Computational Mechanics
https://doi.org/10.1007/s00466-023-02299-7

ORIGINAL PAPER

Adaptive spatiotemporal dimension reduction in concurrent multiscale damage analysis

Shiguang Deng[1] · Diran Apelian[1] · Ramin Bostanabad[2]

Received: 31 August 2022 / Accepted: 17 February 2023

Abstract

Concurrent multiscale damage models are often used to quantify the impacts of manufacturing-induced micro-porosity on the damage response of macroscopic metallic components. However, these models are challenged by major numerical issues including mesh dependency, convergence difficulty, and low accuracy in concentration regions. In this paper, we make two contributions to address these difficulties. Firstly, we develop a novel adaptive assembly-free implicit-explicit (AAF-IE) temporal integration scheme for nonlinear constitutive models. This scheme prevents the convergence issues that implicit algorithms face amid softening. Our AAF-IE scheme autonomously adjusts step sizes to capture intricate history-dependent deformations. It also dispenses with re-assembling the stiffness matrices in elasto-plasticity and damage models which, in turn, dramatically reduces memory footprints. Secondly, we propose an adaptive clustering-based domain decomposition strategy to dramatically reduce the spatial degrees of freedom by agglomerating close-by finite element nodes into a limited number of clusters. Our adaptive clustering scheme has static and dynamic stages that are carried out during offline and online analyses, respectively. The adaptive strategy updates the cluster density based on the spatial discontinuity of the plastic strain. As demonstrated by numerical experiments, the proposed adaptive method strikes a good balance between efficiency and accuracy for fracture simulations. In particular, we use our efficient concurrent multiscale model to quantify the significance of spatially varying microscopic porosity on a macrostructure's softening behavior.

Keywords Reduced-order model · Adaptive spatiotemporal dimension reduction · Multiscale simulation · Continuum damage model · k-means clustering

1 Introduction

Cast alloys have heterogeneous material properties which are primarily due to manufacturing-induced defects. Process-induced pores are critical defects that generally appear due to gas or shrinkage [1, 2]. These pores are non-uniformly distributed in a metallic component and typically possess complex spatially varying morphologies, see Fig. 1a. Porosity deteriorates a component's structural integrity and load-carrying capacity where alloys are ultimately fractured by crack propagations through pores [3–5]. It is, therefore, crucial to quantify the effects of local porosity on a macrostructure's damage response. This quantification is typically achieved via multiscale models that leverage the scale separation between macro-components and micro-pores. While these models are quite powerful, in the presence of softening they become prohibitively expensive, memory demanding, and error prone. In this paper, we aim to address these challenges via an adaptive reduced-order multiscale damage model that predicts the strain-softening behaviors of manufactured alloys with complex local porosity defects.

1.1 Background and relevant works

The influence of micro-porosity on alloys' macroscopic responses can be quantified by various multiscale models [6] such as the homogenization-based methods which leverage the scale-coupling theory. This theory involves the solutions of two nested boundary value problems (BVPs) between macroscopic integration points (IPs) and the associated microstructures: at each increment of the solution

✉ Ramin Bostanabad
raminb@uci.edu

[1] ACRC, Materials Science and Engineering, University of California, Irvine, CA, USA

[2] Mechanical and Aerospace Engineering, University of California, Irvine, CA, USA

Published online: 28 March 2023

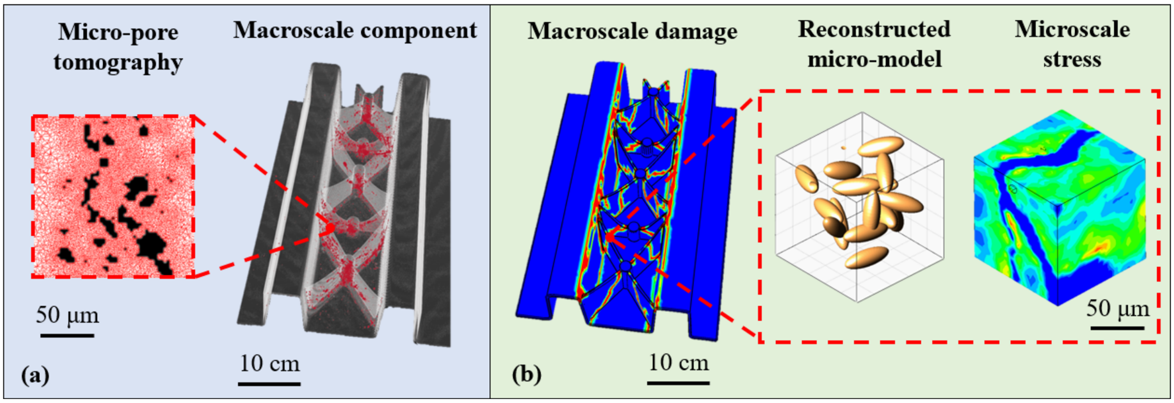

Fig. 1 Multiscale modeling with damage. **a** A metallic component contains non-uniformly distributed microscale pores whose morphologies are identified via 3D X-ray tomography. **b** The distribution of the damage patterns (fracture bands) on the macro-component is strongly affected by its spatially varying porous microstructures

process, macro deformation gradients are used to build and solve an admissible microstructural BVP whose homogenized solutions provide the macro variables for the next increment. The coupled iterations continue till both scales achieve equilibriums. The major advantage of multiscale models is the explicit treatment of microstructure which dispenses with the need to devise phenomenological constitutive laws [7].

The concept of representative volume element (RVE) [8] plays an essential role in multiscale modeling as microscale BVPs are solved in such domains. The size of an RVE should be sufficiently large such that a larger domain with different local morphologies provides the same homogenized behavior. In most practical applications, however, the large RVE size and the embedded morphological details result in high computational costs, especially for the finite element method (FEM) [9]. An efficient alternative to FEM is the method of fast Fourier transformation (FFT) [10] which typically uses a homogeneous auxiliary elastic material as the reference medium to predict the effective responses of heterogeneous microstructures with geometrical or material nonlinearity. FFT relies on voxelated microstructure representation which limits this method's ability to capture fine geometrical details. Additionally, FFT does not apply to microstructures with infinite moduli contrast between material phases since elastic tensors lose coerciveness in such cases and, in turn, result in non-unique full-field solutions [11].

To reduce the simulation costs of FEM and FFT, reduced-order models (ROMs) are developed where the key idea is to systematically reduce the number of unknown variables via offline calculations. Transformed field analysis (TFA) [12] and its later version non-uniform transformed field analysis (NTFA) [13] are two of the earliest ROMs. In these two methods, the number of state variables is reduced by expressing arbitrary strain fields as a function of precomputed eigenstrains.

Clustering-based ROMs are recent approaches for predicting the nonlinear responses of heterogeneous materials. Self-consistent analysis (SCA) [14] presumes that elements with similar elastic responses undergo similar plastic deformations. These clusters are generated by grouping voxels with similar elastic responses and the cluster-to-cluster interaction is quantified by incremental Lippmann–Schwinger (LS) equations. The number of unknown variables in SCA is significantly smaller than that in FEM since it assumes identical states per cluster. Virtual clustering analysis (VCA) [15], a variant of SCA, further improves efficiency by avoiding outer loop iterations and considering boundary terms ignored in LS equations. FEM-cluster-based analysis (FCA) [16] constructs the interaction matrix based on clusters' eigenstrains offline and approximates microstructural effective properties online by following the principle of cluster minimum complementary energy. In deflated clustering analysis (DCA) [17], a clustering-based domain decomposition is universally applied to macro- and micro- domains to accelerate, respectively, high-fidelity calculation of macroscale deformations and effective microscopic responses. Its clusters are created by agglomerating close-by nodes where high-fidelity macroscale solutions are accelerated via deflated methods and effective microstructural responses are expedited in a coarse-graining manner such that nodal states in each cluster take on the same value.

Although ROMs are quite efficient in simulating elastoplastic deformations, their successful application in modeling fracture is rare due to the intricate deformation mechanics. Fracture is generally modeled by two different approaches: (1) a fracture mechanics-based discrete approach that directly simulates displacement discontinuity between fracture interfaces using elastic mediums [18], and (2) a continuum mechanics approach that emulates fractures via strain softening using localized plastic strains. While most ROMs are developed in the realm of continuum mechanics, their applications in the second fracture simulation approach are obstructed by softening-induced numerical instabilities. Specifically, the stiffness matrices of damaged materials lose positive-definiteness amid crack propagation causing ill-conditioned equilibrium equations with imaginary wave speeds [19]. If not properly resolved, solutions of the ill-posed fracture model exhibit spurious mesh dependency that restricts damage elements to single-element wide fracture bands and causes unphysically diminishing energy dissipations upon mesh refinements. Numerical remedies include crack band theory [20], non-local models [21], and phase-field methods [22]. These remedies either introduce mesh-dependent regularizations or modify damage evolution to spread the fracture over finite regions.

Another difficulty in simulating softening via ROMs is the convergence of pure implicit time integration schemes. In these schemes, convergence issues arise due to the non-positive-definite stiffness matrices which cause near-zero or negative gradients in incremental algebraic systems. One way to address this issue is to use explicit time integration schemes which integrate the equations of motion explicitly through time where current kinematic solutions are directly extrapolated from previous steps. However, explicit schemes are very expensive since they require much smaller time steps compared to implicit schemes.

To improve the emulation accuracy of strain-softening, adaptive discretization is often integrated with damage models to enrich spatial interpolations at localized regions while maintaining manageable computational costs. A typical example is adaptive meshing in FEM which is classified as either h-adaptivity or p-adaptivity [23] (in h-adaption element types are unchanged while p-adaption adjusts mesh types by assigning high-order interpolations in critical regions and coarse interpolations elsewhere). Adaptive meshing is also utilized in multiscale damage models [24–26] which often involves identifying the critical regions in the macro-domain where concentrations are expected to develop. The difficulties associated with such an approach include precise anticipation of the critical regions, efficient swap between coarse and fine meshes, and coupling non-matching meshes on the boundaries of subdomains to enforce displacement compatibility. These difficulties can be alleviated by using periodic unit cells with congruent boundaries over different subdomains and enforcing displacement compatibility by Lagrangian multipliers [27]. The adaption strategy is only recently integrated with ROMs [28] where spatial refinements are adaptively performed over material clusters during plastic deformations. These refinements rely on an iterative procedure that, based on a heuristic error metric, rewinds the simulation state to a previous time step where state variables are recalculated on a finer discretization.

1.2 Research gaps and our contributions

We summarize the research gaps in damage modeling for multiscale metallic components as follows:

- Existing methods primarily rely on direct numerical simulations (DNS) to emulate macroscopic damage behaviors with associated microscale heterogeneities. Since these methods are generally memory demanding and computationally expensive, efficient ROMs with high accuracy are needed to quantify the effect of micro pores on the macro component response.
- Most existing ROMs mainly focus on nonlinear elasto-plastic deformations that involve strain-hardening and their successful applications for modeling damage simulations are limited due to softening-induced numerical instabilities. New numerical methods are needed to enable ROMs to resolve softening efficiently and robustly.
- While clustering-based ROMs are highly efficient in approximating microstructures' effective responses, they lose accuracy in concentration regions with highly localized softening. Advancements in ROMs are needed to achieve global and local accuracy.

To fill these gaps, we propose a novel ROM to simulate the damage behavior of metallic alloys with process-induced microscopic porosity. Our method is called adaptive deflated clustering analysis (ADCA) and involves two major contributions. Firstly, it involves a new adaptive assembly-free impl-exp (AAF-IE) temporal integration scheme which resolves the softening-induced convergence issues by preserving the positive-definiteness of the underlying algebraic system amid damage evolution. Our AAF-IE does not require online re-assembly of the global stiffness matrix which dramatically improves computational efficiency. Secondly, we develop a novel adaptive spatial discretization strategy which consists of a static offline clustering stage and an online dynamic stage. While the offline stage assigns high cluster densities in critical regions identified through inexpensive elastic loadings, the online procedure adaptively adjusts cluster densities over crucial areas where softening is expected to initiate and propagate. We integrate the online adaption stage with a novel error metric to automatically detect spatial regions and temporal instances (i.e., there is no need for rewinding the simulation back to an earlier instance) that need finer interpolations. To prevent softening-induced spurious mesh dependencies on multiple scales, we regularize damage evolutions with macroscopic non-local functions and fracture energy stabilized micro-damage models, both of which are systematically integrated with our AAF-IE. It is noted that while we integrate our adaptive techniques with DCA [17], they can be readily integrated with any other clustering-based ROM for efficient multiscale modeling with accurate predictions of localized phenomena.

1.3 Outline of the paper

The remainder of the paper is organized as follows. In Sect. 2, we briefly review some background techniques. Our adaptive multiscale damage method is introduced in Sect. 3 and we demonstrate its efficiency and accuracy via numerical experiments in Sect. 4. The paper is concluded with some final remarks in Sect. 5.

2 Background techniques

The proposed reduced-order multiscale damage model relies on a few background techniques including homogenization theory, damage models, and time integration schemes which are briefly reviewed below.

2.1 Multiscale modeling via first-order homogenization

We use multiscale models to investigate the influence of microscale heterogeneity on the fracture behavior of macroscale components. Our multiscale model relies on the first-order computational homogenization (FOCH) method which assumes distinguishable macro and micro features. Specifically, the sizes of local material heterogeneity (l_{μ}) is assumed to be much smaller than the RVE size (l_{m}) which itself is presumed to be smaller than the macro-characteristic length (l_{M}), that is:

$$l_{\mu} \ll l_{\mathrm{m}} \ll l_{\mathrm{M}} \tag{1}$$

where the subscripts 'M' and 'm' represent macroscale and microscale, respectively.

Solutions of the macro- and micro-structures are coupled via the Hill-Mandel energy condition [29] which implies that the virtual internal work density at a macro IP equals the volume average of the virtual work of its associated RVE:

$$\mathbf{S}_{\mathrm{M}} : \delta\mathbf{E}_{\mathrm{M}} = \frac{1}{|\Omega|}\int_{\Omega} \mathbf{S}_{\mathrm{m}} : \delta\mathbf{E}_{\mathrm{m}} d\Omega \tag{2}$$

where $\mathbf{S}_{\mathrm{M}}$, $\mathbf{E}_{\mathrm{M}}$, $\mathbf{S}_{\mathrm{m}}$ and $\mathbf{E}_{\mathrm{m}}$ are the macroscopic and microscopic stress and strain tensors, respectively. Ω and $|\Omega|$ represent the domain of microstructure and its volume. The symbol ':' represents the double dot product that contracts a pair of repeated indices.

The stress and strain tensors at either scale are computed through equilibrium equations. The equilibrium equations for a macrostructure in the infinitesimal deformation framework are [30]:

$$\mathbf{S}_{\mathrm{M}}(\mathbf{X}) \cdot \nabla_0 + \mathbf{b}_{\mathrm{M}} = \mathbf{0} \quad \forall \mathbf{X} \in \Omega_{0\mathrm{M}} \tag{3}$$

$$\mathbf{u}_{\mathrm{M}}(\mathbf{X}) = \bar{\mathbf{u}}_{\mathrm{M}} \quad \forall \mathbf{X} \in \Gamma^{\mathrm{D}}_{0\mathrm{M}} \tag{4}$$

$$\mathbf{S}_{\mathrm{M}}(\mathbf{X}) \cdot \mathbf{n}_{\mathrm{M}} = \bar{\mathbf{t}}_{\mathrm{M}} \quad \forall \mathbf{X} \in \Gamma^{\mathrm{N}}_{0\mathrm{M}} \tag{5}$$

where $\mathbf{u}_{\mathrm{M}}$ is the unknown macro-displacement, $\bar{\mathbf{u}}_{\mathrm{M}}$ is the prescribed displacement on the Dirichlet boundary $\Gamma^{\mathrm{D}}_{0\mathrm{M}}$, $\bar{\mathbf{t}}_{\mathrm{M}}$ represents the surface traction over the Neumann boundary $\Gamma^{\mathrm{N}}_{0\mathrm{M}}$, and $\mathbf{n}_{\mathrm{M}}$ denotes the outward unit vector to the boundary of the undeformed macro-domain $\Omega_{0\mathrm{M}}$. ∇_0 is the gradient operator with respect to the undeformed configuration.

Similarly, the equilibrium equation in the microscale can be written as the following BVP:

$$\mathbf{S}_{\mathrm{m}}(\mathbf{x}) \cdot \nabla_0 = \mathbf{0} \quad \forall \mathbf{x} \in \Omega_{0\mathrm{m}} \tag{6}$$

$$\mathbf{S}_{\mathrm{m}}(\mathbf{x}) \cdot \mathbf{n}_{\mathrm{m}} = \bar{\mathbf{t}}_{\mathrm{m}} \quad \forall \mathbf{x} \in \Gamma^{\mathrm{N}}_{0\mathrm{m}} \tag{7}$$

where $\bar{\mathbf{t}}_{\mathrm{m}}$ is the prescribed surface traction per unit area on the boundary $\Gamma^{\mathrm{N}}_{0\mathrm{m}}$ of reference microscale domain with the outward unit normal vector $\mathbf{n}_{\mathrm{m}}$. According to the Hill-Mandel condition, macroscale and microscale equations are coupled by equating the macroscopic stress to the homogenized micro-stress of RVE and storing history-dependent state variables for continued analyses in the concurrent multiscale model.

We assume the plastic behavior of the studied alloy system follows a rate-independent isotropic elasto-plastic constitutive law in the microscale as:

$$\begin{cases} \mathbf{S}_{\mathrm{m}} = \mathbb{C}^{\mathrm{el}} : \mathbf{E}^{\mathrm{el}}_{\mathrm{m}} \\ \mathbf{E}^{\mathrm{el}}_{\mathrm{m}} = \mathbf{E}_{\mathrm{m}} - \mathbf{E}^{\mathrm{pl}}_{\mathrm{m}} \\ \mathbf{E}_{\mathrm{m}} = \int \mathrm{d}\mathbf{E}_{\mathrm{m}} \\ \mathbf{E}^{\mathrm{pl}}_{\mathrm{m}} = \int \mathrm{d}\mathbf{E}^{\mathrm{pl}}_{\mathrm{m}} \end{cases} \tag{8}$$

where $\mathbb{C}^{\mathrm{el}}$ represents the fourth-order elasticity tensor. The total microscopic strain ($\mathbf{E}_{\mathrm{m}}$) is additively decomposed into the microscopic elastic strain ($\mathbf{E}^{\mathrm{el}}_{\mathrm{m}}$) and plastic strain ($\mathbf{E}^{\mathrm{pl}}_{\mathrm{m}}$) where the values of the total and plastic strains are integrated by their increments ($\mathrm{d}\mathbf{E}_{\mathrm{m}}$ and $\mathrm{d}\mathbf{E}^{\mathrm{pl}}_{\mathrm{m}}$) at each time step. The plastic state is defined by the yield condition f as:

$$f = f(\mathbf{S}_{\mathrm{m}}, q) = 0 \tag{9}$$

where q is a history-dependent state variable. During plastic flow, the plastic strain increment $\Delta\mathbf{E}^{\mathrm{pl}}_{\mathrm{m}}$ is determined by the plastic flow rule as:

$$\Delta\mathbf{E}^{\mathrm{pl}}_{\mathrm{m}} = \Delta\lambda \frac{\partial Q}{\partial \mathbf{S}_{\mathrm{m}}} \tag{10}$$

where λ is the plastic multiplier and Q refers to the plastic potential. In this work, we use an associated flow rule where $Q = f$.

2.2 Damage in ductile metals

Damage occurs when excessively large loads are applied to materials whose loading-carrying capacity is reduced as a result of progressive degradation of the materials' moduli. To model this degradation in a multiscale setting, we add damage to Eqs. (3)–(7). In particular, we adopt an isotropic continuum damage model which has two major components: an initiation criterion that predicts the onset of softening and an evolution law that traces the crack's progression until rupture.

We select the ductile criterion at either scale as the damage initiation metric which assumes the effective plastic strain ($\overline{E}^{pl}_{d}$) at the onset of fracture is a function of stress states and strain rates. For simplicity, we assume $\overline{E}^{pl}_{d}$ is a user-defined constant and damage is initiated when the following condition is met:

$$\omega_d = \frac{\overline{\mathrm{E}}^{\mathrm{pl}}}{\overline{\mathrm{E}}^{\mathrm{pl}}_{\mathrm{d}}} \geq 1 \tag{11}$$

where ω_d is the damage initiation variable that increases monotonically in elasto-plastic deformations and $\overline{E}^{pl}$ is the equivalent plastic strain. We note that ductile damage initiation can be used in conjunction with other damage initiation criteria. When multiple criteria are specified simultaneously, they are treated independently, and damage evolution starts if any criterion is met.

Upon initiation, damage results in softening of the yield stress and degradation of the stiffness. Correspondingly, the constitutive equation for a damaged elasto-plastic metal is written as:

$$\begin{cases} \mathbf{S} = (1 - D)\mathbf{S}^0 \\ \mathbf{S}^0 = \mathbb{C}^{\mathrm{el}} : \mathbf{E}^{\mathrm{el}} = \mathbb{C}^{\mathrm{el}} : (\mathbf{E} - \mathbf{E}^{\mathrm{pl}}) \end{cases} \tag{12}$$

where $D \in [0, 1]$ is the monotonically increasing damage evolution parameter and $\mathbf{S}^0$ is the stress tensor on a reference material that undergoes the same plastic deformation but in the absence of damage. Since the continuum damage is assumed as isotropic, D is a scalar in Eq. (12). If multiple evolution laws are specified, D would capture the effects of all active damage mechanisms. For an anisotropic damage model, however, D becomes a tensor.

The evolution of D can be specified as a function of equivalent plastic strain and state variables, e.g., as an exponential function [31] of $\overline{E}^{pl}$ and a user-defined non-negative evolution rate parameter (α) as:

$$D(\overline{\mathrm{E}}^{\mathrm{pl}}, \alpha) = 1 - \frac{\overline{\mathrm{E}}^{\mathrm{pl}}_{\mathrm{d}}}{\overline{\mathrm{E}}^{\mathrm{pl}}} \exp(-\alpha(\overline{\mathrm{E}}^{\mathrm{pl}} - \overline{\mathrm{E}}^{\mathrm{pl}}_{\mathrm{d}})) \tag{13}$$

This definition ensures that when $\overline{E}^{pl}$ is sufficiently large compared to $\overline{E}^{pl}_{d}$, the damage variable increases to one so that the stiffness is fully degraded, and the material is entirely ruptured.

Once the maximum degradation is reached at a material point, a choice of removal of the ruptured elements from the computational mesh is enabled. Element deletion prevents further damage accumulation in localized regions and enhances computational efficiency by improving the condition number of damaged stiffness of the underlying algebraic systems.

2.3 Mesh dependency control

With fractures are formulated as strain softening by the isotropic continuum damage model in Eq. (12), softening-induced non-positive definite stiffness causes the BVPs in Eqs. (3)–(7) to be ill-conditioned with unstable convergence and negative wave speeds. In such a scenario, equilibrium solutions lose their objectivity with respect to mesh sizes and exhibit spurious mesh sensitivity, especially when using standard finite elements (FEs) of the first-order continuity. If not properly addressed, the progressive damage model in Eq. (12) would result in unphysical material failures that are localized in a certain region with shrinking fracture energies upon mesh refinements [21]. As discussed below, we adopt two different approaches to control the mesh dependency on the macro and micro domains.

2.3.1 Macroscale mesh dependency control by nonlocal functions

When the damage model of Eq. (12) is directly applied to the damage evolutions in Eq. (13), macrostructures exhibit unphysical material softening in concentration regions where the equilibrium solutions strongly depend on the mesh. This pathological mesh dependency restricts the fractured elements into an unphysical narrow damage band with a single layer of elements. This spurious mesh dependency can be alleviated by introducing the 'fracture band width' as a macroscale material characteristic length in a nonlocal damage model.

We adopt an integral-type nonlocal damage model in Eq. (14) where the nonlocal parameter of a macroscopic point involves the weighted averages of the damage parameters over a finite spatial neighborhood of the point under consideration:

$$\widehat{D}(\mathbf{X}, \mathbf{X}') = \int_{\mathrm{B}} \omega\big(\|\mathbf{X} - \mathbf{X}'\|\big) D(\mathbf{X}') d(\mathbf{X}') \tag{14}$$

where $\widehat{D}(\mathbf{X}, \mathbf{X}')$ is the nonlocal damage parameter at a macroscopic point $\mathbf{X}$ surrounded by neighbor points $\mathbf{X}'$ in a support neighborhood denoted by B, $D(\mathbf{X}')$ is the local damage parameter at $\mathbf{X}'$, and $\omega\big(\|\mathbf{X} - X'\|\big)$ is the nonlocal weighting function. There is no unique way to define $\omega\big(\|\mathbf{X} - X'\|\big)$ and in this study, we adopt a polynomial bell-shaped function [31] as:

$$\omega\big(\|\mathbf{X} - \mathbf{X}'\|\big) = \frac{\omega_\infty\big(\|\mathbf{X} - \mathbf{X}'\|\big)}{\int_{\mathrm{B}} \omega_\infty(\|\mathbf{X} - \mathbf{X}'\|) d(\mathbf{X}')} \tag{15}$$

$$\omega_\infty\big(\|\mathbf{X} - \mathbf{X}'\|\big) = \left\langle 1 - \frac{4\big(\|\mathbf{X} - \mathbf{X}'\|\big)^2}{l_0^2} \right\rangle^2 \tag{16}$$

where the Macauley brackets $\langle \ldots \rangle$ represent a non-negative value defined as $\langle x \rangle = \max(0, x)$, and l_0 is the macroscale material characteristic parameter denoting an interaction radius, that is, the weighting effects diminish when $\|\mathbf{X} - \mathbf{X}'\| > l_0/2$. Thus, in 3D models, the support domain B is a sphere with a radius of $l_0/2$.

The macroscale material characteristic length l_0 determines the width of fracture bands whose value can be measured by high-fidelity numerical simulations using discrete fracture mechanics or via a dedicated experiment with high-resolution digit image correlation analyses. To check mesh independence in fracture analyses, element sizes are generally selected to be much smaller than l_0 and the post-peak damage responses are tracked to show convergence upon mesh refinement, which we demonstrate in Sects. 4.1 and 4.2.

2.3.2 Microscale mesh dependency control by fracture energy

In the presence of fracture, microstructure responses are no longer properly represented by stress–strain relations since they cause damage models to lack objectivity to mesh choice and result in imaginary wave speeds amid damage progression. Using an arbitrary microscale characteristic length (l_1) in a microscale damage model, similar to the macroscopic counterpart (l_0) defined in Eq. (16), helps to stabilize ill-posed strain-softening but the simultaneous applications of l_0 and l_1 in a multiscale model is not physically realistic.

To mitigate mesh dependency on microscopic damage models, the softening part of the constitutive law is converted from the stress–strain relation to a stress-displacement relation by an element characteristic length (l_e). Then, the fracture energy (G_f) is specified as the dissipated energy (after damage initiation) that opens a unit area of the crack as:

$$G_f = \int_{\overline{\mathrm{E}}_0^{\mathrm{pl}}}^{\overline{\mathrm{E}}_{\mathrm{f}}^{\mathrm{pl}}} l_e S_y d\overline{\mathrm{E}}^{\mathrm{pl}} = \int_0^{\overline{u}_{\mathrm{f}}^{\mathrm{pl}}} S_y d\overline{u}^{pl} \tag{17}$$

where G_f is defined on each microscale IP with an l_e, and S_y is the yield stress corresponding to $\overline{E}^{pl}$. The equivalent plastic displacement $\overline{u}^{pl}$ is the fracture work conjugate of S_y in the fracture evolution from damage initiation (marked with the effective plastic strain $\overline{E}_0^{pl}$ and zero plastic displacement) to the final rupture (represented by the effective plastic strain $\overline{E}_f^{pl}$ and the fracture plastic displacement $\overline{u}_f^{pl}$). For FEs with the first-order continuity, l_e equals the length of a line across elements. l_e can also be directly specified as a function of element topology or material orientation [32].

With the effective plastic strains converted to plastic displacements with associated elemental l_e, the damage evolution is therefore defined based on the released energy during the damage propagation in an exponential form [32] of the plastic displacement as:

$$D = 1 - \exp\left(-\frac{1}{G_f} \int_0^{\overline{u}^{pl}} S_y d\overline{u}^{pl} \right) \tag{18}$$

where the damage variable approaches one only asymptotically at infinitely large plastic displacement. In practice, we set D as one when dissipated energies exceed $0.99G_f$. We note that fracture directions are generally not known a priori and, as a result, the elements with high aspect ratios behave differently depending on the direction along which the crack occurs. Thus, continuum damage models typically prefer elements with an aspect ratio close to one.

2.4 Convergence difficulty of implicit solvers

Strain softening and stiffness degradation amid damage evolution cause serious convergence difficulties for implicit time integration schemes because the implicitly integrated algebraic systems exhibit singular or ill-conditioned stiffness matrices for materials with degraded moduli. To demonstrate this, consider a simple isotropic damage model whose constitutive equation can be integrated analytically [33] by the classic implicit backward-Euler integration scheme. The algorithmic tangent operator $\mathbb{C}_{\mathrm{n+1}}^{\mathrm{alg.}}$ in the damage model at time step $(n+1)$ is written as [34]:

$$\mathbb{C}^{\mathrm{alg.}}_{\mathrm{n+1}} \equiv \frac{\partial \mathbf{S}_{\mathrm{n+1}}}{\partial \mathbf{E}_{\mathrm{n+1}}} = (1 - D_{n+1})\mathbb{C}^{\mathrm{el}} - \frac{\mathrm{S}_{\mathrm{n+1}} - \mathrm{H}_{\mathrm{n}}\overline{\mathrm{E}}^{\mathrm{pl}}_{\mathrm{n+1}}}{(\overline{\mathrm{E}}^{\mathrm{pl}}_{\mathrm{n+1}})^3}\mathbf{S}^0_{\mathrm{n+1}} \otimes \mathbf{S}^0_{\mathrm{n+1}} \tag{19}$$

where $\overline{\mathrm{E}}^{\mathrm{pl}}_{\mathrm{n+1}}$, $\mathrm{S}_{\mathrm{n+1}}$, $\mathbf{S}^0_{\mathrm{n+1}}$, H_{n} are the equivalent plastic strain, equivalent stress, referenced (undamaged) stress tensor, and softening modulus, respectively (the subscripts denote the time step). Upon damage initiation, H_{n} becomes negative and hence $\mathbb{C}^{\mathrm{alg.}}_{\mathrm{n+1}}$ loses its positive definiteness in some loading states. The non-positive definite $\mathbb{C}^{\mathrm{alg.}}_{\mathrm{n+1}}$ causes the elemental stiffness matrix ($\mathbf{k}^{\mathrm{e}}_{\mathrm{n+1}}$) to be ill-conditioned with possible negative eigenvalues. As damaged elements agglomerate in certain strain-softening regions, negative eigenvalues enter the global stiffness matrix ($\mathbf{K}_{\mathrm{n+1}}$) via element assembly process:

$$\mathbf{K}_{\mathrm{n+1}} = \frac{\partial \mathbf{F}^{\mathrm{int}}_{\mathrm{n+1}}}{\partial \mathbf{u}_{\mathrm{n+1}}} = \mathrm{A}_{\mathrm{e}}(\mathbf{k}^{\mathrm{e}}_{\mathrm{n+1}}) = \mathrm{A}_{\mathrm{e}}\left(\int_{\Omega_e} \mathbf{B}^{\mathrm{T}} : \mathbb{C}^{a\,\mathrm{lg.}}_{n+1} : \mathbf{B} d\Omega\right) \tag{20}$$

where $\mathbf{F}^{\mathrm{int}}_{\mathrm{n+1}}$ and $\mathbf{u}_{\mathrm{n+1}}$ are the internal forces and nodal displacements at time step $(n+1)$. $\mathrm{A}_{\mathrm{e}}(\dots)$ is the assembling operator and $\mathbf{B}$ is the strain–displacement matrix evaluated within each element domain Ω_e. So, the locally damaged elements cause the global stiffness matrix to become ill-conditioned. Note that softening occurs not only in damage but also in plastic models, and in both cases, the condition number of the global stiffness matrix deteriorates.

The ill-conditioning of the global stiffness matrix reduces the efficiency of the Newton–Raphson procedure. Specifically, when the algebraic system becomes singular in a certain step due to degraded moduli, the Newton–Raphson iteration halts before convergence. Solutions for improving implicit solvers' convergence rates are based on, e.g., viscous regularization schemes [32] and continuation methods [35] that render the tangent stiffness to be positive-definite in sufficiently small steps. However, such remedies fail in many scenarios and the implicit schemes face severe convergence difficulty. To fundamentally resolve the softening-induced numerical instability, we develop an adaptive hybrid time integration scheme in the next section.

3 Proposed adaptive reduced-order multiscale damage model

Our approach has two novel ingredients: (1) an adaptive assembly-free impl-exp time integration scheme that automatically adjusts temporal step sizes amid nonlinear analyses without re-assembling global stiffness matrices of the underlying algebraic system, and (2) an adaptive clustering strategy which alters local spatial discretization to preserve solution accuracy in concentration regions. These temporal and spatial adaption procedures aim to efficiently improve simulation accuracy and they are discussed in Sects. 3.1 and 3.2, respectively. The overall steps of our approach are summarized in Sect. 3.3.

3.1 Adaptive temporal reduction

Time-dependent nonlinear analyses are often integrated via pure implicit schemes, e.g., the Newton–Raphson method, due to their high efficiency and unconditional stability. However, in strain-softening simulations, ill-conditioned algebraic systems with near-zero or negative tangent moduli often prevent iterative solutions from convergence even after many steps. To reduce the overall number of temporal steps, we adopt a proper time integration scheme with robust convergence property and improve it as explained below.

3.1.1 Standard impl-exp scheme

The impl-exp scheme [34, 36] is a hybrid method that is robust and efficient in integrating softened constitutive equations. This method avoids the softening-induced numerical convergence issues discussed in Sect. 2.4 and its algorithm is demonstrated for the elasto-plasticity in Table 1 and the ductile damage model in Table 2.

The basic idea of the impl-exp scheme is to maintain the positive-definiteness of the algorithmic tangent operator by dividing constitutive integration into two stages. In the first stage, the plastic multiplier increment $\Delta\widetilde{\lambda}_{n+1}$ is explicitly extrapolated at time step $(n+1)$ from $\Delta\lambda_n$ to compute an 'explicit' stress $\widetilde{\mathbf{S}}_{\mathrm{n+1}}$ which balances the equilibrium equation between internal and external forces. In the second stage, the 'implicit' stress $\mathbf{S}_{\mathrm{n+1}}$ is obtained in terms of the current strain $\mathbf{E}_{\mathrm{n+1}}$ via the standard implicit backward Euler method. Then, the 'implicit' stress $\mathbf{S}_{\mathrm{n+1}}$ is used in the next step to compute the trial stress for updating yield function, see Table 1. With impl-exp schemes, both tangent operators $\widetilde{\mathbb{C}}^{\mathrm{alg.}}_{\mathrm{n+1}}$ are guaranteed to be positive-definite for the elasto-plasticity and damage models in Tables 1 and 2, respectively.

In an impl-exp scheme, the tangent operator is kept constant in a Newton step (as opposed to a pure implicit scheme,) which dramatically reduces the required iteration numbers within a step. However, since the tangent operator varies between steps, the global stiffness matrix must be re-assembled at the first iteration of each Newton step. This repetitive reassembly results in large memory footprints and slows down the overall process. We address this issue in the next subsection with our new method.

Table 1 Standard impl-exp scheme. Positive-definiteness of the algorithmic tangent operator is preserved by dividing constitutive integration into two stages

Inputs: $\mathbb{C}^{el}$, $\Delta\mathbf{E}_{\mathrm{n+1}}$, $\mathbf{S}_{\mathrm{n}}$, q, Q, $\Delta\lambda_n$	
Trial stress:	$\mathbf{S}_{\mathrm{n+1}}^{\mathrm{trial}} = \mathbf{S}_{\mathrm{n}} + \mathbb{C}^{\mathrm{el}}\Delta\mathbf{E}_{\mathrm{n+1}}$
Yield criterion:	$f = f^{trial}(\mathbf{S}_{\mathrm{n+1}}^{\mathrm{trial}}, q)$
Implicit plastic strain increment:	$\Delta\mathbf{E}_{\mathrm{n+1}}^{\mathbf{pl}} = \Delta\lambda_{n+1}(\partial Q/\partial\mathbf{S}_{\mathrm{n+1}})$
Implicit stress:	$\mathbf{S}_{\mathrm{n+1}} = \mathbf{S}_{\mathrm{n+1}}^{\mathrm{trial}} - \mathbb{C}^{el} : \Delta\mathbf{E}_{\mathrm{n+1}}^{\mathrm{pl}}$
Explicit plastic strain:	$\Delta\tilde{\lambda}_{n+1} = (\Delta t_{n+1}/\Delta t_n)\Delta\lambda_n;$ $\tilde{\mathbf{E}}_{\mathrm{n+1}}^{\mathrm{pl}} = \mathbf{E}_{\mathrm{n+1}}^{\mathrm{pl}} + \Delta\tilde{\lambda}_{n+1}\left(\partial Q/\partial\tilde{\mathbf{S}}_{\mathrm{n+1}}\right)$
Explicit stress:	$\tilde{\mathbf{S}}_{\mathrm{n+1}} = \mathbf{S}_{\mathrm{n+1}}^{\mathrm{trial}} - \mathbb{C}^{el}\Delta\tilde{\lambda}_{n+1}\left(\partial Q/\partial\tilde{\mathbf{S}}_{\mathrm{n+1}}\right)$
Algorithmic tangent operator:	$\tilde{\mathbb{C}}_{n+1}^{alg.} = \frac{\partial\tilde{\mathbf{S}}_{\mathrm{n+1}}}{\partial\mathbf{E}_{\mathrm{n+1}}} =$ $\frac{\partial\left(\mathbb{C}^{el}\left(\mathbf{E}_{\mathrm{n+1}} - \mathbf{E}_{\mathrm{n}}^{\mathrm{pl}}\right) - \mathbb{C}^{el}\Delta\tilde{\lambda}_{n+1}\frac{\partial Q\left(\tilde{\mathbf{S}}_{\mathrm{n+1}}(\mathbf{E}_{\mathrm{n+1}})\right))}{\partial\tilde{\mathbf{S}}_{\mathrm{n+1}}}\right)}{\partial\mathbf{E}_{\mathrm{n+1}}}$
Outputs: $\tilde{\mathbb{C}}_{n+1}^{alg.}$, $\tilde{\mathbf{S}}_{\mathrm{n+1}}$	

Table 2 Standard impl-exp scheme with damage. It is assumed that the material behaves elasto-plastically as described in Table 1

Inputs: $\mathbb{C}^{\mathrm{el}}$, $\mathbf{E}_{\mathrm{n+1}}^{\mathrm{el}}$	
Effective stress:	$\mathbf{S}_{\mathrm{n+1}}^{0} = \mathbb{C}^{\mathrm{el}} : \mathbf{E}_{\mathrm{n+1}}$
Internal variable:	$\Delta\tilde{\lambda}_{n+1} = (\Delta t_{n+1}/\Delta t_n)\Delta\lambda_n$
Damage initiation criterion:	$\omega_d = \overline{E}_{n+1}^{pl}/\overline{E}_d^{pl} = \left(\overline{E}_n^{pl} + \Delta\overline{E}_{n+1}^{pl}\right)/\overline{E}_d^{pl} \geq 1$
Damage parameter:	$\tilde{D}_{n+1} = \tilde{D}_{n+1}\left(D_n, \Delta\tilde{\lambda}_{n+1}\right)$
Damage stress:	$\tilde{\mathbf{S}}_{\mathrm{n+1}} = \left(1 - \tilde{D}_{n+1}\right)\mathbf{S}_{\mathrm{n+1}}^{0}$
Algorithmic tangent operator:	$\tilde{\mathbb{C}}_{n+1}^{alg.} = \partial\tilde{\mathbf{S}}_{\mathrm{n+1}}/\partial\mathbf{E}_{\mathrm{n+1}} = \left(1 - \tilde{D}_{n+1}\right)\mathbb{C}^{\mathrm{el}}$
Outputs: $\tilde{\mathbb{C}}_{n+1}^{alg.}$, $\tilde{\mathbf{S}}_{\mathrm{n+1}}$	

3.1.2 Adaptive assembly-free impl-exp scheme

To improve the computational efficiency of the impl-exp scheme introduced in Sect. 3.1.1, we propose a novel adaptive assembly-free impl-exp (AAF-IE) scheme. Compared to the standard impl-exp, our approach is more efficient because it avoids the re-assembly of the underlying stiffness matrix in runtime and adaptively adjusts temporal steps to improve integration accuracy. The key idea is to estimate plastic strain tensors with explicit extrapolations (instead of expensive implicit estimators) while maintaining equilibrium conditions and satisfying constitutive equations. Specifically, in contrast to the standard impl-exp which updates 'explicit' state variables with respect to a linearly extrapolated plastic multiplier increment $\Delta\widetilde{\lambda}_{n+1}$, we set the extrapolated state variables in AAF-IE to the plastic strain increment tensor, $\Delta\widetilde{\mathbf{E}}_{\mathrm{n+1}}^{\mathrm{pl}}$, as:

$$\begin{aligned}\tilde{\mathbf{S}}_{\mathrm{n+1}}(\Delta\tilde{\mathbf{E}}_{\mathrm{n+1}}^{\mathrm{pl}}) &= \tilde{\mathbf{S}}_{\mathrm{n+1}}^{\mathrm{trial}} - \mathbb{C}^{\mathrm{el}} : \Delta\tilde{\mathbf{E}}_{\mathrm{n+1}}^{\mathrm{pl}} \\ &= \mathbb{C}^{\mathrm{el}} : (\mathbf{E}_{\mathrm{n}} - \mathbf{E}_{\mathrm{n}}^{\mathrm{pl}} + \Delta\mathbf{E}_{\mathrm{n+1}} - \Delta\tilde{\mathbf{E}}_{\mathrm{n+1}}^{\mathrm{pl}}) \\ &= \mathbb{C}^{\mathrm{el}} : \mathbf{E}_{\mathrm{n+1}} - \mathbb{C}^{\mathrm{el}} : \mathbf{E}_{\mathrm{n}}^{\mathrm{pl}} - \mathbb{C}^{\mathrm{el}} : \Delta\tilde{\mathbf{E}}_{\mathrm{n+1}}^{\mathrm{pl}}\end{aligned} \tag{21}$$

$$\Delta\tilde{\mathbf{E}}_{\mathrm{n+1}}^{\mathrm{pl}} = \frac{\Delta t_{\mathrm{n+1}}}{\Delta t_{\mathrm{n}}}\Delta\mathbf{E}_{\mathrm{n}}^{\mathrm{pl}} \tag{22}$$

where $\Delta\mathbf{E}_{\mathrm{n}}^{\mathrm{pl}}$ is the converged implicit plastic strain tensor increment from the previous time step (n). The algorithmic tangent operator for the elasto-plasticity model is simplified as:

$$\begin{aligned}\tilde{\mathbb{C}}_{n+1}^{a\,\mathrm{lg.}} &= \frac{\partial\tilde{\mathbf{S}}_{\mathrm{n+1}}(\Delta\tilde{\mathbf{E}}_{\mathrm{n+1}}^{\mathrm{pl}})}{\partial\mathbf{E}_{\mathrm{n+1}}} \\ &= \frac{\partial(\mathbb{C}^{\mathrm{el}} : \mathbf{E}_{\mathrm{n+1}} - \mathbb{C}^{\mathrm{el}} : \mathbf{E}_{\mathrm{n}}^{\mathrm{pl}} - \mathbb{C}^{\mathrm{el}} : \Delta\tilde{\mathbf{E}}_{\mathrm{n+1}}^{\mathrm{pl}})}{\partial\mathbf{E}_{\mathrm{n+1}}} = \mathbb{C}^{\mathrm{el}}\end{aligned} \tag{23}$$

The advantage of the Eq. (23) comes from the fact that the tangent operator $\widetilde{\mathbb{C}}_{\mathrm{n+1}}^{\mathrm{alg.}}$ is now independent of the plastic potential $Q(\widetilde{\mathbf{S}}_{\mathrm{n+1}}(\mathbf{E}_{\mathrm{n+1}}))$, see the standard impl-exp in Table 1, and it always equals to the elastic tangent modulus $\mathbb{C}^{\mathrm{el}}$ during the entire analysis. As a result, the global stiffness matrix ($\mathbf{K}_{\mathrm{n+1}}$) of the elasto-plastic system in Eq. (20) is only assembled once in the offline stage where its Cholesky decomposition matrices are calculated and then stored for repeated usage in runtime. The constant stiffness matrix avoids re-assembly and therefore significantly reduces memory usage while improving efficiency.

As the material becomes damaged, $\widetilde{D}_{n+1}$ and $\widetilde{\mathbb{C}}_{\mathrm{n+1}}^{\mathrm{alg.}}$ still depend on the linearly extrapolated $\Delta\widetilde{\lambda}_{n+1}$ which itself is a function of the time step increments (Δt_n and Δt_{n+1}) and the implicitly converged $\Delta\lambda_n$. However, since softening is highly localized in a strain-softening zone with a limited number of fractured elements (compared to the whole mesh), the element stiffness matrices of the damaged elements can be incrementally updated in the global stiffness matrix as:

$$\mathbf{K}_{\mathrm{n\,+\,1}} = \mathbf{K}_{\mathrm{n}} + \Delta\mathbf{K}^{\mathrm{d}}_{\mathrm{n\,+\,1}} \tag{24}$$

where $\Delta\mathbf{K}^{\mathrm{d}}_{\mathrm{n+1}}$ corresponds to the entries with damaged material properties. This incremental assembly technique avoids overall re-assembly and significantly reduces memory footprints.

A major limitation of the standard impl-exp is the accuracy loss incurred by large time steps [34]. The accuracy loss comes from the mismatch between the linearly extrapolated state variables and their real values, especially upon transitioning between elasticity and plasticity where the state variables (e.g., effective plastic strain increments) change abruptly. Thus, it is recommended to use sufficiently small steps to reduce the extrapolation error (like explicit integration schemes) which increases the computational expenses.

Noting that large time steps can be used at non-critical moments when material properties change smoothly, we introduce an adaptive integrator in the framework of an assembly-free impl-exp scheme to automatically adjust the time steps. At the arbitrary time instance t_{n+1}, the explicitly extrapolated increment of the plastic strain tensor can be expressed as its implicit counterpart from the last step:

$$\Delta\tilde{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n\,+\,1}} = \tilde{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n\,+\,1}} - \mathbf{E}^{\mathrm{pl}}_{\mathrm{n}} = \frac{\Delta t_{n+1}}{\Delta t_n}\Delta\mathbf{E}^{\mathrm{pl}}_{\mathrm{n}} \tag{25}$$

The Taylor series expansion of the implicit plastic strain at t_{n-1} is:

$$\begin{aligned}\mathbf{E}^{\mathrm{pl}}_{\mathrm{n-1}} &= \mathbf{E}^{\mathrm{pl}}_{\mathrm{n}}(t_n - \Delta t_n)\\ &= \mathbf{E}^{\mathrm{pl}}_{\mathrm{n}} - \dot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}\Delta t_n + \frac{1}{2}\ddot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}(\Delta t_n)^2 - \frac{1}{6}\dddot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}(\Delta t_n)^3 + o((\Delta t_n)^4)\end{aligned} \tag{26}$$

where $\dot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}$, $\ddot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}$ and $\dddot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}$ are the first, second, and third-order time derivatives at the time-step n. The implicit plastic strain increment is then expressed as:

$$\begin{aligned}\Delta\mathbf{E}^{\mathrm{pl}}_{\mathrm{n}} &= \mathbf{E}^{\mathrm{pl}}_{\mathrm{n}} - \mathbf{E}^{\mathrm{pl}}_{\mathrm{n-1}}\\ &= \dot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}\Delta t_n - \frac{1}{2}\ddot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}(\Delta t_n)^2 + \frac{1}{6}\dddot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}(\Delta t_n)^3 + o((\Delta t_n)^4)\end{aligned} \tag{27}$$

By substituting Eq. (27) to (25) and assuming $\Delta t_{n+1} = \Delta t_n$, we arrive at the expansion of the extrapolated increment $\Delta\tilde{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n+1}}$ at time-step n as:

$$\Delta\tilde{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n\,+\,1}} = \dot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}\Delta t_{n+1} - \frac{1}{2}\ddot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}(\Delta t_{n+1})^2 + \frac{1}{6}\dddot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}(\Delta t_{n+1})^3 + o((\Delta t_{n+1})^4) \tag{28}$$

To compare the explicit solution with the implicit one, we expand the implicitly integrated increment $\Delta\mathbf{E}^{\mathrm{pl}}_{\mathrm{n+1}}$ at time-step n via Taylor series:

$$\begin{aligned}\Delta\mathbf{E}^{\mathrm{pl}}_{\mathrm{n\,+\,1}} &= \mathbf{E}^{\mathrm{pl}}_{\mathrm{n\,+\,1}} - \mathbf{E}^{\mathrm{pl}}_{\mathrm{n}}\\ &= \dot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}\Delta t_{n+1} + \frac{1}{2}\ddot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}(\Delta t_{n+1})^2 + \frac{1}{6}\dddot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}(\Delta t_{n+1})^3 + o((\Delta t_{n+1})^4)\end{aligned} \tag{29}$$

By subtracting Eq. (29) from (28), we identify the error between the implicit solution and the assembly-free impl-exp as:

$$\begin{aligned}\tilde{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n\,+\,1}} - \mathbf{E}^{\mathrm{pl}}_{\mathrm{n\,+\,1}} &= (\mathbf{E}^{\mathrm{pl}}_{\mathrm{n}} + \Delta\tilde{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n\,+\,1}}) - (\mathbf{E}^{\mathrm{pl}}_{\mathrm{n}} + \Delta\mathbf{E}^{\mathrm{pl}}_{\mathrm{n\,+\,1}})\\ &= -\ddot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}(\Delta t_{n+1})^2 + o((\Delta t_{n+1})^4)\end{aligned} \tag{30}$$

We create an error bound so that the extrapolated error (e) is bounded for all IPs $\mathbf{X} \in \mathbf{\Omega}$:

$$e(\mathbf{X}) = \left|\tilde{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n\,+\,1}} - \mathbf{E}^{\mathrm{pl}}_{\mathrm{n\,+\,1}}\right|_{\max}(\mathbf{X}) \le \xi E^{\mathrm{ref}} \tag{31}$$

where ξ is the user-defined tolerance with respect to a model-dependent reference value $\mathrm{E}^{\mathrm{ref}}$ and $|x_1, x_2, \ldots|_{\max} = \max(|x_1|, |x_2|, \ldots)$. We now substitute Eq. (30) into (31) and drop the high-order term to arrive at the error upper bound:

$$e = \left|-\ddot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}(\Delta t_{n+1})^2 + o((\Delta t_{n+1})^4)\right|_{\max} \simeq (\Delta t_{n+1})^2\left|\ddot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}}\right|_{\max} \le \xi E^{\mathrm{ref}} \tag{32}$$

To obtain the second-order time derivative of the plastic strain, we expand its first-order derivative at time steps n and $n - 1$ as:

$$\begin{cases}\dot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}} = \dfrac{\mathbf{E}^{\mathrm{pl}}_{\mathrm{n}} - \mathbf{E}^{\mathrm{pl}}_{\mathrm{n-1}}}{\Delta t_{\mathrm{n}}} = \dfrac{\Delta\mathbf{E}^{\mathrm{pl}}_{\mathrm{n}}}{\Delta t_{\mathrm{n}}}\\[2ex] \dot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n\,-1}} = \dfrac{\mathbf{E}^{\mathrm{pl}}_{\mathrm{n\,-1}} - \mathbf{E}^{\mathrm{pl}}_{\mathrm{n-2}}}{\Delta t_{\mathrm{n\,-1}}} = \dfrac{\Delta\mathbf{E}^{\mathrm{pl}}_{\mathrm{n\,-1}}}{\Delta t_{\mathrm{n\,-1}}}\\[2ex] \ddot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}} = \dfrac{\dot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n}} - \dot{\mathbf{E}}^{\mathrm{pl}}_{\mathrm{n-1}}}{\Delta t_{\mathrm{n}}} = \dfrac{1}{\Delta t_{\mathrm{n}}}\left(\dfrac{\Delta\mathbf{E}^{\mathrm{pl}}_{\mathrm{n}}}{\Delta t_{\mathrm{n}}} - \dfrac{\Delta\mathbf{E}^{\mathrm{pl}}_{\mathrm{n\,-1}}}{\Delta t_{\mathrm{n\,-1}}}\right)\end{cases} \tag{33}$$

We now acquire the extrapolation error by substituting Eq. (33) into (32):

$$e \simeq \frac{(\Delta t_{n+1})^2}{\Delta t_n}\left|\left(\frac{\Delta\mathbf{E}^{\mathrm{pl}}_{\mathrm{n}}}{\Delta t_n} - \frac{\Delta\mathbf{E}^{\mathrm{pl}}_{\mathrm{n-1}}}{\Delta t_{n-1}}\right)\right|_{\max} \le \xi E^{\mathrm{ref}} \tag{34}$$

We compute the maximum time step at any integration point $\mathbf{X}$ while keeping the extrapolation error in bound by equating the error to the bound as:

$$\Delta t_{n+1}(\mathbf{X}) \leq \sqrt{\frac{\xi E^{\text{ref}} \Delta t_n}{\left|\left(\frac{\Delta \mathbf{E}_{\text{n}}^{\text{pl}}}{\Delta t_n} - \frac{\Delta \mathbf{E}_{\text{n}-1}^{\text{pl}}}{\Delta t_{n-1}}\right)\right|_{\max}(\mathbf{X})}} \tag{35}$$

To bound the error in the entire domain, we set the critical step size as the minimum value of critical step sizes across all the points at the current time step $(n+1)$, that is:

$$\Delta t_{n+1}^{cri.} = \underset{\mathbf{X}\in\Omega}{\text{MIN}} \sqrt{\frac{\xi E^{\text{ref}} \Delta t_n}{\left|\left(\frac{\Delta \mathbf{E}_{\text{n}}^{\text{pl}}}{\Delta t_n} - \frac{\Delta \mathbf{E}_{\text{n}-1}^{\text{pl}}}{\Delta t_{n-1}}\right)\right|_{\max}(\mathbf{X})}} \tag{36}$$

Equation (36) provides the critical time step in our framework. We point out that $\Delta t_{n+1}^{cri.}$ depends not only on the previous time increments (Δt_n and Δt_{n-1}) but also on the (implicit) plastic strain increments ($\Delta \mathbf{E}_{\text{n}}^{\text{pl}}$ and $\Delta \mathbf{E}_{\text{n}-1}^{\text{pl}}$) from the last two steps. Specifically, if the material property advances smoothly in its previous two steps, the values of plastic strain increments are close and, therefore, $\Delta t_{n+1}^{cri.}$ takes on a large value. In contrast, if the properties change abruptly (e.g., from elasticity to plasticity or vice versa), the difference between the previous two plastic strain increments is large and $\Delta t_{n+1}^{cri.}$ is reduced accordingly. Equation (36) can be further simplified as:

$$\gamma_{n+1} = \frac{\Delta t_{n+1}^{cri.}}{\Delta t_0} \approx \Delta t_n \underset{\mathbf{X}\in\Omega}{\text{MIN}} \sqrt{\frac{\xi E^{\text{ref}}}{\left|\left(\Delta \mathbf{E}_{\text{n}}^{\text{pl}} - \Delta \mathbf{E}_{\text{n}-1}^{\text{pl}}\right)\right|_{\max}(\mathbf{X})}} \tag{37}$$

where γ_{n+1} is the ratio of time increments between the critical current time step and the (user-defined) initial step Δt_0. This simplification assumes the time increments for the last two steps are approximately the same ($\Delta t_n \approx \Delta t_{n-1}$), and their slight difference does not affect the step size at the current time point (t_{n+1}).

3.2 Adaptive spatial reduction

In addition to the adaptive temporal reduction discussed in Sect. 3.1, we develop an adaptive clustering-based spatial discretization method for our ROM. For any clustering-based ROM, domain decomposition generally converts the problem domain from sufficiently fine discretization (e.g., regularly shaped voxel grids from a computed tomography reconstruction or free meshes from a discretization module) into a set of interactive clusters with different shapes and sizes. Since the number of clusters is typically much smaller than the elements in the original discretization, the number of unknown variables is significantly reduced.

Table 3 Position-based clustering data. Elements are grouped in the k-means clustering by the 3D coordinates of their geometric centers

Data points $\boldsymbol{\zeta} = [\zeta_1, \zeta_2, \ldots, \zeta_{ne}]^T$	Features $\boldsymbol{\varphi} = \left[\varphi_1, \varphi_2, \ldots, \varphi_{nf}\right]^T$		
	$\varphi_1 = \text{X}$	$\varphi_2 = \text{Y}$	$\varphi_3 = \text{Z}$
$\zeta_1 = \text{FE}_1$	X_1	Y_1	Z_1
$\zeta_2 = \text{FE}_2$	X_2	Y_2	Z_2
…	…	…	…
$\zeta_{\text{ne}} = \text{FE}_{\text{ne}}$	X_{ne}	Y_{ne}	Z_{ne}

Clustering is an unsupervised learning technique that groups similar data points. Example clustering methods include k-means learning [37], affinity propagation [38], agglomerative clustering [39], and spectral clustering [40]. In this work, we use the k-means clustering technique which, similar to other methods, classifies mesh elements into different groups based on their feature values. The set of these features greatly affects the distinct clustering results. For example, a position-based clustering for grouping a total number of ne elements in a 3D mesh uses the X, Y, and Z coordinates of each element's geometric center as features, see Table 3.

In k-means clustering, cluster seeds[1] are first randomly scattered in the feature space. Then, each element is iteratively assigned to the cluster whose center (i.e., mean of its members) is closest to that element. During the assignment, the cluster means are automatically updated to minimize within-cluster variances by solving the following optimization problem:

$$\mathbf{C} = \underset{\mathbf{C}}{\text{argmin}} \sum_{I=1}^{k} \sum_{n\in C^I} \left\| \boldsymbol{\varphi}_n - \overline{\boldsymbol{\varphi}}_I \right\|^2 \tag{38}$$

where $\boldsymbol{C}$ refers to the created cluster centers $\boldsymbol{C} = \left\{C^1, C^2, \ldots, C^k\right\}$. The variables $\boldsymbol{\varphi}_n$ and $\overline{\boldsymbol{\varphi}}_I$ represent the nth element's feature and the feature mean of the Ith cluster, respectively.

Naïve position-based clustering is a static spatial decomposition that is created once in an offline stage. It does not alter during runtime and results in a set of uniformly distributed material clusters with similar sizes (see [17] and Appendix A for more details). Position-based clustering may lose its data compression efficiency in the presence of nonlinear strain-softening with significant localizations. Effective clustering in such cases requires dense discretization over regions with steep gradients which can arise anywhere in the domain during the deformation (recall that the strains are

[1] The user chooses the number of seeds.

constant in a cluster so capturing high gradients relies on having small clusters).

To accurately capture the softening-induced localizations in clustering, we propose a novel adaptive clustering strategy that consists of a static stage and a dynamic stage which are described in Sects. 3.2.1 and 3.2.2, respectively. The implementation details of these two stages are provided in Algorithm 1.

3.2.1 Offline static stress-informed clustering

To assign more clusters to concentration regions in an RVE where large solution gradients exist, we propose a hierarchical clustering scheme that utilizes elemental elastic responses to guide the position-based clustering. This hierarchical scheme is used only once in the offline stage of our ROM and has two levels which firstly groups elements with the similar elastic response and then decomposes each group into some clusters.[2] The rationale behind using materials' elastic behaviors in clustering is that they are highly correlated with plastic responses. For instance, a highly stressed material point during elastic reaction is likely to obtain large plastic strains and accumulate strain concentrations during plastic deformation. Hence, our contribution in this section is the development of a hierarchical clustering scheme that creates more clusters in critical regions identified by elastic stress responses.

At the first level, to obtain the scalar elastic stress response at each element we begin by applying a set of six orthogonal loads to the RVE. In each of the six cases, we deform the RVE by a homogeneous elastic macroscopic strain tensor and then calculate the microscale Von-Mises stress at each element's center. Next, we condense the vector containing the six elemental Von-Mises stresses (one from each load case) into a scalar via:

$$S^n = \left\| [S_1^n,\ S_2^n,\ S_3^n,\ S_4^n,\ S_5^n,\ S_6^n] \right\|_2 \tag{39}$$

where S_k^n, $(k = 1, \ldots, 6)$ is the Von-Mises stress at the nth element when the RVE is subject to the kth orthogonal load and S^n is the L2-norm of these stresses representing the local stress intensity at an FE in the deformed RVE. The format of the resulting dataset for the stress intensity is demonstrated in Table 4 for a 3D RVE whose mesh has ne elements.

Once the dataset of stress intensities is built, we start to divide the elements of a meshed RVE into multiple groups (the number of groups is chosen a priori) where we assume that the elements in a group have similar stress intensity and that each group has approximately the same number of elements. We note that the elements in the same group can be anywhere in the RVE, i.e., a group can be topologically disconnected, see Fig. 2.

Table 4 Stress-based clustering data. FEs are agglomerated via k-means clustering by their stress intensity values

Data points $\boldsymbol{\zeta} = [\zeta_1, \zeta_2, \ldots, \zeta_{ne}]^T$	Feature $\varphi = S^n$
$\zeta_1 = \mathrm{FE}_1$	S^1
$\zeta_2 = \mathrm{FE}_2$	S^2
…	…
$\zeta_{\mathrm{ne}} = \mathrm{FE}_{\mathrm{ne}}$	S^{ne}

After the groups are constructed, we then decompose each of them into multiple clusters where we assign more clusters to groups with higher stress intensities. That is, the second level of our hierarchical clustering approach aims to selectively decompose the groups such that concentration regions (marked with higher stress intensities) are discretized with more clusters. For illustration, consider Fig. 2 where the elements in a generic RVE are segmented into five groups where the groups with higher stress intensities are decomposed with more clusters, and as a result, high stress-induced local phenomena in these groups can be captured more accurately.

We introduce the clustering split factor (s_f), to better control the number of clusters in each group. Specifically, we first sort the groups in an ascending order based on their stress intensity values. Then, we compute the number of clusters in a group via a power function whose exponent is s_f:

$$N_{g_i}^{cl} = \frac{\theta_{g_i}}{\sum_{i=1}^{N_g} \theta_{g_i}} N^{cl} \tag{40}$$

$$\theta_{g_i} = \left(\frac{I_{g_i}}{N_g} \right)^{s_f} \tag{41}$$

where $N_{g_i}^{cl}$ is the number of clusters in the group g_i, θ_{g_i} is the fraction of total clusters assigned to g_i, I_{g_i} is the group's sorted index in the ascending order of stress intensities, N^{cl} is the prescribed total number of clusters, and N_g is the total number of groups. If the computed $N_{g_i}^{cl}$ is decimal, we round it up to the nearest integer and adjust the number of clusters in other groups accordingly such that $N^{cl} = \sum_{i=1}^{N_g} N_{g_i}^{cl}$.

We note that s_f is defined by the user and represents the contrast of cluster densities across the groups. This contrast is illustrated in Fig. 3 where the cluster numbers per group are controlled by adjusting s_f. We observe that a large s_f promotes increasing cluster numbers in groups with high-stress intensities. In contrast, a small s_f lowers the sensitivity of cluster density to stress intensity. Specifically, when $s_f = 0$ the numbers of clusters in each group are the same, and when $s_f > 0$ more clusters are generated in groups with

[2] While in both levels of our approach we are agglomerating elements, for clarity we reserve the terms group and cluster for the first and second levels, respectively.

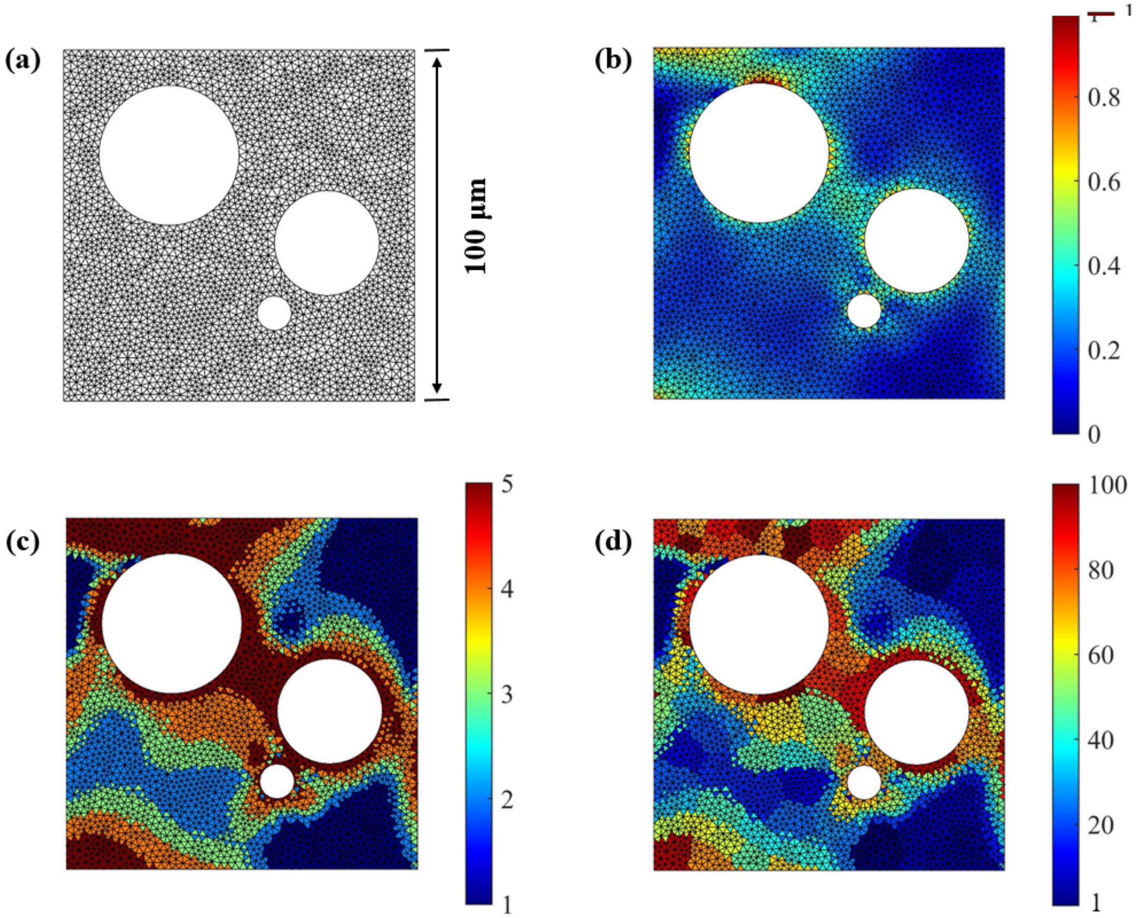

Fig. 2 Schematic 2D illustration of the offline stress-informed hierarchical clustering. **a** A generic 2D RVE is discretized with 5000 triangle elements. This RVE contains three circular pores with radii of 20 µm, 15 µm, and 5 µm. **b** Element stress intensities are computed via offline orthogonal loads where intensity values are scaled to [0, 1] to aim in visualization. **c** RVE elements are firstly agglomerated into five groups based on elemental intensity values where each group has 1000 elements. The intensity monotonically increases from group 1 to group 5. **d** Elements are then classified into a total of 100 clusters where the groups 1, 2, 3, 4, and 5 contain 10, 15, 20, 25, and 30 clusters, respectively

higher average intensity values. Also observing that an excessively large split factor can dramatically reduce clusters in the groups with low-stress intensities. For example, when $s_f = 10$ most clusters are created in the group with the highest intensity, and many groups with low intensities only contain one cluster. One cluster cannot capture solution gradients, and if local phenomena occur in those regions during the online stage (overlooked in offline elastic tests), the lack of sufficient clusters will result in dramatic local inaccuracy. Therefore, we do not recommend large s_f and we set the default value to 1.0 which makes the number of clusters in each group to be linearly proportional to the sorting index of stress intensities. We illustrate the effects of s_f on the accuracy of our ROM in Sect. 4.2.

We point out that the total number of clusters, N^{cl}, is determined by the user and then they are distributed across the groups by Eq. (40). If N^{cl} is too small, too few clusters in a group may not sufficiently represent sharp solution gradients in a group. Hence, we require the computed cluster number from Eq. (40) to be at least as large as $N_{g_i}^{Cl*}$ which is the smallest number that well-behaved clustering requires, that is:

$$N_{g_i}^{cl} = \max(N_{g_i}^{cl},\ N_{g_i}^{Cl*}) \tag{42}$$

There are multiple techniques to compute $N_{g_i}^{Cl*}$ including gap statistics [41], elbow method [42], Silhouette coefficient [43], and Calinski–Harabasz index [44]. We adopt the Calinski–Harabasz index in this work which is computed as:

$$CH_k = \frac{V_b}{V_w} \times \frac{N_{elem}^{g_i} - k}{k - 1} \tag{43}$$

where CH_k is the Calinski–Harabasz index of k clusters, V_b is the overall between-cluster variance, V_w is the overall

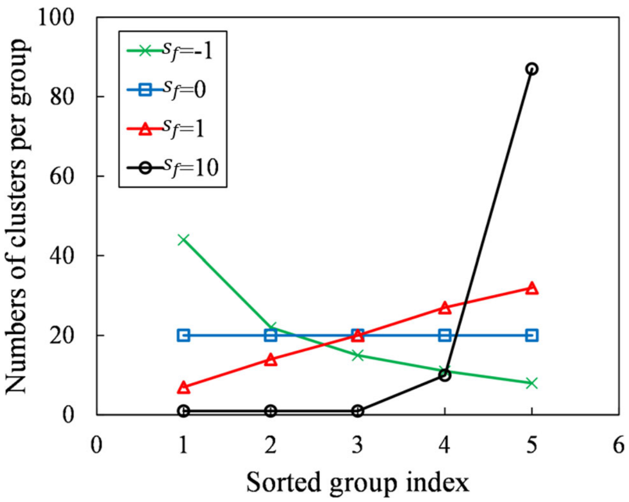

Fig. 3 Effects of split factor. The numbers of clusters in groups with high-stress intensities (marked by high sorted indices) rapidly increase as a positive s_f grows. In contrast, when n is negative, more clusters are assigned to non-critical groups with smaller intensity values. This figure is generated by distributing 100 clusters between five groups

within-cluster variance, and $N_{elem}^{g_i}$ is the number of elements in the ith group. The overall between-cluster variance and the overall within-cluster variance are defined as:

$$V_b = \sum_{j=1}^{k} N_{elem}^{cl_j} \left\| m_j - m \right\|^2 \tag{44}$$

$$V_w = \sum_{j=1}^{k} \sum_{m_e \in cl_j} \left\| m_e - m_j \right\|^2 \tag{45}$$

where $N_{elem}^{cl_j}$ is the number of elements in the jth cluster of the ith group and m_j denotes the centroid of the jth cluster. m and m_e are the overall mean of feature values for all elements in the ith group and feature of one element, respectively. Since a good clustering has a large between-cluster variance and a small within-cluster variance, i.e., a large Calinski–Harabasz index value, the optimal cluster number for the ith group can be determined by the following optimization problem:

$$N_{g_i}^{Cl*} = \underset{k}{\text{argmax}}(CH_k) \tag{46}$$

where the applied cluster number for the ith group in k-means clustering is obtained by comparing $N_{g_i}^{Cl*}$ with $N_{g_i}^{cl}$, see Eqs. (40) and (42).

The hierarchical clustering is performed once in the offline stage of our ROM. In the next section, we augment this static offline clustering with a dynamic counterpart that evolves with iterative solutions of the nonlinear equilibrium equations of plasticity and damage.

3.2.2 Online dynamic clustering

We propose an online dynamic spatial decomposition method to accommodate the evolution of equilibrium equations during the initiation and propagation of strain hardening and softening. The dynamic method augments the static offline clustering of Sect. 3.2.1 with online refinements over critical regions in an RVE as it is deformed. In a multiscale simulation, the dynamic clustering is performed for all the RVEs associated with the macro-IPs.

While an RVE is deformed, we check a dynamic clustering initiation condition at each load increment. If the condition is not met, we solve the micro-equilibrium equations based on the existing clustering. Otherwise, we activate the dynamic refinement and calculate an error indicator that selects the "parent" clusters that must be decomposed into multiple "children" clusters. This decomposition relies on a remeshing procedure for dividing the clusters and transferring the state variables between the parent and children clusters. We stop the online decomposition once the total number of child clusters exceeds the user-defined total number of clusters (N_{user}^{cl}), that is:

$$\sum_{t=1}^{n+1} N_t^{cl} \leq N_{user}^{cl} \tag{47}$$

where N_t^{cl} is the number of new clusters created at time step t. Details of each step are discussed below.

In this work, we assume adaption starts and continues as long as the total number of existing clusters is less than the prescribed upper limit that is chosen by the user. If this condition is satisfied, we select the parent clusters that must be refined based on their equivalent plastic strain values. Specifically, we decompose the clusters whose plastic strains are either the highest in the RVE (spatial anomaly) or quite different from the plastic strains of the neighboring clusters (spatial discontinuity). We note that our adaption scheme is integrated with DCA [17] whose clusters are topologically connected (unlike the approach developed in [28]). Consequently, finding a cluster's neighbors and computing its spatial discontinuity is simply done by checking the connectivity of agglomerated elements followed by comparing their equivalent plastic strains.

The clusters with spatial anomalies are likely located in a strain softening region while those with spatial discontinuities tend to be at the boundary of damage bands surrounded by fractured and intact elements. We consider both cases as indicators of concentration regions which require higher degrees of interpolations (i.e., more cluster decomposition) for capturing solution gradients accurately.

Compared to existing adaptive methods such as the one in [45], the novelty of our approach is that we augment the two spatial softening indicators discussed above with a temporal metric to choose the proper time steps for spatial refinements. Specifically, we systematically integrate our dynamic clustering with the new AAF-IE scheme proposed in Sect. 3.1.2 to identify the transition time of material properties between elasticity and plasticity. The identification of the critical time instances helps to improve fracture modeling since increasing spatial interpolations at material transition stages are crucial to predict accurate strain states that precede plasticity and damage.

Once the parent clusters are selected at critical transition time instances, we refine them by re-meshing which controls the number of children clusters created at each step. This re-meshing, similar to the h-adaptivity of FEM, decomposes a parent cluster into multiple small clusters while maintaining the original boundaries. Since the parent clusters are in concentration regions with high solution gradients, this decomposition improves the solution accuracy by enriching spatial interpolations.

We divide each parent cluster by the position-based clustering which agglomerates material points based on their coordinates to multiple child clusters with topologically connected boundaries. For simplicity, we assume each parent cluster is decomposed into two child clusters, and neither of the new clusters can be further divided. Furthermore, we use the AAF-IE metric in Eq. (37) to approximate the number of clusters that must be added at the current time instance t_{n+1} as a function of the time increment Δt_n and the plastic strain increments ($\Delta \mathbf{E}^{\mathrm{pl}}_{\mathrm{n}}$ and $\Delta \mathbf{E}^{\mathrm{pl}}_{\mathrm{n-1}}$) from previous steps, that is:

$$N^{cl}_{n+1} = N^{cl}_0 \Big/ \gamma_{n+1} \approx N^{cl}_0 / \underset{\mathbf{X}\in\Omega}{\mathrm{MIN}}\, \Delta t_n \sqrt{\frac{\xi E^{\mathrm{ref}}}{\left|(\Delta \mathbf{E}^{\mathrm{pl}}_{\mathrm{n}} - \Delta \mathbf{E}^{\mathrm{pl}}_{\mathrm{n-1}})\right|_{\max}(\mathbf{X})}} \tag{48}$$

where N^{cl}_0 is the prescribed number of new clusters when the adaption step starts.

The underlying idea of Eq. (48) is to have the number of new clusters be inversely proportional to the AAF-IE's error that estimates the variation of material responses amid transitions. As a result, we provide the transition phase (when extrapolation errors are expected, see Sect. 3.1) with smaller time steps and finer spatial discretization. In Sect. 4.2 we demonstrate the advantage of dynamic clustering over a simple approach where a fixed number of clusters are added per load increment.

A major advantage of our dynamic clustering strategy is that parent and their child clusters are hierarchically related. That is, upon refinement, child clusters inherit the same history-dependent state variables from their parents and we only need to locally adjust the reduced stiffness matrix in DCA via the incremental assembly by Eq. (24). The implementation details of our dynamic and static clustering methods are outlined in Algorithm 1.

Algorithm 1 Structure of our adaptive clustering approach

```
procedure Develop microscopic fully adaptive clusters for highly localized phenomena
    ▷Offline stage: stress-informed static clustering
    if $t \leftarrow 0$    ▷At the initial time step
        Read the microstructure's FE mesh and material properties
        Apply six orthogonal loadings via elastic macro-strains and compute stress norm for each element
        Agglomerate micro-elements to groups based on element stress norm value
        Calculate the number of cluster seeds in each group according to their average stress values
        Check if the number of cluster seeds per group is optimal by the Calinski-Harabasz criterion
        if Calinski-Harabasz criterion is satisfied, then
            Generate the static stress-informed clusters for each group by the optimal seed numbers
        else
            Adjust group seed numbers and re-check Calinski-Harabasz criterion
    end if
    ▷Online stage: dynamic clustering
    if $t > 0$    ▷Increment the time steps in online computing
        for j $\leftarrow 0, N$ do    ▷Loop over $N$ load steps
            Read macroscopic deformation information
            Solve micro-equilibrium solutions
            Check clustering adaption criterion
            if clustering adaption is satisfied, then
                Identify target decomposing clusters via spatial error indicator
                Determine critical adaption time steps via temporal error indicator
            else
                Proceed to the next load step
            end if
            Estimate each group's new cluster numbers via temporal error estimator
            Compare the accumulated new cluster number to the prescribed upper limit
            if the current cluster number < upper limit, then
                Decompose target clusters and generate new clusters
                Transfer state variables from target clusters to their child clusters
                Update reduced stiffness matrix on entries associated with cluster refinement
            else
                Proceed to the next load step
            end if
        end for
    end if
end procedure
```

3.3 Workflow of our multiscale damage model

As demonstrated in Fig. 4 our model starts with the offline stress-informed clustering (see Sect. 3.2.1) which solves the multiscale nonlinear analysis by incrementing load steps to estimate the macroscale deformation gradients and microscale effective responses. This estimation is done via a hybrid integration scheme which avoids softening-induced solution divergence (see Sects. 2.1 and 2.4). In this process, we perform online adaptive temporal adjustment (see Sect. 3.1.2) and spatial decomposition (see Sect. 3.2.2) to improve the analysis accuracy in regions where strain-localized softening behaviors appear.

4 Numerical experiments

In this section, we study the effect of porosity on the softening behavior of metallic components made of aluminum alloy A356. We consider the alloy as the primary material phase and the pores as the secondary phase. Except for porosity, other polycrystalline microscopic features such as grain boundaries are out of the scope of this study. We perform the spatial and temporal adaptive reductions introduced in Sects. 3.1 and 3.2 on the primary material phase in all micro-, macro-, and multi-scale simulations. The values of elastic modulus (M) and Poisson's ratio (υ) of A356 are given as:

$$M = 5.70\text{e}4\,\text{MPa}, \quad \upsilon = 0.33 \tag{49}$$

A356's elastoplastic hardening behavior is assumed to be isotropic and follow an associated plastic flow rule with the Von-Mises yield surface defined as:

$$S \leq S_Y\left(\overline{E}^{pl}\right) \tag{50}$$

where S is the Von-Mises equivalent stress and the yield stress S_Y is governed by a predefined hardening law that depends on the equivalent plastic strain $\overline{E}^{pl}$. The material's hardening behavior is assumed as piecewise linear as demonstrated in Fig. 5. To model strain softening, the alloy is modeled as a ductile metal with the fracture strain (E_f) and fracture energy (G_f) given as:

$$E_f = 6.67\text{e} - 2, \quad G_f = 1.92\text{E4 N/m} \tag{51}$$

As discussed in Sect. 3, our ROM framework employs temporal and spatial reductions to accelerate damage simulations. Correspondingly, in Sect. 4.1, we study a macroscopic 3D plate to demonstrate the benefits of the proposed AAF-IE approach discussed in Sect. 3.1 which achieves temporal reduction by reducing required time steps in intricate nonlinear analyses. Next, in Sect. 4.2, we test the adaptive spatial clustering techniques discussed in Sect. 3.2 to deform 3D microstructures under complex loading conditions. Finally, in Sect. 4.3 the two reduction techniques are integrated within the first-order computational homogenization scheme to quantify the effects of spatially varying micro-porosity on macro-component damage behaviors. In all experiments, the accuracy of the proposed ROM is verified against direct numerical simulations (DNS) based on FEM.

The proposed method is implemented in Matlab. All macro- and micro-scale simulations in Sects. 4.1 and 4.2 are conducted on a 64-bit Windows desktop with the following hardware: four Intel i5-8250U CPU cores running at 1.8 GHz with 16 GB installed RAM. The concurrent multiscale simulation in Sect. 4.3 is performed on a high-performance cluster with 30 CPU cores (AMD EPYC processor running at 4.1 GHz) and 192 GB RAM.

4.1 Macroscale experiments

We test the AAF-IE method on the macroscale 3D plate shown in Fig. 6a. Due to symmetry, we deform only one-quarter of the model by applying a Dirichlet boundary condition ($\overline{u} = 0.8$mm) on the top surface and symmetric boundary conditions on the left and bottom surfaces, see Fig. 6b. To demonstrate the efficiency of AAF-IE, we first consider an elastoplastic deformation with hardening but without fracture. To obtain a benchmark solution, we mesh the model with 6000 linear tetrahedron elements and compute the solution via an implicit scheme. The obtained

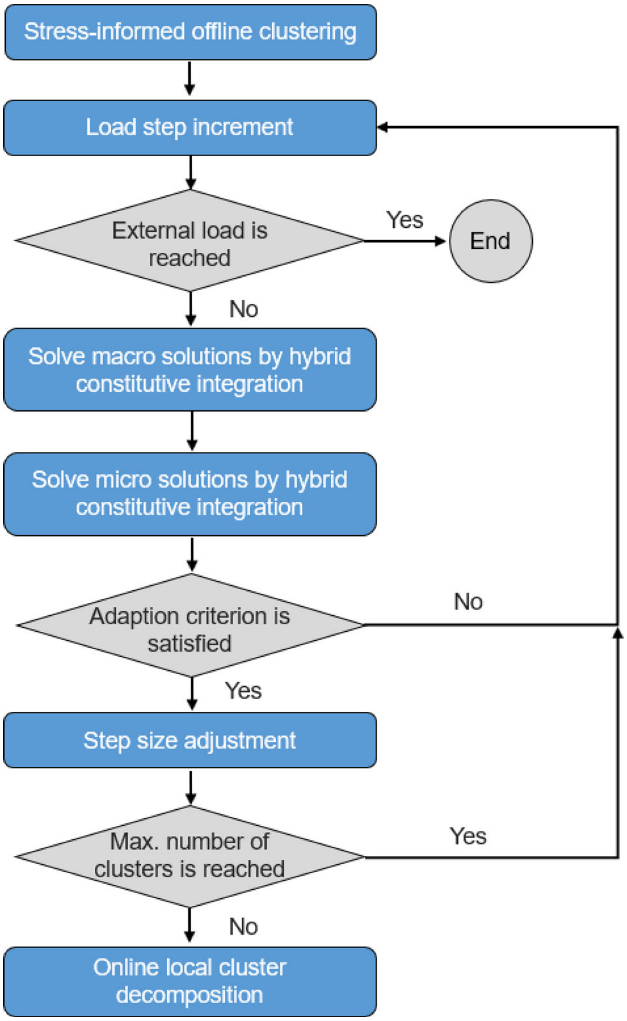
Stress-informed offline clustering
Load step increment
External load is reached
Yes
End
No
Solve macro solutions by hybrid constitutive integration
Solve micro solutions by hybrid constitutive integration
Adaption criterion is satisfied
No
Yes
Step size adjustment
Max. number of clusters is reached
Yes
No
Online local cluster decomposition

Fig. 4 Flowchart of our model. Our approach has offline and online stages where the latter stage allocates clusters to regions where high strain concentrations appear during the simulation

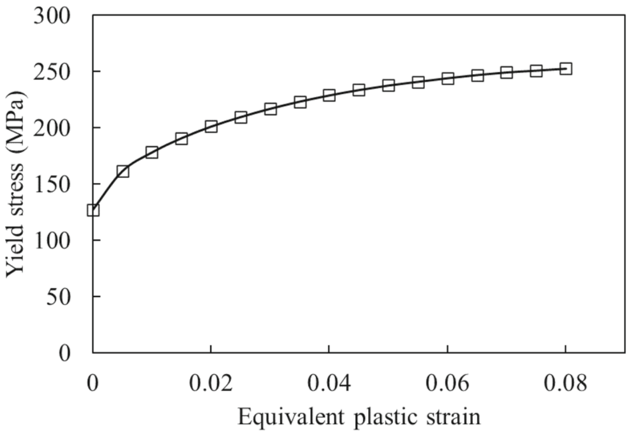

Fig. 5 Hardening behavior of A356. We use a piecewise linear hardening behavior

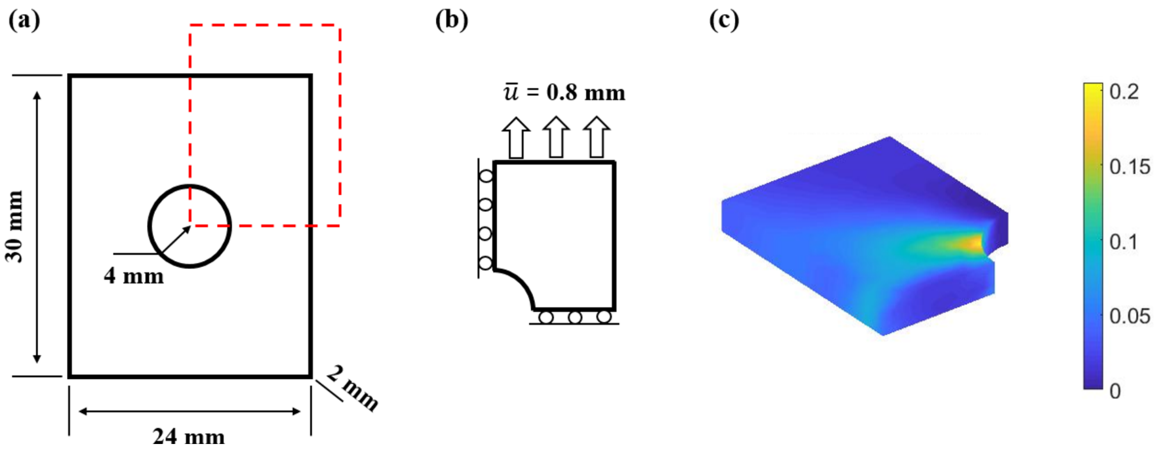

Fig. 6 Macroscale model. **a** Geometry and the dimensions of the 3D plate which has a hole in its center. Due to symmetry, only the red dashed part is modeled. **b** Boundary conditions are applied to the studied model. **c** Distribution of the equivalent plastic strain after work hardening (no damage, obtained via DNS)

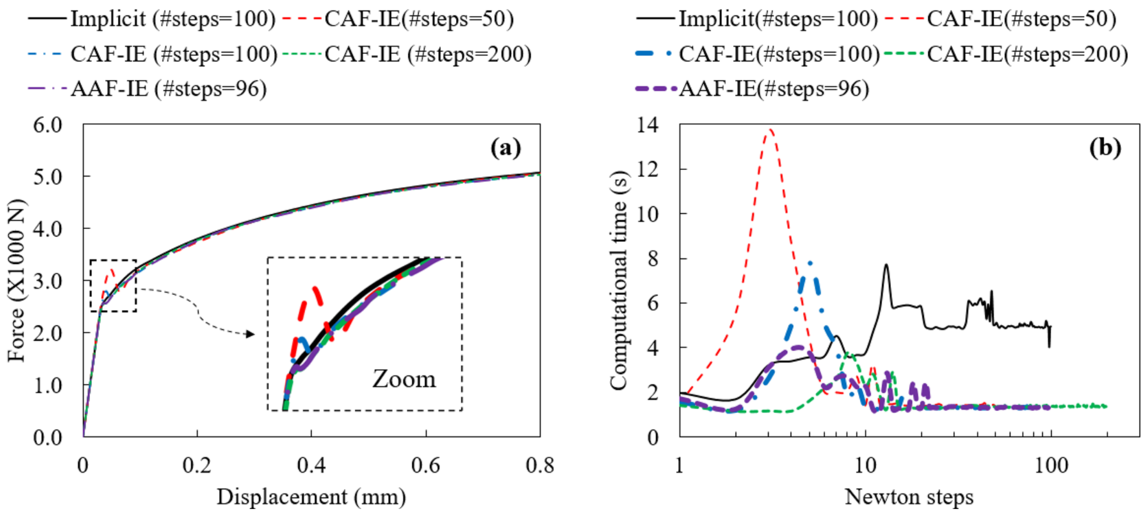

Fig. 7 Accuracy and efficiency assessment of IE schemes. **a** Comparison of load–displacement responses for a strain-hardening simulation between the implicit solver, CAF-IE with different constant steps, and AAF-IE. **b** The computational time of each Newton step for different schemes. A close-up comparison of CAF-IE schemes is shown without other solutions

equivalent plastic strain distribution is shown in Fig. 6c where the elements with the highest strain values are observed near the surface of the middle hole.

The AAF-IE's load–displacement response is demonstrated in Fig. 7a where the implicit solution is included as a benchmark. Since the current state variable (increment of plastic strain tensor) in an impl-exp scheme is linearly extrapolated from its previous values, see Sect. 3.1, prediction accuracy is affected by its step sizes at material transition. In particular, the impl-exp scheme's extrapolation errors are significant when using large steps in the transition. For example, in the constant-step assembly-free impl-exp (CAF-IE) schemes, decreasing the number of steps from 200 to 50 results in larger time steps and higher extrapolation errors, see Fig. 7a. On the contrary, the extrapolation error of our AAF-IE scheme is reasonably small even though we use much fewer time steps. This is because our adaptive schemes automatically adjust timestep sizes such that smaller steps are utilized in the transition of property phases while large steps are used at other non-critical times.

We note from Fig. 7a that during the material transition phase fewer constant steps of CAF-IE result in larger step sizes and, as a result, exhibit higher extrapolation errors (e.g., comparison of CAF-IE with 50 steps and 200 steps). With a larger extrapolation error, more iterations are required in CAF-IE towards convergence at each step which, in turn, results in higher computational time at each Newton step as shown in Fig. 7b. Comparatively, our AAF-IE adaptively reduces its step size to lower the required number of iterations

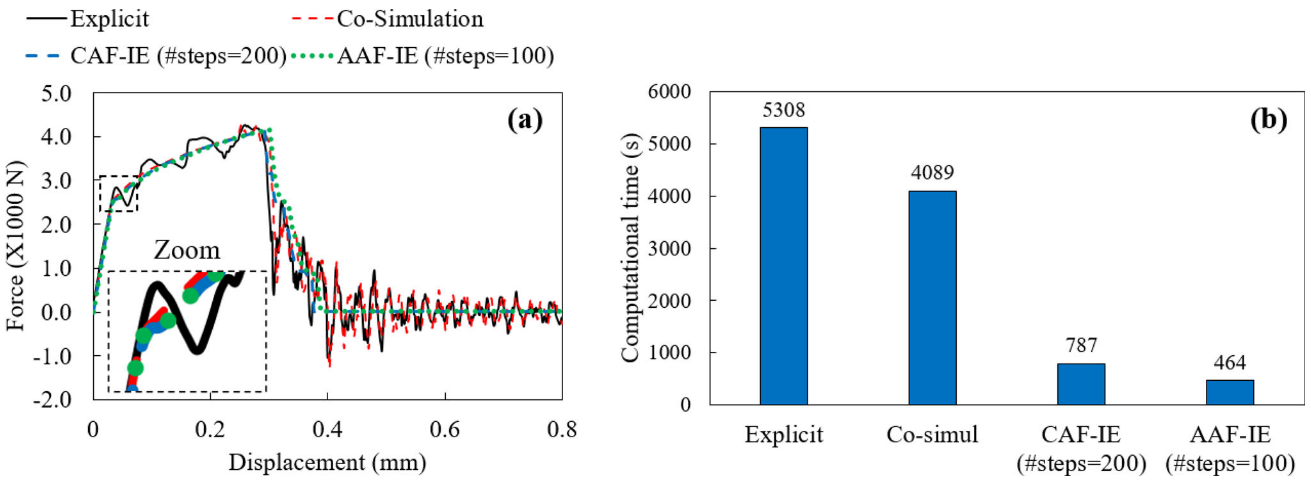

Fig. 8 Fracture simulations via different solvers. **a** Comparison of load–displacement responses between explicit solver, co-simulation scheme, IE solver with constant steps, and our AAF-IE solver. **b** Comparison of total computational time between different solvers

and computational time per step. After the transition phase, extrapolation errors are greatly reduced such that fewer iterations and smaller computational time are required at each step. Comparatively, for the implicit scheme, its computational cost per iteration is much higher than the other schemes because it requires the underlying stiffness matrices to be re-assembled at every iteration.

Note that the number of steps in CAF-IEs (i.e., 50, 100, 200 in Fig. 7b) are prescribed by the user where smaller numbers result in longer per-step computational costs since the steps are larger and involve more iterations. Meanwhile, the number of steps in AAF-IE (i.e., 96) is automatically determined by the algorithm to achieve the same level of solution accuracy as the CAF-IE with 200 steps, see Fig. 7a. The reason AAF-IE needs fewer steps than CAF-IE is that AAF-IE can adjust its temporal discretization where steps are enlarged during non-critical time instances and, as a result, the number of steps decreases.

We now demonstrate the performance of AAF-IE in the presence of softening by comparing it against two fracture simulations that use explicit and co-simulation schemes, see Fig. 8a. The explicit solver is conditionally stable but is quite slow since sufficiently small steps are required to ensure stability. The co-simulation approach applies the implicit and explicit solvers sequentially to simulate strain hardening and softening, respectively. It is more efficient than the pure explicit solver, but its efficiency depends on the fraction of deformation that includes softening.

The load–displacement curves obtained from different methods are illustrated in Fig. 8a where we observe that our AAF-IE and CAF-IE with 200 constant steps achieve very similar results. However, the solutions obtained via explicit and co-simulation methods indicate significant fluctuations in both hardening and softening. Reducing such erroneous fluctuations often requires applying fictitious damping forces which can result in unphysical solutions if they are excessive and implemented improperly. In addition, it is noteworthy that our AAF-IE (i.e., 100 steps) only requires half of the time steps as that of CAF-IE (i.e., 200 steps), indicating a dramatic temporal reduction.

We compare the total computational time of the four different solvers in Fig. 8b where we observe that CAF-IE and AAF-IE are significantly faster than the explicit and co-simulation procedures. This computational efficiency is primarily because both CAF-IE and AAF-IE are unconditionally stable and hence their enlarged steps reduce the overall simulation time. We also note that AAF-IE converges to the same solution as CAF-IE but with almost half of the steps and a 59% reduction in costs. Compared to the explicit solver widely applied in fracture mechanics, our AAF-IE scheme achieves a more accurate (smooth) solution with an acceleration factor of 11.4. We, therefore, apply AAF-IE to the following studies.

We prevent the softening solutions from mesh dependency by applying a non-local function (see Sect. 2.3) to the macroscale model in Fig. 6. Specifically, we assume the strain localization band has a width[3] of 1.5 mm and then test for mesh independency by discretizing the model with four mesh levels and comparing the resulting load–displacement responses. As shown in Fig. 9, the softening curves converge as the number of elements increases. With 13,000 elements, mesh independency is observed for both post-failure load–displacement responses and fracture energies (i.e., the area under the load–displacement curve after damage initiation).

We can further verify mesh independency by comparing the equivalent plastic strain distributions and damage patterns across the four mesh levels, see Fig. 10. As the number

[3] A more realistic softening bandwidth value is typically determined via experiments or high-fidelity simulations.

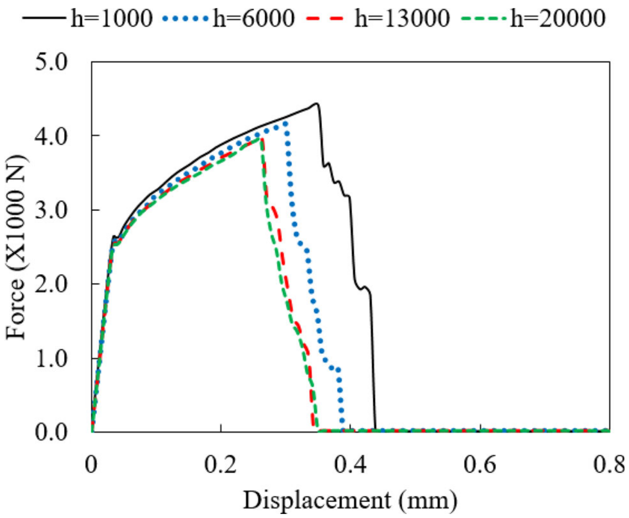

Fig. 9 Mesh independency study on history variables. Load–displacement responses converge as the number of elements (h) increases

of elements increases in Fig. 10a–d, we observe similar strain localizations. In particular, the direction of the softening band in each plot is orthogonal to the direction of the applied Dirichlet boundary condition. This band marks the damaged elements and extends from the inner hole to the outer surface. During damage evolution, the elements inside the band accumulate significant plastic strain and so their damage variables possess much higher values than their surrounding elements, see Fig. 10e–h. This observation explains the striking similarity between the strain localization bands (top row) with the fracture bands (bottom row) in Fig. 10.

It is noteworthy that in Fig. 10e, the mesh is so coarse that the element sizes are almost the same as the prescribed fracture band width. In this scenario, the non-local function in Eq. (14) has little effect and the fracture band contains one single layer of damaged elements. However, as discretization become increasingly finer in Fig. 10f–h, the non-local function achieves convergent softening solutions by averaging the damage variables of elements within the prescribed width of fracture bands. Across these cases, we also observe that multiple layers of damaged elements are contained within the fracture bands which indicates that the macroscale damage patterns are independent of mesh sizes and produce consistent fracture energies.

4.2 Microscale experiments

In this section, we integrate the temporal and spatial adaption techniques into our ROM whose performance is then tested on various porous microstructures. In all experiments, microscale pores are modeled as prolate ellipsoids with two identical minor axes. We use the following four descriptors to characterize the morphologies and spatial distribution of pores in a microstructure: pore volume fraction (V_f), number of pores (N_p), aspect ratio between major and minor axes (A_r), and the average spatial distance between the two nearest pores ($\overline{r}_d$, units in ¯m). In addition, we simulate the microscale damage evolutions via the stabilized microdamage model proposed in [31] (see also Appendix B).

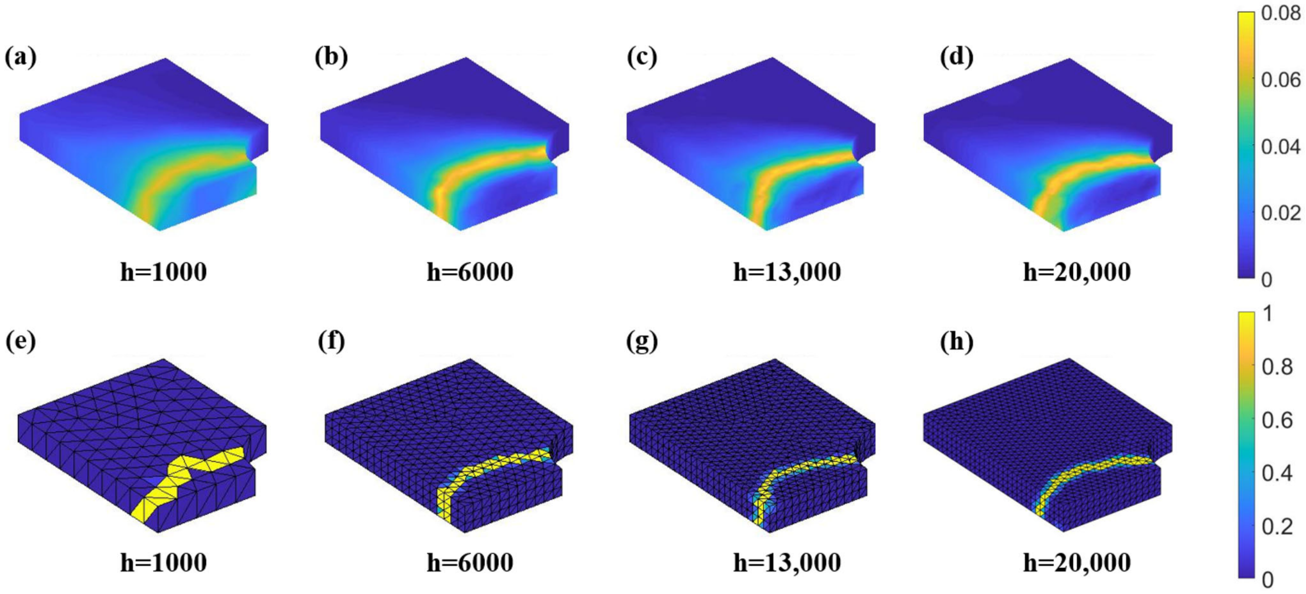

Fig. 10 Macroscale mesh independency study of field variables. **a**–**d** Distributions of equivalent plastic strains which converge as the number of elements increases. **e**–**f** Damaged elements and the distributions of fracture bands are independent of the mesh

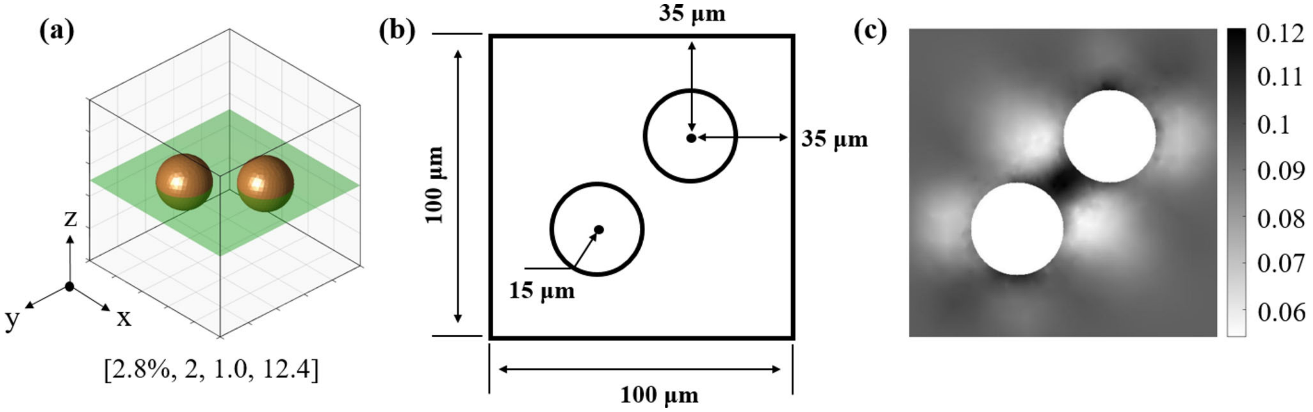

Fig. 11 A simple 3D microstructure with two pores. **a** The microstructure contains two identical spherical pores on the neutral z-plane. **b** Dimension of the microstructure (on the neutral z-plane). **c** Distribution of the equivalent plastic strain via DNS

4.2.1 Effect of different clustering methods

In this experiment, we apply our ROM to the microstructure in Fig. 11a which has two identical spherical pores located on the neutral z-plane. The dimensions of the microstructure's cross-section at the neutral z-plane are shown in Fig. 11b. Figure 11c shows the distribution of the equivalent plastic strain obtained via DNS (FEM with 33,000 elements) when this microstructure is subject to the macroscopic deformation gradient in Eq. (52).

$$\mathbf{F} = \begin{bmatrix} 1.1 & 0 & 0 \\ 0 & 0.95 & 0 \\ 0 & 0 & 0.95 \end{bmatrix} \tag{52}$$

We start by checking the sensitivity of the microscale damage solutions to mesh size in DNS. To this end, we discretize the microstructure by four different meshes which have 2000, 8000, 15,000, and 33,000 elements. By comparing their homogenized stress–strain responses in Fig. 12a, we find that the microscale mesh dependency is well controlled by using the fracture energy-constrained microscale damage model (see Sect. 2.3.2 and Appendix B). Specifically, as the number of elements increases, the effective responses converge to the solution with 33,000 elements. To emphasize the importance of the prescribed fracture energy on damage responses, we add two curves in Fig. 12 that correspond to a simulation with larger fracture energy ($G_f^* = 1.92e6\text{N/m}$), see Eq. (51) for G_f. As these curves indicate, the post-damage behavior and fracture energy are quite sensitive to the prescribed G_f. Particularly, a larger fracture energy enables material points to withstand higher external loads before fracture.

We now compare DNS with our ROM with the position-based clustering. The results are provided in Fig. 12b and indicate that as the number of clusters increases, the predicted ultimate tensile strength (UTS) and fracture toughness approach to those of DNS. Specifically, with 1600 clusters, ROM's homogenized response is identical to that of the DNS. It is noteworthy that the number of clusters represents the levels of spatial discretizations. A small cluster number generally results in coarse discretization with diffusive (low) plastic strain fields at each cluster. As the magnitude of the local plastic strain is smaller than DNS, ROMs exhibit delayed fracture initiations, higher UTS, and larger material toughness. This delayed damage response is gradually resolved as we increase cluster numbers. We also note that the ROM solutions are obtained by the microscale damage model (see Appendix B) which uses the same G_f as in DNS. Similar to above, the ROM with a larger fracture energy G_f^* is able to endure higher macro-strain before softening, and thus exhibit delayed damage behavior.

We next demonstrate the efficacy of the stress-informed clustering by preprocessing the same RVE with six offline orthogonal loads as discussed in Sect. 3.2.1. The resulting Von-Mises stresses are used to compute the elemental stress intensity scalar whose spatial distribution is demonstrated in Fig. 13a. Following our hierarchical approach, we divide the elements into three groups where each group has approximately the same number of elements, see Fig. 13b. We choose three different values for the total number of clusters (i.e., $k = 400, 800,$ and 1600) and then follow the procedures outlined in Sect. 3.2.1 to assign these clusters to the three groups. After hierarchical clustering (which is in the offline stage and done only once), we predict the fracture behavior in the online stage via our ROM. The results are summarized in Fig. 13c and indicate that, with the same number of clusters, stress-informed clustering provides higher accuracy than the naïve position-based clustering in Fig. 12b. Specifically, in Fig. 13c, we note that when the number of clusters increases to 800, the ROM's solution converges to that of DNS.

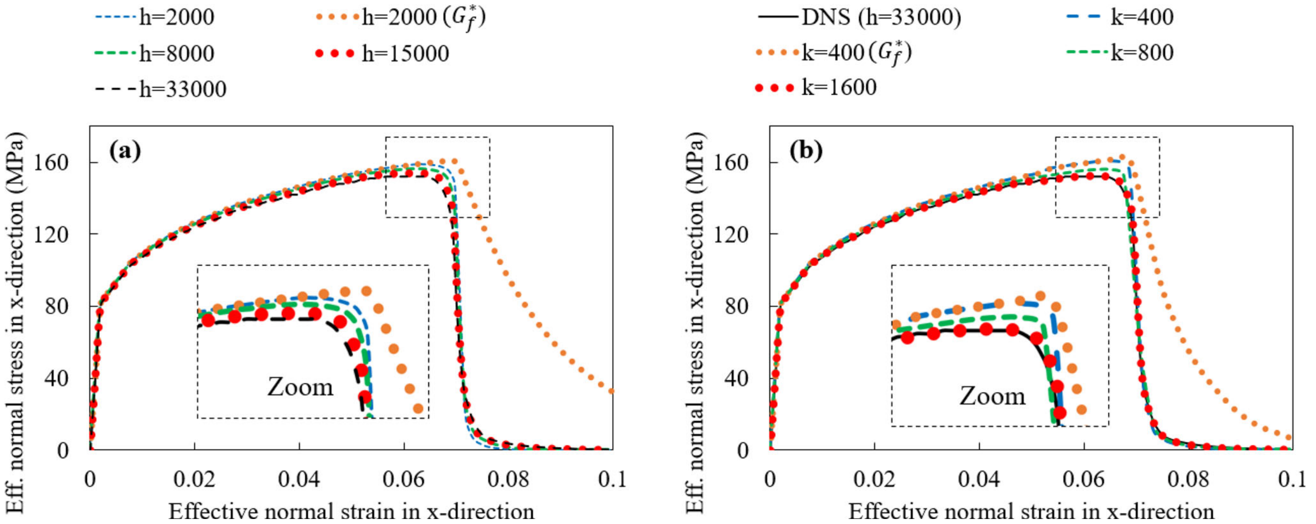

Fig. 12 Homogenized results by DNS and ROM. **a** Mesh independency study in DNS (h: number of elements) where the evolution of the damage variable is controlled by fracture energy G_f. Fracture behaves very distinctively when using G_f^*. **b** Results of ROM homogenizations via a position-based clustering (k: number of clusters) where the impacts of G_f is demonstrated by providing the curve corresponding to G_f^* for k = 400

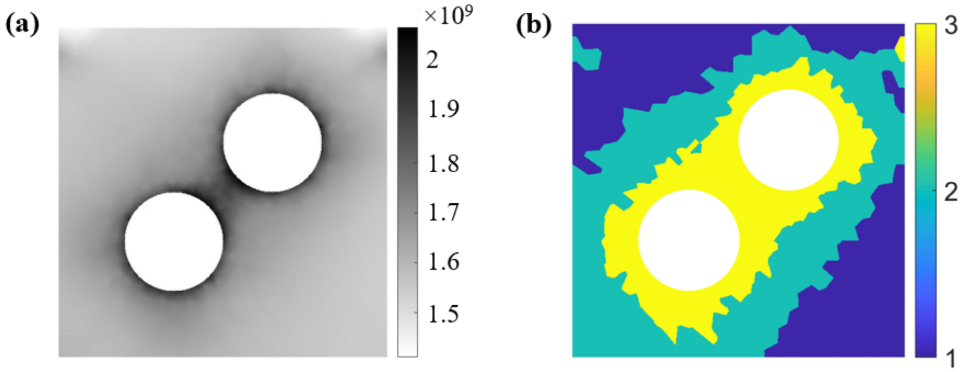

Fig. 13 Stress-informed clustering. **a** Distribution of stress intensities which is computed from the six orthogonal offline loadings. **b** Elements are agglomerated into three groups according to their stress intensities where each group has approximately the same number of elements. Both figures are plotted on the neutral z-plane of the 3D RVE in Fig. 11a

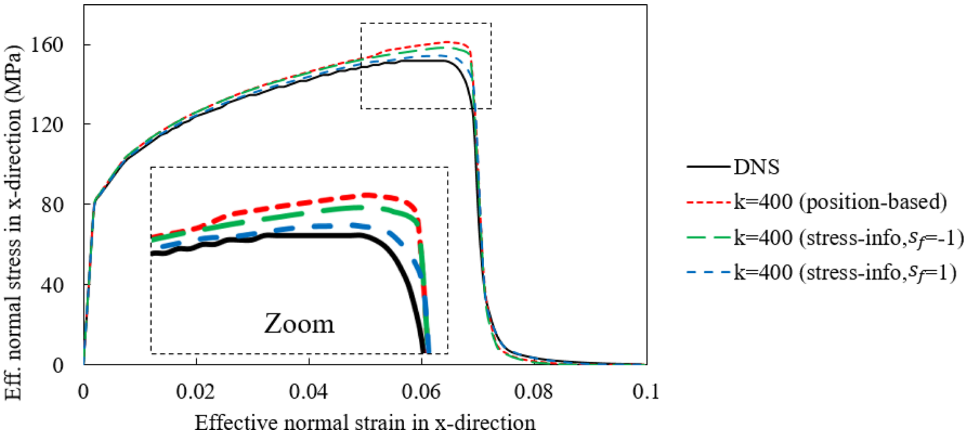

Fig. 14 Effective responses of stress-informed clustering. Compared to position-based clustering results, stress-informed solutions are improved using the same number of clusters (k = 400). A higher value of clustering split factor in the stress-informed method results in more accurate solutions

The clustering split factor in Eq. (40) is an important parameter that affects the results of our hierarchical clustering. To demonstrate its impacts, we compare the solutions of ROMs with the same 400 clusters but different values of s_f in Fig. 14. As discussed in Sect. 3.2.1, a positive s_f promotes more clusters in the groups with high stress intensities, while a negative s_f works oppositely. For instance, $s_f = 1.0$ results in 67, 133, and 200 clusters in groups 1, 2, and 3, respectively, while $s_f = -1.0$ genrates 219, 109, and 72 clusters in those groups. Comparing the effective results from the two s_f, we observe a significant improvement when assigning more clusters in critical regions ($s_f = 1.0$). Therefore, we choose the default value of the heuristic split factor as 1.0 in our following tests.

To further distinguish the position-based and static stress-informed clusterings, we compare their cluster distributions and equivalent plastic strains in Fig. 15 where the RVE is decomposed by 400 and 1600 clusters. Compared to the position-based method, the stress-informed approach allocates more clusters to the highly-stressed regions (around the bridge region connecting the two pores) and fewer clusters to the non-critical regions, compare the top row in Fig. 15a, c with Fig. 15b, d. As a result of such distributions, the stress-informed method produces large (small) clusters in non-critical (stressed) regions which increase solution accuracy as small clusters enable capturing high gradients. This improvement is illustrated in the bottom row of Fig. 15 where stress-informed clustering captures strain concentrations better than the position-based counterpart. We note that, while the results based on these two clustering methods differ, they are both quite close to DNS results (e.g., compare the equivalent plastic strains in Fig. 15 with DNS results in Fig. 11c).

We now compare the performance of the above two clustering methods with the online dynamic approach (see Sect. 3.2.2) which aims to refine clustering as an RVE is deformed and softening appears. The first step of the dynamic approach is to determine the number of new clusters per time step. To this end, we use our error-control adaptive scheme for new cluster generations relying on the AAF-IE's error metric in Eq. (34), and we compare its performance against a naïve approach where the same number of new clusters are created per step. We set the initial and maximum numbers of clusters as 400 and 800, respectively. Both approaches use position-based clustering to create the initial 400 clusters but the additional 400 clusters are generated either by the error-controlled scheme or uniformly at each step.

In Fig. 16a, we illustrate the adaptive Newton step sizes on the left axis and find that they are reduced to their minimum at the fifth step when material properties transit from elasticity to plasticity. The steps gradually grow and plateau when most material points enter the plastic regimes. The number of new clusters added in each step is recorded on the right axis in Fig. 16a and indicates that, while the simple approach adds 8 clusters in each step (a total of 400 additional clusters), the adaptive scheme adds all the new clusters during the first eight steps when phase transitions occur.

We compare the effective strain–stress responses between the two approaches in Fig. 16b where it is clear that the adaptive scheme provides higher accuracy. The primary reason behind this result is that in history-dependent deformations the adaptive scheme reduces the errors that appear early in the simulations (due to insufficient discretizations) and are accumulated as the deformation progresses.

To further improve accuracy, we now use the fully adaptive strategy which leverages both the offline stress-informed clustering and the online dynamic clustering. As illustrated in Fig. 16b the solution of the fully adaptive clustering is more accurate than the previous two cases (all approaches have the same total number of clusters) and matches the solution obtained via DNS.

Hereafter, for brevity, we refer to the clusterings without error control, with error control, and with full adaption as clustering type-1, 2, and 3, respectively. In the top row of Fig. 17, we plot the spatial distributions of the cluster centroids across the three methods. We observe that the new clusters from type-2 and especially type-3 methods are quite compact and largely located near pore surfaces. This behavior is because when material properties change abruptly during phase transition, AAF-IE adaptively reduces time steps and generates more clusters at each step. The new clusters are created around pore surfaces where high solution gradients appear. As shown in the bottom row of Fig. 17, this compact clustering behavior improves the accuracy in capturing concentrations.

We analyze the computational costs of type-3 (full adaption) clustering in Fig. 18a where the offline and online stages consume 54.9 and 291.2 s, respectively. The online step time increases in early steps due to phase transition and then it drops as materials are yielded. The step time rises again in late steps due to damage evolutions where expensive microscopic damage models are activated to map effective responses between the reference microstructures used in damage modeling, see Appendix B.

Finally, we compare the online computational costs of different clustering approaches against the clock time of DNS in Fig. 18b. We note that as the clustering types advance from the position-based method to offline stress-informed and fully-adaptive approaches, the required number of clusters for matching DNS decreases. In addition, compared to DNS which relies on the pure implicit scheme with re-assembling stiffness matrix at each iteration, our adaptive ROM with temporal reductions (by AAF-IE) and spatial reductions (via adaptive clustering) needs less than 3% computational time.

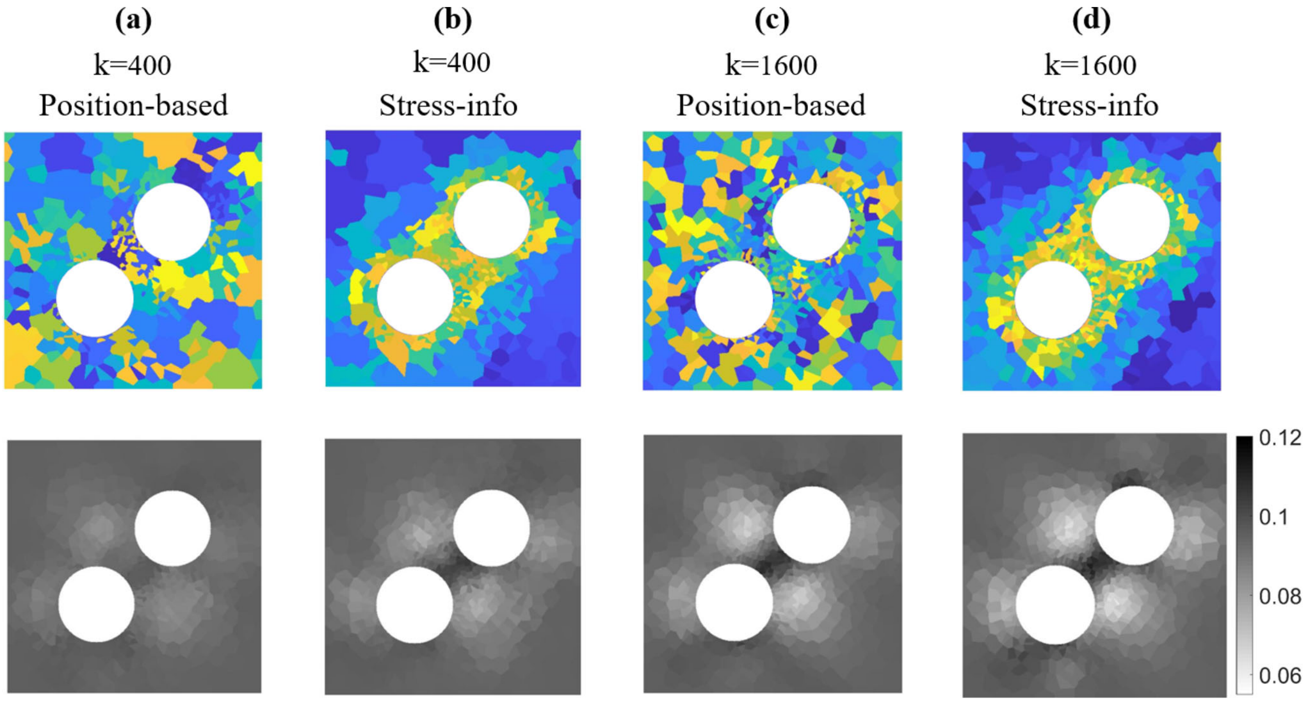

Fig. 15 Comparisons of clustering and effective plastic strains. The top and bottom rows indicate, respectively, the cluster distribution and the equivalent plastic strain field on the neutral z-plane

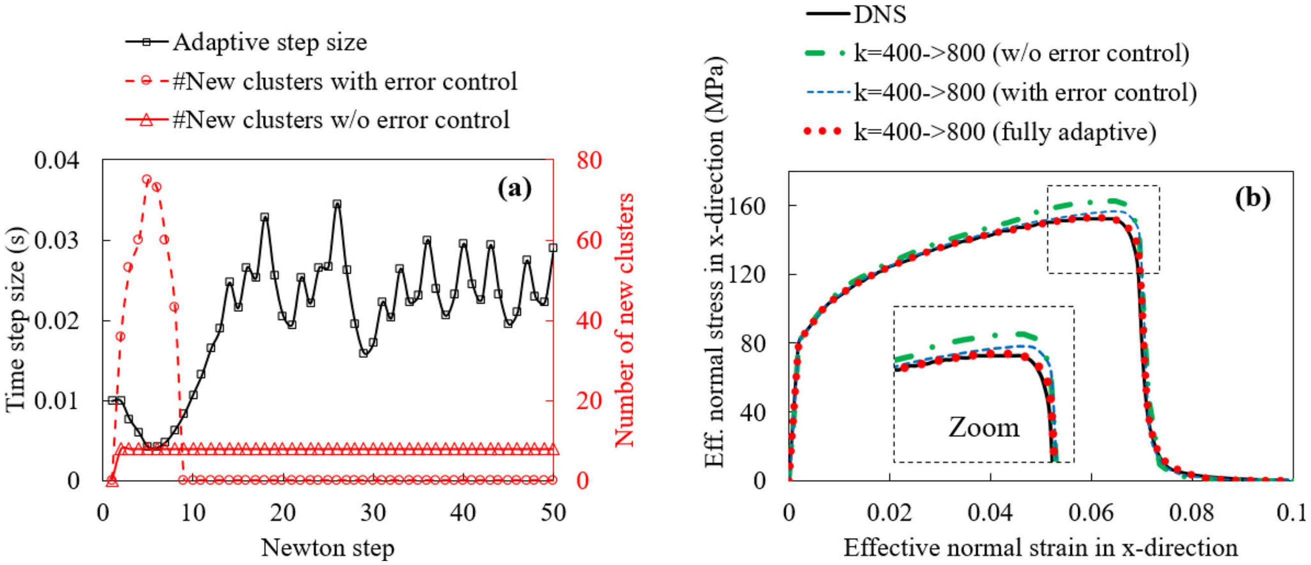

Fig. 16 Adaptive clustering with the error-control scheme. **a** Adaptive Newton step sizes and the number of new clusters per step. **b** Accuracy improvements of the error-control adaptive schemes on homogenization results where the fully adaptive approach is the combination of offline stress-informed clustering with an online error-controlled adaptive counterpart

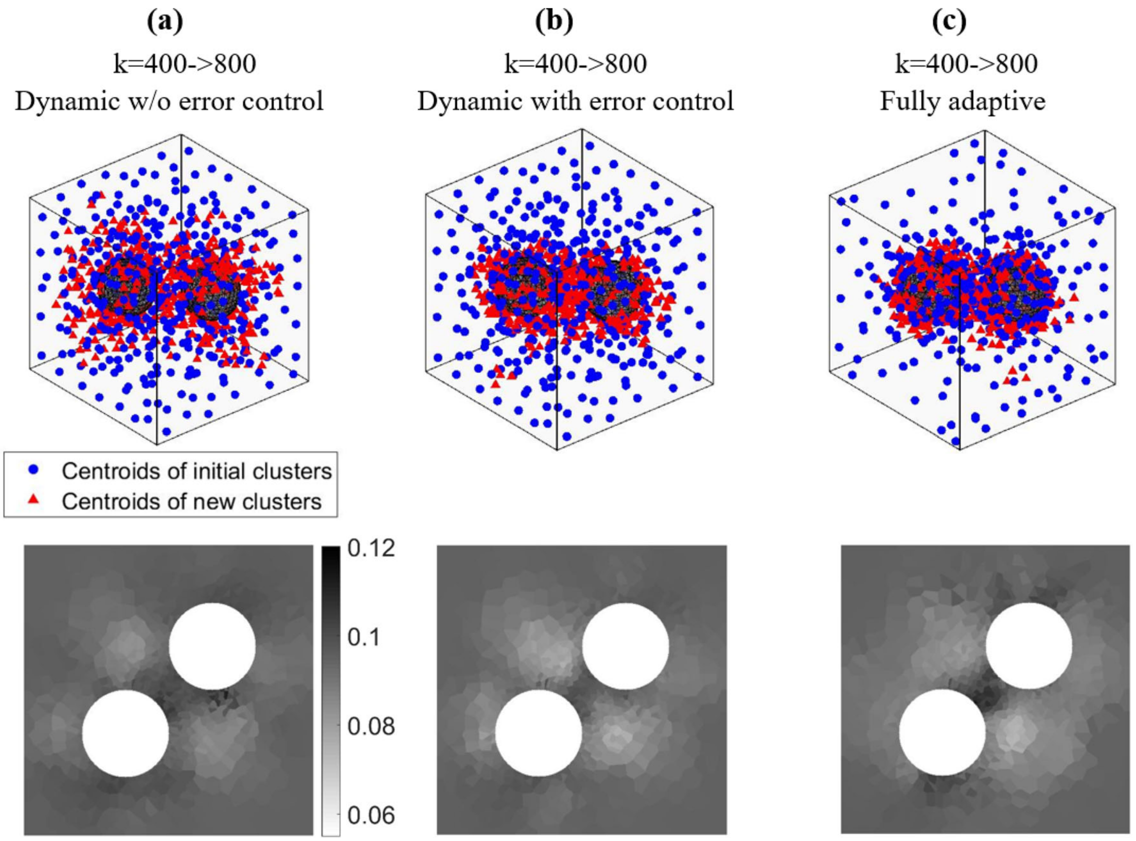

Fig. 17 Different clustering approaches and effective plastic strain distributions. The top row indicates the spatial positions of the initial (offline) and dynamic (online) clusters across the three different clustering approaches. The plots in the bottom row illustrate the distribution of the effective plastic strains on the neutral z-plane for each of the approaches

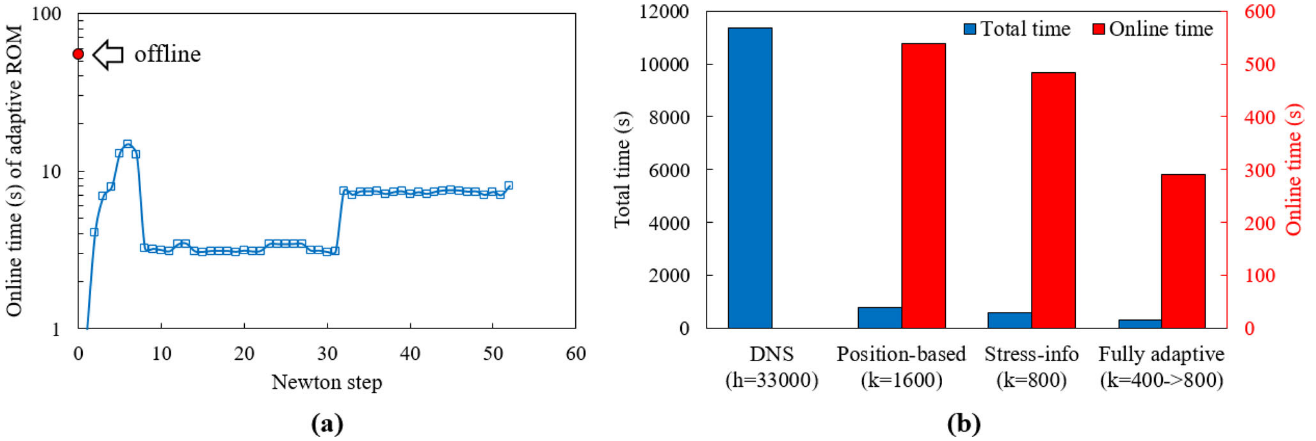

Fig. 18 Time analyses on the fully adaptive method. **a** Online solution time per step of the fully adaptive ROM. **b** Time comparisons between DNS and ROMs (h: number of elements, k: number of clusters) where different clustering approaches converge to the same DNS solutions

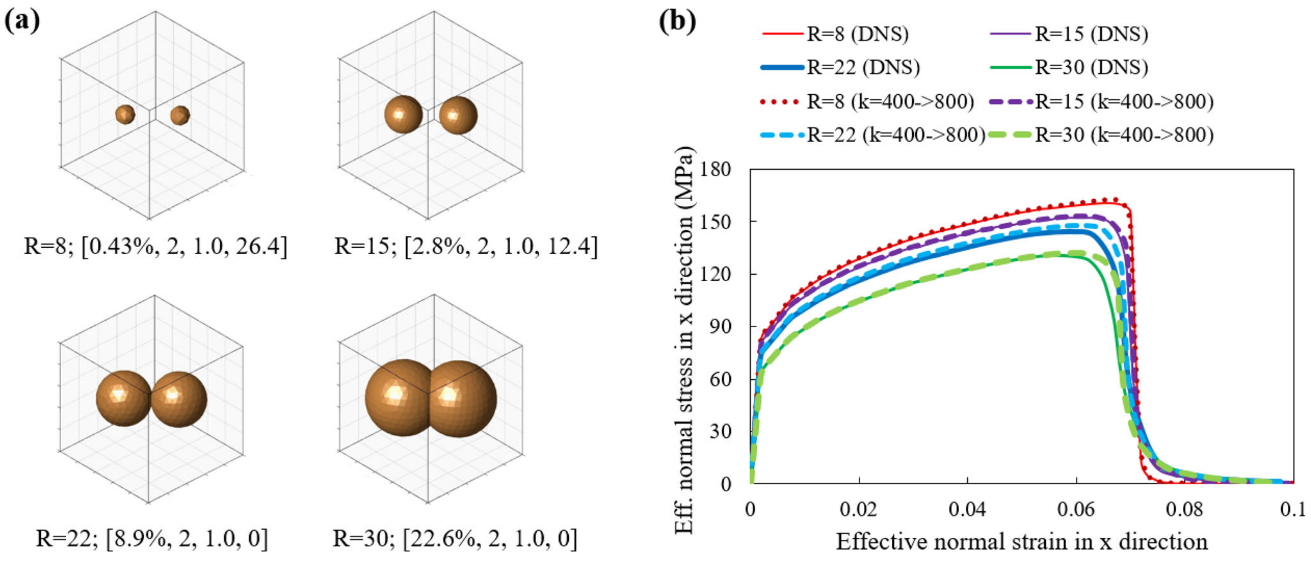

Fig. 19 Microstructures with different porosity volume fractions. **a** Morphologies of the microstructures with distinct pore radii (R) and descriptors $[V_f, N_p, A_r, \bar{r}_d]$. **b** Homogenized stress and strain curves

Table 5 Accuracy analyses. Errors of ROMs' toughness and UTS as compared to DNS

Pore radii (μm)	Error of toughness (%)	Error of UTS (%)
8	0.34	1.24
15	0.71	0.65
22	2.27	2.02
30	2.19	0.76

4.2.2 Effects of porosity volume fraction

Porosity volume fraction significantly affects alloys' failure behaviors [46]. To investigate its influence, we keep the positions and shapes of the two pores in the previous subsection identical but change their radii to 8 μm, 15 μm, 22 μm, and 30 μm. As shown in Fig. 19a, these four cases correspond to microstructures with porosity volume fractions of 0.43%, 2.8%, 8.9%, and 22.6%, respectively.

We deform these four RVEs using the same macro-deformation gradient in Eq. (52) and illustrate their effective stress–strain responses in Fig. 19b. We observe that with the increase of porosity volume fractions, the values of both toughness and UTS are significantly reduced. Specifically, the toughness and UTS of the microstructure with 22.6% porosity are only 79.2% and 80.9% of the one with 0.43% porosity. In addition, by comparing our fully adaptive ROM against DNS, we find our ROM achieves high fidelity with solution errors smaller than 3%, as listed in Table 5.

4.2.3 Microstructure with complex morphology

In this section, we test the performance of our ROM using the complex RVE in Fig. 20a which is subject to the deformation gradient in Eq. (52). The distributions of the equivalent plastic strain obtained by DNS and ROM with fully adaptive clustering are compared in Fig. 20b–e. In this simulation, the ROM is accelerated by both temporal and spatial adaption techniques. We see that ROM's solutions (strain distributions and localizations) converge to DNS as more clusters are used. However, compared to the RVE in Fig. 11a which has relatively simple pore geometries, more clusters are needed in this test to capture local porosity morphology to ensure ROM's solution converges to that of DNS.

The homogenized responses are compared between DNS and our ROM in Fig. 21a. When the initial number of clusters increases to 3200, the ROM's stress–strain curve matches DNS in terms of both UTS and fracture toughness. To further investigate the discrepancy between DNS and ROM, we plot an error histogram in Fig. 21b which measures the difference in the effective plastic strains between DNS and ROM at each element. It is observed that compared to fewer clusters (e.g., k = 200 → 300), the ROMs with more clusters have lower prediction errors. Predicted toughness values of DNS and ROM are compared in Table 6 where we see that the error drops from 1.61 to 0.56% as the total cluster numbers increase from 300 to 3300.

Finally, we compare the computational costs of DNS and ROMs in Fig. 22. We note that the offline costs are excluded from the time analyses since they are only performed once and their results can be reused for any other deformation. We add that with the highest adaptive cluster numbers (k

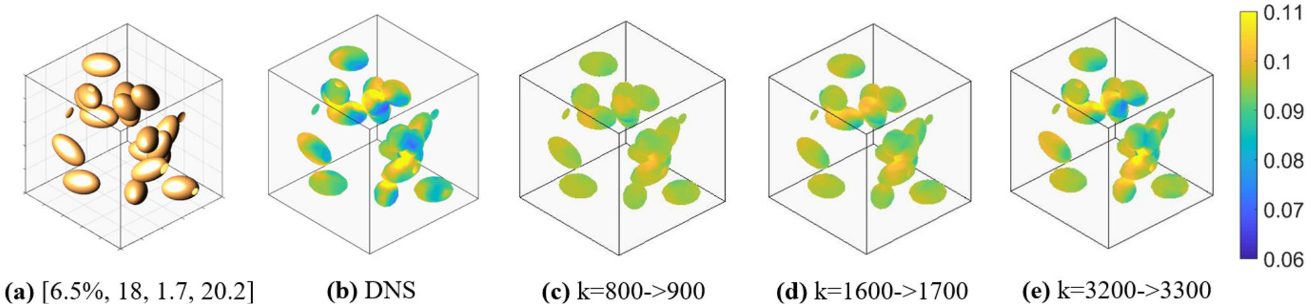

Fig. 20 Microstructure with complex pore morphology. **a** The studied microstructure and its pore descriptors $[V_f,\ N_p,\ A_r,\ \overline{r}_d]$. **b** Distribution of equivalent plastic strains emulated via DNS. **c–e** Distributions of equivalent plastic strains are simulated by adaptive ROMs with different numbers of clusters where the first and second numbers indicate the initial and total number of clusters, respectively

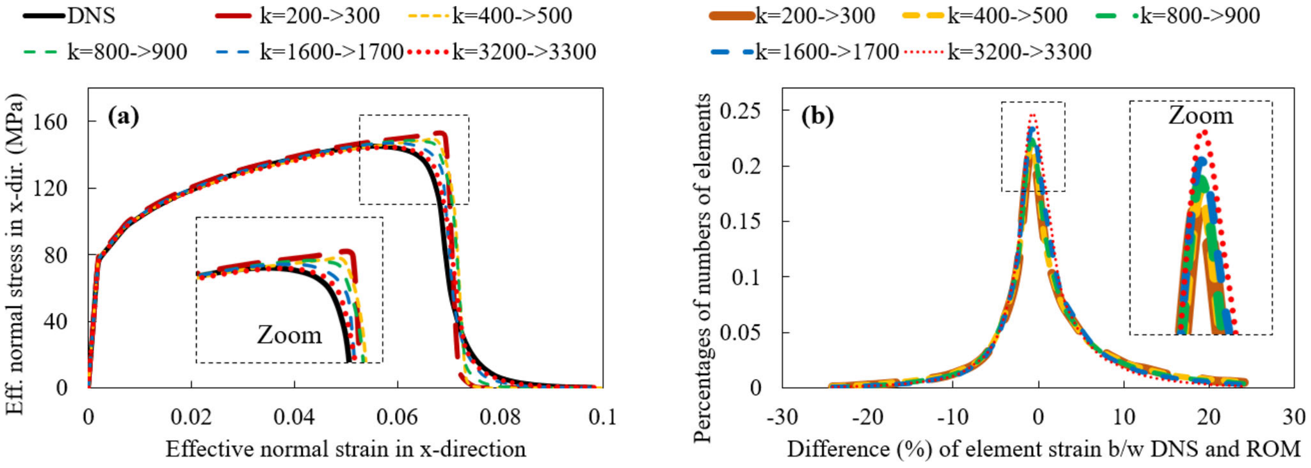

Fig. 21 Accuracy analyses on effective stresses. **a** Comparisons of homogenization results between DNS and adaptive ROMs. **b** Element-to-element strain comparisons between DNS and ROMs

Table 6 Accuracy analyses on toughness. Comparison of material toughness simulated by DNS and ROMs for the studied microstructure in Figure 20a

Method	Toughness (MJ/mm^3)	Error (%)
DNS	8.89	–
ROM (k = 200→300)	9.04	1.61
ROM (k = 400→500)	9.01	1.29
ROM (k = 800→900)	9.02	1.42
ROM (k = 1600→1700)	8.95	0.61
ROM (k = 3200→3300)	8.84	0.56

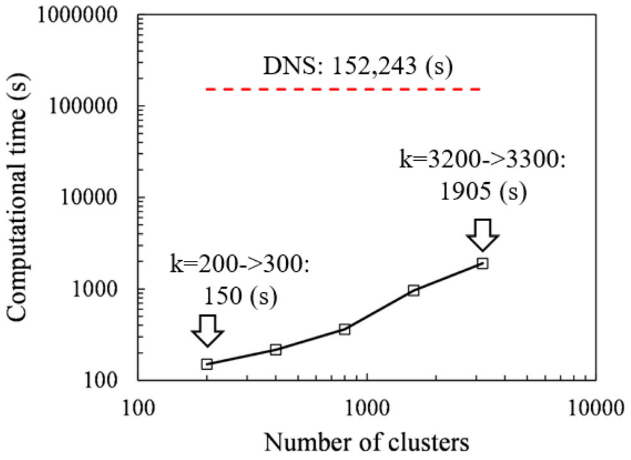

Fig. 22 Time comparisons between DNS and our fully adaptive ROMs. Computational time is compared between DNS and ROMs with different numbers of adaptive clusters for the microstructure in Fig. 20

= 3200→3300), the ROM's online time only accounts for about 1.3% of that of DNS; indicating an acceleration factor of 79.9 in this example.

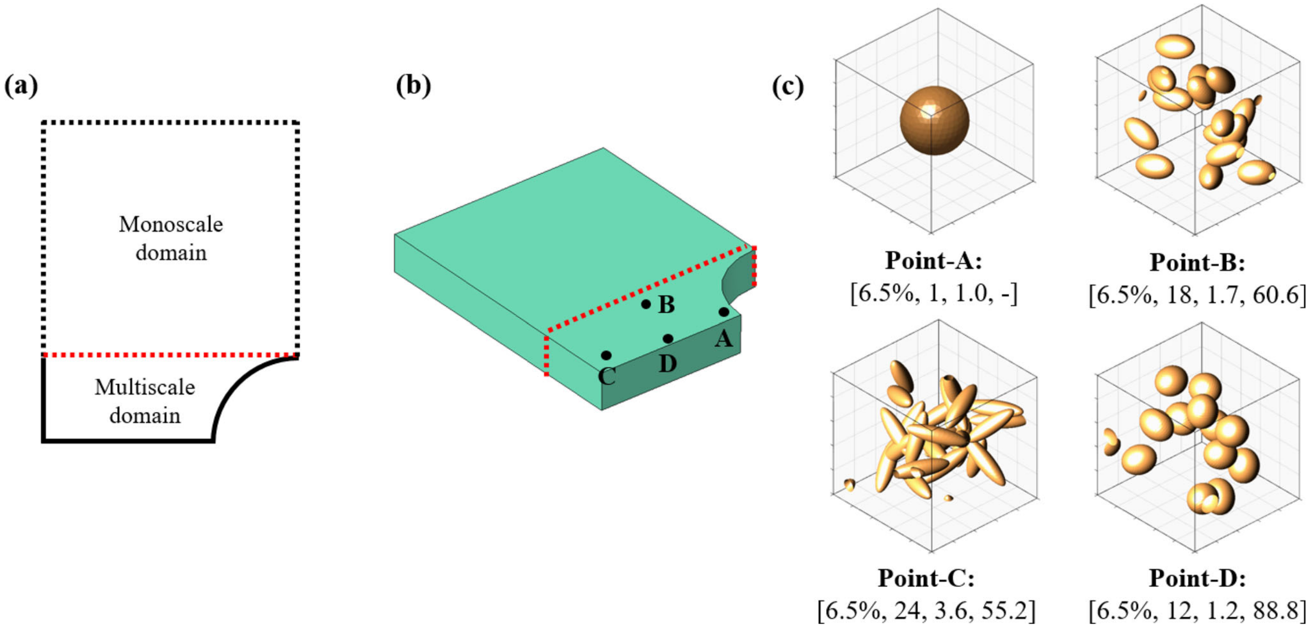

Fig. 23 Multiscale model. **a** The macroscale component comprises a monoscale region without any porosity and a multiscale region containing micro-pores. **b** All integration points are assigned with porous microstructures in the multiscale region with four highlighted points. **c** The four points are assigned with different microstructures, respectively, where their pore descriptors are listed in the vectors $[V_f, N_p, A_r, \bar{r}_d]$

4.3 Multiscale experiment

In this section, we integrate the temporal and spatial adaptation schemes with a FOCH-based concurrent multiscale framework to investigate the influence of micro-pores on the damage behavior of a macro component. Our multiscale study is performed on the same 3D plate model in Fig. 6 which is now considered the macro-structure, see Fig. 23a. We assume the same Dirichlet boundary conditions are applied on the plate as in Fig. 6b. We also presume the plate consists of a monoscale region without any porosity and a multiscale region with microscopic pores, see Fig. 23b. To avoid softening-induced mesh sensitivity, we follow the mesh dependency study in Sect. 4.1 by discretizing the plate with 13,000 linear tetrahedral elements and applying the non-local function with the fracture bandwidth of 1.5 mm. Under this discretization, the mono- and multiscale regions are meshed by 10,560 and 2440 elements, respectively.

To model local morphologies, we associate all macro-IPs in the multiscale region with spatially varying porous microstructures. We highlight four IPs in the multiscale region whose corresponding microstructures are visualized in Fig. 23c. Although the four RVEs share the same porosity volume fraction of 6.5%, their local morphologies are significantly different and vary from having one spherical pore to having multiple randomly dispersed overlapping pores. These four types of RVEs are randomly assigned to the macro-IPs. To ensure consistent fracture energies across scales [19], we enforce the sizes of RVEs to be the same as the sizes of macroscale elements.

We mesh the microstructures of our multiscale model based on the complexity of their morphologies. Specifically, we mesh the four microstructures associated with macro-points A, B, C, and D with 15,000, 78,000, 103,300, and 60,400 elements, respectively. Correspondingly, our multiscale model is discretized with a total of roughly 156 million elements. Since the computational cost of simulating such a model via DNS (FE^2) is prohibitively high, we only utilize our ROM where we assign 300 adaptive clusters to the microstructure with a single spherical pore and 900 clusters to all other microstructures.

The computational time of our concurrent multiscale simulation is about 59.4 h. Based on the time comparison between DNS and ROMs in Sects. 4.1 and 4.2, the estimated time for DNS (FE^2) is more than 52,519.9 h (2188.3 days) given the same computational resources. That is, our method achieves an acceleration factor of 884.2 in this multiscale study.

We demonstrate the multiscale model's load–displacement response in Fig. 24a where the plate's ultimate load carrying capacity is 3659.1 N at the displacement of 0.24 mm, which is an 8.5% decline from its dense counterpart in Fig. 9 where the plate ruptured at the displacement of 0.34 mm. We observe a localized fracture band that lies in the multiscale

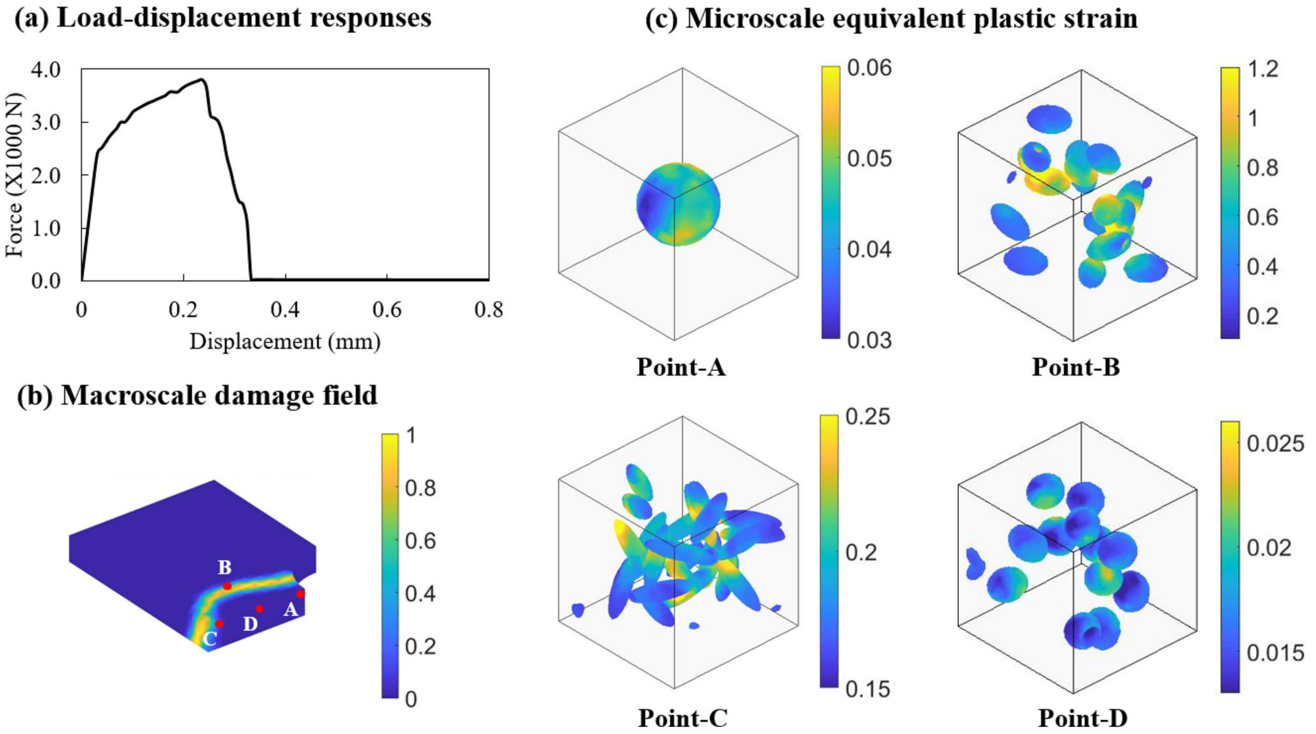

Fig. 24 Results of the adaptive ROM. **a** Macroscale load–displacement responses. **b** Macroscale damage patterns with a fully developed fracture band connecting mid-hole to outer surfaces. **c** Microscale distributions of the equivalent plastic strains are associated with the four integration points on the macroscale model

region and is oriented orthogonal to the Dirichlet boundaries in Fig. 24b. Two of the highlighted material points (B and C) are close to the fracture band while the other two points (A and D) are distant from the softening regions. The microscale equivalent plastic strain fields in the corresponding four RVEs are demonstrated in Fig. 24c where we observe that microstructures B and C have accumulated much larger strains (note the axis scales). In each of these RVEs, we notice that high plastic strains concentrate in regions where pores are closely packed.

5 Conclusion

We introduce adaptive deflated clustering analysis in this paper which is a novel ROM that can simulate the damage behavior of metallic alloys with process-induced spatially varying microscopic pores. Our method is comprised of two new adaption strategies that are readily extensible for any clustering-based ROMs: a novel AAF-IE temporal adaption scheme and a new spatial clustering approach. Our AAF-IE not only alleviates softening materials' instabilities by preserving the positive definiteness of the underlying algebraic system's stiffness matrices, but also improves the overall efficiency by avoiding stiffness matrices' reassembly during the online process. In addition, our adaptive spatial discretization approach significantly improves the ROM's prediction accuracy over critical regions by selectively modifying local interpolations. Specifically, the new clustering approach automatically adjusts cluster densities according to online solutions and resizes temporal steps to allocate more computational resources at crucial time instances. In our ROM we use a macroscale non-local function and a microscopic energy-based damage formulation to resolve softening-induced pathological mesh dependencies and ensure fracture energy consistency across scales.

We test the performance of our adaptive ROM via several numerical experiments. Our ROM results show significant improvements in robustness, efficiency, and local accuracy compared to DNS. We also apply the proposed method to a concurrent multiscale model to simulate the multiscale softening responses and indicate that components with spatially varying porosity defects have lower ultimate load-carrying capacity than their dense counterparts.

We note that There are some fundamental differences between our ROM with existing techniques. As an example, contrary to SCA which groups elements based on their mechanical responses, our ROM agglomerates elements based on their geometrical proximity. Another major difference is our novel hybrid integration scheme that avoids softening-induced solution divergences. Additionally, SCA is typically applied to composite or polymer materials [14, 15] with strong or weak inclusions where the property ratio between material phases (e.g., moduli) is reasonably small. Our ROM is used to simulate alloys with pores where the moduli difference between material and void is infinite. Compared with strong or weak inclusions, it is much harder to simulate microstructures with pores (especially for an FFT-based approaches whose computational efficiency significantly deteriorates due to the infinite property contrast between material phases).

Given the promising results from this work, our ROM can be extended to finite strains and extreme load conditions in the future. Another research direction of interest is to use our ROM as an efficient database generator for complex material responses, followed by deep learning approaches [47–50] to directly correlate materials' local morphologies with their responses. It is also noted that in this work the plasticity behavior of the metallic matrix is based on the J2 plasticity model as opposed to, e.g., crystal plasticity [51–53]. Adding crystal plasticity to our multiscale model is an interesting idea which we plan to explore in our future works.

Acknowledgements The authors thank the ACRC consortium members. Specifically, we appreciate Randy Beals from Magna International, and Chen Dai from VJ Technologies for providing us with the W-profile plate samples and X-ray computed tomography scanning and data generation, respectively. Ramin Bostanabad also acknowledges NSF funding (award numbers OAC-2211908 and OAC-2103708).

Appendix A. Deflated clustering analysis

A computationally efficient reduced-order multiscale model, named deflated clustering analysis (DCA), is proposed in [17] to simulate the elastoplastic behaviors of metallic alloys with process-induced, spatially varying porosity defects. DCA relies on the clustering-based domain decomposition that universally applies to multiscale domains to accelerate the accurate computations of macroscale deformations and effective microscopic responses.

Compared to the classic FE^2 approach, which solves equilibrium equations on both macro- and micro-scales via computationally expensive FEM, the numerical advantage of DCA is to accelerate emulations by systematically agglomerating neighboring materials points to clusters and projecting solution variables into clustering-based deflation spaces for efficient nonlinear simulations, see Fig. 25. While both macro- and micro- computations follow general FEM formulation with displacements as unknown variables, the acceleration mechanisms on macro- and micro-domains are different.

Macroscale clustering is integrated with the deflation method to accelerate the conjugate gradient (CG) process of the underlying algebraic systems at each Newton iteration. The deflation method aims to remove the near-zero eigenvalues from the algebraic system's stiffness matrices, and it simultaneously lowers their conditional number which, in turn, increases the computational efficiency substantially. An incremental deflation method is also utilized to avoid any unnecessary re-assembly of the deflation matrix during runtime. FEM solutions are recovered from the deflation space per CG iteration, and their convergence errors are checked based on prescribed tolerance. By enforcing the same CG convergence criteria between FEM and deflation method, the macroscale ROM converges to the same displacement results as those of FEM without losing solution accuracy. Therefore, the local deformation gradients at each macroscale integration point are highly accurate for the sequent microscale analyses.

For microscale, clustering aims to generate coarse computational grids, similar to the coarse-graining process, where node-to-node interactions of FEM are condensed to the cluster interactions in the reduced cluster-based mesh. For the microscale equilibrium equations, utilization of more clusters results in a larger algebraic system with more eigenmodes and sophisticated deformations but with the expense of higher degrees of freedom and longer simulation time. In essence, the microscale clustering analysis targets to provide homogenized responses at macro-integration points by coarse-graining responses from micro-clusters. The efficiency of the microscopic ROM is much improved than FEM since degrees of freedom are dramatically reduced from large numbers of elements to a few clusters. Although it loses local accuracy as the solution variables are assumed uniform within each cluster, it is sufficient for the sake of homogenizations.

Detailed steps of applying DCA in microscale simulations are summarized in Algorithm A.1. We note that following the construction of a clustering-based reduced mesh and stiffness matrix, microstructural local stress and strain fields are averaged at the centroid of each cluster. In other words, instead of computing distinct solution fields at different elements, close-by material points (in one cluster) are assumed to share the same solutions. We also stress that DCA in [17] is dependent on naïve position-based clustering, and its accuracy on localized solutions can be improved by the proposed adaptive clustering technique in this work.

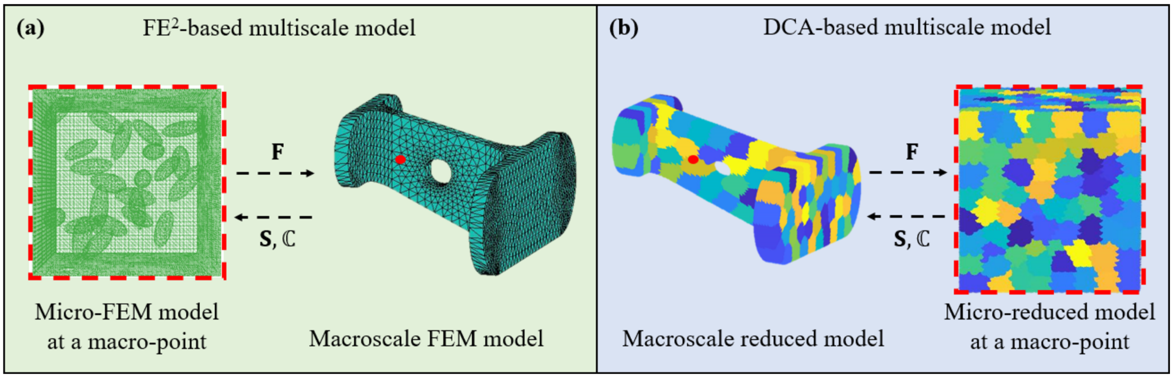

Fig. 25 Comparison of the classic multiscale method with DCA. **a** Classic FE^2-based multiscale model assumes each macroscale material point is associated with a microstructure where the deformation gradients (**F**) are passed from the macro- to micro-domains, and the effective stress (**S**) and moduli (ℂ) are passed backward. **b** DCA speeds up nonlinear simulations on multiple scales via clustering-based data compression to systematically reduce unknown variables where different clusters are marked by distinct colors

Algorithm A.1 Structure of the microscale cluster-based ROM

```
procedure Solve for the homogenized microstructural responses at the $i^{th}$ nonlinear increment
    ▷Initialization
    if $i \leftarrow 1$, then
        Generate FE mesh on the microstructure
        Compute the microstructure geometric center $x_{center}$ from nodal coordinates $x$
        Create node-based clusters via domain decomposition
        Construct cluster-based reduced mesh
        Initialize material properties with elastic material properties
        Develop reduced stiffness matrix on cluster-based mesh
    else
        Update material properties from the last increment
        Update the cluster-based constitutive model
    end if
    Read macroscopic deformation gradient $\mathbf{F}^{\mathrm{i}}(\mathbf{X})$
    Compute incremental homogeneous displacements on FE mesh: $\mathbf{u}_0^{\mathrm{i}}(\mathbf{x}) = (\mathbf{F}^{\mathrm{i}}(\mathbf{X}) - \mathbf{I}) \cdot (\mathbf{x} - \mathbf{x}_0)$
    Project the homogeneous displacements from FE mesh to cluster-based mesh
    ▷Starting Newton iterations to solve micro-equilibrium equations on the reduced mesh
    Set Newton iteration convergence criteria $\epsilon = 10^{-3}$
    for $\mathrm{j} \leftarrow 0, N$ do        ▷Loop over $N$ Newton iterations
        Compute the microscale internal force vector $\boldsymbol{f}_j^{int}(\boldsymbol{u}_j^i(\boldsymbol{x}))$
        Solve micro-equilibrium for microscopic displacement fluctuation $\tilde{\boldsymbol{u}}_j(\boldsymbol{x})$
        Update microscale displacements: $\boldsymbol{u}_{j+1}^i(\boldsymbol{x}) = \boldsymbol{u}_j^i(\boldsymbol{x}) + \tilde{\boldsymbol{u}}_j(\boldsymbol{x})$
        Postprocess for microstructural stress and strain fields
        ▷check for convergence
        if iteration residual $< \epsilon$ , then
            Compute homogenized stress tensor and material modulus
            return
        end if
    end for
end procedure
```

Appendix B. Stabilized micro-damage model

Strain softening causes severe convergence difficulty to implicit integration procedures, e.g., the Newton–Raphson method, see Sect. 2.4. Although our impl-exp integration scheme can successfully resolve softening-induced instability, the global stiffness matrix of the underlying algebraic system still needs to update the entries corresponding to damaged elements amid crack propagations. And as large numbers of elements enter damage realms, the updates are expensive. To further improve the microscale computational efficiency, we adopt the stabilized micro-damage model [31] which is reviewed in this section.

We first demonstrate the classic progressive damage model in Fig. 26a where the damage manifests itself in both material moduli degradation and yield stress reduction. Specifically, as an arbitrary material point is subject to strain softening, its damaged yield stress ($\mathbf{S}_{\mathrm{m}}^{\mathrm{d}}$) and degraded modulus ($\mathbb{C}_{\mathrm{m}}^{\mathrm{d}}$) are computed by projecting its undamaged reference stress ($\mathbf{S}_{\mathrm{m}}^{\mathrm{ref}}$) and the intact elastic modulus ($\mathbb{C}^{\mathrm{el}}$) by damage variables, see Sect. 2.2. We point out that this model focuses on the damage evolutions on a single integration point, and the subscript ‘m’ presents an arbitrary material point in a microstructure.

In contrast, the stabilized micro-damage model [31] aims to emulate an RVE’s softening by considering its overall response. Its advantage is that it projects the damage evolutions from plastic hardening by introducing three additional reference RVEs with partially shared state variables, and the whole procedure is simply based on pure implicit schemes without convergence issues. To further improve this method’s efficiency, we replace its implicit scheme with our AAF-IE for computing RVE’s effective damage responses.

For an arbitrary RVE subject to strain softening, its homogenized stress–strain response can be demonstrated by the blue curve in Fig. 26b where its damaged homogenized stress $\mathbf{S}_{\mathrm{M}}^{\mathrm{d}}$ is computed as:

$$\mathbf{S}_{\mathrm{M}}^{\mathrm{d}} = \mathbb{C}_{\mathrm{M}}^{\mathrm{d}} : \mathbf{E}_{\mathrm{M}}^{\mathrm{el}} = \mathbb{C}_{\mathrm{M}}^{\mathrm{d}} : (\mathbf{E}_{\mathrm{M}} - \mathbf{E}_{\mathrm{M}}^{\mathrm{pl}}) \tag{B.1}$$

where $\mathbb{C}_{\mathrm{M}}^{\mathrm{d}}$ is the homogenized elastic modulus of the damaged RVE, $\mathbf{E}_{\mathrm{M}}$, $\mathbf{E}_{\mathrm{M}}^{\mathrm{el}}$ and $\mathbf{E}_{\mathrm{M}}^{\mathrm{pl}}$ are the RVE’s effective total strain, elastic strain, and plastic strain, respectively, where the subscript ‘M’ indicates the associated variable is homogenized for overall macroscale property. The un-damaged stress of the first reference RVE can be computed by Eq. (B.2), see the red curve in Fig. 26b, where we assume the referenced RVE shares the same elastoplastic property as its original counterpart but in absence of damage.

$$\mathbf{S}_{\mathrm{M}}^{1} = \mathbb{C}_{\mathrm{M}}^{\mathrm{el}} : \mathbf{E}_{\mathrm{M}}^{\mathrm{el}} = \mathbb{C}_{\mathrm{M}}^{\mathrm{el}} : (\mathbf{E}_{\mathrm{M}} - \mathbf{E}_{\mathrm{M}}^{\mathrm{pl}}) \tag{B.2}$$

where $\mathbb{C}_{\mathrm{M}}^{\mathrm{el}}$ is the shared effective elastic moduli of the original and the first reference RVEs. Combining Eq. (B.1) and (B.2), the stress states on the two RVEs are related by:

$$\mathbf{S}_{\mathrm{M}}^{1} = \mathbb{C}_{\mathrm{M}}^{\mathrm{el}} : (\mathbb{C}_{\mathrm{M}}^{\mathrm{d}})^{-1} : \mathbf{S}_{\mathrm{M}}^{\mathrm{d}} \tag{B.3}$$

Let homogenized stress of the first RVE ($\mathbf{S}_{\mathrm{M}}^{1}$) identical to that of the second RVE ($\mathbf{S}_{\mathrm{M}}^{2}$) which we assume to have the same material properties as the original RVE but it only deforms elastically, see the yellow curve in Fig. 26b. The elastic macroscale strain of the second RVE is then computed by inverting the elastic modulus of Eq. (B.2) as:

$$\mathbf{E}_{\mathrm{M}}^{\mathrm{el}} = \left(\mathbb{C}_{\mathrm{M}}^{\mathrm{el}}\right)^{-1} : \mathbf{S}_{\mathrm{M}}^{1} = \left(\mathbb{C}_{\mathrm{M}}^{\mathrm{el}}\right)^{-1} : \mathbf{S}_{\mathrm{M}}^{2} \tag{B.4}$$

where we assume microscale stress of the second RVE to satisfy elastic constitutive equation as:

$$\mathbf{S}_{\mathrm{m}}^{2} = \mathbb{C}^{\mathrm{el}} : \mathbf{E}_{\mathrm{m2}}^{\mathrm{el}} \tag{B.5}$$

where $\mathbf{E}_{\mathrm{m2}}^{\mathrm{el}}$ represents the local elastic strain at an arbitrary integration point in the second referenced RVE, and the homogenized elastic strain can be computed via the volume averaging process as:

$$\mathbf{E}_{\mathrm{M}}^{\mathrm{el}} = \frac{1}{|\Omega|}\int_{\Omega} \mathbf{E}_{\mathrm{m2}}^{\mathrm{el}} d\Omega \tag{B.6}$$

where Ω and $|\Omega|$ are the domain and volume of the original RVE which are shared among all reference RVEs. We assume the local value of the elastic strain ($\mathbf{E}_{\mathrm{m2}}^{\mathrm{el}}$) is shared between the second RVE and the third RVE, which only deforms elastically but its effective elastic modulus is identical to that of the damaged original RVE ($\mathbb{C}_{\mathrm{M}}^{\mathrm{d}}$), see the green curve in Fig. 26b. Under this assumption, the local stress of the third RVE ($\mathbf{S}_{\mathrm{m}}^{3}$) is computed by:

$$\mathbf{S}_{\mathrm{m}}^{3} = (1 - \mathrm{D}_{\mathrm{m}})\mathbb{C}^{\mathrm{el}} : \mathbf{E}_{\mathrm{m2}}^{\mathrm{el}} = (1 - \mathrm{D}_{\mathrm{m}})\mathbb{C}^{\mathrm{el}} : \mathbf{E}_{\mathrm{m3}}^{\mathrm{el}} \tag{B.7}$$

where D_{m} is the local damage parameter at a microscale integration point. Its value is determined by the state variable from the first reference RVE. Towards the end, the elastic microscale stress in the third RVE is homogenized, which equals the effective stress of the original damaged RVE as:

$$\mathbf{S}_{\mathrm{M}}^{\mathrm{d}} = \frac{1}{|\Omega|}\int_{\Omega} \mathbf{S}_{\mathrm{m}}^{3} d\Omega \tag{B.8}$$

In addition, an effective damage parameter can be defined by the homogenized stresses from the original and the first RVEs as:

$$\mathrm{D}_{\mathrm{M}} = 1 - \frac{\left\|\mathbf{S}_{\mathrm{M}}^{\mathrm{d}} : \mathbf{S}_{\mathrm{M}}^{1}\right\|}{\left\|\mathbf{S}_{\mathrm{M}}^{1} : \mathbf{S}_{\mathrm{M}}^{1}\right\|} \tag{B.9}$$

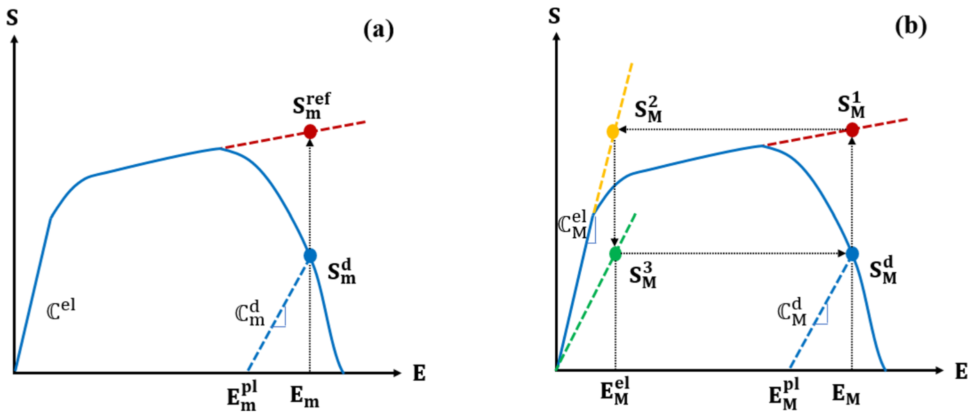

Fig. 26 Comparison of damage models. **a** Classic progressive damage model at each integration point of RVE. **b** Stabilized micro-damage model computes the homogenized damage responses of the original RVE by introducing three additional reference RVEs that partially share their state variables

where $\mathrm{D_M}$ is the homogenized damage parameter that indicates the damage status of the macroscopic integration point associated with the studied RVE. Similar to the single-scale damage parameter defined in Sect. 2.2, $\mathrm{D_M}$ ranges between [0, 1]. The macro-point is considered fully ruptured when $\mathrm{D_M} = 1$, even if only parts of the elements in the associated RVE are damaged.

References

1. Ammar HR, Samuel AM, Samuel FH (2008) Porosity and the fatigue behavior of hypoeutectic and hypereutectic aluminum–silicon casting alloys. Int J Fatigue 30(6):1024–1035. https://doi.org/10.1016/j.ijfatigue.2007.08.012
2. Catalina AV, Sen S, Stefanescu DM, Kaukler WF (2004) Interaction of porosity with a planar solid/liquid interface. Metall Mater Trans A 35(5):1525–1538. https://doi.org/10.1007/s11661-004-0260-z
3. Stefanescu DM (2015) Science and engineering of casting solidification, 3rd edn. Springer, Cham
4. Deng S, Soderhjelm C, Apelian D, Suresh K (2021) Estimation of elastic behaviors of metal components containing process induced porosity. Comput Struct 254:106558. https://doi.org/10.1016/j.compstruc.2021.106558
5. Deng S, Soderhjelm C, Apelian D, Suresh K (2022) Second-order defeaturing estimator of manufacturing-induced porosity on structural elasticity. Int J Num Methods Eng 123(19):4483–4517
6. Geers MGD, Kouznetsova VG, Matouš K, Yvonnet J (2017) Homogenization methods and multiscale modeling: nonlinear problems. In: Stein E, de Borst R, Hughes TJR (eds) Encyclopedia of computational mechanics, 2nd edn. American Cancer Society, Atlanta, pp 1–34. https://doi.org/10.1002/9781119176817.ecm2107
7. Collot J (2001) Review of new process technologies in the aluminum die-casting industry. Mater Manuf Process 16(5):595–617. https://doi.org/10.1081/AMP-100108624
8. Hill R (1965) A self-consistent mechanics of composite materials. J Mech Phys Solids 13(4):213–222. https://doi.org/10.1016/0022-5096(65)90010-4
9. Cook RD, Malkus DS, Plesha ME, Witt RJ (2001) Concepts and applications of finite element analysis, 4th edn. Wiley, New York
10. de Geus TWJ, Vondřejc J, Zeman J, Peerlings RHJ, Geers MGD (2017) Finite strain FFT-based non-linear solvers made simple. Comput Methods Appl Mech Eng 318:412–430. https://doi.org/10.1016/j.cma.2016.12.032
11. To Q-D, Bonnet G (2020) FFT based numerical homogenization method for porous conductive materials. Comput Methods Appl Mech Eng 368:113160. https://doi.org/10.1016/j.cma.2020.113160
12. Dvorak GJ (1992) Transformation field analysis of inelastic composite materials. Proc R Soc Lond Ser A Math Phys Sci 437(1900):311–327. https://doi.org/10.1098/rspa.1992.0063
13. Roussette S, Michel JC, Suquet P (2009) Nonuniform transformation field analysis of elastic–viscoplastic composites. Compos Sci Technol 69(1):22–27. https://doi.org/10.1016/j.compscitech.2007.10.032
14. Liu Z, Bessa MA, Liu WK (2016) Self-consistent clustering analysis: an efficient multi-scale scheme for inelastic heterogeneous materials. Comput Methods Appl Mech Eng 306:319–341. https://doi.org/10.1016/j.cma.2016.04.004
15. Tang S, Zhang L, Liu WK (2018) From virtual clustering analysis to self-consistent clustering analysis: a mathematical study. Comput Mech 62(6):1443–1460. https://doi.org/10.1007/s00466-018-1573-x
16. Cheng G, Li X, Nie Y, Li H (2019) FEM-Cluster based reduction method for efficient numerical prediction of effective properties of heterogeneous material in nonlinear range. Comput Methods Appl Mech Eng 348:157–184. https://doi.org/10.1016/j.cma.2019.01.019
17. Deng S, Soderhjelm C, Apelian D, Bostanabad R (2022) Reduced-order multiscale modeling of plastic deformations in 3D alloys with spatially varying porosity by deflated clustering analysis. Comput Mech 70:517–548
18. Bazant ZP, Planas J (1997) Fracture and size effect in concrete and other quasibrittle materials, 1st edn. CRC Press, New York
19. Bazant ZP (2010) Can multiscale-multiphysics methods predict softening damage and structural failure? JMC. https://doi.org/10.1615/IntJMultCompEng.v8.i1.50
20. Bažant ZP, Oh BH (1983) Crack band theory for fracture of concrete. Mat Constr 16(3):155–177. https://doi.org/10.1007/BF02486267
21. Bažant ZP, Jirásek M (2002) Nonlocal integral formulations of plasticity and damage: survey of progress. J Eng Mech 128(11):1119–1149. https://doi.org/10.1061/(ASCE)0733-9399(2002)128:11(1119)
22. Miehe C, Hofacker M, Schänzel L-M, Aldakheel F (2015) Phase field modeling of fracture in multi-physics problems. Part II. Coupled brittle-to-ductile failure criteria and crack propagation in thermo-elastic–plastic solids. Comput Methods Appl Mech Eng 294:486–522. https://doi.org/10.1016/j.cma.2014.11.017
23. Zienkiewicz OC, Taylor RL, Zhu JZ (2013) The finite element method: its basis and fundamentals, 7th edn. Butterworth-Heinemann, Oxford

24. Rodrigues EA, Manzoli OL, Bitencourt LAG, Bittencourt TN, Sánchez M (2018) An adaptive concurrent multiscale model for concrete based on coupling finite elements. Comput Methods Appl Mech Eng 328:26–46. https://doi.org/10.1016/j.cma.2017.08.048
25. Lamichhane BP, Wohlmuth BI (2004) Mortar finite elements for interface problems. Computing 72(3):333–348. https://doi.org/10.1007/s00607-003-0062-y
26. Wohlmuth BI (2000) A mortar finite element method using dual spaces for the Lagrange multiplier. SIAM J Numer Anal 38(3):989–1012. https://doi.org/10.1137/S0036142999350929
27. Boyd S, Vandenberghe L (2004) Convex optimization, 1st edn. Cambridge University Press, Cambridge
28. Ferreira BP, Pires FMA, Bessa MA (2021) Adaptive clustering-based reduced-order modeling framework: fast and accurate modeling of localized history-dependent phenomena. arXiv:2109.11897 [cond-mat]. Accessed 10 Jan 2022 [Online]. arXiv:abs/2109.11897
29. Otero F, Oller S, Martinez X (2018) Multiscale computational homogenization: review and proposal of a new enhanced-first-order method. Arch Comput Methods Eng 25(2):479–505. https://doi.org/10.1007/s11831-016-9205-0
30. Belytschko T, Liu WK, Moran B, Elkhodary K (2014) Nonlinear finite elements for continua and structures, 2nd edn. Wiley, Chichester
31. Liu Z, Fleming M, Liu WK (2018) Microstructural material database for self-consistent clustering analysis of elastoplastic strain softening materials. Comput Methods Appl Mech Eng 330:547–577. https://doi.org/10.1016/j.cma.2017.11.005
32. ABAQUS/Standard User's Manual, Version 6.9./Smith, Michael (2009). Dassault Systèmes Simulia Corp, Providence
33. Simo JC, Ju JW (1987) Strain and stress based continuum damage models. Int J Solids Struct 23(7):821–840. https://doi.org/10.1016/0020-7683(87)90083-7
34. Oliver J, Huespe AE, Cante JC (2008) An implicit/explicit integration scheme to increase computability of non-linear material and contact/friction problems. Comput Methods Appl Mech Eng 197(21):1865–1889. https://doi.org/10.1016/j.cma.2007.11.027
35. Ciampi V (1997) M. A. Crisfield, Non-linear finite element analysis of solids and structures. Meccanica 6(32):586–587. https://doi.org/10.1023/A:1004259118876
36. Prazeres PGC, Bitencourt LAG, Bittencourt TN, Manzoli OL (2016) A modified implicit–explicit integration scheme: an application to elastoplasticity problems. J Braz Soc Mech Sci Eng 38(1):151–161. https://doi.org/10.1007/s40430-015-0343-3
37. Likas A, Vlassis N, Verbeek JJ (2003) The global k-means clustering algorithm. Pattern Recogn 36(2):451–461. https://doi.org/10.1016/S0031-3203(02)00060-2
38. Wang K, Zhang J, Li D, Zhang X, Guo T (2008) Adaptive affinity propagation clustering. arXiv:0805.1096 [cs]. Accessed 10 Jan 10 [Online]. arXiv:abs/0805.1096
39. Ackermann MR, Blömer J, Kuntze D, Sohler C (2014) Analysis of agglomerative clustering. Algorithmica 69(1):184–215. https://doi.org/10.1007/s00453-012-9717-4
40. von Luxburg U (2007) A tutorial on spectral clustering. Stat Comput 17(4):395–416. https://doi.org/10.1007/s11222-007-9033-z
41. Giancarlo R, Scaturro D, Utro F (2008) Computational cluster validation for microarray data analysis: experimental assessment of Clest, Consensus Clustering, Figure of Merit, Gap Statistics and Model Explorer. BMC Bioinform 9(1):462. https://doi.org/10.1186/1471-2105-9-462
42. Integration K-Means Clustering Method and Elbow Method For Identification of The Best Customer Profile Cluster - IOPscience. https://iopscience.iop.org/article/10.1088/1757-899X/336/1/012017/meta. Accessed 10 Jan 2022
43. Aranganayagi S, Thangavel K (2007) Clustering categorical data using silhouette coefficient as a relocating measure. In: International conference on computational intelligence and multimedia applications (ICCIMA 2007), vol 2, pp 13–17. https://doi.org/10.1109/ICCIMA.2007.328
44. Łukasik S, Kowalski PA, Charytanowicz M, Kulczycki P (2016) Clustering using flower pollination algorithm and Calinski–Harabasz index. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp 2724–2728. https://doi.org/10.1109/CEC.2016.7744132.
45. Ainsworth M, Oden JT (1997) A posteriori error estimation in finite element analysis. Comput Methods Appl Mech Eng 142(1):1–88. https://doi.org/10.1016/S0045-7825(96)01107-3
46. Hardin RA, Beckermann C (2013) Effect of porosity on deformation, damage, and fracture of cast steel. Metall Mater Trans A 44(12):5316–5332. https://doi.org/10.1007/s11661-013-1669-z
47. Shakoor M, Gao J, Liu Z, Liu WK (2019) A data-driven multiscale theory for modeling damage and fracture of composite materials. In: Griebel M, Schweitzer MA (eds) Meshfree methods for partial differential equations IX. Springer, Cham, pp 135–148. https://doi.org/10.1007/978-3-030-15119-5_8
48. Liu Z (2020) Deep material network with cohesive layers: multi-stage training and interfacial failure analysis. Comput Methods Appl Mech Eng 363:112913. https://doi.org/10.1016/j.cma.2020.112913
49. Xie Y, Li S (2021) A stress-driven computational homogenization method based on complementary potential energy variational principle for elastic composites. Comput Mech 67(2):637–652. https://doi.org/10.1007/s00466-020-01953-8
50. Bishara D, Xie Y, Liu WK, Li S (2023) A state-of-the-art review on machine learning-based multiscale modeling, simulation, homogenization and design of materials. Arch Comput Methods Eng 30(1):191–222. https://doi.org/10.1007/s11831-022-09795-8
51. Xie Y, Li S (2021) Finite temperature atomistic-informed crystal plasticity finite element modeling of single crystal tantalum (α-Ta) at micron scale. Int J Numer Methods Eng 122(17):4660–4697. https://doi.org/10.1002/nme.6741
52. Zhang L-W, Xie Y, Lyu D, Li S (2019) Multiscale modeling of dislocation patterns and simulation of nanoscale plasticity in body-centered cubic (BCC) single crystals. J Mech Phys Solids 130:297–319. https://doi.org/10.1016/j.jmps.2019.06.006
53. Xie Y, Li S (2021) Geometrically-compatible dislocation pattern and modeling of crystal plasticity in body-centered cubic (BCC) crystal at micron scale. Comput Model Eng Sci 129(3):1419–1440. https://doi.org/10.32604/cmes.2021.016756

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.