticity problem,

$$-\nabla \cdot (\mathbb{D} : \nabla_s \boldsymbol{u}(\boldsymbol{x})) = \boldsymbol{f}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega \tag{34}$$

$$\boldsymbol{u}(\boldsymbol{x})|_{\partial\Omega} = \boldsymbol{g}(\boldsymbol{x})\,, \tag{35}$$

with the elastic material tensor $\mathbb{D}$ and the vector solution $\boldsymbol{u} = (u_x\ u_y)^T$, whose components u_x and u_y are displacement functions in x- and y- directions, respectively.

As illustrated in Fig. 2, the domain Ω is discretized by coarse-scale elements, $\Omega = \cup_i \Omega_i$, with element size H, and by an underlying fine-scale mesh of element size h. The coarse-scale interpolation of the displacement vector is

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_{i=1}^{m} \begin{pmatrix} \phi_{ixx} \\ \phi_{ixy} \end{pmatrix} u_{ix} + \sum_{i=1}^{m} \begin{pmatrix} \phi_{ixy} \\ \phi_{iyy} \end{pmatrix} u_{iy} = \sum_{i=1}^{m} \left(\boldsymbol{\phi}_{ix} u_{ix} + \boldsymbol{\phi}_{iy} u_{iy} \right)\,, \tag{36}$$

where m is the number of coarse-scale nodes. The vectors $\boldsymbol{\phi}_{ix}$ and $\boldsymbol{\phi}_{iy}$ are the two multiscale basis functions corresponding to x- and y- directions at node i. The two components ϕ_{ixx} and ϕ_{ixy} of multiscale basis function $\boldsymbol{\phi}_{ix}$, for instance, represent the multiscale displacement in x-direction and the associated multiscale Poisson effect in y-direction, respectively. u_{ix} and u_{iy} are the nodal displacement coefficients in x- and y- directions at node i.

To find the two multiscale basis function vectors $\boldsymbol{\phi}_{ix}$ and $\boldsymbol{\phi}_{iy}$, we firstly consider the decomposition of the solution $\boldsymbol{u}$ [26, 27],

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}_a(\boldsymbol{x}) + \boldsymbol{u}_b(\boldsymbol{x})\,. \tag{37}$$

The harmonic part $\boldsymbol{u}_a$ is the solution of the following local boundary value problem:

$$-\nabla\cdot(\mathbb{D}:\nabla_s\boldsymbol{u}_a(\boldsymbol{x})) = \boldsymbol{0},\quad \boldsymbol{x}\in\Omega_i \tag{38}$$

$$\boldsymbol{u}_a(\boldsymbol{x})|_{\partial\Omega_i} = \boldsymbol{u}(\boldsymbol{x})|_{\partial\Omega_i}\,, \tag{39}$$

and the bubble part $\boldsymbol{u}_b$ is the solution of the following local boundary value problem:

$$-\nabla\cdot(\mathbb{D}:\nabla_s\boldsymbol{u}_b(\boldsymbol{x})) = \boldsymbol{f}(\boldsymbol{x}),\quad \boldsymbol{x}\in\Omega_i \tag{40}$$

$$\boldsymbol{u}_b(\boldsymbol{x})|_{\partial\Omega_i} = \boldsymbol{0}\,. \tag{41}$$

The two local problems are defined individually on each coarse-scale element domain Ω_i. The boundary conditions for $\boldsymbol{u}_a$ are the exact (unknown) solution components on element interfaces, while homogeneous boundary conditions are imposed on $\boldsymbol{u}_b$. It is straightforward to show that $\boldsymbol{u}_a$ and $\boldsymbol{u}_b$ are orthogonal [26], satisfying

$$\int_{\Omega_i}\nabla\boldsymbol{u}_b:\mathbb{D}:\nabla\boldsymbol{u}_a\,\mathrm{d}\Omega = 0\,. \tag{42}$$

Substituting the approximation of $\boldsymbol{u}_a$ based on (36) into (38), we obtain

$$-\sum_{j=1}^{m}\left(\nabla\cdot(\mathbb{D}:\nabla_s\boldsymbol{\phi}_{jx})\,u_{a,jx}\right)-\sum_{j=1}^{m}\left(\nabla\cdot(\mathbb{D}:\nabla_s\boldsymbol{\phi}_{jy})\,u_{a,jy}\right)=\boldsymbol{0},\quad \boldsymbol{x}\in\Omega_i \tag{43}$$

At each node of the coarse-scale mesh, we can determine the two multiscale basis functions by solving the following element-wise boundary problems:

$$-\nabla\cdot(\mathbb{D}:\nabla_s\boldsymbol{\phi}_{jx}) = \boldsymbol{0}\quad\text{and}\quad -\nabla\cdot(\mathbb{D}:\nabla_s\boldsymbol{\phi}_{jy}) = \boldsymbol{0},\quad \boldsymbol{x}\in\Omega_i\,, \tag{44}$$

using the fine-scale mesh and simple boundary conditions illustrated in Fig. 3. For the multiscale basis function $\boldsymbol{\phi}_{jx} = (\phi_{jxx}\ \ \phi_{jxy})^T$, we set ϕ_{jxx} to the corresponding standard basis function with a unit displacement in x-direction at node j while fixing ϕ_{jxy} to zero on element interfaces. The equivalent procedure is applied for $\boldsymbol{\phi}_{jy}$.

Using the multiscale basis $\boldsymbol{\phi}_{jx}$ and $\boldsymbol{\phi}_{jy}$ with $j = 1, ..., m$, the MsFEM searches the numerical solution $\boldsymbol{u}_{ms}^H \in V^H$ such that

$$a\left(\boldsymbol{u}_{ms}^H(\boldsymbol{x}),\boldsymbol{v}(\boldsymbol{x})\right) = \langle\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{v}(\boldsymbol{x})\rangle\,,\quad \text{for all } \boldsymbol{v}(\boldsymbol{x})\in V^H \tag{45}$$

$$V^H = \text{span}\left\{\boldsymbol{\phi}_{1x}(\boldsymbol{x}),\boldsymbol{\phi}_{1y}(\boldsymbol{x}),\boldsymbol{\phi}_{2x}(\boldsymbol{x}),\boldsymbol{\phi}_{2y}(\boldsymbol{x}),...,\boldsymbol{\phi}_{mx}(\boldsymbol{x}),\boldsymbol{\phi}_{my}(\boldsymbol{x})\right\}\subset H_0^1(\Omega)\,,$$

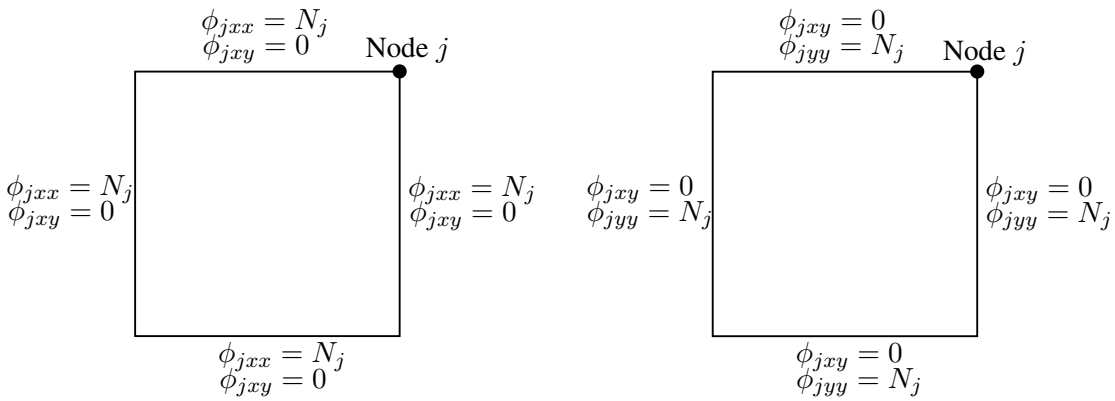


Figure 3: Local boundary conditions of the multiscale basis functions at node j in a coarse-scale element.

where $a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \nabla \boldsymbol{u}(\boldsymbol{x}) : \mathbb{D} : \nabla \boldsymbol{v}(\boldsymbol{x}) \, \mathrm{d}\Omega$ and $\langle \boldsymbol{f}, \boldsymbol{v} \rangle = \int_\Omega \boldsymbol{f}(\boldsymbol{x}) \cdot \boldsymbol{v}(\boldsymbol{x}) \, \mathrm{d}\Omega$.

The bubble part $\boldsymbol{u}_b$ is found by solving (40) separately in each coarse-scale element domain. The complete multiscale finite element solution, $\bar{\boldsymbol{u}}$, is obtained by adding the local bubble part $\boldsymbol{u}_b$ to the multiscale solution $\boldsymbol{u}^H_{ms}$,

$$\bar{\boldsymbol{u}}(\boldsymbol{x}) = \boldsymbol{u}^H_{ms}(\boldsymbol{x}) + \boldsymbol{u}_b(\boldsymbol{x}) \, . \tag{46}$$

We note that this procedure directly extends to three dimensions and can be readily adapted to other vector problems. We emphasize that a proper MsFEM implementation uses extraction-type coefficient matrices to assemble global entities entirely from element-wise entities [23, 33].

3.2. Oversampling

Already in the first publications on the MsFEM [22, 55], it was shown that local constraints on coarse-scale element boundaries have a significant impact on the accuracy of the multiscale solution (46). It turned out that the error can be particularly high in the thin layers along the boundaries of each local element domain, a phenomenon that was termed scale resonance. To improve the solution accuracy with respect to the boundary conditions in Fig. 3, the oversampling technique was proposed. The basic idea of oversampling is to first construct multiscale basis functions for each coarse-scale element on a larger domain that exceeds the element, called the oversampling domain. These multiscale basis functions are then restricted to the actual element domain.

We again illustrate the procedure for the two-dimensional case. Figure 4 illustrates the oversampling domain for one coarse element Ω_i marked in red. To construct the multiscale basis functions, we consider the oversampling domain Ω_{os}, extended by length d_{os} from the original element domain Ω_i. We firstly compute temporary multiscale basis functions $\boldsymbol{\Phi}_{jx}$ and $\boldsymbol{\Phi}_{jy}$, $j = 1, ..., m_e$, defined on Ω_{os}. Here, m_e is the number of coarse nodes in each coarse-scale element. We then construct the actual multiscale basis functions $\boldsymbol{\phi}_{jx}$, $j = 1, ..., m_e$, defined on the coarse-scale element. This is achieved by the linear combination

$$\boldsymbol{\phi}_{jx} = \sum_{k=1}^{m_e} \left(\tilde{c}_{jxk} \boldsymbol{\Phi}_{kx} + \tilde{c}_{jyk} \boldsymbol{\Phi}_{ky} \right) \, , \tag{47}$$

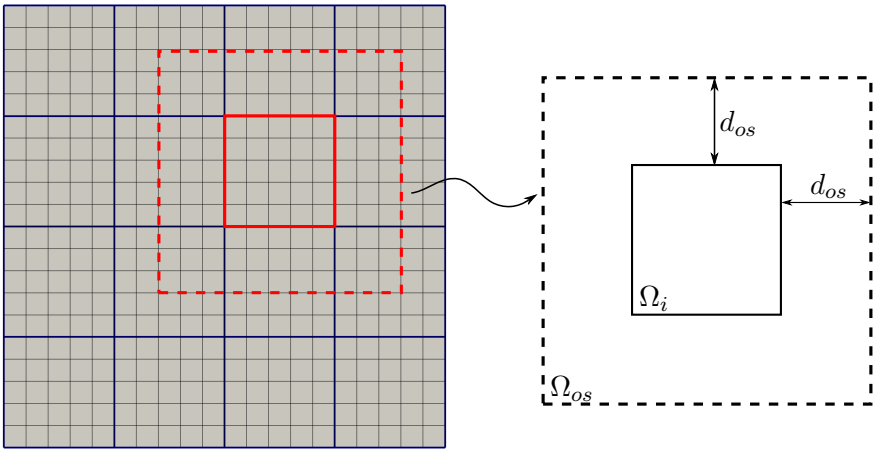


Figure 4: Oversampling domain.

where $\tilde{c}_{jxk}$ and $\tilde{c}_{jyk}$ are constants that we can determine by enforcing the condition

$$\boldsymbol{\phi}_{jx}(\boldsymbol{x}_k) = (\delta_{jk}\,,\,0)^T\,. \tag{48}$$

The location $\boldsymbol{x}_k$ refers to the position of the k^{th} coarse-scale node. The same procedure can be applied for $\boldsymbol{\phi}_{jy}$ with the constants $\tilde{c}_{jxk}$ and $\tilde{c}_{jyk}$ that follow from the condition $\boldsymbol{\phi}_{jy}(\boldsymbol{x}_k) = (0\,,\,\delta_{jk})^T$.

3.3. Residual-driven iterative correction

The oversampling method improves accuracy across coarse-scale element interfaces, but does not enable the best possible accuracy. An alternative is the residual-driven corrector scheme recently introduced by the authors in [33]. It iteratively improves the accuracy of multiscale basis functions until they are able to represent the best-possible fine-mesh solution at element interfaces.

3.3.1. The local corrector problem

Let us consider the support $\tilde{\Omega}_i$ of the multiscale basis function $\boldsymbol{\phi}_i$ with node i and boundary $\partial\tilde{\Omega}_i$, illustrated as the gray-shaded region in Fig. 5. To improve the accuracy of this basis function, we introduce a corrector solution $\boldsymbol{u}_c$ which can be determined from the following local boundary value problem:

$$-\nabla\cdot(\mathbb{D}:\nabla_s\boldsymbol{u}_c(\boldsymbol{x})) = \boldsymbol{r}(\boldsymbol{x}),\quad \boldsymbol{x}\in\tilde{\Omega}_i \tag{49}$$

$$\boldsymbol{u}_c(\boldsymbol{x})|_{\partial\tilde{\Omega}_i} = \boldsymbol{0},\quad \boldsymbol{x}\in\partial\tilde{\Omega}_i\,. \tag{50}$$

We observe that the corrector solution is directly driven by the local residual defined as

$$\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) - \bar{\boldsymbol{f}}(\boldsymbol{x})\,, \tag{51}$$

where

$$\bar{\boldsymbol{f}}(\boldsymbol{x}) := -\nabla\cdot(\mathbb{D}:\nabla_s\bar{\boldsymbol{u}}(\boldsymbol{x}))\,, \tag{52}$$

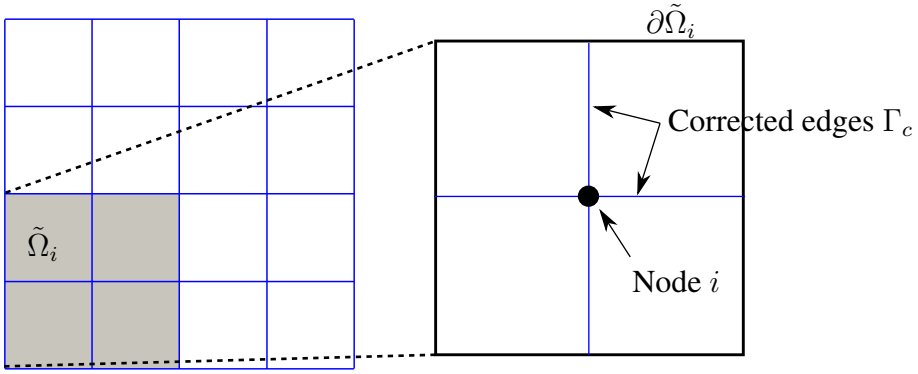


Figure 5: Local corrector region that corresponds to the support $\tilde{\Omega}_i$ of the multiscale basis function ϕ_i.

with $\bar{\boldsymbol{u}}$ being the current solution to be corrected. The corrector solution of each local corrector problem can be found via finite element discretizations in a straightforward way. To this end, we cast (49) and (50) into the following variational form:

$$\int_{\tilde{\Omega}_i} \nabla \boldsymbol{v}^h : \mathbb{D} : \nabla \boldsymbol{u}_c^h \, \mathrm{d}\Omega = \int_{\tilde{\Omega}_i} \boldsymbol{v}^h \cdot \boldsymbol{f} \, \mathrm{d}\Omega - \int_{\tilde{\Omega}_i} \nabla \boldsymbol{v}^h : \mathbb{D} : \nabla \bar{\boldsymbol{u}}^h \, \mathrm{d}\Omega \,, \tag{53}$$

where $\boldsymbol{v}^h$ and $\boldsymbol{u}_c^h$ are finite element approximations of the test functions and the corrector solution for the coarse-scale node i. They are represented on the corresponding local corrector region $\tilde{\Omega}_i$ by the fine mesh available in each coarse-scale element in the sense of Fig. 2. The current approximation of the multiscale solution $\bar{\boldsymbol{u}}^h$ comes from previous MsFEM computations, where the initial solution is delivered by the pristine MsFEM. Since local constraints are acting on element interfaces, the residual $\boldsymbol{r}$ in (51) is nonzero only on element interfaces.

The corrector solution can now be used to improve accuracy of the multiscale basis functions. For the elasticity problem, however, there is no mechanism in the formulation that specifies how to distribute a single corrector solution onto multiple basis functions in an optimal way. To resolve this issue, we employ the concept of corrector degrees of freedom that are introduced at each coarse-scale node as illustrated in Fig. 6. At each coarse-scale node i, this new degree of freedom, $u_{i,cor}$, absorbs the corrector solutions during corrector iterations. The correction effect is thus incorporated automatically and optimally in the Galerkin sense.

Remark 1: It is interesting to note that in scalar problems, we can obtain a new multiscale basis function ϕ_i^{new} at node i by directly adding the corrector solution to the current basis function:

$$\phi_i^{new} = \phi_i u_i + u_c \,, \tag{54}$$

where ϕ_i and u_i are the scalar multiscale basis function at node i and its solution coefficient, respectively. We note that one has to immediately set u_i to one after the update (54) [33].

3.3.2. Local imposition of global Dirichlet boundary conditions

In the pristine MsFEM, Dirichlet boundary conditions are imposed via the coarse-scale nodes only. The accuracy of the boundary interpolation therefore directly depends on the local constraints

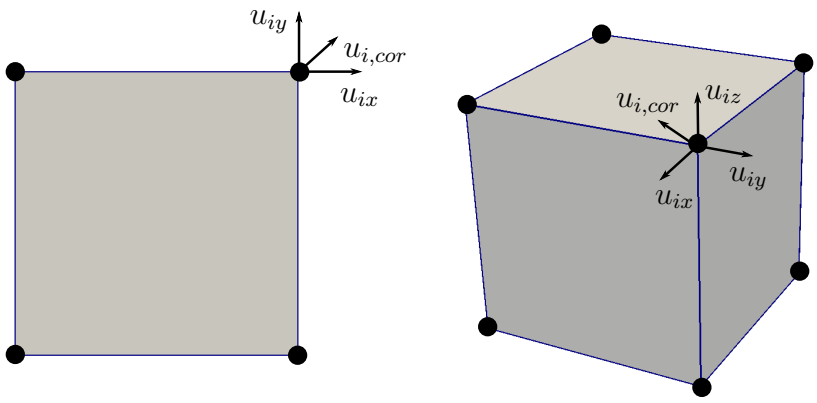


Figure 6: The concept of corrector degrees of freedom leads to one additional degree of freedom at each vertex node.

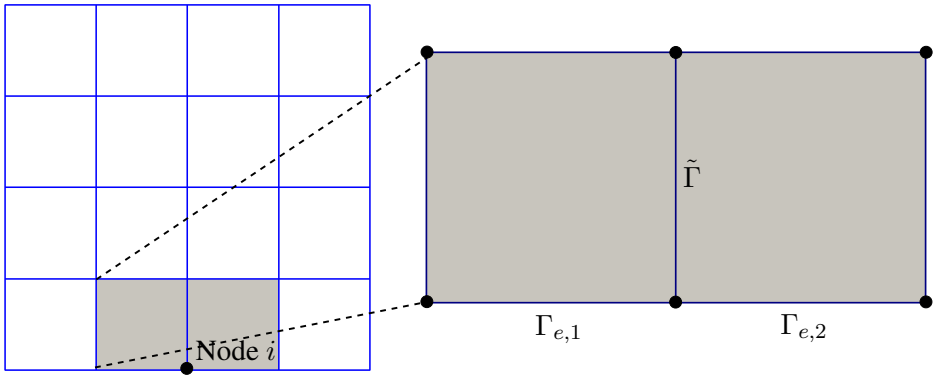


Figure 7: Corrector problem involving the global Dirichlet boundary.

on coarse-scale element boundaries. As a consequence, the interpolation at the global domain boundary only converges to the fine-mesh solution when we correct boundary multiscale basis functions to satisfy the correct Dirichlet boundary conditions at the fine-scale mesh.

To this end, we consider the corrector region shown in Fig. 7, where $\tilde{\Gamma}_D = \Gamma_{e,1} \cup \Gamma_{e,2}$ is a part of the global boundary $\partial\Omega$ with Dirichlet boundary conditions. In the fine-mesh discretization of the corrector problem corresponding to node i, we impose the following boundary condition

$$\boldsymbol{u}_c(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x}) - \bar{\boldsymbol{u}}(\boldsymbol{x}), \ \boldsymbol{x} \in \tilde{\Gamma}_D \tag{55}$$

At all other interior boundaries of the local corrector domain, homogeneous boundary conditions according to (50) are imposed.

3.3.3. Iterative algorithm, automatic adaptivity and robust parallelization

At this point, all components are in place to establish the iterative corrector scheme. The computational procedure is summarized in Fig. 8 and Algorithm 3. There are two important aspects that need to be discussed in conjunction with the iterative algorithm. First, the availability of the residual throughout the computational procedure opens the door for an automatic and natural adaptivity strategy. Since the corrector scheme is driven by the residual, it automatically detects

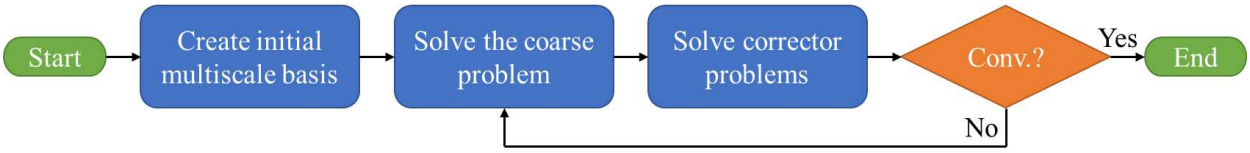


Figure 8: Iterative corrector scheme

Algorithm 3: Iterative corrector scheme

Result: Complete multiscale solution $\bar{\boldsymbol{u}}$

Solve (40) for the bubble part $\boldsymbol{u}_b$;
Solve (44) for the initial multiscale basis functions;
while $\|\bar{\boldsymbol{u}}_k - \bar{\boldsymbol{u}}_{k-1}\| \geq tol.$ **do**
    Solve macroscale problem (45) for $\boldsymbol{u}_{ms}^H$;
    Update multiscale solution $\bar{\boldsymbol{u}}_k = \boldsymbol{u}_{ms}^H + \boldsymbol{u}_b$;
    for $i = 1 \rightarrow$ *numberOfCorrectorProblems* **do**
        Compute $\boldsymbol{r}$, being the right-hand side of (53);
        % Automatic fine-mesh adaptivity;
        if $\|\boldsymbol{r}\| \geq tol.$ **then**
            Solve corrector problem (53) for $\boldsymbol{u}_c$;
            Use local corrector solution $\boldsymbol{u}_c$ to update $\bar{\boldsymbol{u}}_k$;
        end
    end
end

regions with high residuals where localization occurs, for instance fracture regions or plastic shear bands, and locally improves the accuracy of the multiscale solution. Regions where the residual is small are automatically excluded from the computation. This mechanism reduces the computational cost significantly, which is especially important in the presence of localization phenomena, where the error can be expected to concentrate in a few coarse-scale elements. We will illustrate and further examine this important aspect in Section 5 via numerical examples.

The second aspect concerns the parallelization of Algorithm 3. We first note that updating a multiscale basis function changes the multiscale solution $\bar{\boldsymbol{u}}$ in the local support of that multiscale basis function. To avoid race conditions in a parallel corrector scheme, we group multiscale basis functions and associated nodes in such a way that the corrector regions of each group do not overlap and multiscale basis functions of each group do not directly affect each other's supports. The grouping procedure has been coined corrector patterning in [33]. Figure 9 illustrates four possible corrector patterns of a two-dimensional mesh with 4×4 elements. We can work on all nodes in one pattern in parallel and then move to the next pattern. This guarantees that no overlapping corrector regions are being considered at the same time. For an extension of the idea of corrector patterns to three-dimensional problems and a simple algorithm to create these patterns, we refer to [33].

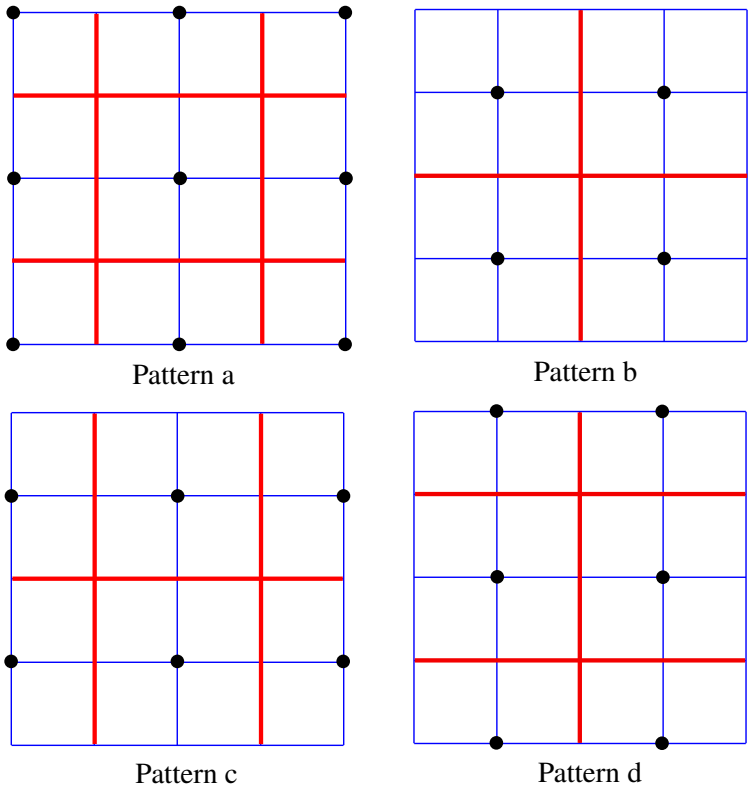


Figure 9: Four corrector patterns in 2D (the red lines show the local boundaries of non-overlapping corrector regions).

4. Extension to multifield and nonlinear problems

In this section, we introduce algorithmic and variational concepts that allow the efficient extension of the multiscale finite element method to nonlinear problems. We demonstrate that for MsFEM analysis of problems beyond linear scalar-field equations a number of additional aspects need to be taken into account. We first focus on the somewhat simpler multifield case before proceeding to full nonlinearity, using phase-field fracture and plasticity as model problems. Since both involve localization, constraints at coarse-scale element interfaces significantly affect accuracy, requiring effective interelement correction. For each of the two cases, we first discuss a general MsFEM procedure and then focus on particular aspects with respect to the extensions of oversampling and residual-driven iterative correction.

4.1. Multifield MsFEM, illustrated for phase-field fracture

In order to apply the MsFEM to a multifield problem, we require several different multiscale bases, each specifically tailored to one of the multiple solution fields. For the example of phase-field fracture, this requires an extension of Algorithm 1.

4.1.1. General procedure

For the pristine MsFEM, we suggest the algorithmic approach outlined in Fig. 10. In each multifield staggered iteration, we determine multiscale basis functions from element-wise local

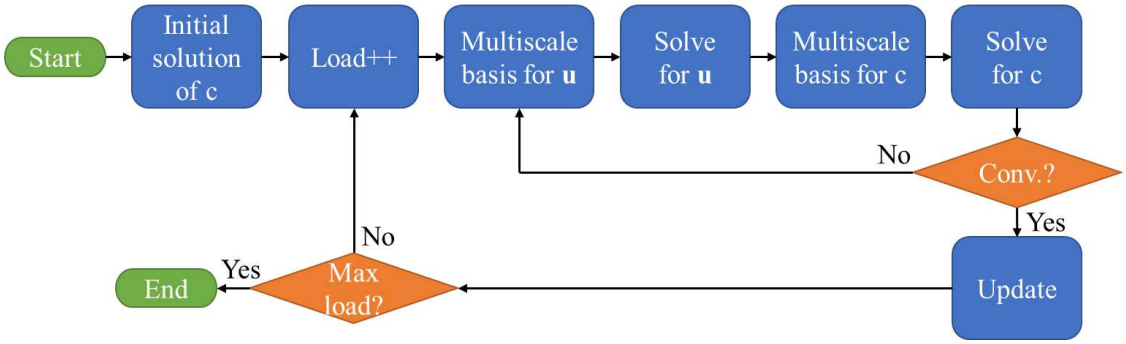


Figure 10: Computational procedure for the solution of a phase-field fracture problem with the pristine MsFEM without interelement correction and the MsFEM with oversampling. In each multifield iteration, a new set of multiscale basis functions must be computed.

problems and then find the new solution from the macroscale problem for the displacements $\boldsymbol{u}$ and the phase-field c independently. We repeat the staggered iterations at constant load, until both fields have sufficiently converged (see also Algorithm 1).

To determine the multiscale basis functions for the displacements $\boldsymbol{u}$, we derive the following local variational problem from (20):

$$\int_{\Omega_i} \nabla \boldsymbol{\phi}_j : \left(\mathbb{D}^+ + \mathbb{D}^-\right) : \nabla \delta \boldsymbol{\phi}_j \, \mathrm{d}\Omega = \mathbf{0} \,, \tag{56}$$

with boundary conditions according to Fig. 3, where $\delta \boldsymbol{\phi}_j$ are the virtual multiscale basis functions at node j. We emphasize that $\boldsymbol{\phi}_j$ can be either $\boldsymbol{\phi}_{jx}$ or $\boldsymbol{\phi}_{jy}$. The bubble part of the multiscale solution in each coarse-scale element can be determined from the following local variational statement:

$$\int_{\Omega_i} \nabla \boldsymbol{u}_b : \left(\mathbb{D}^+ + \mathbb{D}^-\right) : \nabla \delta \boldsymbol{u}_b \, \mathrm{d}\Omega = \int_{\Omega_i} \boldsymbol{f} \cdot \delta \boldsymbol{u}_b \, \mathrm{d}\Omega + \int_{(\Gamma_N)_i} \hat{\boldsymbol{t}} \cdot \delta \boldsymbol{u}_b \, \mathrm{d}\Omega \,, \tag{57}$$

where $\delta \boldsymbol{u}_b$ is the virtual displacement of the bubble part and $(\Gamma_N)_i$ is the part of the element boundary that coincides with the global Neumann boundary.

For the phase-field c, we employ the same procedure. Using the simple boundary conditions of Fig. 3, we can determine the multiscale basis functions ϕ_{cj} in each coarse-sale element as the solution of the following local variational problem:

$$\int_{\Omega_i} \left(\frac{4 l_0 (1-\kappa) \mathcal{H}}{\mathcal{G}_c} + 1 \right) \phi_{cj} \, \delta \phi_{cj} \, \mathrm{d}\Omega + \int_{\Omega_i} 4 l_0^2 \, \nabla \phi_{cj} \cdot \nabla \delta \phi_{cj} \, \mathrm{d}\Omega = 0 \,, \tag{58}$$

in which $\delta \phi_{cj}$ represents the virtual multiscale basis function at node j. Using homogeneous boundary conditions at element boundaries, we can determine the bubble part c_b of the phase-field as the solution of the following local variational problem:

$$\int_{\Omega_i} \left(\frac{4 l_0 (1-\kappa) \mathcal{H}}{\mathcal{G}_c} + 1 \right) c_b \, \delta c_b \, \mathrm{d}\Omega + \int_{\Omega_i} 4 l_0^2 \, \nabla c_b \cdot \nabla \delta c_b \, \mathrm{d}\Omega = \int_{\Omega_i} \delta c_b \, \mathrm{d}\Omega \tag{59}$$

where δc_b is the virtual bubble part of the phase-field. We note that the variational statements (58) and (59) can be easily derived by transforming the phase-field equation (21) into its strong form [48, 49] and following the same steps that are detailed in Section 3.1 for the elasticity equation.

We observe that the variational statements (56) through (59) change in each multifield staggered iteration, since they either depend on the displacements $\boldsymbol{u}$ or the phase-field c that are continuously updated as the solutions of the two decoupled macroscale problems. Therefore, in the pristine MsFEM the multiscale basis functions need to be recomputed in each multifield iteration. The computation of multiscale basis functions (offline step) can be efficiently parallelized, but still constitutes a computational burden when repeated many times.

4.1.2. Oversampling of the phase-field bubble part

For the MsFEM with oversampling, the algorithmic approach of Fig. 10 holds in the same way. When applying the oversampling method for interelement correction in phase-field fracture, however, the following issue needs to be addressed. In the completely undamaged state, the history field $\mathcal{H}$ is zero and the equation (21) can be rewritten as

$$\int_\Omega \frac{1}{4l_0^2}\ (c-1)\ \delta c\ \mathrm{d}\Omega + \int_\Omega \nabla c \cdot \nabla \delta c\ \mathrm{d}\Omega = 0\,. \tag{60}$$

Its exact solution $c = 1$ can be easily found by inspection. Based on this observation, we require that all multiscale basis functions and the bubble part can sum up to one, such that the complete multiscale solution of the phase-field is able to represent the undamaged state exactly.

Following (60), we can find multiscale basis functions in the undamaged state by solving the following local variational problem:

$$\int_\Omega \frac{1}{4l_0^2}\ (\phi_{cj}-1)\ \delta\phi_{cj}\ \mathrm{d}\Omega + \int_\Omega \nabla\phi_{cj} \cdot \nabla\delta\phi_{cj}\ \mathrm{d}\Omega = 0 \tag{61}$$

on each coarse-scale element domain. If we employ the oversampling technique, we firstly solve (61) with the simple boundary conditions on the oversampling domain Ω_{os}. We then construct the actual multiscale basis functions by a linear combination in analogy to (47).

In the case of $l_0 \ll 1$, the first term in (61) drives the solution to zero inside the oversampling domain. Therefore, the values of the actual multiscale basis functions are close to zero on element interfaces (except at the corresponding coarse-scale node). Figure 11a illustrates the shape of these basis functions by a two-dimensional example. We note that the multiscale basis function plotted on the coarse-scale element domain was computed on the oversampling domain shown in Fig. 4 and scaled to one at the corresponding coarse-scale node according to (47).

The solution of the bubble part in the undamaged state can be obtained by solving the following local variational problem:

$$\int_{\Omega_i} \frac{1}{4l_0^2}\ (c_b-1)\ \delta c_b\ \mathrm{d}\Omega + \int_{\Omega_i} \nabla c_b \cdot \nabla\delta c_b\ \mathrm{d}\Omega = 0 \tag{62}$$

on each coarse-scale element domain with homogeneous boundary conditions on element interfaces. The bubble solution is driven to $c_b = 1$ inside the element domain by the first term in (62). Figure 11b illustrates the shape of the bubble part, plotting it over the coarse-scale element domain.

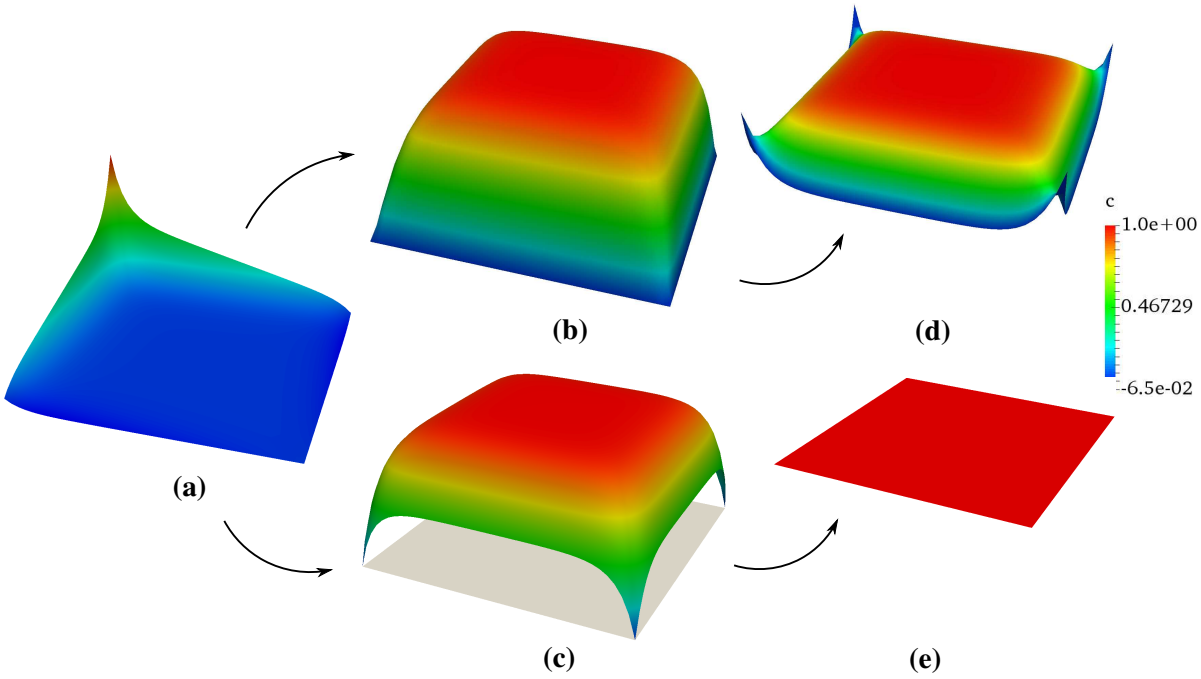


Figure 11: Phase-field oversampling: (a) one multiscale basis function; (b) bubble part of the solution computed on the coarse-scale element domain; (c) bubble part computed on the oversampling domain; (d), (e) sums of multiscale basis and bubble part for each variant. Note that all functions are plotted on the coarse-scale element domain.

When adding all multiscale basis functions and the bubble part together, we obtain the result plotted in Fig. 11d. We observe that we obtain the correct solution inside the element domain, but a significant deviation from one near the element boundaries. The reasons for this deviation can be summarized as follows: (i) the mass term in (60) is multiplied by a large penalty-type parameter when $l_0 \ll 1$; and (ii) the multiscale basis functions are constructed based on the oversampling domain while the bubble part must be computed only on the element domain.

To overcome this issue, we suggest that we firstly construct the bubble solution on the oversampling domain and then extract the actual bubble part ϕ_b from the following relation:

$$\phi_b = \Phi_b - \sum_{k=1}^{m_e} \Phi_b(\boldsymbol{x}_k)\phi_k, \quad \boldsymbol{x} \in \Omega_i \,. \tag{63}$$

Here, the bubble solution Φ_b is obtained on the oversampling domain and the multiscale basis functions ϕ_k are obtained by the oversampling technique for scalar-field problems in analogy to (47). Figure 11c illustrates the shape of the new oversampled bubble part plotted on the coarse-scale element. Figure 11e demonstrates that when we add all multiscale basis functions and the bubble part, we obtain a constant sum of one. Therefore, the undamaged state can be represented exactly.

4.1.3. Residual-driven correction for phase-field fracture

The displacements $\boldsymbol{u}$ and the phase-field c are vector and scalar fields, respectively. For both fields, we apply the concept of additional corrector degrees of freedom described in Section 3.3.1.

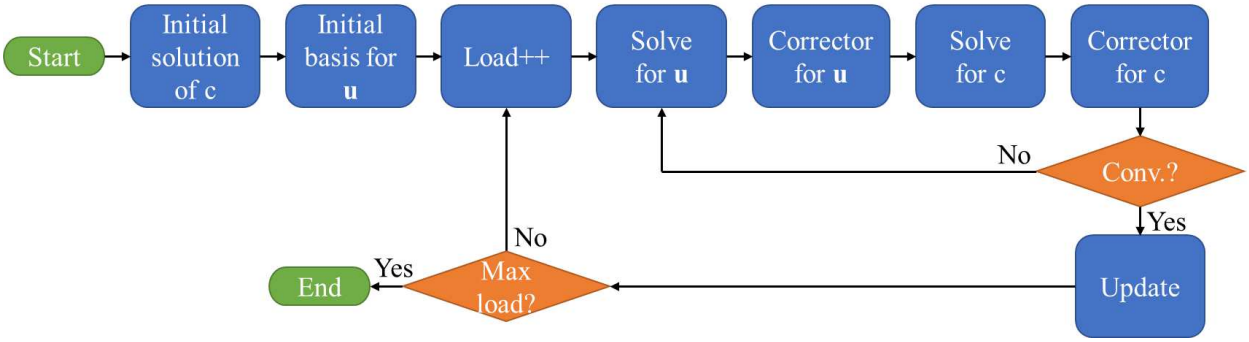


Figure 12: Computational procedure for phase-field fracture with MsFEM and residual-driven correction.

The residual-driven corrector scheme described in Section 3.3 requires an additional set of nested corrector iterations. When we combine multifield staggered iterations and multiscale corrector iterations in a naive way, we therefore arrive at an algorithmic complexity of $\mathcal{O}(n_{cor} \cdot n_{itr})$, where n_{itr} and n_{cor} are the characteristic numbers of multifield staggered iterations and multiscale corrector iterations, respectively.

We now introduce the following algorithmic ideas summarized in Fig. 12 and Algorithm 4 that significantly improve the computational efficiency. The first idea is to synergistically combine multifield staggered iterations and multiscale corrector iterations, carrying out both iterative procedures concurrently. Thus, we do one multiscale corrector iteration per multifield staggered iteration (and vice versa), so that the algorithmic complexity is reduced to $\mathcal{O}(\max(n_{cor}, n_{itr}))$. The second idea is to maintain the initial multiscale basis functions throughout the staggered procedure. Thus, the modification of the displacement and phase-field solution is completely up to the corrector and the associated degrees of freedom. At each coarse-scale node, we only have one corrector degree of freedom irrespective of the dimensionality of the problem. Therefore, the localization caused by crack propagation is solely provided by the corrector basis functions. This completely eliminates the computation of multiscale basis functions from the iterative procedure and the offline step to be repeated only consists of the residual-driven corrector scheme.

Remark 2: For a scalar field, we can directly add corrector solutions to update multiscale basis functions as mentioned in Remark 1, eliminating the corresponding corrector degrees of freedom. In the present case, this could be employed for the phase-field, with the advantage that the number of degrees of freedom of the phase-field macroscale problem would be reduced by half.

Remark 3: We emphasize again that in each multifield staggered iteration, we employ residual-based fine-mesh adaptivity, only finding corrector solutions for those corrector patches, for which the norm of the local residual is above a user-defined threshold. The effectiveness of this approach for propagating localization phenomena such as phase-field fracture will be illustrated in Section 5.3.

4.2. Nonlinear MsFEM, illustrated for plasticity

In the next step, we transfer these ideas to fully nonlinear problems, where the multifield staggered scheme must be replaced by consistent linearization and the Newton-Raphson algorithm. To outline the MsFEM solution procedure for a plasticity problem, we consider the nonlinear

Algorithm 4: Staggered multiscale corrector scheme for phase-field fracture

Result: Displacements $\boldsymbol{u}$, phase-field c

Solve (58) for initial multiscale basis functions ϕ_{cj} of the phase-field;
Solve (59) for bubble part c_b;
if *initial crack* **then**
    Solve (21) with initial $\mathcal{H}$ using Algorithm 3, current ϕ_{cj} and c_b;
end
Solve (56) for initial multiscale basis functions $\boldsymbol{\phi}_j$ of the displacements;
Solve (57) for bubble part $\boldsymbol{u}_b$;
while *load* $\leq$ *maxload* **do**
    % Multifield staggered iterations;
    while c *and* $\boldsymbol{u}$ *not converged* **do**
        Solve macroscale problem (68) to determine multiscale displacement solution;
        % One multiscale corrector for $\boldsymbol{u}$;
        for $i = 1 \rightarrow$ *numberOfCorrectorProblems* **do**
            Compute displacement residual $\boldsymbol{r}_u$ on corrector domain (right-hand side of (53));
            % Automatic fine-mesh adaptivity;
            if $\|\boldsymbol{r}_u\| \geq tol.$ **then**
                Solve (53), use corrector solution to update multiscale displacement solution;
            end
        end
        Solve macroscale problem (68) to determine multiscale phase-field solution;
        % One multiscale corrector for c;
        for $i = 1 \rightarrow$ *numberOfCorrectorProblems* **do**
            Compute phase-field residual $\boldsymbol{r}_c$ on corrector domain;
            % Automatic fine-mesh adaptivity;
            if $\|\boldsymbol{r}_c\| \geq tol.$ **then**
                Find corrector in the sense of (53), update multiscale phase-field solution;
            end
        end
    end
    Update history variable $\mathcal{H}$;
    Load increment *load*++;
end

equation that can be obtained by combining (23) and (26), together with the set of boundary conditions (24) and (25).

4.2.1. General procedure

We first consider again the case of the pristine MsFEM with simple interface constraints. To solve the nonlinear boundary value problem numerically, the general computational procedure (for any finite element method) is illustrated in Algorithm 2. We can observe that after each Newton

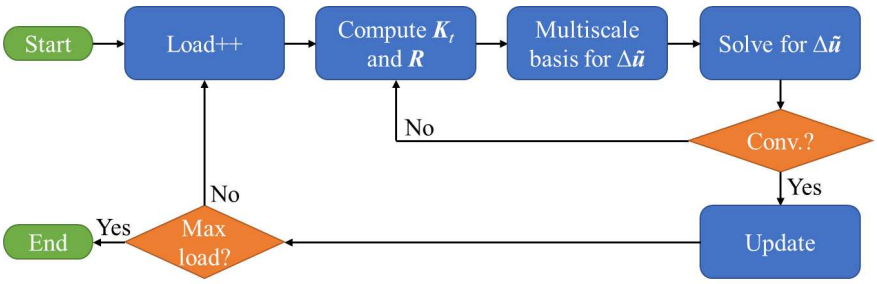


Figure 13: Computational procedure for the solution of a nonlinear plasticity problem with the pristine MsFEM without interelement correction and the MsFEM with oversampling. In each Newton iteration, a new set of multiscale basis functions must be computed in the plastic zone.

iteration, the material tensor $\mathbb{D}(\boldsymbol{u}, \boldsymbol{\alpha})$ changes due to the update of the displacement vector $\boldsymbol{u}$ and the state variables $\boldsymbol{\alpha}$. The potential change of $\mathbb{D}$, however, requires that makes that in each Newton iteration, the multiscale basis functions in the plastic zone need to be recomputed again.

Based on this observation, we adapt the MsFEM algorithm presented in [23, 56] as illustrated in Fig. 13. Using (32), we first compute multiscale basis functions as the solution of the following local variational problem on each coarse-scale element domain,

$$\int_{\Omega_i} \boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B} \, \mathrm{d}\Omega \, \tilde{\boldsymbol{\phi}}_j = \boldsymbol{0} \,, \tag{64}$$

with the simple boundary conditions shown in Fig. 3 and the consistent tangent matrix $\boldsymbol{D}$ that can be obtained as detailed in Appendix A. The vector $\tilde{\boldsymbol{\phi}}_j$ contains the discrete coefficients of the multiscale basis function $\boldsymbol{\phi}_j$ at node j, which can be either $\boldsymbol{\phi}_{jx}$ or $\boldsymbol{\phi}_{jy}$. Having obtained multiscale basis functions, we can use (45) to find the increment $\Delta \boldsymbol{u}_{ms}^H$ in the current iteration. We then proceed to compute the bubble part of the solution from the following local variational problem in each coarse-sale element domain,

$$\int_{\Omega_i} \boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B} \, \mathrm{d}\Omega \, \Delta\tilde{\boldsymbol{u}}_b = \int_{\Omega_i} \boldsymbol{N}^T \boldsymbol{f} \, \mathrm{d}\Omega + \int_{\Gamma_{Ni}} \boldsymbol{N}^T \hat{\boldsymbol{t}} \, \mathrm{d}\partial\Omega - \int_{\Omega_i} \boldsymbol{B}^T \boldsymbol{\sigma}(\boldsymbol{\alpha}, \boldsymbol{u}) \, \mathrm{d}\Omega \,, \tag{65}$$

with homogeneous boundary conditions on element interfaces. The vector $\Delta\tilde{\boldsymbol{u}}_b$ contains the discrete coefficients of the increment of the bubble part $\Delta \boldsymbol{u}_b$. With these two parts of the solution, we can obtain the complete multiscale solution and proceed to updating the state variables.

Remark 4: The computational procedure introduced in [23, 56] is slightly different from the algorithm we illustrate here. In that approach, one first computes and stores multiscale basis functions with some predefined state variables. In each iteration, one then obtains the current state variables and finds the current multiscale basis functions by interpolating between the stored multiscale basis functions. In localization problems, however, this approach is difficult to apply and likely to lead to inaccurate results.

4.2.2. Residual-driven correction for plasticity

For the MsFEM with residual-driven correction, we follow the same arguments as in Section 4.1.3 for phase-field fracture, arriving at the multiscale Newton-Raphson procedure summarized

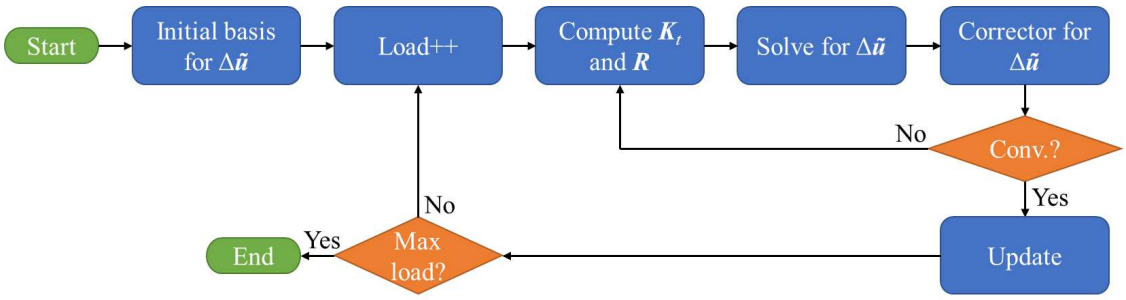


Figure 14: Computational procedure for nonlinear plasticity with the MsFEM and residual-driven correction.

Algorithm 5: Newton-Raphson algorithm for plasticity

Result: Displacements $\boldsymbol{u}$

Solve (64) and (65) with $\boldsymbol{D}^{el}$ for the multiscale basis functions and bubble solution;
while *load* $\leq$ *maxload* **do**
  % Newton iterations;
  while $\boldsymbol{u}$ *not converged* **do**
    Compute macroscale $\boldsymbol{r}$ from (29), check convergence;
    Compute $\boldsymbol{K}_t$ from (33), using the multiscale basis of (68);
    Solve linearized macroscale system for $\Delta\tilde{\boldsymbol{u}}$;
    for $i = 1 \rightarrow$ *numberOfCorrectorProblems* **do**
      Compute displacement residual $\boldsymbol{r}_u$ on corrector domain from (66);
      % Automatic fine-mesh adaptivity;
      if $\|\boldsymbol{r}_u\| \geq tol.$ **then**
        Find $\Delta\boldsymbol{u}_c$ by solving $\boldsymbol{K}_t^{cor}\Delta\boldsymbol{u}_c = \boldsymbol{r}_u$, update the multiscale solution;
      end
    end
  end
  Update internal variables $\boldsymbol{\alpha}$;
  Load increment *load*++;
end

in Fig. 14 and Algorithm 5. We firstly compute the initial multiscale basis functions with the initial elastic material tensor $\boldsymbol{D}^{el}$. The multiscale basis functions remain unchanged during the complete loading history. Thus, the modification of the displacement solution relies on the corrector solutions and their degrees of freedom. In particular, any localization caused by the perfect plasticity model is solely concentrated in the corrector solution. This completely eliminates the computation of multiscale basis functions from the Newton iterations and the offline step contains only the residual-driven corrector scheme. We again synergistically combine Newton and multiscale corrector iterations to circumvent the additional set of nested iterations. We thus apply only one corrector iteration per Newton iteration (or vice versa), reducing the algorithmic complexity from $\mathcal{O}(n_{itr} \cdot n_{cor})$ to $\mathcal{O}(\max(n_{itr}, n_{cor}))$, where n_{itr} and n_{cor} denote now the characteristic numbers of

Newton and corrector iterations, respectively. In this context, we note that the local residual $\boldsymbol{r}_u$ in the corrector domain is computed as

$$\boldsymbol{r}_u = \boldsymbol{f}^{ext} - \boldsymbol{f}^{int} - \boldsymbol{K}_t^{cor} \Delta \tilde{\boldsymbol{u}}_{cur}^{cor} \tag{66}$$

with the tangent stiffness matrix $\boldsymbol{K}_t^{cor}$ and the current corrector solution $\Delta \tilde{\boldsymbol{u}}_{cur}^{cor}$.

Remark 5: Algorithm 5 again relies on the natural mechanism for automatic fine-mesh adaptivity, computing fine-mesh solutions via the correction scheme only in corrector domains where the norm of the local residual is above a user-defined threshold.

4.3. Orthogonality of multiscale, bubble and corrector basis functions

We recall that in the pristine MsFEM, each bubble basis function is orthogonal to all other multiscale basis functions (see (42) in Section 3.1). The corresponding degrees of freedom can be determined independently from decoupled element-wise problems. Therefore, the macroscale Galerkin projection (45) only considers the multiscale basis functions at vertex nodes to find the harmonic part. To obtain the complete multiscale solution, both parts are added together.

Orthogonality, however, cannot always be guaranteed. In this work, one case is the MsFEM with oversampling for the phase-field equation (see Section 4.1.2). It is straightforward to see in Fig. 11c that the bubble part computed on the oversampling domain cannot be orthogonal to the multiscale basis, as it does not satisfy homogeneous Dirichlet boundary conditions at coarse-scale element boundaries. Without orthogonality, however, the harmonic and bubble parts cannot be treated separately without introducing an error. We can resolve this issue by introducing one additional bubble degree of freedom in each coarse-scale element. The basis function of each bubble coefficients, $\boldsymbol{\phi}_{bj}$, is the initial bubble solution in the j^{th} coarse-scale element. To find the complete multiscale solution $\bar{\boldsymbol{u}}$, we employ all multiscale basis functions in the macroscale Galerkin projection as follows:

$$\begin{aligned} a\left(\bar{\boldsymbol{u}}(\boldsymbol{x}), \boldsymbol{v}(\boldsymbol{x})\right) &= \left\langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{v}(\boldsymbol{x}) \right\rangle, \text{for all } \boldsymbol{v}(\boldsymbol{x}) \in \bar{V}^H \\ \bar{V}^H &= \text{span}\left\{\boldsymbol{\phi}_{1x}(\boldsymbol{x}), \boldsymbol{\phi}_{1y}(\boldsymbol{x}), ..., \boldsymbol{\phi}_{mx}(\boldsymbol{x}), \boldsymbol{\phi}_{my}(\boldsymbol{x}), \boldsymbol{\phi}_{b1}(\boldsymbol{x}), ..., \boldsymbol{\phi}_{bn_e}(\boldsymbol{x})\right\} \subset H_0^1(\Omega), \end{aligned} \tag{67}$$

where n_e denotes the number of coarse-scale elements.

We then move on to the MsFEM with residual-driven correction, where we consider independent corrector degrees of freedom. Their multiscale basis functions are identified as the solutions of the local corrector problems. Since they are in general not orthogonal to the multiscale basis, they need to be independently considered in the macroscale Galerkin projection (67) that follows as:

$$\begin{aligned} a\left(\bar{\boldsymbol{u}}(\boldsymbol{x}), \boldsymbol{v}(\boldsymbol{x})\right) &= \left\langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{v}(\boldsymbol{x}) \right\rangle, \text{for all } \boldsymbol{v}(\boldsymbol{x}) \in \bar{V}^H \\ \bar{V}^H &= \text{span}\left\{\boldsymbol{\phi}_{1x}(\boldsymbol{x}), \boldsymbol{\phi}_{1y}(\boldsymbol{x}), ..., \boldsymbol{\phi}_{mx}(\boldsymbol{x}), \boldsymbol{\phi}_{my}(\boldsymbol{x}), \boldsymbol{\phi}_{1,cor}(\boldsymbol{x}), ..., \boldsymbol{\phi}_{m,cor}(\boldsymbol{x})\right\} \subset H_0^1(\Omega), \end{aligned} \tag{68}$$

in which $\boldsymbol{\phi}_{j,cor}$ is the multiscale corrector basis function at node j.

We emphasize that (68) does not include bubble basis functions. In linear problems, the bubble basis functions are orthogonal to all other basis functions [33], and therefore do not have

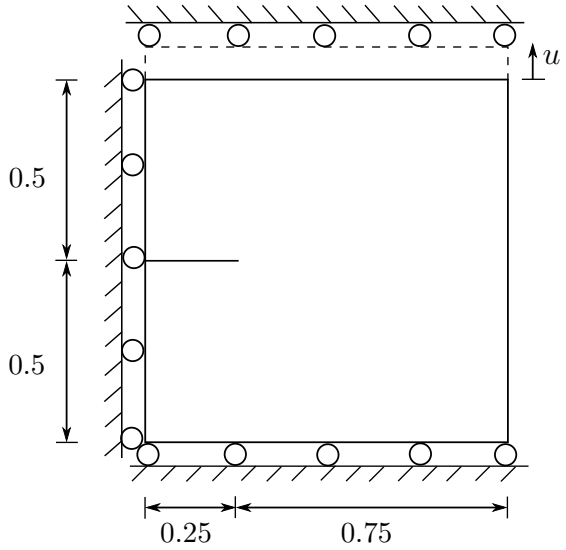


Figure 15: Geometry and boundary conditions for a notched specimen under uniform displacement constraints.

to be included in (68). In nonlinear problems, the multiscale basis functions and the bubble part are constructed with the initial material matrix, but this material matrix changes during nonlinear iterations. Therefore, orthogonality does not generally hold. However, the fine-mesh basis functions that represent the corrector solutions contain the bubble part as a subspace. Since they are recomputed in each Newton iteration, the corresponding corrector basis functions automatically cover the missing bubble part with full accuracy.

Remark 6: It is worthwhile to note that in problems without body forces, we can neglect all bubble degrees of freedom.

Remark 7: For linear problems, the residual is nonzero only on element interfaces (see Section 3.3.1). In our MsFEM framework for nonlinear problems, the residual is nonzero everywhere in local corrector regions, since corrector solutions also cover changes in the bubble part.

5. Numerical examples

In this section, we illustrate the methods discussed above with four numerical examples, covering both phase-field fracture and plasticity benchmarks. We place particular emphasis on demonstrating the computational efficiency of residual-driven correction in the context of nonlinear localization and highlight its significant advantages over the standard FEM, the pristine MsFEM with simple boundary conditions and the MsFEM with oversampling. We finally illustrate the applicability of the MsFEM with residual-driven correction to large-scale problems of practical relevance by a failure analysis of a complex foam structure given by imaging data.

5.1. Fracture of a single edge notched specimen

We start with the classical benchmark for phase-field fracture, a plane-strain square domain with a horizontal notch as illustrated in Fig. 15, on which uniform vertical displacement boundary

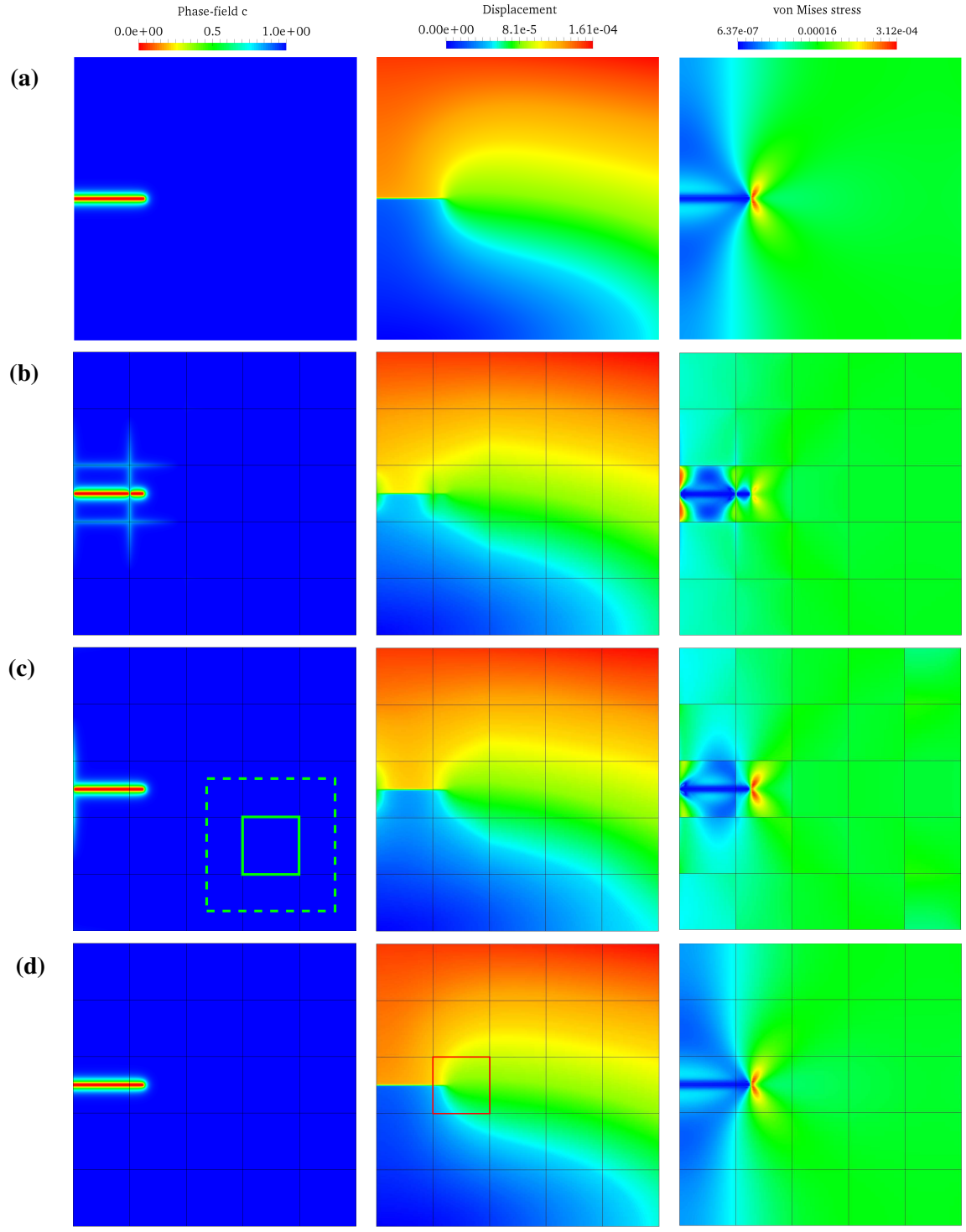


Figure 16: Phase-field, displacement, and von Mises stress solutions after the first load step: (a) standard fine-mesh FEM (reference), (b) pristine MsFEM with simple boundary conditions, (c) MsFEM with oversampling (dashed green line = oversampling domain), and (d) MsFEM with residual-driven correction.

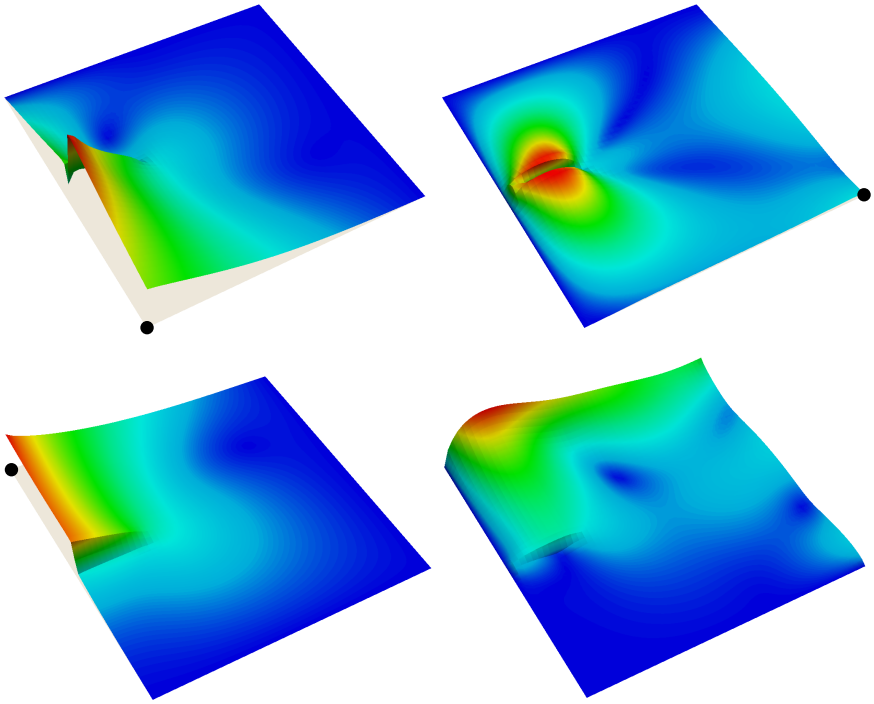


Figure 17: Multiscale corrector basis functions of the coarse-scale element that contains the crack tip, representing the displacement solution. The red-to-blue color scale refers to their magnitude.

conditions are imposed. The model parameters are Young's modulus $E = 1.0$, Poisson's ratio $\nu = 0.3$, the critical energy release rate $\mathcal{G}_c = 5 \times 10^{-4}$, and the length-scale parameter $l_0 = 6 \times 10^{-3}$. To achieve a resolution sufficient for this length scale, we target a fine-scale mesh of 250×250 linear finite elements. Applying the staggered algorithm, we increment the top displacement by $\Delta u = 1.5 \times 10^{-4}$ in each load step.

In the following, we compare the results of four different approaches: (a) the standard FEM using the full fine mesh, (b) the pristine MsFEM with simple boundary conditions at coarse-scale element interfaces, (c) the MsFEM with the oversampling technique, and (d) the MsFEM with residual-driven correction. All MsFEM solution are computed with 5×5 coarse-scale elements, each of which contains 50×50 fine-scale elements, leading to an equivalent fine-scale mesh of 250×250 elements for the complete domain. Oversampling is based on an extension of $d_{os} = 0.12$, which corresponds to 60% of the coarse-scale element size H. Figure 16 plots the phase-field, displacement and von Mises stress solution for all four methods.

Figure 16a plots the standard fine-mesh FEM results, which serve as a reference. Figure 16b plots the solutions obtained with the MsFEM and simple boundary conditions. We observe that the associated local constraints significantly affect the accuracy of the solution, preventing the representation of localized features across coarse-scale element interfaces. In Fig. 16c, we observe that the oversampling technique can improve the solution quality, but still provides inaccurate stress results near element interfaces, in particular in coarse-scale elements near the domain boundary. The solid and dashed green lines in Fig. 16c illustrate a coarse-scale element and its oversampling domain, respectively.

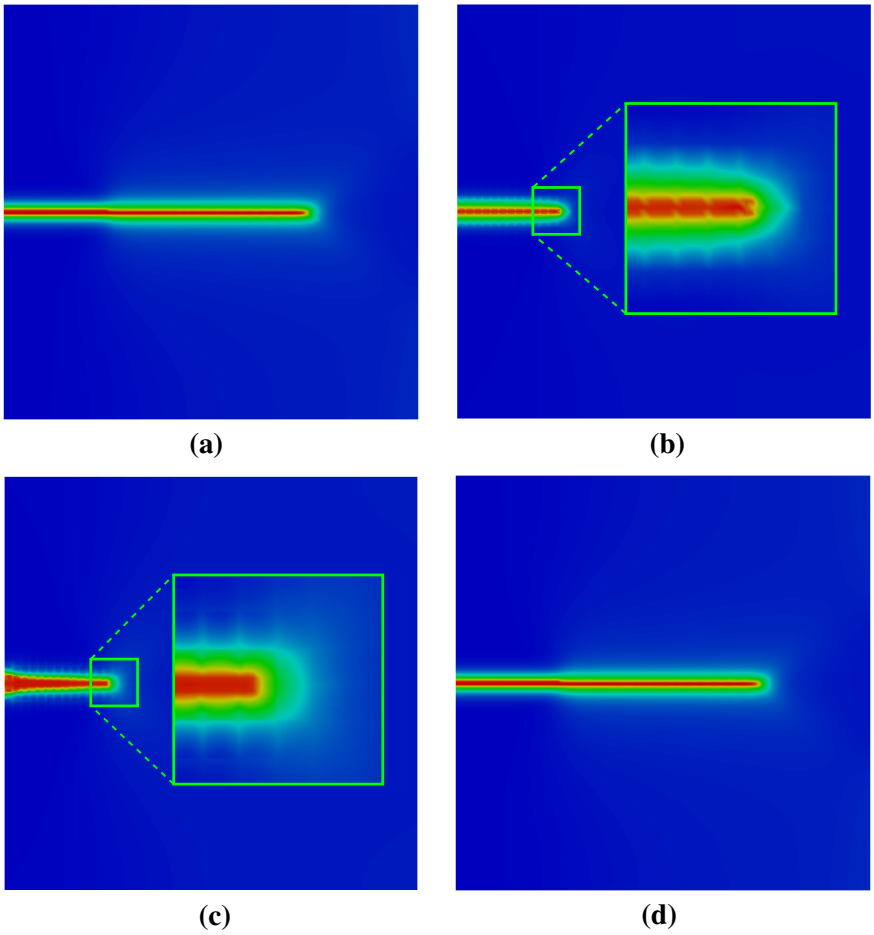
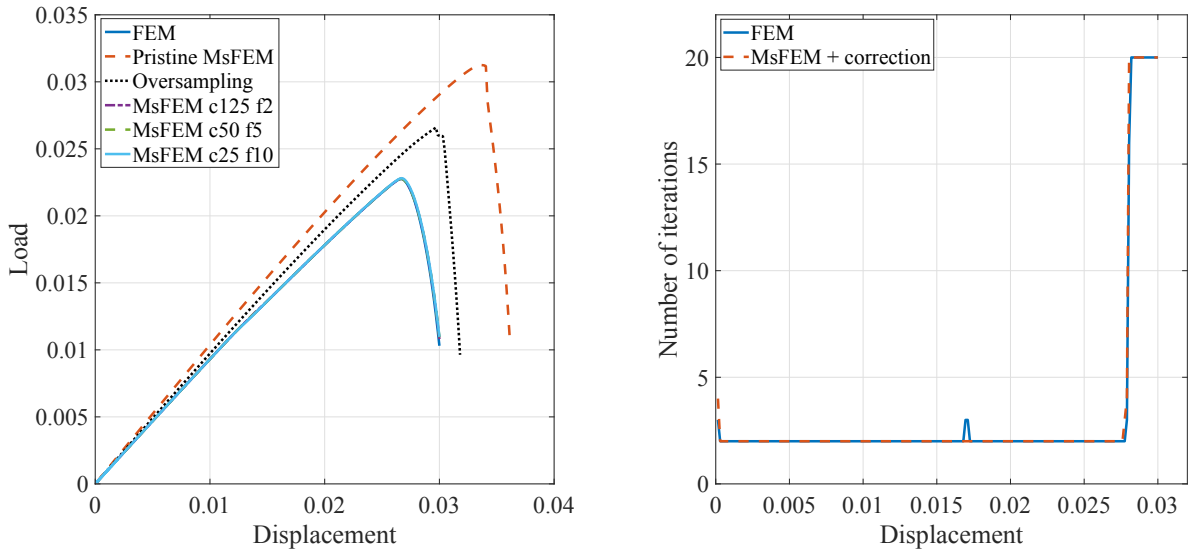


Figure 18: Phase-field solutions at top displacement $u = 0.03$: (a) standard fine-mesh FEM (reference), (b) pristine MsFEM, (c) MsFEM with oversampling, and (d) MsFEM with residual-driven correction. The MsFEM solutions are based on 50×50 coarse-scale elements, each of which holds 5×5 fine-scale elements.

The results plotted in Fig. 16d are indistinguishable from the reference solution, indicating that the residual-driven corrector scheme achieves full fine-mesh accuracy. In particular, the results accurately represent localized features across coarse-scale element interfaces up to the fine-mesh resolution. Since we keep the initial multiscale basis functions the same during the computation, the corrector solution alone is responsible for representing localized solution features near element interfaces. This aspect is illustrated by Fig. 17 that plots the multiscale corrector basis functions of the displacement solution, taken from the coarse-scale element that contains the crack tip.

In the next step, we decrease the ratio between coarse-scale and fine-scale mesh sizes, considering a coarse-scale mesh of 50×50 elements, each of which covers 5×5 fine-scale elements. Figure 18 plots fracture patterns obtained with each of the four approaches for a top displacement of $u = 0.03$, where the standard fine-mesh FEM solution in Fig. 18a serves again as our reference. In Fig. 18b, we observe that the pristine MsFEM with simple boundary conditions fails to properly represent crack propagation across element interfaces. When taking a close look at the damage region, we can see oscillations in the phase-field caused by the local constraints. For this mesh ratio,



(a) Load-displacement behavior.

(b) Number of iterations per load step.

Figure 19: Comparison of accuracy and computational efficiency in terms of load-displacement behavior and number of staggered iterations for the notched specimen example.

we encounter the same situation in Fig. 18c for the MsFEM with oversampling. More importantly, in both methods, the crack represented by the phase-field solution does not freely propagate. This indicates that the local constraints lead to spurious stiffening of the solution behavior. In Fig. 18d, we see that the residual-driven corrector scheme completely removes these limitations. At any given top displacement, it yields the same fracture pattern as the standard fine-mesh FEM, allowing the crack to freely propagates across element interfaces.

The superior accuracy of the MsFEM with residual-driven correction is confirmed by Fig. 19a that plots the load-displacement curves obtained with the standard fine-mesh FEM (reference), the pristine MsFEM with simple boundary conditions, the MsFEM with oversampling, and the MsFEM with residual-driven correction. All MsFEM schemes use 50×50 coarse-scale elements with 5×5 fine-scale elements per coarse-scale element. We observe that the pristine MsFEM is significantly stiffer than the reference solution. The oversampling technique reduces the stiffening error, but its load-displacement curve still shows a significant increase in stiffness with respect to the reference. The curve obtained with the MsFEM with residual-driven correction lies on top of the reference curve in the complete loading range. This excellent solution behavior does not depend on the mesh size ratio between coarse-scale and fine-scale meshes. This is illustrated by varying the mesh size ratio to 125×125 and 25×25 coarse-scale elements with just 2×2 and 10×10 fine-scale elements in each coarse-scale element, respectively. In Fig. 19a, the corresponding load-displacement curves are still exactly reproducing the fine-mesh reference solution.

In the next step, we quantify the computational efficiency of the residual-driven iterative corrector scheme that we implement based on Algorithm 4. Here, we stop the multifield staggered

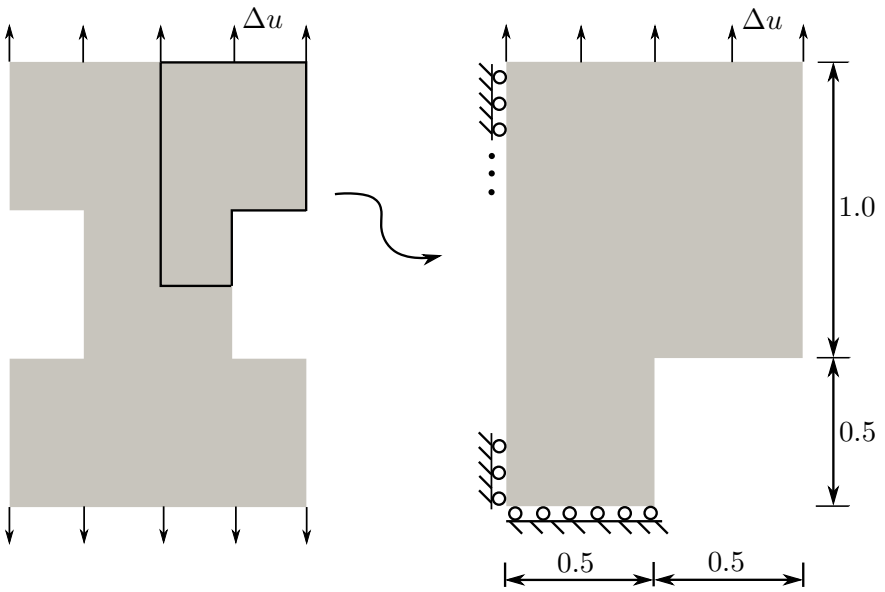


Figure 20: Geometry and boundary conditions of the two-dimensional plane-strain tension test.

iterations in each load step as soon as the relative error measure

$$\text{Relative error} = \frac{\|\boldsymbol{u}_{k+1} - \boldsymbol{u}_k\|}{\|\boldsymbol{u}_k\|} \tag{69}$$

falls below 10^{-3}, with $\boldsymbol{u}_{k+1}$ and $\boldsymbol{u}_k$ being the discrete displacement solution coefficients of the k^{th} and $(k+1)^{th}$ multifield staggered iteration, respectively. We note that equation (69) is computed for coefficients of the fine-scale basis that can be efficiently evaluated from macroscale quantities through coefficient matrices, see for instance [23, 33]. We recall that we only do one residual-driven corrector step in each multifield staggered iteration to limit the increase in the total number of iterations. We observe in Fig. 19b that in almost all load steps, the standard fine-mesh FEM and the MsFEM with residual-based correction require the same number of multifield staggered iterations. This confirms that the synergistic combination of residual-driven correction with multifield staggered iterations does not significantly increase the total number of iterations.

5.2. Tension test with plastic shear band formation

The next example consists of the plane-strain elastoplastic domain shown in Fig. 20 with Young's modulus $E = 100$, Poisson's ratio $\nu = 0.3$, yield stress $\sigma_y = 2$, and the dimensions given in Fig. 20. We apply a uniform displacement Δu in vertical direction at the top and bottom edges of the specimen. Due to symmetry, we only consider one quarter of the complete domain. In each load step, we apply a displacement increment of 1.75×10^{-3}. In the analysis, we employ a uniform fine mesh with 200×300 four-node quadrilateral elements, in which the maximum number of elements in the horizontal and vertical directions are 200 and 300, respectively.

Figure 21a plots the evolution of the plastic strains and shear band formation for different load levels. We computed these results with the MsFEM with residual-driven correction on a

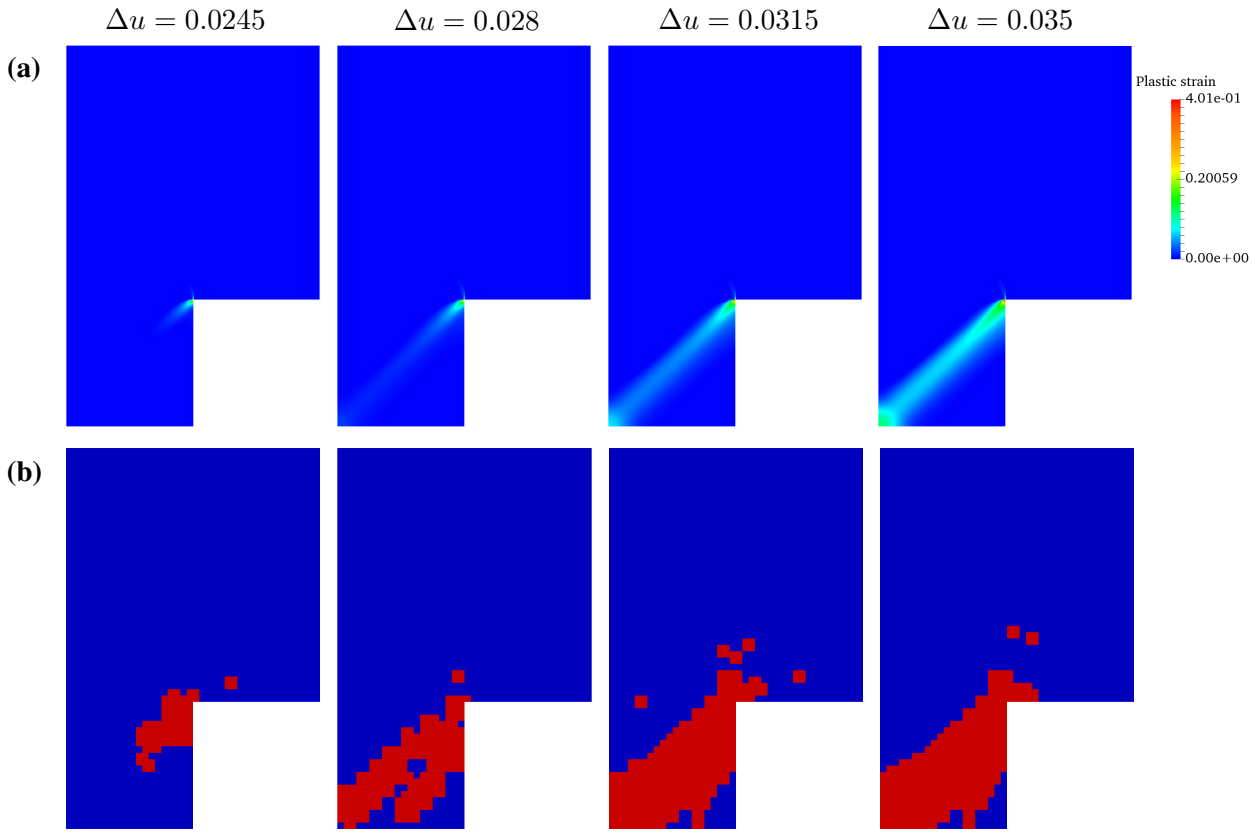
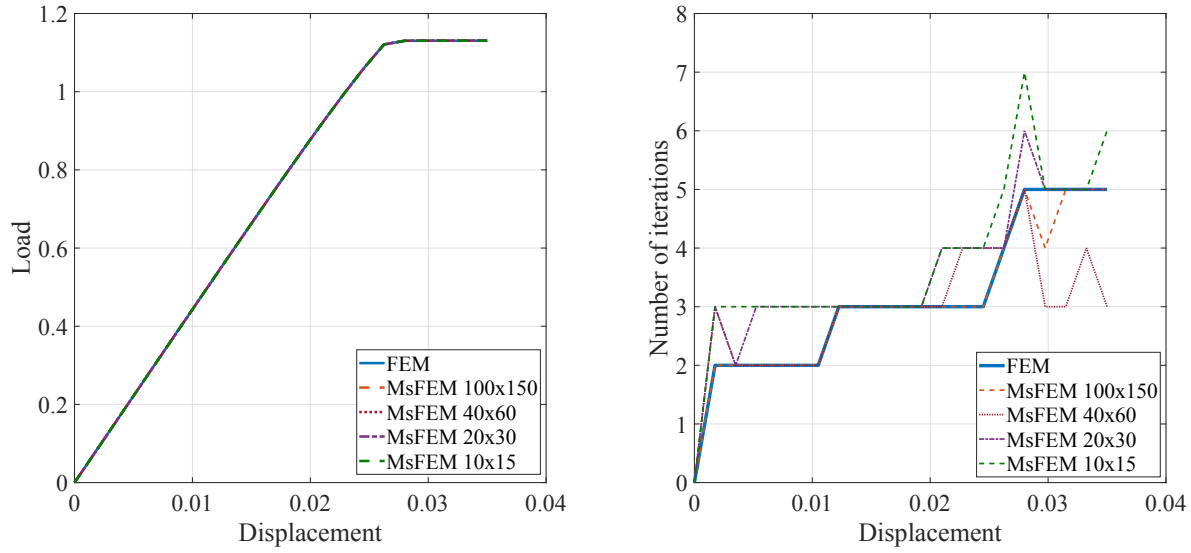


Figure 21: (a) Shear band formation through localized plastic strain, and (b) local corrector regions based on automatic residual-based adaptivity (marked in red).



(a) Load-displacement behavior.

(b) Number of Newton iterations in each load step.

Figure 22: Accuracy and computational efficiency with different coarse-scale meshes for the 2D tension test example. We note that all MsFEM meshes and the FEM mesh have the same fine-mesh resolution.

coarse-scale mesh with 40×60 linear elements, each of which contains 5×5 fine-scale elements. As expected, we obtain exactly the same results as with the fine-mesh FEM. In both computations, we stop the Newton iterations as soon as the relative error of the macroscale problem defined in (69) is less than 10^{-3}. We only solve a local corrector problem when the norm of the corresponding local residual is $\|\boldsymbol{r}_u\| > 10^{-5}$. This automatically drives fine-mesh adaptivity, where corrector basis functions are only evaluated in regions of high residual. The corresponding active local corrector regions are plotted in Fig. 21b for different load levels. The corrector scheme thus offers an attractive adaptive strategy for localization problems. We will further investigate this aspect in the following examples.

Table 1: Number of iterations in each load step: comparison of different coarse-scale meshes and the standard FEM (all at the same fine-mesh resolution). The numbers in the brackets show the difference to the fine-mesh FEM.

Load step	FEM	MsFEM (different coarse-scale mesh size)			
	200×300	100×150	40×60	20×30	10×15
1	2	2	3(+1)	3(+1)	3(+1)
2	2	2	2	2	3(+1)
3	2	2	2	3(+1)	3(+1)
4	2	2	2	3(+1)	3(+1)
5	2	2	2	3(+1)	3(+1)
6	2	2	2	3(+1)	3(+1)
7	3	3	3	3	3
8	3	3	3	3	3
9	3	3	3	3	3
10	3	3	3	3	3
11	3	3	3	3	3
12	3	3	3	4(+1)	4(+1)
13	3	3	4(+1)	4(+1)	4(+1)
14	3	3	4(+1)	4(+1)	4(+1)
15	4	4	4	4	5(+1)
16	5	5	5	6(+1)	7(+2)
17	5	4(−1)	3(−2)	5	5
18	5	5	3(−2)	5	5
19	5	5	4(−1)	5	5
20	5	5	3(−2)	5	6(+1)

We now focus on assessing the number of iterations of the MsFEM with residual-driven correction, when a fully nonlinear problem is considered. In particular, we are interested in whether we need significantly more iterations to obtain the same accuracy level as the fine-mesh FEM solution. For our MsFEM calculations, we employ the following four coarse-scale meshes: 100×150, 40×60, 20×30, and 10×15 elements. Each contains a local fine mesh of 2×2, 5×5, 10×10, and 20×20 elements, respectively. Using the same parameters as before, we plot the load-displacement curves for all meshes in Fig. 22a. We observe that all of the results are practically indistinguishable from the load-displacement curve obtained with the full-resolution FEM.

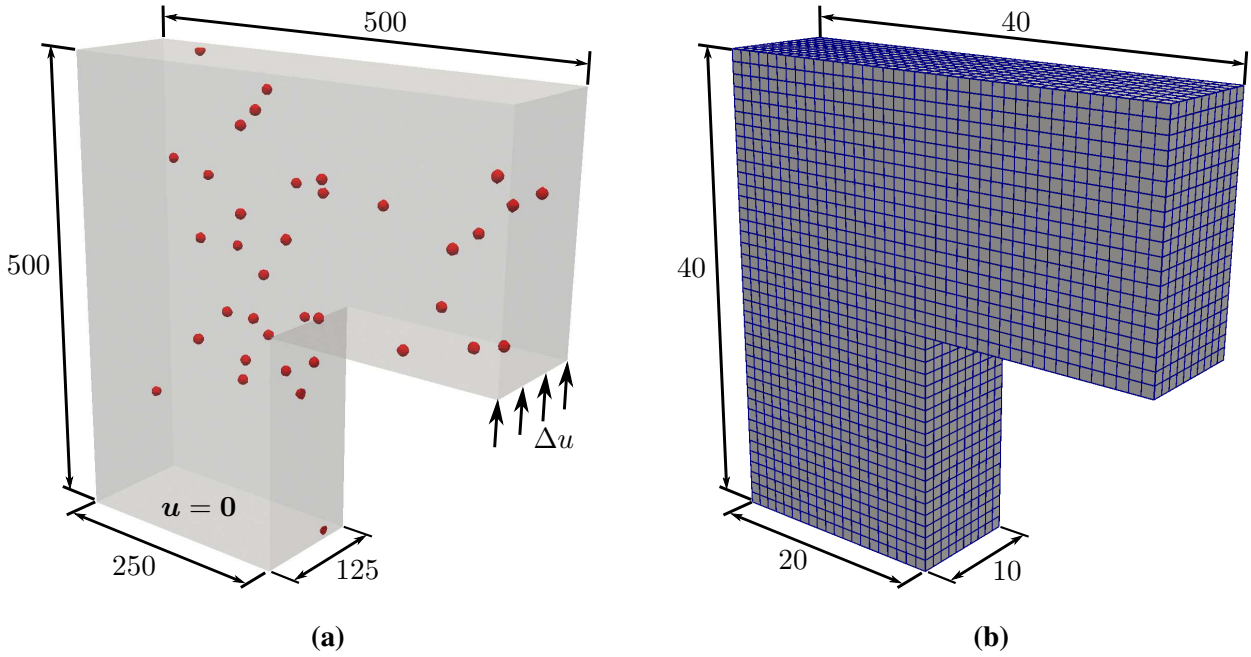


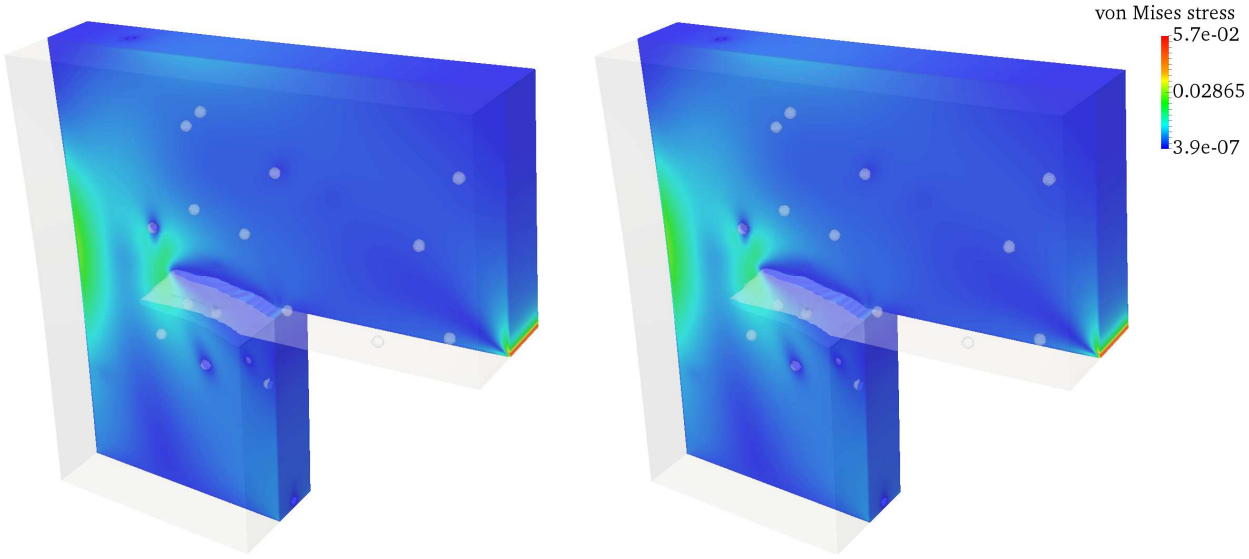
Figure 23: L-shaped specimen with randomly distributed fine-scale defects: (a) Geometry and boundary conditions and (b) coarse-scale MsFEM mesh. The numbers in (a) refer to the dimensions of the specimen, the numbers in (b) refer to the number of coarse-scale elements along each side.

Figure 22b and Table 1 report the number of Newton iterations for each load step that is required for convergence. In the table, the numbers in the brackets show the difference between the MsFEM with residual-driven correction and the standard FEM. We observe that the synergistic combination of Newton and corrector iterations does not lead to a significant increase in the total number of iterations. In most load steps, we converge with the same number of iterations as (and in some cases even with a smaller number of iterations than) the standard full-resolution FEM. The maximum increase is two iterations. We notice that as a general rule, slightly more iterations are required for larger coarse-scale elements that cover more fine-mesh elements.

5.3. Fracture of an L-shaped specimen with random defects

In the next step, we assess the effectiveness of residual-driven adaptivity and the computational cost of the MsFEM with residual-driven correction by considering the three-dimensional L-shaped specimen shown in Fig. 23a. As boundary conditions, we fix all displacements at the base and apply the vertical displacement Δu at the cantilever edge. Assuming the phase-field fracture model, we consider a number of randomly distributed fine-scale defects (marked as red circles) that will influence the propagation of the crack path. Material properties are: Young's modulus $E = 1.0$, Poisson's ratio $\nu = 0.3$, a critical energy release rate $\mathcal{G}_c = 0.001$, and a length-scale parameter $l_0 = 5$. The spherical defects with radius l_0 are generated by randomly assigning initial values to the history field $\mathcal{H}$.

We use a coarse-scale discretization with $12,000$ hexahedral elements as illustrated in Fig. 23b. Each coarse-scale element contains a local fine-scale mesh with $5 \times 5 \times 5$ hexahedral elements.



(a) **(b)**

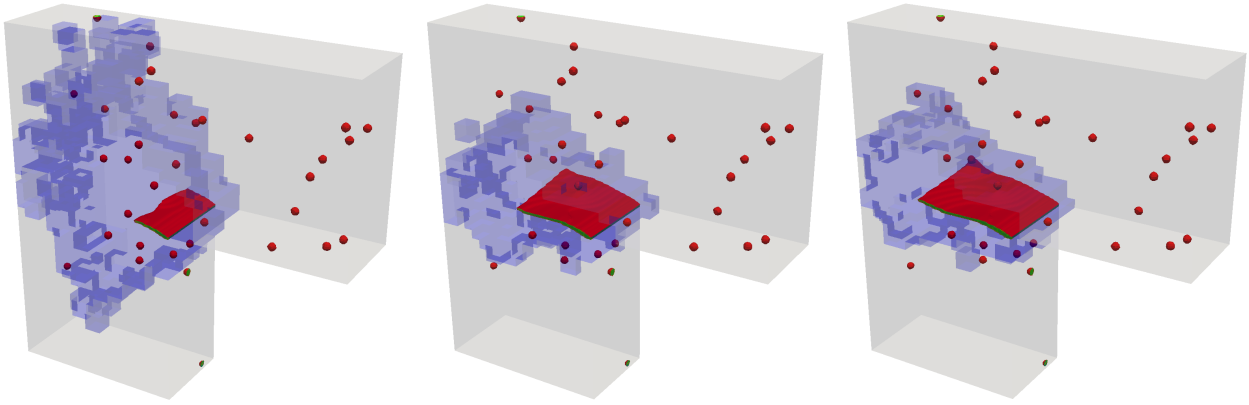
Figure 24: Von Mises stress plotted on the center plane of the deformed body at $\Delta u = 6.25$: (a) the fine-mesh FEM, and (b) the MsFEM with residual-driven correction. We note that the domain around the diffuse crack where the phase-field value is below 0.65 is not visualized.

Therefore, the equivalent full fine-scale mesh has 1.5 million elements with $4,651,353$ degrees of freedom for the displacement field and $1,550,451$ degrees of freedom for the phase-field solution. In each load step, we increase the vertical displacement Δu by an increment of 0.125. We prevent crack propagation in the area where the displacement load is applied. The staggered iterations in each load step are stopped when the relative error defined in (69) falls below 10^{-3}. In addition, we restrict the maximum number of staggered iterations per load step to 20.

For the vector-field displacement solution, we use the local corrector solutions as macroscale basis functions with independent degrees of freedom as given in (68). For the scalar phase-field solution, we update the multiscale basis functions in each iteration by the corrector solution (see Remark 2 and [33]). As a result, the macroscale system for the displacements has $56,364$ degrees of freedom, while the macroscale system has only $14,091$ degrees of freedom. Furthermore, we apply different tolerance criteria to control residual-based adaptivity. In this example, we solve a local corrector problem when $\|\boldsymbol{r}_c\| > 10^{-2}$ for the phase-field and $\|\boldsymbol{r}_u\| > 10^{-3}$ for the displacements, where $\boldsymbol{r}_c$ and $\boldsymbol{r}_u$ are the local residual vectors for each case. When the norms of the residuals are

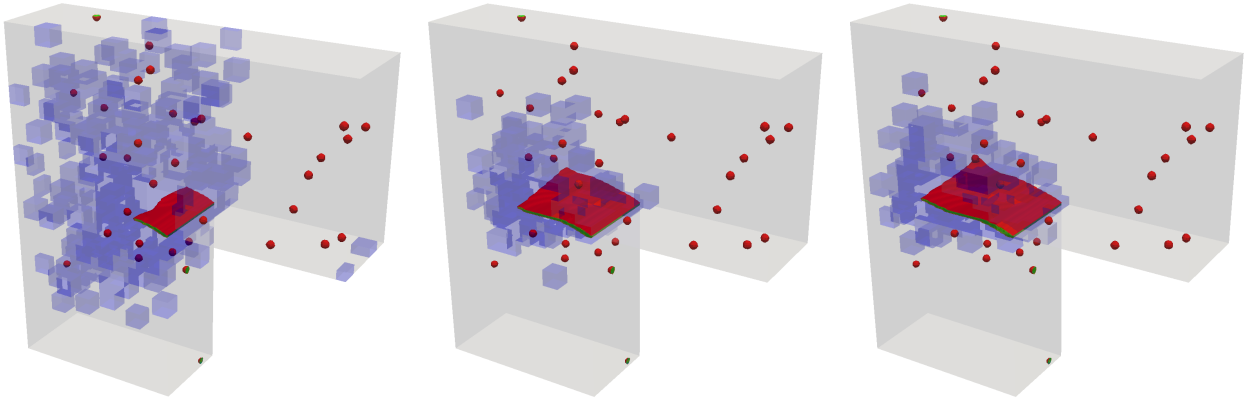
Table 2: Percentage of the domain covered by corrector regions with respect to the total L-shaped domain.

Displacement load	For phase-field system [%]			For displacement system [%]		
	1st iter.	3rd iter.	5th iter.	1st iter.	3rd iter.	5th iter.
$\Delta u = 5.75$	50.2	36.7	34.9	22.4	22.0	21.2
$\Delta u = 6.25$	25.0	18.3	18.8	11.7	11.4	11.1
$\Delta u = 7.5$	21.5	17.0	16.3	9.9	9.2.	9.4



(a) **(b)** **(c)**

Figure 25: Corrector regions (marked in purple) for phase-field at three different loading states: (a) $\Delta u = 5.75$, (b) $\Delta u = 6.25$, (c) $\Delta u = 7.5$.



(a) **(b)** **(c)**

Figure 26: Corrector regions (marked in purple) for the displacements at three different loading states: (a) $\Delta u = 5.75$, (b) $\Delta u = 6.25$, (c) $\Delta u = 7.5$.

below these criteria, we skip the corresponding corrector problem. Figure 24 plots the von Mises stress obtained with the fine-mesh FEM and the MsFEM with residual-driven correction. We see that the stress field and the fracture pattern are exactly the same. In particular, we can observe that the effect of the fine-scale spherical defects that results in a slightly bent crack path are fully captured in the MsFEM solution.

We first focus on the effectiveness of residual-driven adaptivity in the corrector scheme. Table 2 reports the percentage of the total L-shaped domain that requires local correction for different loading states and for the phase-field and displacement solutions. Figures 25 and 26 plot the local corrector regions active in the last iteration of different load steps during crack propagation for both the phase-field and displacement solutions. It is obvious that the residual-based adaptive strategy automatically reduces the number of corrector regions, effectively limiting the computational cost of the corrector scheme.

Beyond this basic aspect, the adaptive corrector scheme provides additional advantages. When the crack is about to propagate, both solution fields still change significantly in areas further away from the crack. When the crack is propagating, significant changes happen predominantly near the localization regions. We observe in Figs. 25 and 26 that the corrector scheme is able to automatically adapt to this phenomenon, detecting a larger corrector region at the onset of crack propagation and restricting the corrector region to the vicinity of the crack later on. We note that this differs from phase-field driven adaptivity, where refinement is a-priori concentrated only in the localization region (see e.g. [57–59]). In many adaptive multifield discretization methods, the same adaptivity pattern must be applied to all fields and tailoring adaptivity to the specific features of each field is not possible. Adaptive residual-driven correction allows to adjust the residuals $\boldsymbol{r}_c$ and $\boldsymbol{r}_u$, so that adaptivity can be controlled for each field independently, although both fields are still represented on the same mesh. This property constitutes a significant strength for the efficient discretization of multifield diffuse interface methods in general (see for instance [60, 61]).

We now focus on the computational cost advantage of the MsFEM with residual-driven correction with respect to the pristine MsFEM and the full-resolution FEM. To this end, we compare the memory usage and computational time of the three methods for the following three different mesh densities. For the MsFEM schemes, we consider $10 \times 10 \times 2$, $20 \times 20 \times 5$, and $40 \times 40 \times 10$ coarse-scale elements, with respect to the element numbering given in Fig. 23b, each of which contains $5 \times 5 \times 5$ fine-scale elements. For standard FEM, this results in full-resolution meshes with $50 \times 50 \times 10$, $100 \times 100 \times 25$, and $200 \times 200 \times 50$ linear hexahedral elements. The total number of degrees of freedom for the fine-mesh FEM corresponds to the numbers given at the beginning of this subsection. All methods were implemented within the same in-house finite element code that was run on the same machine[1] provided by the Minnesota Supercomputing Institute (MSI) with a shared memory parallelization based on OpenMP [62]. For the solution of the local and global linear systems in the MsFEM, we use the direct solver Pardiso provided by Intel's MKL Library[2]. For the solution of the global linear system in the full-resolution FEM, we employ either the same direct solver or an iterative CG solver with ILU preconditioning provided by the Trilinos library[3].

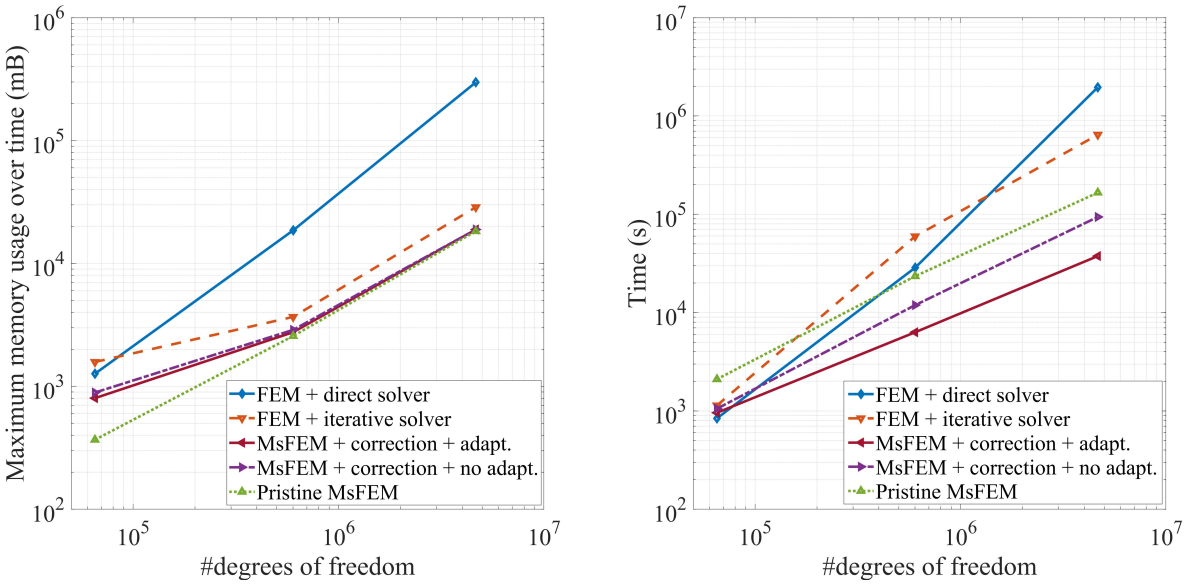
Figure 27a plots the memory usage for different MsFEM variants and the full-resolution FEM. We observe that all MsFEM variants require only half the memory than the standard FEM with the iterative CG solver. Our numerical experiments with the full-resolution FEM confirm that for problems with more than a few million degrees of freedom, using a direct solver leads to prohibitively large memory requirements. We note that for the coarsest discretization, the maximum memory required depends on the size of the local problem. For small discretizations, the pristine MsFEM therefore requires less memory than the MsFEM with residual-driven correction, since it considers the fine mesh in one coarse-scale element only, while the corrector scheme considers the fine mesh in a patch of several coarse-scale elements. For the larger discretization, the memory required mainly depends on storing multiscale basis functions and element matrices and assembling the global matrix, such that all MsFEM variants require the same memory.

The computing time of all methods is illustrated in Fig. 27b. We notice that for the coarsest

[1] 12 processes (Intel Haswell E5-2680v3) with 512 RAM on MSI's cluster Mesabi; https://www.msi.umn.edu/

[2] Intel Math Kernel Library (MKL); https://software.intel.com/en-us/mkl

[3] The Trilinos project, developed at Sandia National Labs; https://trilinos.org/



(a) Memory usage. **(b)** Computing time.

Figure 27: Computational cost of the MsFEM with residual correction as compared to the pristine MsFEM and the full-resolution standard FEM.

discretization, the computing time of all methods remains within the same order while for the larger discretizations the differences become significant. We observe that the MsFEM with the corrector scheme enables significant savings in computing time. For the largest discretization, the MsFEM with residual-driven adaptive correction is approximately 15 times faster than the full-resolution FEM with the iterative solver and 50 times faster the full-resolution FEM with the direct solver. Focusing on the MsFEM with residual correction, we observe that the residual-based adaptivity in the corrector scheme decreases the overall computing time by almost a factor of three.

A core observation is that the MsFEM with residual correction is faster than the pristine MsFEM when an iterative computational solution procedure such as the staggered algorithm or a Newton-Raphson method needs to be applied. In the pristine MsFEM, all multiscale basis functions in each iteration have to be recomputed, where each node has three degrees of freedom and therefore requires the solution of three local problems. In the algorithmic framework for residual-driven correction that we introduce in this work, we do not recompute multiscale basis functions, but use the corrector solutions as basis functions during the iteration. This is significantly less expensive, as each node requires the computation of only one local problem. We can therefore conclude that for nonlinear problems, the MsFEM with residual-driven correction yields the best possible accuracy and is computationally more efficient than the pristine MsFEM.

5.4. Imaging-based failure analysis of a metal foam structure

The last example considers the failure analysis of a metal foam whose geometry is given by microCT scans, demonstrating the applicability of our approach for a more complex engineering

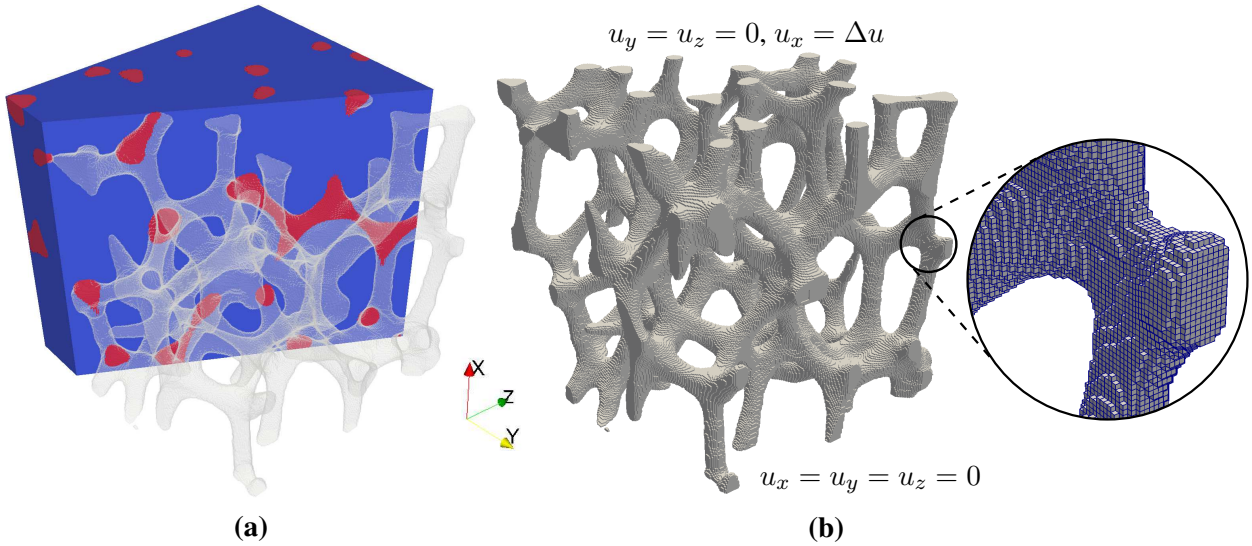


Figure 28: MicroCT-based foam structure: (a) geometry given by imaging data; (b) the full-resolution voxel mesh.

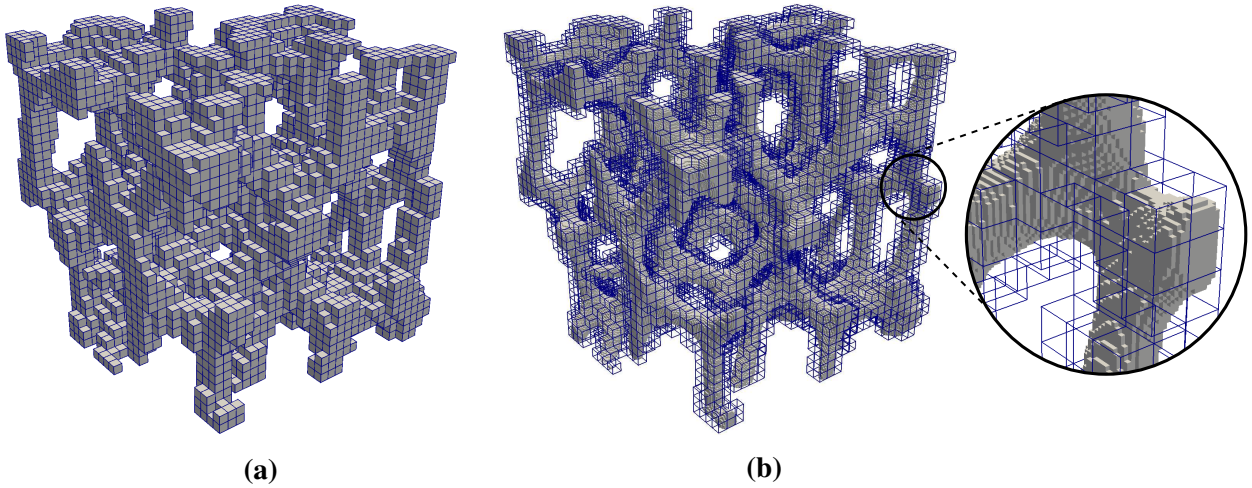


Figure 29: MsFEM discretization: (a) coarse-scale mesh and (b) fine-scale voxel meshes.

problem. The microCT-based foam structure that was extracted from the original imaging data via variational segmentation (see [63] for further details) is illustrated in Fig. 28a. The embedding cube consists of 256 volumetric pixels (voxels) in each direction, where the size of each voxel in each spatial direction is $\Delta_x = \Delta_y = \Delta_z = 75\mu$m. The material properties are: Young's modulus $E = 100$ GPa, Poisson's ratio $\nu = 0.3$, and the yield stress $\sigma_y = 1$ GPa. Considering the color value in each voxel, we can determine whether it belongs to the foam or is void. Deleting all voxels that are outside the physical domain and identifying each voxel as one hexahedral element, we can construct the voxel (fine) mesh shown in Fig. 28b [64–67]. It consists of $2,077,685$ eight-node hexahedral elements with $2,384,192$ nodes (or $7,152,576$ degrees of freedom for the displacement field). We fix all components of the displacement at the bottom of the structure while applying the vertical displacement Δu at the top, leaving the other two components unconstrained. In each load

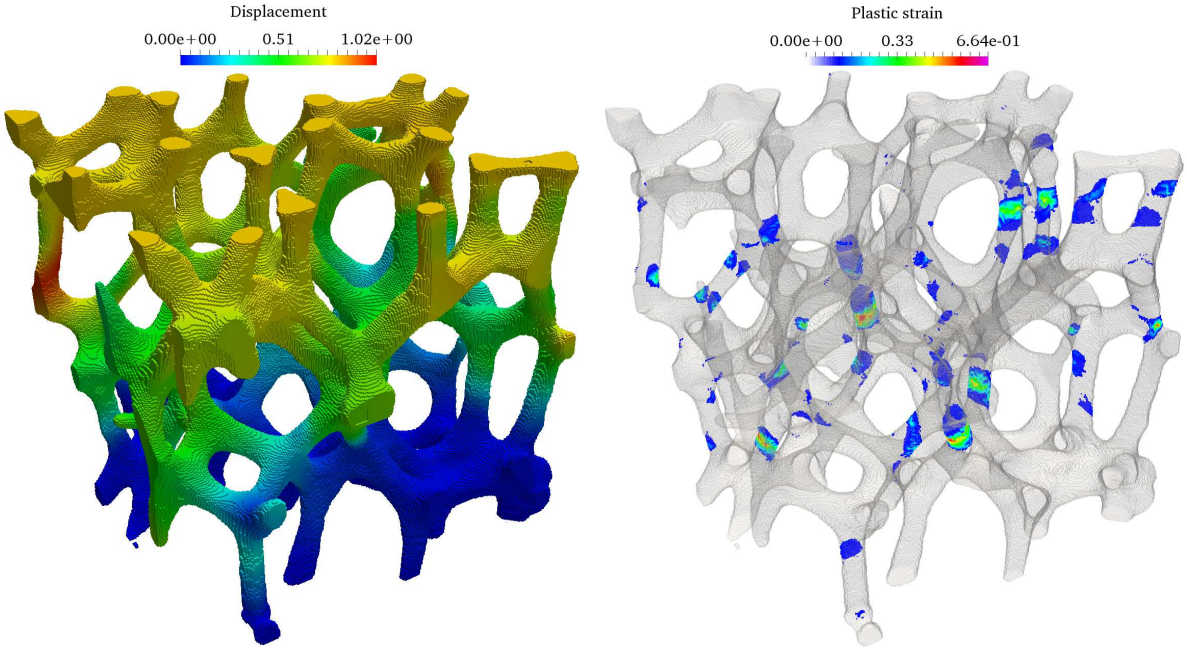


Figure 30: Results obtained with the full-resolution FEM at a top displacement of $\Delta u = 0.8$: (a) magnitude of the displacement field and (b) plastic strains (visualized only for plastic strains larger than 0.1).

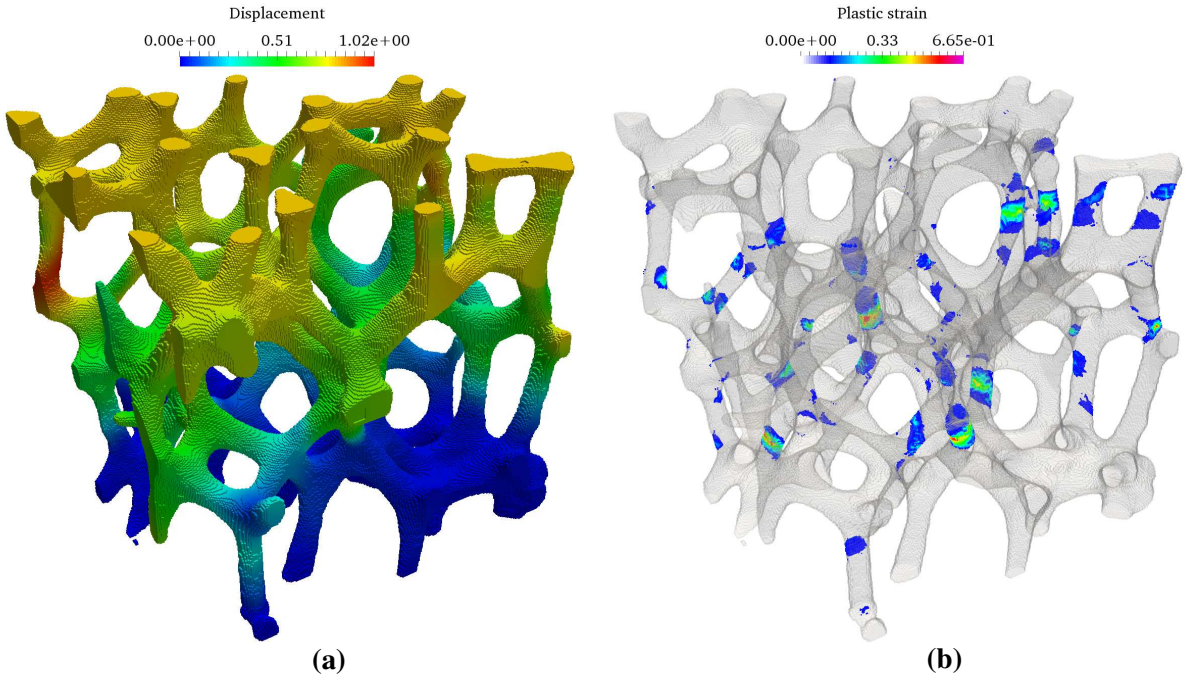


Figure 31: Results obtained with the MsFEM and residual-driven correction at a top displacement of $\Delta u = 0.8$: (a) magnitude of the displacement field and (b) plastic strains (visualized only for plastic strains larger than 0.1).

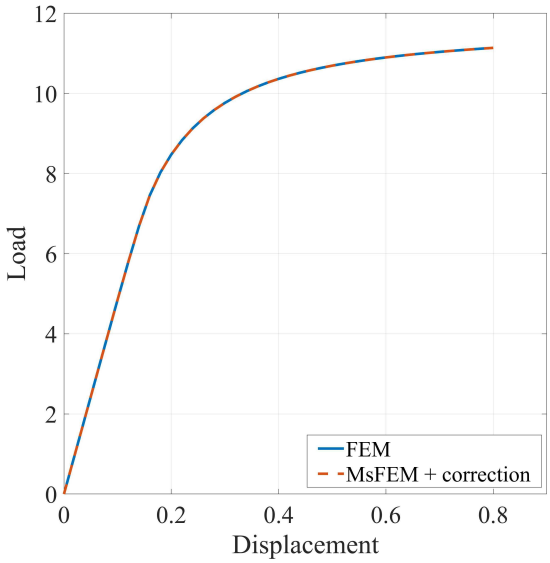


Figure 32: Load-displacement behavior of the foam structure under uniform top displacement.

step, we apply a vertical displacement increment of 20 μm.

For MsFEM analysis with residual-driven correction, we employ a coarse-scale mesh with $8,342$ hexahedral elements and $14,946$ nodes (see Fig. 29a), corresponding to $68,126$ degrees of freedom for the displacement field. We can see that the size of the macroscale problem is significantly smaller than the size of the full-resolution voxel mesh. Each coarse-scale element contains a fine mesh with $8 \times 8 \times 8$ hexahedral elements. Since we eliminate all void voxels, the number of fine-scale elements depends on the location of the coarse-scale elements. Figure 29b shows the non-void voxel domain embedded in the coarse-scale mesh.

We use the same computer system described in Section 5.3. For the full-resolution voxel FEM, we employ the parallelized CG solver with ILU preconditioning to solve the global system. For the MsFEM with residual-driven correction, we employ the direct solver for both local and global systems. Figures 30 and 31 plot the displacement solution and the plastic strains for the displacement load $\Delta u = 0.8$ obtained with voxel FEM and the MsFEM. We observe that both displacement and strain solutions are indistinguishable between the two methods. This is confirmed by Fig. 32 that plots the load-displacement curves for both methods. Figure 33 illustrates the importance of residual-based fine-mesh adaptivity for the efficiency of the corrector scheme. In this example, correction is only initiated if the norm of the local residual in each corrector region satisfies $\|\boldsymbol{r}_u\| > 10^{-5}$. One can see in Fig. 33 that the corrector scheme automatically focuses on localization areas to improve the accuracy in these regions.

Considering the associated computational cost, the voxel FEM requires 308.3 hours with a maximum memory usage of 45.5 GB of memory, while the MsFEM requires only 19.5 hours and 32.4 GB of memory. We conclude that the MsFEM with adaptive residual-driven correction yields the same fine-scale fidelity as the full-resolution voxel FEM, but at a significantly reduced computational cost. As in the previous example, the MsFEM is about fifteen times faster than the

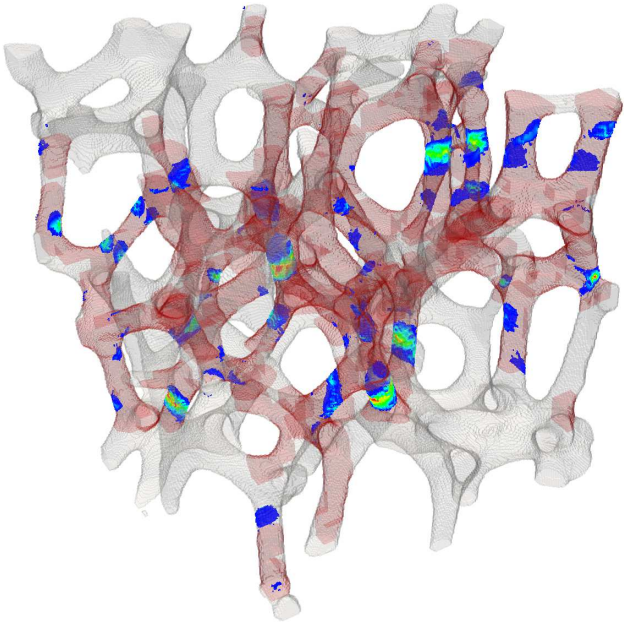


Figure 33: Adaptive corrector regions (marked in red) at the last load step (top displacement $\Delta u = 0.8$).

voxel FEM with the iterative solver.

6. Summary and conclusions

In this article, we discussed a series of algorithmic and variational extensions that enable efficient residual-driven correction for multifield and nonlinear localization problems. The core contributions of this article are threefold: First, we showed that the synergistic combination of n_{itr} staggered or Newton iterations and n_{cor} corrector iterations successfully reduces the algorithmic complexity of staggered multifield or Newton-Raphson procedures from $\mathcal{O}(n_{itr} \cdot n_{cor})$ to $\mathcal{O}(\max(n_{itr}, n_{cor}))$. Second, we introduced the concept of corrector basis functions and associated degrees of freedom in the macroscale Galerkin projection. This idea eliminates the repeated recomputation of multiscale basis functions during staggered or Newton iterations, while preserving the ability to achieve full fine-mesh accuracy. Third, we discussed the opportunistic use of the local residual as an error indicator that is available as part of the corrector scheme. This enables fully automatic corrector adaptivity that further decreases the computational cost of residual-driven correction.

We illustrated through examples from phase-field fracture and plasticity that the MsFEM with residual-driven adaptive correction not only achieves full fine-scale accuracy, but also is computationally more efficient than the pristine MsFEM. The method is general and can be applied to any other continuum localization model such as gradient damage or strain softening laws. Our results therefore open a new pathway for the effective application of the multiscale finite element method to a wide range of multiscale problems that combine interacting mechanisms across multiple scales, missing scale separation, nonlinearity and localization. Our numerical experiments also confirmed that for this class of problems, standard oversampling does not guarantee accurate results in the pristine MsFEM.

A competitive implementation of the multiscale finite element method on modern heterogeneous architectures is based on a hybrid concept [68, 69]. The macroscale system is solved with an efficient preconditioned iterative solver across distributed memory compute nodes, while shared memory parallelism is employed locally on each multicore node to compute multiscale and corrector basis functions for clusters of macroscale elements. We are confident that the computational advantages demonstrated in this article will transfer to this setting, which will be a target of future work.

Acknowledgments: The authors gratefully acknowledge support from the National Science Foundation via the NSF grant CISE-1565997. The first author (L.H. Nguyen) was partially supported by a Doctoral Dissertation Fellowship awarded by the University of Minnesota for the academic year 2017-2018. The second author (D. Schillinger) gratefully acknowledges support from the National Science Foundation through the NSF CAREER Award No. 1651577. The authors also acknowledge the Minnesota Supercomputing Institute (MSI) of the University of Minnesota for providing computing resources that have contributed to the research results reported within this article (https://www.msi.umn.edu/).

Appendix A.

In the context of the Newton-Raphson method, we would like to find the solution at load step $n+1$, where the elastic strain ε_n^{el} from the previous load step n is known. At each quadrature point, we compute the current state variables at the Newton-Raphson iteration $k+1$ with $\Delta\varepsilon$ from the previous iteration k:

1. Evaluate elastic trial state

$$\begin{aligned}
\varepsilon_{n+1,k+1}^{el\ trial} &= \varepsilon_n^{el} + \Delta\varepsilon \\
p_{n+1,k+1}^{trial} &= K\ \varepsilon_{v\ n+1,k+1}^{e\ trial} \\
\boldsymbol{s}_{n+1,k+1}^{trial} &= 2G\ \varepsilon_{d\ n+1,k+1}^{el\ trial} \\
q_{n+1,k+1}^{trial} &= \sqrt{1.5\ \boldsymbol{s}_{n+1,k+1}^{trial} \cdot \boldsymbol{s}_{n+1,k+1}^{trial}}
\end{aligned}$$

2. Check plastic admissibility

$$\begin{aligned}
&\text{If } q_{n+1,k+1}^{trial} - \sigma_y \leqslant 0 \\
&\text{Then set } (\cdot)_{n+1,k+1} = (\cdot)_{n+1,k+1}^{trial} \text{ and } \boldsymbol{D}^{ep} = \boldsymbol{D}^{el}
\end{aligned}$$

3. If the above is not the case, solve the equation

$$\Phi(\Delta\gamma) = q_{n+1,k+1}^{trial} - 3G\ \Delta\gamma - \sigma_y = 0$$

for $\Delta\gamma$ and update the state variables

$$
\begin{aligned}
p_{n+1,k+1} &= p^{trial}_{n+1,k+1} \\
\boldsymbol{s}_{n+1,k+1} &= \left(1 - \frac{\Delta\gamma 3G}{q^{trial}_{n+1,k+1}}\right) \boldsymbol{s}^{trial}_{n+1,k+1} \\
\boldsymbol{\sigma}_{n+1,k+1} &= \boldsymbol{s}_{n+1,k+1} + p_{n+1,k+1}\mathbf{I} \\
\boldsymbol{\varepsilon}^{el}_{n+1,k+1} &= \tfrac{1}{2G}\, \boldsymbol{s}_{n+1,k+1} + \tfrac{1}{3}\, \varepsilon^{etrial}_{v\ n+1,k+1}\mathbf{I} \\
\boldsymbol{D}^{ep} = \boldsymbol{D}^{el} - \frac{\Delta\gamma 6G^2}{q^{trial}_{n+1,k+1}}\mathbf{I}_d &+ 6G^2\left(\frac{\Delta\gamma}{q^{trial}_{n+1,k+1}} - \frac{1}{3G+H}\right)\bar{\boldsymbol{N}}_{k+1}\otimes\bar{\boldsymbol{N}}_{k+1}
\end{aligned}
$$

In the above equations, ε, ε^{el}_v, and ε^{el}_d represent the strain tensor, the volumetric strain tensor, and the deviatoric strain tensor, respectively. K and G are the bulk and shear modulus. p, $\boldsymbol{s}$, q, σ_y are the volumetric stress tensor, the deviatoric stress tensor, the von Mises equivalent stress, and the yield stress, respectively. γ is the plastic multiplier and H is the hardening modulus, which we assume to be zero in the scope of this paper. $\mathbf{I}$ is the second-order identity tensor and $\mathbf{I}_d$ is the deviatoric projection tensor. $\bar{\boldsymbol{N}}$ is the unit plastic flow tensor. $\boldsymbol{D}^{ep}$ is the infinitesimal elastoplastic consistent tangent operator and $\boldsymbol{D}^{el}$ is the infinitesimal elasticity tensor. We refer the interested reader to [39] for more details.

References

[1] F. Feyel and J.L. Chaboche. FE^2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials. *Computer Methods in Applied Mechanics and Engineering*, 183(3):309 – 330, 2000.

[2] V. Kouznetsova, W.A.M. Brekelmans, and F.P.T. Baaijens. An approach to micro-macro modeling of heterogeneous materials. *Computational Mechanics*, 27(1):37–48, 2001.

[3] J.C. Michel and P. Suquet. Nonuniform transformation field analysis. *International Journal of Solids and Structures*, 40(25):6937 – 6955, 2003.

[4] C. Oskay and J. Fish. Eigendeformation-based reduced order homogenization for failure analysis of heterogeneous materials. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1216 – 1243, 2007.

[5] T.I. Zohdi and P. Wriggers. *An introduction to computational micromechanics*. Springer, 2008.

[6] J. Fish. *Practical multiscaling*. John Wiley & Sons, 2013.

[7] V.P. Nguyen, M. Stroeven, and L.J. Sluys. Multiscale continuous and discontinuous modeling of heterogeneous materials: A review on recent developments. *Journal of Multiscale Modelling*, 03(04):229–270, 2011.

[8] T.I. Zohdi. Homogenization methods and multiscale modeling. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–24, 2017.

[9] T. Hettich, A. Hund, and E. Ramm. Modeling of failure in composites by X-FEM and level sets within a multiscale framework. *Computer Methods in Applied Mechanics and Engineering*, 197(5):414–424, 2008.

[10] M.G.D. Geers, V.G. Kouznetsova, and W.A.M. Brekelmans. Multi-scale computational homogenization: Trends and challenges. *Journal of Computational and Applied Mathematics*, 234(7):2175–2182, 2010.

[11] K. Matouš, M.G.D. Geers, V.G. Kouznetsova, and A. Gillman. A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials. *Journal of Computational Physics*, 330:192–220, 2017.

[12] M.G.D. Geers, V.G. Kouznetsova, K. Matouš, and J. Yvonnet. Homogenization methods and multiscale modeling: Nonlinear problems. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–34, 2017.

[13] A. Hund and E. Ramm. Locality constraints within multiscale model for non-linear material behaviour. *International Journal for Numerical Methods in Engineering*, 70(13):1613–1632, 2007.

[14] J. Mergheim. A variational multiscale method to model crack propagation at finite strains. *International Journal for Numerical Methods in Engineering*, 80(3):269–289, 2009.
[15] T.J. Massart, R.H.J. Peerlings, and M.G.D. Geers. An enhanced multi-scale approach for masonry wall computations with localization of damage. *International Journal for Numerical Methods in Engineering*, 69(5):1022–1059, 2007.
[16] E.W.C. Coenen, V.G. Kouznetsova, and M.G.D. Geers. Multi-scale continuous–discontinuous framework for computational-homogenization–localization. *Journal of the Mechanics and Physics of Solids*, 60(8):1486–1507, 2012.
[17] E. Bosco, V.G. Kouznetsova, E.W.C. Coenen, M.G.D. Geers, and A. Salvadori. A multiscale framework for localizing microstructures towards the onset of macroscopic discontinuity. *Computational Mechanics*, 54(2):299–319, 2014.
[18] R. Krause and E. Rank. Multiscale computations with a combination of the *h*-and *p*-versions of the finite element method. *Computer Methods in Applied Mechanics and Engineering*, 192(35):3959–3983, 2003.
[19] S. Löhnert and T. Belytschko. A multiscale projection method for macro/microcrack simulations. *International Journal for Numerical Methods in Engineering*, 71(12):1466–1482, 2007.
[20] D. Schillinger, A. Düster, and E. Rank. The *hp-d* adaptive finite cell method for geometrically nonlinear problems of solid mechanics. *International Journal for Numerical Methods in Engineering*, 89:1171–1202, 2012.
[21] L.H. Nguyen and D. Schillinger. A multiscale predictor/corrector scheme for efficient elastoplastic voxel finite element analysis, with application to CT-based bone strength prediction. *Computer Methods in Applied Mechanics and Engineering*, 330:598 – 628, 2018.
[22] T.Y. Hou and X.-H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of Computational Physics*, 134(1):169–189, 1997.
[23] Y. Efendiev and T.Y. Hou. *Multiscale finite element methods: theory and applications*. Springer, 2009.
[24] U.L. Hetmaniuk and R.B. Lehoucq. A special finite element method based on component mode synthesis. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(3):401–420, 2010.
[25] V.M. Calo, E.T. Chung, Y. Efendiev, and W.T. Leung. Multiscale stabilization for convection-dominated diffusion in heterogeneous media. *Computer Methods in Applied Mechanics and Engineering*, 304:359–377, 2016.
[26] T.Y. Hou and P. Liu. Optimal local multi-scale basis functions for linear elliptic equations with rough coefficients. *Discrete and Continuous Dynamical Systems*, 36(8):4451–4476, 2016.
[27] T.Y. Hou, F.N. Hwang, P. Liu, and C.C. Yao. An iteratively adaptive multi-scale finite element method for elliptic pdes with rough coefficients. *Journal of Computational Physics*, 336:375 – 400, 2017.
[28] Y. Efendiev, J. Galvis, and T.Y. Hou. Generalized multiscale finite element methods (GMsFEM). *Journal of Computational Physics*, 251:116–135, 2013.
[29] L.J. Durlofsky, Y. Efendiev, and V. Ginting. An adaptive local-global multiscale finite volume element method for two-phase flow simulations. *Advances in Water Resources*, 30(3):576 – 588, 2007.
[30] Y. Efendiev, V. Ginting, T. Hou, and R. Ewing. Accurate multiscale finite element methods for two-phase flow simulations. *Journal of Computational Physics*, 220(1):155 – 174, 2006.
[31] H. Hajibeygi, G. Bonfigli, M.A. Hesse, and P. Jenny. Iterative multiscale finite-volume method. *Journal of Computational Physics*, 227(19):8604 – 8621, 2008.
[32] P. Henning and D. Peterseim. Oversampling for the multiscale finite element method. *Multiscale Modeling & Simulation*, 11(4):1149–1175, 2013.
[33] L.H. Nguyen and D. Schillinger. A residual-driven local iterative corrector scheme for the multiscale finite element method. *Journal of Computational Physics*, 377:60–88, 2019.
[34] E.T. Chung, Y. Efendiev, and W.T. Leung. Residual-driven online generalized multiscale finite element methods. *Journal of Computational Physics*, 302:176 – 190, 2015.
[35] T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 2000.
[36] B. Bourdin, G.A. Francfort, and J.-J. Marigo. The variational approach to fracture. *Journal of Elasticity*, 91(1-3):5–148, 2008.
[37] C. Miehe, M. Hofacker, and F. 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):2765–2778, 2010.
[38] M. Ambati, T. Gerasimov, and L. De Lorenzis. A review on phase-field models of brittle fracture and a new fast hybrid formulation. *Computational Mechanics*, 55(2):383–405, 2015.
[39] E.A. de Souza Neto, D. Perić, and D.R.J. Owen. *Computational Methods for Plasticity: Theory and Applications*. Wiley, 2008.
[40] R. De Borst and H.B. Mühlhaus. Gradient-dependent plasticity: Formulation and algorithmic aspects. *International Journal for Numerical Methods in Engineering*, 35(3):521–539, 1992.
[41] R.H.J. Peerlings, R. de Borst, W.A.M. Brekelmans, and J.H.P. de Vree. Gradient enhanced damage for quasi-brittle materials. *International Journal for Numerical Methods in Engineering*, 39(19):3391–3403, 1996.
[42] A. Benallal and J.J. Marigo. Bifurcation and stability issues in gradient theories with softening. *Modelling and Simulation in Materials Science and Engineering*, 15(1):S283, 2007.
[43] E. Lorentz and V. Godard. Gradient damage models: Toward full-scale computations. *Computer Methods in Applied Mechanics and Engineering*, 200(21):1927 – 1944, 2011.
[44] J. Oliver. Modelling strong discontinuities in solid mechanics via strain softening constitutive equations. part 1: Fundamentals. *International Journal for Numerical Methods in Engineering*, 39(21):3575–3600, 1996.
[45] Z.P. Bazant and J. Planas. *Fracture and Size Effect in Concrete and Other Quasibrittle Materials*. CRC Press, 1998.
[46] R. De Borst. Numerical aspects of cohesive-zone models. *Engineering Fracture Mechanics*, 70(14):1743–1757, 2003.
[47] K. Park, G.H. Paulino, and J.R. Roesler. A unified potential-based cohesive model of mixed-mode fracture. *Journal of the Mechanics and Physics of Solids*, 57(6):891 – 908, 2009.
[48] D. Schillinger, M.J. Borden, and H.K. Stolarski. Isogeometric collocation for phase-field fracture models. *Computer Methods in Applied Mechanics and Engineering*, 284:583–610, 2015.
[49] M.J. Borden, C.V. Verhoosel, M.A. Scott, Hughes T.J.R., and C.M. Landis. A phase-field description of dynamic brittle fracture. *Computer Methods in Applied Mechanics and Engineering*, 217–220:77–95, 2012.
[50] G.A. Francfort and J.-J. Marigo. Revisiting brittle fracture as an energy minimization problem. *Journal of the Mechanics and Physics of Solids*, 46(8):1319–1342, 1998.
[51] B. Bourdin, G.A. Francfort, and J-J. Marigo. Numerical experiments in revisited brittle fracture. *Journal of the Mechanics and Physics of Solids*, 48(4):797 – 826, 2000.
[52] C. Miehe, F. Welschinger, and M. Hofacker. Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field FE implementations. *International Journal for Numerical Methods in Engineering*, 83(10):1273–1311, 2010.
[53] P. Wriggers. *Nonlinear finite element methods*. Springer, 2008.
[54] H.W. Zhang, J.K. Wu, J. Lü, and Z.D. Fu. Extended multiscale finite element method for mechanical analysis of heterogeneous materials. *Acta Mechanica Sinica*, 26(6):899–920, 2010.
[55] T.Y. Hou, X.H. Wu, and Z. Cai. Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients. *Mathematics of Computation*, 68:913–943, 1999.
[56] Y. Efendiev, V. Ginting, and T.Y. Hou. Multiscale finite element methods for nonlinear problems and their applications. *Communications in Mathematical Sciences*, 2(4):553–589, 2004.
[57] T. Gerasimov, N. Noii, O. Allix, and L. De Lorenzis. A non-intrusive global/local approach applied to phase-field modeling of brittle fracture. *Advanced Modeling and Simulation in Engineering Sciences*, 5(1):14, 2018.
[58] P. Hennig, M. Ambati, L. De Lorenzis, and M. Kästner. Projection and transfer operators in adaptive isogeometric analysis with hierarchical b-splines. *Computer Methods in Applied Mechanics and Engineering*, 334:313–336, 2018.
[59] M. Dittmann, F. Aldakheel, J. Schulte, P. Wriggers, and C. Hesch. Variational phase–field formulation of non-linear ductile fracture. *Computer Methods in Applied Mechanics and Engineering*, 342:71–94, 2018.
[60] L.H. Nguyen, S.K.F. Stoter, T. Baum, J.S. Kirschke, M. Ruess, Z. Yosibash, and D. Schillinger. Phase-field boundary conditions for the voxel finite cell method: Surface-free stress analysis of CT-based bone structures. *International Journal for Numerical Methods in Biomedical Engineering*, doi:10.1002/cnm.2880, 2017.
[61] L.H. Nguyen, S.K.F. Stoter, M. Ruess, M.A. Uribe Sanchez, and D. Schillinger. The diffuse Nitsche method: Dirichlet constraints on phase-field boundaries. *International Journal for Numerical Methods in Engineering*,

113(4):601–633, 2018.

[62] B. Chapman, G. Jost, and R. Van Der Pas. *Using OpenMP: portable shared memory parallel programming*. The MIT Press, 2008.

[63] T. Gangwar, J. Calder, T. Takahashi, J.E. Bechtold, and D. Schillinger. Robust variational segmentation of 3d bone ct data with thin cartilage interfaces. *Medical Image Analysis*, 47:95–110, 2018.

[64] S.J. Hollister and B.A. Riemer. Digital-image-based finite element analysis for bone microstructure using conjugate gradient and Gaussian filter techniques. *Proc. SPIE 2035, Mathematical Methods in Medical Imaging II*, doi:10.1117/12.146616, 1993.

[65] P. Müller and P. Rüegsegger. Three-dimensional finite element modelling of non-invasively assessed trabecular bone structures. *Medical Engineering & Physics*, 17(2):126 – 133, 1995.

[66] K. Terada, T. Miura, and N. Kikuchi. Digital image–based modeling applied to the homogenization analysis of composite materials. *Computational Mechanics*, 20(4):331–346, 1997.

[67] J.H. Keyak, S.A. Rossi, K.A. Jones, and H.B. Skinner. Prediction of femoral fracture load using automated finite element modeling. *Journal of Biomechanics*, 31(2):125–133, 1997.

[68] A. Heinlein, U. Hetmaniuk, A. Klawonn, and O. Rheinbach. The approximate component mode synthesis special finite element method in two dimensions: Parallel implementation and numerical results. *Journal of Computational and Applied Mathematics*, 289:116–133, 2015.

[69] P. Bastian, C. Engwer, J. Fahlke, M. Geveler, D. Göddeke, O. Iliev, O. Ippisch, R. Milk, J. Mohring, S. Müthing, M. Ohlberger, D. Ribbrock, and S. Turek. Advances concerning multiscale methods and uncertainty quantification in EXA–DUNE. In *Software for Exascale Computing – SPPEXA 2013-2015*, pages 25–43. Springer, 2016.