Computational Mechanics (2021) 67:1293–1306
https://doi.org/10.1007/s00466-021-01982-x

ORIGINAL PAPER

Multiresolution clustering analysis for efficient modeling of hierarchical material systems

Cheng Yu[1] · Orion L. Kafka[1] · Wing Kam Liu[1]

Received: 22 September 2020 / Accepted: 24 January 2021 / Published online: 21 March 2021

Abstract
Direct representation of material microstructure in a macroscale simulation is prohibitively expensive, if even possible, with current methods. However, the information contained in such a representation is highly desirable for tasks such as material/alloy design and manufacturing process control. In this paper, a mechanistic machine learning framework is developed for fast multiscale analysis of material response and structure performance. The new capabilities stem from three major factors: (1) the use of an unsupervised learning (clustering)-based discretization to achieve significant order reduction at both macroscale and microscale; (2) the generation of a database of interaction tensors among discretized material regions; (3) concurrent multiscale response prediction to solve the mechanistic equations. These factors allow for an orders-of-magnitude decrease in the computational expense compared to FE^n, $n \geq 2$. This method provides sufficiently high fidelity and speed to reasonably conduct inverse modeling for the challenging tasks mentioned above.

Keywords Data-driven · Unsupervised learning · Reduced order modeling · Concurrent multiscale · Materials design

1 Introduction

1.1 Hierarchical materials design

Modern advanced materials are manufactured with complicated microstructural features at multiple length scales to achieve better properties and performance than previous generations of materials. Figure 1 shows the system chart for Nickel-based superalloys, adapted from the computational materials design framework first proposed by Olson [1]. The performance of components, e.g. aeroengine disks, in a high temperature environment relies on the material properties such as strength, toughness, creep resistance and fatigue resistance. These properties are determined by the interactions among the hierarchical microstructures: polycrystals, carbides at grain boundaries, primary and secondary γ' precipitates, as well as byproducts such as voids, inclusions and residual stress, which are further controlled by a sequence of processing steps. For example, excellent creep resistance might be achieved by optimizing the volume fraction and morphology of precipitates so that they impede dislocation movement as well as grain boundary migration in the most efficient way. Hierarchical materials design thus requires fast evaluation of macroscopic properties as a function of hierarchical microstructures.

On the one hand, the involvement of microstructural features at multiple length scales significantly broadens the materials design space. On the other, this challenges mechanistic computational prediction of macroscopic properties and performance (in-service behavior) of material systems. The objective of this work is to address this concern by developing a physics-based data-driven multiscale modeling framework that can predict macroscopic properties/performance while capturing complex microstructures at several length scales.

1.2 Multiscale methods: hierarchical and concurrent

The hierarchical (also known as serial coupling or parameter passing) technique [2,3] has been used to model complex material systems. It often has three stages: the bottom-up stage, the macroscale analysis stage and the top-down stage. The bottom-up stage is used to develop macroscale constitutive models. It starts from the finest scale by calibrating an upper scale material model with average stress-strain

✉ Wing Kam Liu
w-liu@northwestern.edu

[1] Department of Mechanical Engineering, Northwestern University, Evanston, IL 60208, USA

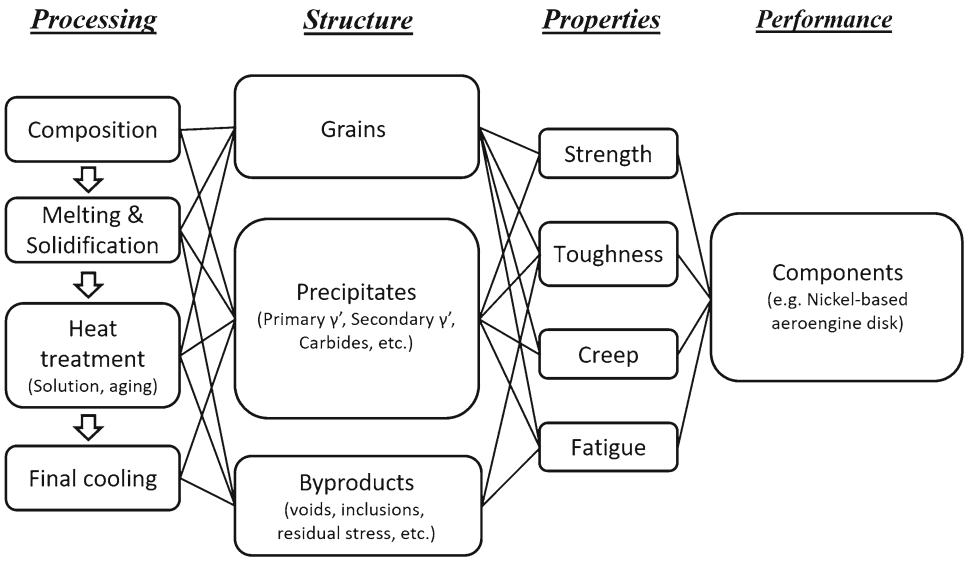

Fig. 1 System chart for Nickel-based superalloys: blocks indicate desired property objectives, hierarchical microstructures and sequential processing steps needed for design; links between blocks indicate computational models

response computed from a microstructural volume element (MVE, sometimes referred to as representative volume element (RVE) [4] or statistical volume element (SVE) [5]), and repeats this process until the macroscale constitutive law is calibrated. The macroscale analysis stage predicts part performance using the calibrated macroscale constitutive law. Then, in the top-down stage, a detailed solution in a lower scale MVE can be computed with the deformation response of a material point in the upper scale as boundary conditions. However, all these hierarchical approaches rely on the calibration of assumed constitutive laws at each scale through homogenization operations, during which microscale details are inevitably lost or neglected. In addition, the top-down stage propagates errors through the boundary conditions.

One alternative is the concurrent (or two-way coupling) technique, which requires the simulations at all scales to be performed simultaneously. Every integration or material point of an arbitrary scale (except the finest scale) is linked to a MVE at a finer scale, which is solved on-the-fly to provide the stress-strain responses for that integration point during the analysis. Thus no assumed constitutive law is needed other than at the finest scale. In this regard, FE-FE (or FE^2) [6] and FE-FFT [7] have been proposed to model a two-scale material system. However, the computational cost is tremendous if more than two scales are considered.

One assumption the above mentioned multiscale methods often take is scale separation, meaning that the characteristic length of deformation at the part scale is much larger than that of the constitutive microstructural features. To regularize the case when scale separation breaks down, generalized continuum theories [8–11] have been proposed, which introduce extra degrees of freedom (e.g. strain gradients) to avoid spurious mesh-size dependency and RVE-size dependency. These theories were later generalized to an arbitrary number of scales, resulting in the multiresolution continuum theory (MCT) [12–14]. These generalized continuum theories were used most in the hierarchical multiscale framework, although archetype-blending continuum (ABC) [15,16] was proposed to maintain some information about microscale constituents. The second-order computational homogenization methods [17,18] maintain microscale information concurrently but again suffer from the high computational cost. The need to compute detailed mechanical response in multiscale methods motivates the development of order-reduction methods.

1.3 Clustering: an efficient discretization method

A myriad of reduced-order methods have been developed for efficient multiscale simulations, for example, the transformation field analysis (TFA) [19], the nonuniform transformation field analysis (NTFA) [20], the eigen-deformation based method [21], and the proper orthogonal decomposition (POD) [22]. A detailed review can be found in [23]. Recently, the self-consistent clustering analysis (SCA) proposed by Liu et al. [24] has been shown to maintain high accuracy and effciency. The key idea of SCA is the clustering-based discretization theory for order reduction rather than the traditional domain-based discretization, as illustrated in Fig. 2. The domain-based approach decomposes the domain directly, leading to a Riemann integral; while the clustering-based approach analyzes the *a priori* response of interest through an unsupervised machine learning method to decompose the domain into "clusters", resulting in the so-called Lebesgue integral. The key differences of SCA from the other reduced-order methods mentioned above lie in: (1) the clustering-based discretization, resulting in potentially higher efficiency in representing heterogeneous material systems; (2) the flexibility of using a self-consistent scheme

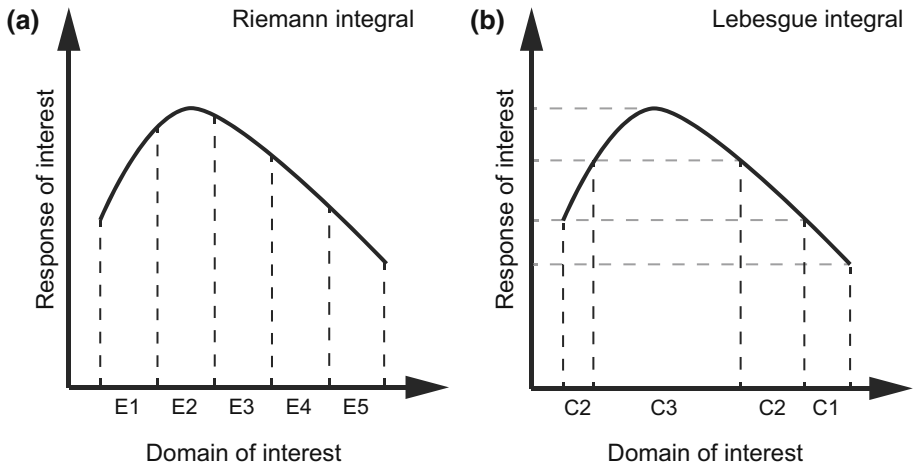

Fig. 2 Discretization of partial differential equations through **a** domain-based decomposition and **b** response-based decomposition clustering e.g. with unsupervised learning

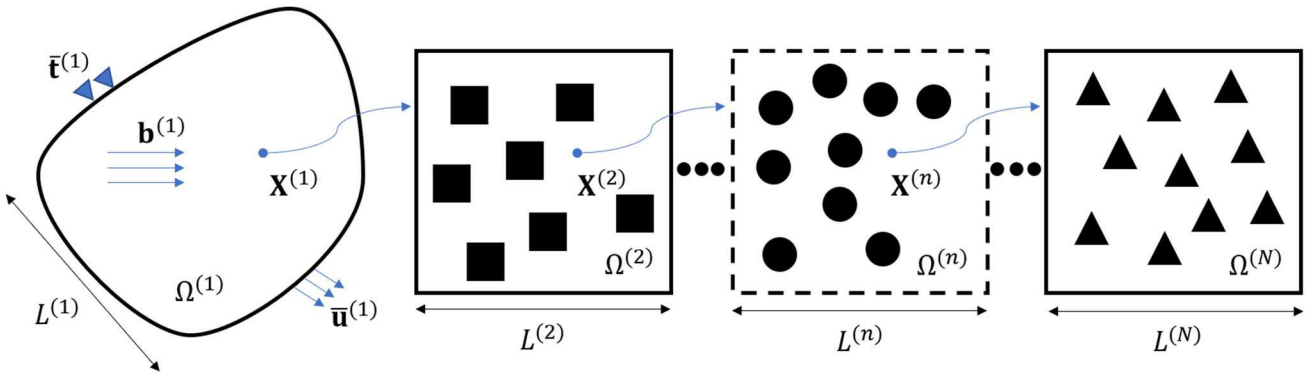

Fig. 3 A schematic of the N-scale problem of interest where the characteristic length scales satisfy $L^{(1)} \gg L^{(2)} \gg \cdots \gg L^{(n)} \gg \cdots \gg L^{(N)}$. Certain material points in a scale are associated with a finer scale microstructure volume element. Microstructure features are represented by the squares, circles and triangles

in the online stage for higher accuracy. This method has been used in FE-SCA type concurrent simulations for strain softening materials [25], polycrystalline materials [26–28], and composite materials [29–31]. However, in all the concurrent simulation approaches (FE-FE, FE-FFT, FE-SCA) mentioned above, the finite element method is used for the macroscale analysis, which slows down computational prediction of structure-property-performance relationships in a materials design practice. This work will address this issue by extending the clustering-based discretization theory to macroscale modeling and further to N-scale concurrent simulation of material systems, resulting in the multiresolution clustering analysis (MCA) framework.

1.4 Outline of the paper

Section 2 presents the general integral equations for an arbitrary N-scale problem at finite strains and their clustering-based discretization. A two-stage (offline/online) solution scheme is given in Sect. 3. The accuracy and efficiency of MCA in the context of an example problem is shown and discussed in Sect. 4. Concluding remarks are provided in Sect. 5.

2 Multiresolution formulation for hierarchical material systems

2.1 Fully coupled *N*-scale problem description

A fully coupled N-scale problem of interest is schematically shown in Fig. 3. Certain material points in a scale are associated with a finer scale microstructure volume element. The characteristic lengths of each scale are assumed to be well separated, meaning that $L^{(1)} \gg L^{(2)} \gg \cdots \gg L^{(n)} \gg \cdots \gg L^{(N)}$, where $L^{(n)}$ is the characteristic length of the n^{th} scale with $n = 1, \cdots, N$.

The first scale involves a component or part with general boundary conditions and body forces. The equilibrium boundary value problem (BVP) of the first scale is expressed by

$$\begin{cases} \nabla^{(1)} \cdot \mathbf{P}^{(1)}(\mathbf{X}^{(1)}) + \mathbf{b}^{(1)} = \mathbf{0}, \quad \forall \mathbf{X}^{(1)} \in \Omega^{(1)} \\ \mathbf{F}^{(1)} = \mathbf{I}_2 + \nabla^{(1)}\mathbf{u}^{(1)} \\ \mathbf{u}^{(1)} = \bar{\mathbf{u}}^{(1)}, \quad \forall \mathbf{X}^{(1)} \in \partial\Omega_{\mathrm{u}}^{(1)} \\ \mathbf{t}^{(1)} = \bar{\mathbf{t}}^{(1)}, \quad \forall \mathbf{X}^{(1)} \in \partial\Omega_{\mathrm{t}}^{(1)} = \partial\Omega^{(1)} \setminus \partial\Omega_{\mathrm{u}}^{(1)} \end{cases} \tag{1}$$

where $\mathbf{P}$ is the first Piola-Kirchhoff stress (PK1 stress); $\mathbf{F}$ is the deformation gradient; $\mathbf{b}$ is the body force; $\mathbf{I}_2$ is the second order identity tensor; $\mathbf{u}$ is the displacement with prescribed values $\bar{\mathbf{u}}$ on the displacement boundary $\partial\Omega_{\mathrm{u}}$; $\mathbf{t} = \mathbf{P} \cdot \mathbf{n}$ is the traction with prescribed values $\bar{\mathbf{t}}$ on the traction boundary $\partial\Omega_{\mathrm{t}}$. The normal vector $\mathbf{n}$ is on the domain boundary. The gradient operator is denoted by ∇. The superscript (1) denotes the first scale. The other scales (scale 2 to N) involve representative microstructure volume elements (MVEs) with periodic boundary conditions. The equilibrium BVPs defined on these MVEs are given by

$$\begin{cases} \nabla^{(n)} \cdot \mathbf{P}^{(n)} = \mathbf{0}, \quad \forall \mathbf{X}^{(n)} \in \Omega^{(n)} \\ \mathbf{F}^{(n)} = \mathbf{I}_2 + \nabla^{(n)}\mathbf{u}^{(n)} \\ \mathbf{u}^{(n)} \text{ periodic on } \partial\Omega^{(n)} \\ \mathbf{t}^{(n)} \text{ anti-periodic on } \partial\Omega^{(n)} \end{cases} \tag{2}$$

where the superscript (n) denotes the n^{th} scale with $n = 2, ..., N$. To establish the coupling relationship among these BVPs at different scales, we denote $\mathrm{V}^{(n)}$ as a subset of $\Omega^{(n)}$ whose material points are represented by finer scale microstructures. It should be noted that $\mathrm{V}^{(N)} = \varnothing$. If the first scale is the component (part) scale where all material points are associated with finer scale microstructures, we have $\mathrm{V}^{(1)} = \Omega^{(1)}$. Thus, the coupling between any two consecutive scales through homogenization of PK1 stress and deformation gradient is given by

$$\begin{cases} \mathbf{F}^{(n-1)}(\mathbf{X}^{(n-1)}) = \frac{1}{|\Omega^{(n)}|}\int_{\Omega^{(n)}} \mathbf{F}^{(n)}(\mathbf{X}^{(n)})\mathrm{d}\mathbf{X}^{(n)}, \\ \mathbf{P}^{(n-1)}(\mathbf{X}^{(n-1)}) = \frac{1}{|\Omega^{(n)}|}\int_{\Omega^{(n)}} \mathbf{P}^{(n)}(\mathbf{X}^{(n)})\mathrm{d}\mathbf{X}^{(n)}, \end{cases} \tag{3}$$

where $\mathbf{X}^{(n-1)}$ is an arbitrary material point in $\mathrm{V}^{(n-1)}$; $\Omega^{(n)}$ is the n^{th} scale MVE domain associated with $\mathbf{X}^{(n-1)}$; and $n = 2, ..., N$. The constitutive equations for material points at a scale that does not have finer microstructural features are assumed to be known *a priori* and are represented by a generic constitutive law:

$$\mathbf{P}^{(n)} = \mathbf{f}^{(n)}(\mathbf{F}^{(n)}, \mathbf{w}^{(n)}), \quad \forall \mathbf{X}^{(n)} \in \Omega^{(n)} \setminus \mathrm{V}^{(n)} \quad \text{with } n = 1, ..., N \tag{4}$$

where $\mathbf{w}$ collectively denotes the necessary state variables. Note that different constitutive equations can be used to represent the physics involved at each scale for which a constitutive law is used.

Eqs. (1–4) together describe the fully coupled N-scale problem to be solved in this work.

2.2 Generalized Lippmann-Schwinger integral equation

The derivation in Sect. 2.1 shows that each scale entails a boundary value problem, which is rewritten in the following general form:

$$\begin{cases} \nabla \cdot \mathbf{P}(\mathbf{X}) + \mathbf{b} = \mathbf{0}, \quad \forall \mathbf{X} \in \Omega \\ \mathbf{F} = \mathbf{I}_2 + \nabla\mathbf{u}, \quad \forall \mathbf{X} \in \Omega \\ \mathbf{u} = \bar{\mathbf{u}}, \quad \forall \mathbf{X} \in \partial\Omega_{\mathrm{u}} \\ \mathbf{t} = \bar{\mathbf{t}}, \quad \forall \mathbf{X} \in \partial\Omega_{\mathrm{t}} = \partial\Omega \setminus \partial\Omega_{\mathrm{u}} \end{cases} \tag{5}$$

where the superscript defining the operative scale is omitted to simplify the notation.

By introducing an arbitrary comparison medium with elastic stiffness $\mathbf{C}^0(\mathbf{X})$, the solution $\mathbf{F}(\mathbf{X})$ to problem Eq. (5) can be obtained through the superposition of a reference deformation solution and an eigenstress deformation solution: $\mathbf{F}(\mathbf{X}) = \tilde{\mathbf{F}}(\mathbf{X}) + \nabla\hat{\mathbf{u}}(\mathbf{X})$. Here, $\tilde{\mathbf{F}}(\mathbf{X})$ is the deformation solution to the following reference problem:

$$\begin{cases} \nabla \cdot \tilde{\mathbf{P}}(\mathbf{X}) + \mathbf{b} = \mathbf{0}, \quad \forall \mathbf{X} \in \Omega \\ \tilde{\mathbf{P}}(\mathbf{X}) = \mathbf{C}^0(\mathbf{X}) : \left(\tilde{\mathbf{F}}(\mathbf{X}) - \mathbf{I}_2\right), \quad \forall \mathbf{X} \in \Omega \\ \tilde{\mathbf{u}} = \bar{\mathbf{u}}, \quad \forall \mathbf{X} \in \partial\Omega_{\mathrm{u}} \\ \tilde{\mathbf{t}} = \bar{\mathbf{t}}, \quad \forall \mathbf{X} \in \partial\Omega_{\mathrm{t}} \end{cases} \tag{6}$$

and $\hat{\mathbf{u}}(\mathbf{X})$ is the displacement solution to the following eigenstress problem:

$$\begin{cases} \nabla \cdot \hat{\mathbf{P}}(\mathbf{X}) = \mathbf{0}, \quad \forall \mathbf{X} \in \Omega \\ \hat{\mathbf{P}}(\mathbf{X}) = \mathbf{C}^0(\mathbf{X}) : \nabla\hat{\mathbf{u}}(\mathbf{X}) + \boldsymbol{\tau}(\mathbf{X}), \quad \forall \mathbf{X} \in \Omega \\ \hat{\mathbf{u}} = \mathbf{0}, \quad \forall \mathbf{X} \in \partial\Omega_{\mathrm{u}} \\ \hat{\mathbf{t}} = \mathbf{0}, \quad \forall \mathbf{X} \in \partial\Omega_{\mathrm{t}} \end{cases} \tag{7}$$

where $\boldsymbol{\tau}(\mathbf{X}) = \mathbf{P}(\mathbf{X}) - \mathbf{C}^0(\mathbf{X}) : (\mathbf{F}(\mathbf{X}) - \mathbf{I}_2)$. The displacement solution of the eigenstress problem Eq. (7) can be expressed as

$$\begin{aligned} \hat{u}_i(\mathbf{X}) &= \int_{\Omega} G_{ij}(\mathbf{X}, \mathbf{X}')\tau_{jk,k'}(\mathbf{X}')d\mathbf{X}' \\ &= \int_{\partial\Omega} G_{ij}(\mathbf{X}, \mathbf{X}')\tau_{jk}(\mathbf{X}')n_k(\mathbf{X}')d\mathbf{X}' \\ &\quad - \int_{\Omega} G_{ij,k'}(\mathbf{X}, \mathbf{X}')\tau_{jk}(\mathbf{X}')d\mathbf{X}' \end{aligned}$$

$$= \int_{\Omega} \big(G_{ij}(\mathbf{X}, \mathbf{X}')n_k(\mathbf{X}')\delta_S(\mathbf{X}') - G_{ij,k'}(\mathbf{X}, \mathbf{X}')\big)\,\tau_{jk}(\mathbf{X}')d\mathbf{X}', \tag{8}$$

where $G_{ij}(\mathbf{X}, \mathbf{X}')$ is the Green's function that satisfies the same boundary condition as in Eq. (7). Note that k' in Eq. (2.2) denotes that the derivative is taken w.r.t. $\mathbf{X}'$ and it still denotes the same index with k and they follow the Einstein summation convention. $\delta_S(\mathbf{X})$ is the delta function defined on boundary $\partial\Omega$:

$$\delta_S(\mathbf{X}) = \begin{cases} \lim_{\delta\to 0} \frac{1}{\delta} & \text{if } \mathbf{X} \in \partial\Omega_\delta \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $\partial\Omega_\delta$ is a volume such that it includes the boundary $\partial\Omega$ and its orthogonal cross-section has a specially defined measure of δ, see [32] for details. This allows the conversion from the surface integral to volume integral in Eq. (2.2).

Then the deformation gradient can be obtained with

$$\hat{u}_{i,l}(\mathbf{X}) = -\int_{\Omega} \Gamma_{iljk}(\mathbf{X}, \mathbf{X}')\tau_{jk}(\mathbf{X}')d\mathbf{X}' \tag{10}$$

where the Green's operator $\boldsymbol{\Gamma}$ is defined as

$$\Gamma_{iljk} = G_{ij,k'l}(\mathbf{X}, \mathbf{X}') - G_{ij,l}(\mathbf{X}, \mathbf{X}')n_k(\mathbf{X}')\delta_S(\mathbf{X}'). \tag{11}$$

The tensor form of the strain solution to the eigenstress problem Eq. (7) is

$$\nabla\hat{\mathbf{u}}(\mathbf{X}) = -\int_{\Omega} \boldsymbol{\Gamma}(\mathbf{X}, \mathbf{X}') : \boldsymbol{\tau}(\mathbf{X}')d\mathbf{X}'. \tag{12}$$

Then the solution of problem Eq. (5) can be expressed as

$$\mathbf{F}(\mathbf{X}) = \tilde{\mathbf{F}}(\mathbf{X}) - \int_{\Omega} \boldsymbol{\Gamma}(\mathbf{X}, \mathbf{X}') : \Big(\mathbf{P}(\mathbf{X}') - \mathbf{C}^0(\mathbf{X}') : (\mathbf{F}(\mathbf{X}') - \mathbf{I}_2)\Big)\,d\mathbf{X}'. \tag{13}$$

This is the *generalized Lippmann-Schwinger equation* (L-S equation). It should be noted that the above derivation relies on the introduction of the reference elastic stiffness $\mathbf{C}^0(\mathbf{X})$. This ensures the availability of the Green's function, so that the generalized L-S equation holds for finite strain cases as well as nonlinear constitutive relationships between $\mathbf{P}$ and $\mathbf{F}$.

In the special case of a homogeneous reference stiffness tensor, i.e. $\mathbf{C}^0(\mathbf{X}') = constant$, and periodic boundary conditions, Eq. (13) reduces to the classical L-S equation [33]:

$$\mathbf{F}(\mathbf{X}) = \mathbf{F}^0 - \int_{\Omega} \boldsymbol{\Gamma}(\mathbf{X}, \mathbf{X}') : (\mathbf{P}(\mathbf{X}') - \mathbf{C}^0 : \mathbf{F}(\mathbf{X}'))d\mathbf{X}', \tag{14}$$

where $\mathbf{F}^0$ is the far field deformation gradient. This can be shown by noting that the solution to the reference problem Eq. (6) is equal to the far field deformation gradient $\mathbf{F}^0$, and the following property of the Green's operator [34]:

$$\int_{\Omega} \boldsymbol{\Gamma}(\mathbf{X}, \mathbf{X}') : (\mathbf{C}^0 : \mathbf{I}_2)d\mathbf{X}' = \mathbf{0}. \tag{15}$$

For practical convenience, the incremental form of Eq. (13) is given by

$$\Delta\mathbf{F}(\mathbf{X}) = \Delta\tilde{\mathbf{F}}(\mathbf{X}) - \int_{\Omega} \boldsymbol{\Gamma}(\mathbf{X}, \mathbf{X}') : (\Delta\mathbf{P}(\mathbf{X}') - \mathbf{C}^0(\mathbf{X}') : \Delta\mathbf{F}(\mathbf{X}'))d\mathbf{X}'. \tag{16}$$

where $\Delta\mathbf{F}$ and $\Delta\mathbf{P}$ are the incremental deformation gradient and PK1 stress respectively.

2.2.1 Coupling between scales

Based on the above derivation, the fully coupled integral equations of the N-scale problem described in Sect. 2.1 are given in the following incremental form:

$$\Delta\mathbf{F}^{(n)}(\mathbf{X}) = \Delta\tilde{\mathbf{F}}^{(n)}(\mathbf{X}) - \int_{\Omega^{(n)}} \boldsymbol{\Gamma}^{(n)}(\mathbf{X}, \mathbf{X}') : \Big(\Delta\mathbf{P}^{(n)}(\mathbf{X}') - \mathbf{C}^{0,(n)}(\mathbf{X}') : \Delta\mathbf{F}^{(n)}(\mathbf{X}')\Big)d\mathbf{X}' \tag{17}$$

where $n = 1, ..., N$; $\Delta\tilde{\mathbf{F}}^{(1)}$ is the solution to

$$\begin{cases} \nabla^{(1)} \cdot \Delta\tilde{\mathbf{P}}^{(1)}(\mathbf{X}^{(1)}) + \mathbf{b}^{(1)}(\mathbf{X}^{(1)}) = \mathbf{0}, & \forall\mathbf{X}^{(1)} \in \Omega^{(1)} \\ \Delta\tilde{\mathbf{P}}^{(1)}(\mathbf{X}) = \mathbf{C}^{0,(1)}(\mathbf{X}^{(1)}) : \Delta\tilde{\mathbf{F}}^{(1)}(\mathbf{X}^{(1)}), & \forall\mathbf{X}^{(1)} \in \Omega^{(1)} \\ \Delta\tilde{\mathbf{u}}^{(1)}(\mathbf{X}^{(1)}) = \Delta\bar{\mathbf{u}}^{(1)}, & \forall\mathbf{X}^{(1)} \in \partial\Omega_{\mathrm{u}}^{(1)} \\ \Delta\tilde{\mathbf{t}}^{(1)}(\mathbf{X}^{(1)}) = \Delta\bar{\mathbf{t}}^{(1)}, & \forall\mathbf{X}^{(1)} \in \partial\Omega_{\mathrm{t}}^{(1)}; \end{cases} \tag{18}$$

$\Delta\tilde{\mathbf{F}}^{(n)}$, $n = 2, ..., N$ is the solution to

$$\begin{cases} \nabla^{(n)} \cdot \Delta\tilde{\mathbf{P}}^{(n)}(\mathbf{X}^{(n)}) = \mathbf{0}, & \forall\mathbf{X}^{(n)} \in \Omega^{(n)} \\ \Delta\tilde{\mathbf{P}}^{(n)}(\mathbf{X}^{(n)}) = \mathbf{C}^{0,(n)}(\mathbf{X}^{(n)}) : \Delta\tilde{\mathbf{F}}^{(n)}(\mathbf{X}^{(n)}), & \forall\mathbf{X}^{(n)} \in \Omega^{(n)} \\ \frac{1}{|\Omega^{(n)}|}\int_{\Omega^{(n)}} \Delta\tilde{\mathbf{F}}^{(n)}(\mathbf{X}^{(n)})\mathrm{d}\mathbf{X}^{(n)} = \Delta\mathbf{F}^{(n-1)}(\mathbf{X}^{(n-1)}), & \forall\mathbf{X}^{(n-1)} \in \mathrm{V}^{(n-1)} \\ \Delta\tilde{\mathbf{u}}^{(n)} \text{ periodic on } \partial\Omega^{(n)} \\ \Delta\tilde{\mathbf{t}}^{(n)} \text{ anti-periodic on } \partial\Omega^{(n)} \end{cases} \tag{19}$$

and

$$\Delta\mathbf{P}^{(n-1)} = \frac{1}{|\Omega^{(n)}|}\int_{\Omega^{(n)}} \Delta\mathbf{P}^{(n)}\mathrm{d}\mathbf{X}^n, \quad n = 2, ..., N. \tag{20}$$

Note that in the special case of a homogeneous reference stiffness tensor, i.e. $\mathbf{C}^{0,(n)}(\mathbf{X}) = constant$, Eq. (19) is equivalent to

$$\Delta\tilde{\mathbf{F}}^{(n)} = \Delta\mathbf{F}^{(n-1)}, \quad n = 2, ..., N. \tag{21}$$

2.3 Clustering-based discretization of the Lippmann-Schwinger equation

At each scale $n, n = 1, ..., N$, suppose the domain $\Omega^{(n)}$ is decomposed into $N_c^{(n)}$ compact, non-overlapping clusters. Also assume that all variables are uniform in each cluster $\Omega^{I^{(n)}}, I^{(n)} = 1, ..., N_c^{(n)}$ of scale n and represented by their volume averages $\square^{I^{(n)}} = \frac{1}{|\Omega^{I^{(n)}}|}\int_{\Omega^{I^{(n)}}} \square(\mathbf{X}^{(n)})d\mathbf{X}^{(n)}$. Note that $\square^{I^{(n)}} = \square^{I^{(1)},...,I^{(n-1)},I^{(n)}}$ indicates the average value of a variable (scalar, vector or tensor) in cluster $I^{(n)}$ associated with cluster $I^{(1)}, ..., I^{(n-1)}$ in successively coarser scales. Thus any variable $\square^{(n)}(\mathbf{X}^{(n)})$ can be approximated by

$$\square^{(n)}(\mathbf{X}^{(n)}) = \sum_{I^{(n)}=1}^{N_c^{(n)}} \square^{I^{(n)}} \chi^{I^{(n)}}(\mathbf{X}^{(n)})d\mathbf{X}^{(n)}, \tag{22}$$

where

$$\chi^{I^{(n)}}(\mathbf{X}^{(n)}) = \begin{cases} 1 & \text{if } \mathbf{X}^{(n)} \in \Omega^{I^{(n)}} \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

Using the above representation, the fully coupled integral equations in Eq. (17) can be discretized as

$$\Delta\mathbf{F}^{I^{(n)}} = \Delta\tilde{\mathbf{F}}^{I^{(n)}} - \sum_{J^{(n)}=1}^{N_c^{(n)}} \mathbf{D}^{I^{(n)},J^{(n)}} : \Delta\boldsymbol{\tau}^{J^{(n)}}, \quad I^{(n)} = 1, ..., N_c^{(n)}, \quad n = 1, ..., N \tag{24}$$

where the incremental eigenstress $\Delta\boldsymbol{\tau}^{J^{(n)}} = \Delta\mathbf{P}^{J^{(n)}} - \mathbf{C}^{0,J^{(n)}} : \Delta\mathbf{F}^{J^{(n)}}$, and

$$\Delta\tilde{\mathbf{F}}^{I^{(n)}} = \frac{1}{|\Omega^{I^{(n)}}|}\int_{\Omega^{I^{(n)}}} \Delta\tilde{\mathbf{F}}(\mathbf{X}^{(n)})d\mathbf{X}^{(n)}, \quad I^{(n)} = 1, ..., N_c^{(n)}, \quad n = 1, ..., N \tag{25}$$

and $\mathbf{D}^{I^{(n)},J^{(n)}}$ is the so-called interaction tensor between cluster $I^{(n)}$ and cluster $J^{(n)}$, which is given by

$$\mathbf{D}^{I^{(n)},J^{(n)}} = \frac{1}{|\Omega^{I^{(n)}}|}\int_{\Omega^{I^{(n)}}} \chi^{I^{(n)}}(\mathbf{X}^{(n)})(\boldsymbol{\Gamma}^{(n)} * \chi^{J^{(n)}})(\mathbf{X}^{(n)})d\mathbf{X}^{(n)}. \tag{26}$$

The clustering-based discretized version of the fully coupled N-scale integral equations, given in Eq. (24), will be solved with the MCA framework to be introduced in the next section.

3 Two-stage (offline/online) solution

This section describes the MCA solution scheme based on the Lebesgue integral concept mentioned earlier, which consists of an "offline" stage to generate a clustering database and an "online" stage that makes use of the database to predict material responses at multiple length scales. The efficiency and accuracy of MCA will be shown through numerical examples that involve several length scales.

3.1 Offline stage: clustering database generation

The offline stage for clustering database generation consists of three primary steps: (1) data collection, (2) unsupervised learning (e.g. clustering), and (3) pre-computation of the interaction tensors among clusters.

3.1.1 Deformation concentration field

MCA reduces the degrees of freedom to be solved by taking advantage of the mechanical response similarity of material points. This similarity is found by clustering the field data of some mechanical response. In this work, linear elastic mechanical responses are chosen as the clustering data, because they well indicate local material behaviors and are computationally less expensive to obtain.

At microstructural scales, the deformation concentration tensor can be used. It is defined by

$$\mathbf{A}(\mathbf{X}) = \frac{\partial\mathbf{F}(\mathbf{X})}{\partial\mathbf{F}^0}, \forall\mathbf{X} \in \Omega \tag{27}$$

where $\mathbf{F}^0$ is the macroscopic deformation corresponding to the boundary conditions of the MVE, $\mathbf{F}(\mathbf{X})$ is the local deformation at point $\mathbf{X}$ in the MVE domain Ω. In two dimensions, $\mathbf{A}(\mathbf{X})$ has $(2 \times 2)^2 = 16$ independent components, requiring direct numerical simulations (DNS) under four orthogonal loading conditions to determine. In three dimensions, $\mathbf{A}(\mathbf{X})$ has $(3 \times 3)^2 = 81$ independent components, requiring DNS under nine orthogonal loading conditions to uniquely define. For small strain problems, the strain concentration tensor can be used, and only six (or three) orthogonal loading conditions for DNS are needed in three (or two) dimensions.

At the macro-scale, the clustering data depends on the applied boundary conditions and loading conditions. Thus, some prior knowledge of the possible loading paths would benefit the applicability of the proposed method. In the case of fixed boundary conditions, the deformation gradient (or strain) field calculated under the same boundary conditions can be used for clustering. In the case of moving boundary conditions, a database of deformation gradient (or strain) field under all possible loading paths needs to be prepared.

3.1.2 Clustering

Once the deformation concentration field is prepared, the next step is to decompose the domain into clusters each containing similar data points. This can be done using the k-means clustering algorithm [24]. For a predefined number of clusters N_c, k-means clustering solves a minimization problem defined by

$$\mathbf{S}^{\star} = \underset{\mathbf{S}}{\operatorname{argmin}} \sum_{J=1}^{N_c} \sum_{m \in S^J} ||\mathbf{A}^m - \bar{\mathbf{A}}^J||^2. \tag{28}$$

where $\mathbf{S} = \{S^1, S^2, ..., S^{N_c}\}$ is a non-overlapping partition of the material points in a domain of interest; $\mathbf{S}^{\star}$ is the partition that achieves minimization; $\mathbf{A}^m$ is the deformation concentration tensor at material point m; $\bar{\mathbf{A}}^J$ is the deformation concentration tensor averaged within cluster J. Similarity is measured by the Frobenius norm, denoted by $||\bar{\boxdot}||$. It should be noted that more advanced clustering methods, e.g. self-organizing maps [35], could also be used in lieu of k-means clustering.

3.1.3 Interaction tensors

The interaction tensor describes the impact one cluster has on each of the other clusters. Once the clustering process is completed, the interaction tensor can be explicitly computed. Importantly, the integral part only has to be computed once during the offline stage. Three ways to compute the interaction tensor are [36]:

Fast Fourier transform With periodic boundary conditions and a homogeneous reference material with isotropic stiffness, the Green's operator has a simple expression in Fourier space [37,38], given by

$$\hat{\Gamma}^0_{ijkl}(\boldsymbol{\xi}) = \frac{\delta_{ik}\xi_j\xi_l}{2\mu^0\,|\boldsymbol{\xi}|^2} - \frac{\lambda^0}{2\mu^0\left(\lambda^0 + 2\mu^0\right)}\frac{\xi_i\xi_j\xi_k\xi_l}{|\boldsymbol{\xi}|^4} \tag{29}$$

where $\hat{\Gamma}^0_{ijkl} = \mathcal{F}(\boldsymbol{\Gamma}^0)$ is the Fourier transform of a periodic Green's operator $\boldsymbol{\Gamma}^0$; λ^0 and μ^0 are the Lamé's constants of the isotropic stiffness tensor; $\boldsymbol{\xi}$ is the Fourier point. The expression of Green's operators for anisotropic stiffness tensors are more complex and can be found in [39]. Eq. (29) is preferred because the elastic constants are separated from the Fourier variables, so that the elastic constants can be easily updated in the online stage to achieve self-consistency and improve accuracy. Then the interaction tensor can be calculated with

$$\boldsymbol{D}^{I,J} = \frac{1}{|\Omega^I|}\int_{\Omega} \chi^I(\mathbf{X})\mathcal{F}^{-1}\left(\mathcal{F}(\chi^J)\mathcal{F}(\boldsymbol{\Gamma}^0)\right)\mathrm{d}\mathbf{X}, \quad \forall I, J \in \{1, ..., N_c\} \tag{30}$$

using the fast Fourier transform (FFT) technique. The computational complexity is $O\left((N_c)^2(N_F)\log(N_F)\right)$, where N_F is the number of Fourier points used in the FFT calculation.

Numerical integration With an infinite homogeneous reference material, the Green's operator can be expressed in real space as shown in [35] and [40]. Numerical integration is the most straightforward method to compute the integral equation given in Eq. (26). The computational complexity is $O\left((N_I)^2\right)$, where N_I is the number of integration points used. In [40], a fast method is proposed to approximate D^{IJ}.

Finite element method Based on the physical interpretation of the interaction tensor, the finite element method can also be used. By applying uniform unit eigenstress component kl in the J^{th} cluster, the average strain can be computed for all clusters, resulting in D^{IJ}_{ijkl} for all $I = 1, ..., N_c$. Thus, the computational complexity is $O\left(6(N_c)(N_e)\right)$, where N_e is the number of finite elements used.

For microstructure scale solutions ($n > 1$), we use FFT or numerical integrations; for the part scale ($n = 1$), we use the finite element method. A fast approximation of interaction tensors based on coarse background grid was developed in [40].

3.2 Online stage: iterative solution scheme

In the online stage, the clustering-based discretization problem derived in Eq. (24) is solved, with proper constitutive equations for each of the individual phases not represented by a smaller scale.

Note that $\Delta\tilde{\mathbf{F}}^{I^{(n)}}$ can be obtained in the offline stage by solving the elastic problem (6), and $\Delta\mathbf{P}^{J^{(n)}}$ is a function of $\Delta\mathbf{F}^{J^{(n)}}$ through the local constitutive equations and/or homogenization of a finer scale MVE associated with the J^{th} cluster. The unknowns for Eq. (24) are

$$\{\Delta\mathbf{F}\} = \left\{\Delta\mathbf{F}^{I^{(1)},...,I^{(n)}},\ I^{(n)} = 1, ...N_c^{(n)}, n = 1, ..., N\right\}. \tag{31}$$

Newton's iterative method is used to solve Eq. (24) for each scale in the online stage, where the residual $\{\mathbf{r}\}^{(n)}$ is given component-wise as:

$$\mathbf{r}^{I^{(n)}} = \Delta\mathbf{F}^{I^{(n)}} + \sum_{J=1}^{N_c} \mathbf{D}^{I^{(n)},J^{(n)}} : \Delta\boldsymbol{\tau}^{J^{(n)}} - \Delta\tilde{\mathbf{F}}^{I^{(n)}}. \tag{32}$$

The system Jacobian $\{\mathbf{M}\}^{(n)(n)}$ at scale (n) is defined component-wise as:

$$\mathbf{M}^{I^{(n)}J^{(n)}} = \delta_{I^{(n)}J^{(n)}}\mathbf{I}_4 + \mathbf{D}^{I^{(n)}J^{(n)}} : \Delta\mathbf{C}^{J^{(n)}}, \quad \text{for } I^{(n)}, J^{(n)} = 1, ..., N_c^{(n)}, \tag{33}$$

where $\Delta \mathbf{C}^{J^{(n)}} = \mathbf{C}_{\text{eff}}^{J^{(n)}} - \mathbf{C}^{0,J^{(n)}}$. Here $\mathbf{C}_{\text{eff}}^{J^{(n)}} = \frac{\partial \Delta \mathbf{P}^{J^{(n)}}}{\partial \Delta \mathbf{F}^{J^{(n)}}}$ is the effective tangent stiffness tensor of the material in the cluster $J^{(n)}$ and is an output of a finer scale problem. $\mathbf{I}_4$ is a 4th rank identity tensor defined by $\mathbf{I}_{4,klmn} = \delta_{km}\delta_{ln}$, and $\delta_{I^{(n)}J^{(n)}}$ is the Kronecker delta in terms of indices $I^{(n)}$ and $J^{(n)}$. It is suggested to use the self-consistent scheme [28] to update the effective tangent stiffness tensor of an MVE to achieve higher accuracy in the online stage.

A recursive algorithm to solve Eq. (24) is given in Boxes I and II in Appendix 1. The algorithm in Box I controls Newton iterations at the part scale, while a recursive subroutine in Box II handles consecutive scale coupling. Scale coupling passes deformation gradient increments from a cluster at the coarser scale (e.g. scale 1) to the corresponding MVE at the finer scale (e.g. scale 2) as loading conditions, and returns the homogenized stress increments and the effective tangent stiffness tensor.

The number of unknowns in Eq. (31) is no more than $9 \times \sum_{n=1}^{N} \prod_{k=1}^{n} N_{\text{c}}^{(k)}$ with the extreme case where all clusters of all scales except the finest scale are associated with a finer scale MVE. The computational complexity of Newton's method is $O\left(729 \sum_{n=1}^{N} \prod_{k=1}^{n-1} N_{\text{c}}^{(k)} \left(N_{\text{c}}^{(n)}\right)^3\right)$, if a direct solver such as the LU decomposition method is used for the linearized equation systems. As a comparison, the number of unknowns when using concurrent multiscale FEM is $3\left(\sum_{n=1}^{N} \prod_{k=1}^{n-1} N_{\text{i}}^{(k)} N_{\text{n}}^{(n)}\right)$, resulting in a computational complexity of $O\left(3\left(\sum_{n=1}^{N} \prod_{k=1}^{n-1} N_{\text{i}}^{(k)} N_{\text{n}}^{(n)}\right)\right)$, assuming an linear scaling in the best scenario. Here, $N_{\text{i}}^{(k)}$ is the number of integration points used at the k^{th} scale; $N_{\text{n}}^{(n)}$ is the number of nodes used at the n^{th} scale. For a three-scale problem, if $N_{\text{i}}^{(1)}$, $N_{\text{i}}^{(2)}$ and $N_{\text{n}}^{(3)}$ is $O\left(10^5\right)$, and $N_{\text{c}}^{(1)}$, $N_{\text{c}}^{(2)}$ and $N_{\text{c}}^{(3)}$ is $O\left(10\right)$, then the computational complexity of MCA is $O(10^{11})$, which is significantly less than that of concurrent multiscale FEM $O(10^{15})$.

4 Numerical example: modeling three-scale particle-reinforced composites

The MCA framework is applicable to problems that involve an arbitrary number of length scales. In this section, we demonstrate its accuracy and efficiency by modeling a three-scale particle-reinforced composite.

4.1 Problem setting

A particle-reinforced composite system shown in Fig. 4 is used to demonstrate the accuracy and efficiency of the proposed method. A plate with three holes made of a composite whose matrix is reinforced by cubic particles (primary) at the micrometer scale with volume fraction of 5% and spherical particles (secondary) at the nanomemeter scale with volume fraction of 20%. This might be thought of as a rough model of a precipitate strengthen alloy, for example. Table 2 lists seven different numerical cases conducted. The first case is a homogeneous coupon with only the matrix phase, under uniaxial loading. In the second case, 5% volume fraction of cubic primary particles are added to the matrix. In the third case, 20% volume fraction of spherical secondary particles are added further to the matrix (but not the primary particles). In the fourth to seventh cases, the shape of the secondary particles are changed to ellipsoids with shape and volume fraction fixed but different angles ϕ between the major axis and the loading direction.

Table 1 Elastic properties of the matrix and particle phases

Phases	Young's moduli (MPa)	Poisson's ratios
Matrix	1000	0.3
Particle	2000	0.2

The matrix phase is modeled as an elasto-plastic material with J2 yielding and linear hardening rules. Both the primary and secondary particle phases are modeled with pure elasticity. The elastic properties of the two phases are given in Table 1. It should be noted that these constitutive relationships are used for demonstration purposes. More complex ones could be used depending on the problem of interest.

The von Mises yield surface is given by

$$f = \bar{\sigma} - \sigma_Y(\bar{\boldsymbol{\varepsilon}}^{\text{p}}) \leq 0, \tag{34}$$

where $\bar{\sigma}$ is the von Mises equivalent stress, and the yield stress σ_Y is a linear function of the equivalent plastic strain $\bar{\varepsilon}^{\text{p}}$:

$$\sigma_Y(\bar{\varepsilon}^{\text{p}}) = 6 + 500\bar{\varepsilon}^{\text{p}}\ \text{MPa}. \tag{35}$$

4.2 Clustering and interaction database generation

To generate the interaction database for MCA, the three-step method introduced in Sect. 3.1 is used. In the first step, linear elastic analysis for each individual scale is conducted separately. Uniaxial loading is applied at the part scale (scale 1) until 1% elongation is reached. FEM is used to get the strain field with the material properties of the matrix phase given in Table 1. Six orthogonal loading cases were applied at the microstructure scales (scales 2 and 3) to get the corresponding strain concentration fields using the FFT method. It should be noted that high stiffness contrast between matrix and particles could result in spurious oscillations in the elastic strain field, as reported in [37]. A simple resolution used in this work is postprocessing the deformation gradient field

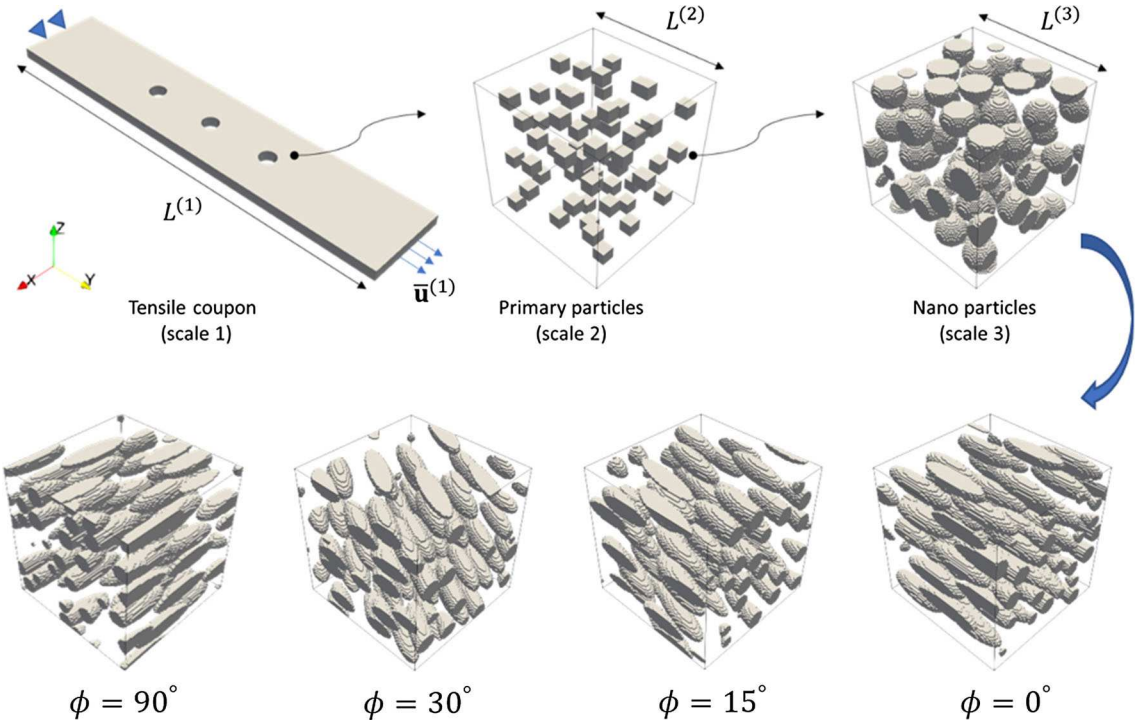

Fig. 4 A three-scale numerical example: **a** a plate with three holes made of a particle-reinforced composite; **b** cubic primary particles at the micrometer scale with volume fraction of 5%; **c** spherical secondary particles at the nanometer scale with volume fraction of 20%. Ellipsoidal secondary particles with volume fraction of 20% and different angles ϕ between the major axis and the loading direction (y axis direction) will also be tested: **d** $\phi = 90°$; **e** $\phi = 30°$; **f** $\phi = 15°$; **g** $\phi = 0°$. The scale lengths $L^{(1)}$, $L^{(2)}$, and $L^{(3)}$ are assumed to be 10 mm, 200 μm, and 10 μm respectively, so that the scales are well separated

Table 2 Numerical cases for modeling the particle reinforced composite system

Cases	Material description	Number of scales
Case 1	Matrix material only	1
Case 2	Matrix + 5% cubic primary particles	2
Case 3	Matrix + 5% cubic primary particles + 20% spherical secondary particles	3
Case 4	Matrix + 5% cubic primary particles + 20% ellipsoidal secondary particles with $\phi = 90°$	3
Case 5	Matrix + 5% cubic primary particles + 20% ellipsoidal secondary particles with $\phi = 30°$	3
Case 6	Matrix + 5% cubic primary particles + 20% ellipsoidal secondary particles with $\phi = 15°$	3
Case 7	Matrix + 5% cubic primary particles + 20% ellipsoidal secondary particles with $\phi = 0°$	3

by performing local averaging over the nearest 27 voxels (including each voxel itself). Alternative methods to solve this issue can be found in [41]. Figure 5a, b and c shows the DNS mesh used to generate strain concentration fields for each scale; Fig. 5d, e and f show contours of the strain component ε_{yy} normalized by their corresponding far field values. In the second step, the strain and strain concentration fields thus obtained were used to get a cluster-based domain decomposition at each scale with the k-means clustering method. The resulting clusters at each scale of the composite system are visualized in Fig. 5g, h and i. With this, the degrees of freedom to be solved is reduced from $O(10^{16})$ to $O(10^{3})$. In step three, the interaction tensors, see Fig. 5j, k and l, among these clusters at each scale were calculated based on Eq. (26) with FEM used for the part scale and FFT for microstructure scales. This multiscale interaction database will be used in the online stage of MCA.

4.3 Single-scale (part scale) MCA verification

For test case 1, Fig. 6 shows the reaction force-displacement curves computed using single-scale MCA with differing numbers of clusters, and compares that with the reference FEM solution using a fine mesh. The MCA result is close

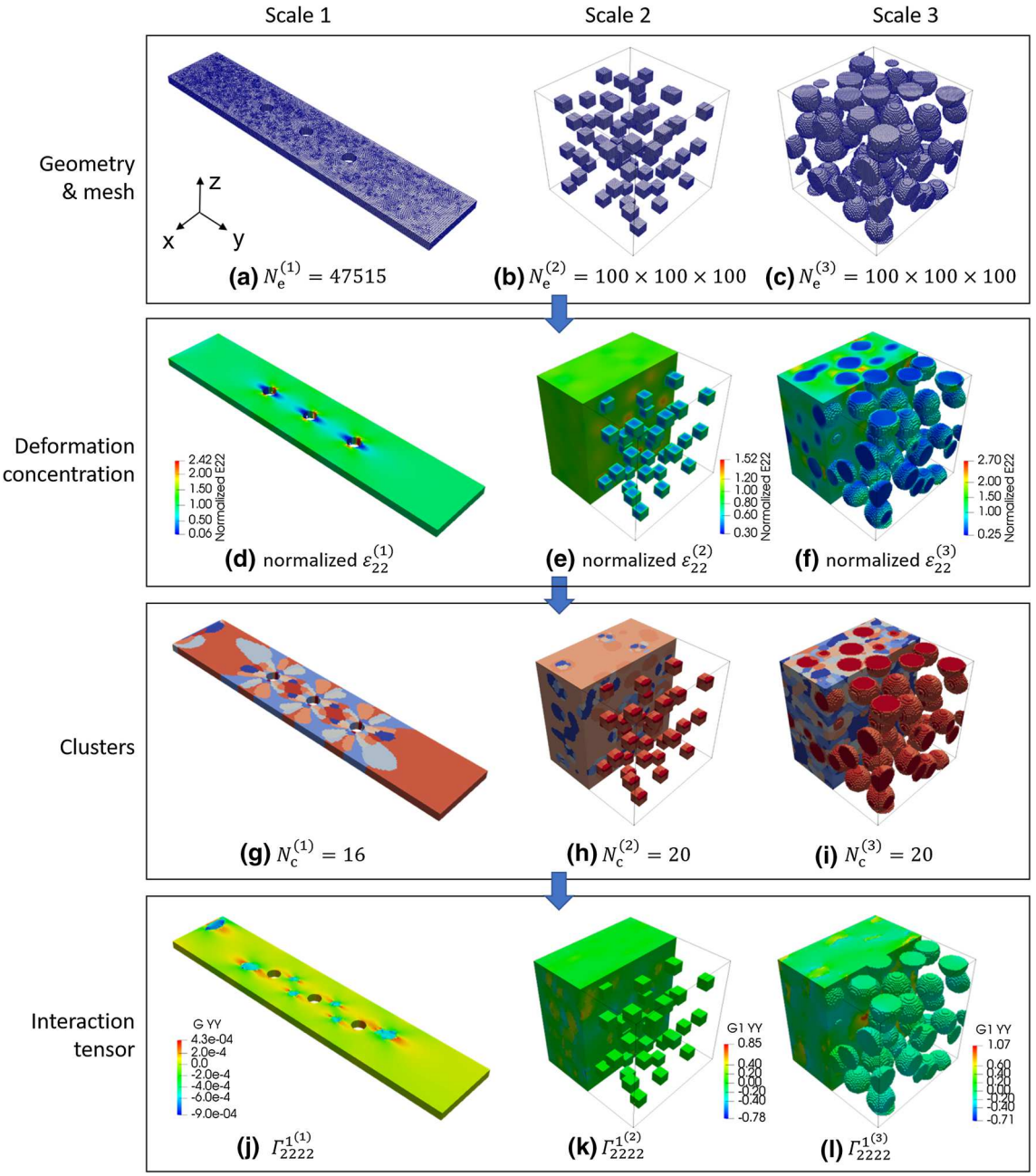

Fig. 5 Example database generated in the offline stage: **a, b, c** geometry and mesh, **d, e, f** deformation concentration obtained through linear elastic analysis, **g, h, i** clusters obtained using the *k*-means clustering method and **j, k, l** interaction tensors obtained using FEM and FFT

to the FEM result even for a small number of clusters and gets closer to the FEM result as the number of clusters increases. Figure 7 shows a comparison of the strain component ε_{22} distribution around the middle hole. As the number of clusters increases, the contour of ε_{22} computed using MCA approaches the reference FEM solution. This can be also seen from Fig. 8, which plots strain ε_{22} along the dashed line in Fig. 7.

4.4 Two-scale MCA verification

For test case 2, Fig. 9 shows the load-displacement curves computed using two-scale MCA with different numbers of clusters at the part scale with comparison to that using FE-SCA. Again, the MCA result is close to the FEM result even for a small number of clusters and gets closer to the FEM result as the number of clusters increases.

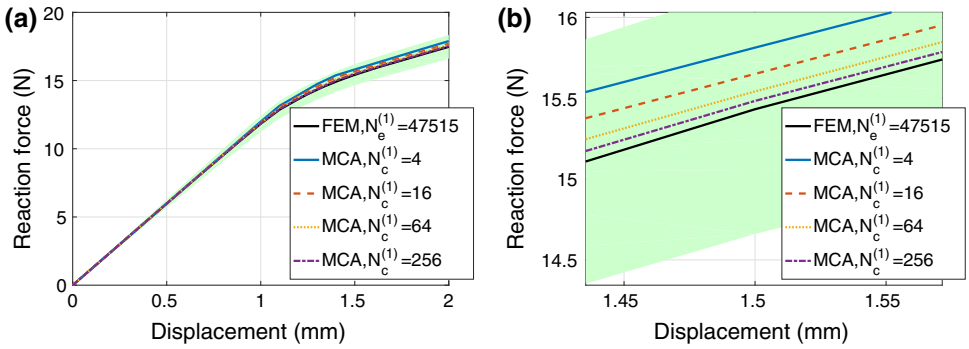

Fig. 6 **a** Comparison of reaction force-displacement curves for the macroscale problem computed using FEM and MCA with different numbers of clusters; **b** An enlarged view. The green shading indicates an area within 5% percent of the FEM solution

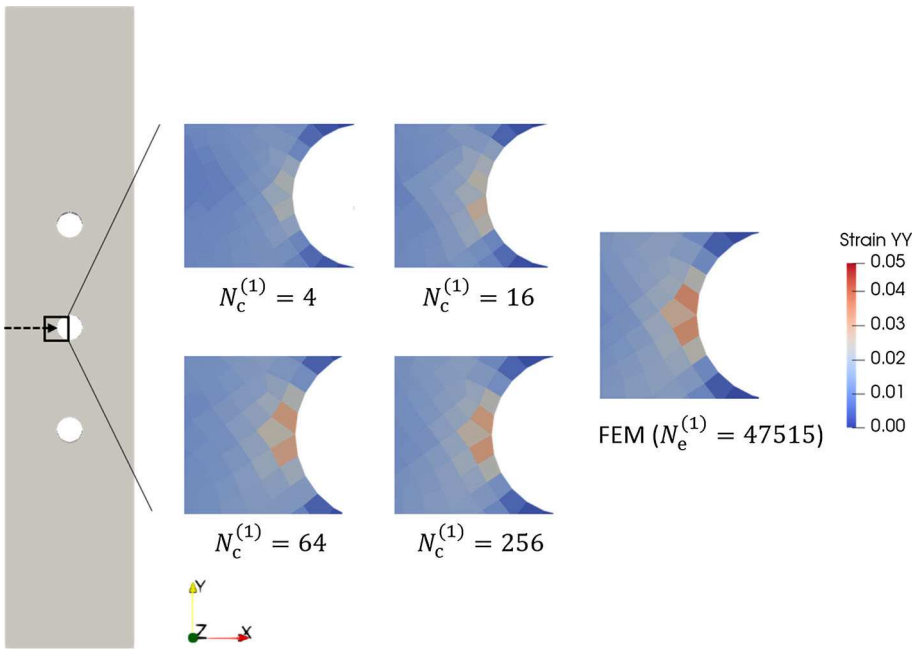

Fig. 7 Comparison between FEM and MCA with a varying number of clusters of local strain component YY distribution around the middle hole. The MCA solution approaches that of FEM as the number of clusters increases, showing the ability of MCA to capture strain concentrations while still reducing the number of degrees of freedom substantially

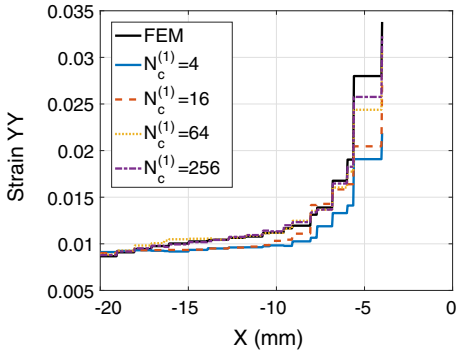

Fig. 8 Comparison of strain component YY along the dashed line in Fig. 7

4.5 Three-scale MCA for strengthening effect prediction

The result of test case 3 is compared with that of test cases 1 and 2 in Fig. 10a, which shows the strengthening effect of adding primary and secondary particles. Significant increase in both effective stiffness and yield strength is achieved by adding 20% volume fraction of secondary particles. The results of test cases 4-7 are compared with those of test case 3 in Fig. 10b, which shows that the stiffness and yield strength could be increased further by changing the secondary particles to ellipsoids and aligning them with the loading direction.

4.6 CPU time

Table 3 compares the degrees of freedom (DoFs) and the CPU time used for different methods. Note that simulations with FE^2 and FE^3 are not practical to conduct due to their tremendous computational cost. For the two scale problems, the speedup is calculated by comparing to the FE-SCA method. For a tolerance of $tol_{\text{newton}} = 10^{-4}$ in Box I and II, the Newton's method typically converges in less than 10 iterations. It is seen that the CPU time in the one-time offline stage of MCA is relatively expensive. However, once the offline database is prepared, MCA shows a tremendous speedup

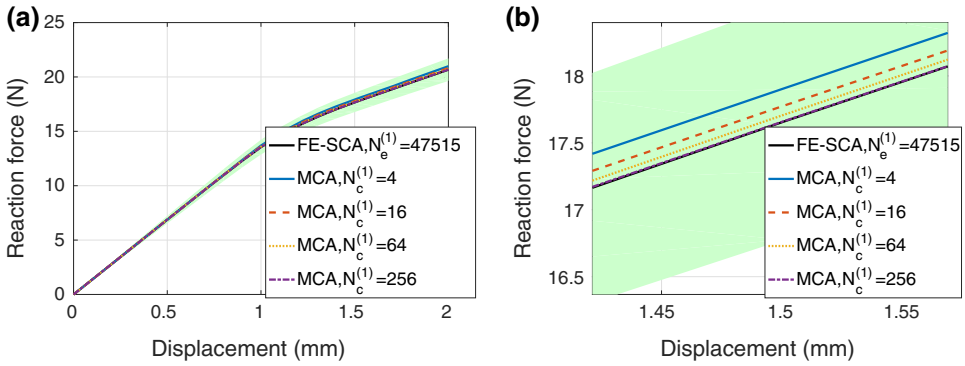

Fig. 9 **a** Comparison of reaction force-displacement curves computed using FE-SCA and two-scale MCA with a different number of clusters at the part scale and 20 clusters at the primary particle scale; **b** An enlarged view. The green shading indicates an area within 5% percent of the FE-SCA solution

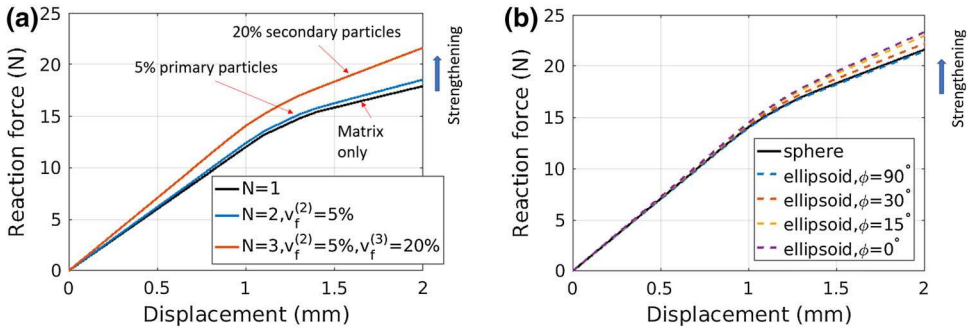

Fig. 10 **a** Load-displacement curve predicted using MCA showing that adding primary and secondary particles increases both stiffness and yield strength; **b** Load-displacement curve predicted using MCA showing that decreasing the angle between the inclusion major axis and the loading direction increases both stiffness and yield strength

Table 3 CPU time of MCA compared with different methods showing orders-of-magnitude speedup in the online stage

Methods	DoFs	Offline (s)	Online (s)	Speedup
FE	1.76×10^5	0	254.6	1
Single-scale MCA	36	207.34	0.0075	33952
FE^2	1.47×10^{11}	0	NA	NA
FE-SCA	3.44×10^7	17,032	3266.8	1
Two-scale MCA	756	17,239	0.13	25129
FE^3	1.47×10^{17}	0	NA	1
Three-scale MCA	15156	34271	26.4	NA

(more than 25,000 times faster) compared to the traditional finite-element-based methods.

5 Conclusion and future work

A multiresolution clustering analysis method is proposed for properties and performance prediction by concurrently modeling material behaviors at multiple length scales. The key idea of this method is to solve a set of fully coupled governing partial differential equations using the clusters generated from unsupervised machine learning at multiple length scales and a precomputed database of interaction tensors among these clusters. This method features an unprecedented balance of accuracy and efficiency by combining the advantages of both physics-based modeling and data-science based order reduction. Potential application to materials design is demonstrated with a particle reinforced composite, roughly analogous to a precipitate strengthened alloy, under uniaxial tensile loading. The example results show that the composite stiffness and yield strength could be improved by adding primary and secondary particles, and changing particle shapes. Refined material models can be used within this efficient multiscale modeling framework to discover more structure-property relationships, guiding hierarchical material design.

Theoretically, MCA works for material systems that involve an arbitrary number of discrete scales as long as continuum and scale separation assumptions can be made. However, attention must be paid to microstructural modeling and design at the nanoscale. For example, there are strong interactions between nanoparticles and dislocations resulting in a size effect in precipitation strengthened alloy systems. One way to capture the size effect would be to introduce a strain-gradient formulation of the Lippmann-Schwinger equation. Furthermore, problems with moving boundaries (e.g. moving contact between the roller and the part in the

rolling process) and microscale problems with significantly evolving microstructures (e.g. micro cracks) require special considerations. For example, one could adopt the arbitrary Lagrangian Eulerian method [42,43] in a moving contact problem where the clusters are fixed while materials points are allowed to flow in and out of a cluster. To accurately capture evolving microstructures, adaptive clustering methods might be used in a similar sense to the adaptive finite element methods [44] along with a fast method [40] to update the interaction tensors.

Acknowledgements Cheng Yu, Orion L. Kafka, and Wing Kam Liu were supported by the United States National Science Foundation under Grant No. MOMS/CMMI-1762035 and the award 70NANB14H012 from U.S. Department of Commerce, National Institute of Standards and Technology as part of the Center for Hierarchical Materials Design (CHiMaD). Orion L. Kafka also thanks the United States National Science Foundation for their support through the NSF Graduate Research Fellowship Program under financial award number DGE-1324585.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

A Recursive algorithm for solving N-scale L-S equations

Box I
Algorithm for MCA online stage

1. Initial conditions and initilization:
   (a) Set $k = 0$, $\{\mathbf{F}\}_k^{(1)} = \{\mathbf{I}_2\}$, $\{\mathbf{P}\}_k^{(1)} = \mathbf{0}$, $\{\Delta\mathbf{F}\}_k^{(1)} = \mathbf{0}$ and $\{\Delta\mathbf{F}\}_{\text{new}}^{(1)} = \{\Delta\mathbf{F}\}_k^{(1)}$
   (b) Set the reference stiffness $\mathbf{C}^{0,(1)}$
   (c) Load the interaction tensor $\{\mathbf{D}\}^{(1)}$
2. Newton iterations:
   (a) For each $\Delta\mathbf{F}^{I^{(n)}}$, $I^{(n)} = 1, , ..., N_c^{(1)}$, call the subroutine *Concurrent* to form $\{\Delta\mathbf{P}\}_{\text{new}}^{(1)}$ and $\{\mathbf{C}\}_{\text{eff}}^{(1)}$
   (b) Compute the residual $\{\mathbf{r}\}^{(1)}$
   (c) Compute the system Jacobian $\{\mathbf{M}\}^{(1)} = \partial\{\mathbf{r}\}^{(1)}/\partial\{\Delta\mathbf{F}\}^{(1)}$
   (d) Solve the linear equation $\{\mathbf{M}\}^{(1)}\{\delta\mathbf{F}\}^{(1)} = -\{\mathbf{r}\}^{(1)}$ for $\{\delta\mathbf{F}\}^{(1)}$
   (e) $\{\Delta\mathbf{F}\}_{\text{new}}^{(1)} \leftarrow \{\Delta\mathbf{F}\}_{\text{new}}^{(1)} + \{\delta\mathbf{F}\}^{(1)}$
   (f) if $\max_{J=1}^{N_c}\{||\delta\mathbf{F}^{J^{(1)}}||\} < tol_{\text{newton}}$ is not met, go to 2(a)
3. $\{\mathbf{F}\}_k^{(1)} \leftarrow \{\mathbf{F}\}_k^{(1)} + \{\Delta\mathbf{F}\}_{\text{new}}^{(1)}$, $\{\mathbf{P}\}_k^{(1)} \leftarrow \{\mathbf{P}\}_k^{(1)} + \{\Delta\mathbf{P}\}_{\text{new}}^{(1)}$, $k \leftarrow k + 1$ and update state variables
4. Repeat 2-3 until simulation complete

Box II
Recursive subroutine for RVE analysis: *Concurrent*($\Delta\mathbf{F}^{I^{(n)}}$,$N$,$\mathbf{C}^{0,I^{(n)}}$)

1. If $n = N$, call a user material subroutine to get $\Delta\mathbf{P}^{I^{(n)}}$ and $\mathbf{C}^{I^{(n)}}$; Set $\mathbf{C}_{\text{eff}}^{I^{(n)}} \leftarrow \mathbf{C}^{I^{(n)}}$ and go to 5
2. Initial conditions and initilization:
   (a) Set $\{\Delta\mathbf{F}\}_{\text{new}}^{(n+1)} = \Delta\mathbf{F}^{I^{(n)}}$
   (b) Set the reference stiffness $\mathbf{C}^{0,I^{(n+1)}} = \mathbf{C}^{0,I^{(n)}}$
   (c) Load the interaction tensor $\{\mathbf{D}\}^{(n+1)}$
3. Newton iterations:
   (a) Call the subroutine *Concurrent* to get $\{\Delta\mathbf{P}\}_{\text{new}}^{(n+1)}$ and $\{\mathbf{C}\}_{\text{eff}}^{(n+1)}$
   (b) Compute the residual $\{\mathbf{r}\}^{(n+1)}$
   (c) Compute the system Jacobian $\{\mathbf{M}\}^{(n+1)} = \partial\{\mathbf{r}\}^{(n+1)}/\partial\{\Delta\mathbf{F}\}^{(n+1)}$
   (d) Solve the linear equation $\{\mathbf{M}\}^{(n+1)}\{\delta\mathbf{F}\}^{(n+1)} = -\{\mathbf{r}\}^{(n+1)}$ for $\{\delta\mathbf{F}\}^{(n+1)}$
   (e) $\{\Delta\mathbf{F}\}_{\text{new}}^{(n+1)} \leftarrow \{\Delta\mathbf{F}\}_{\text{new}}^{(n+1)} + \{\delta\mathbf{F}\}^{(n+1)}$
   (f) If $\max_{J=1}^{N_c}\{||\delta\mathbf{F}^{J^{(n+1)}}||\} < tol_{\text{newton}}$ is not met, go to 3(a)
4. Average $\{\Delta\mathbf{P}\}_{\text{new}}^{(n+1)}$ to get $\Delta\mathbf{P}^{I^{(n)}}$; Get effective tangent stiffness $\mathbf{C}_{\text{eff}}^{I^{(n)}}$
5. Return $\Delta\mathbf{P}^{I^{(n)}}$ and $\mathbf{C}_{\text{eff}}^{I^{(n)}}$

References

1. Olson GB (1997) Computational design of hierarchically structured materials. Science 277(5330):1237–1242
2. Hao S, Liu WK, Moran B, Vernerey F, Olson GB (2004) Multi-scale constitutive model and computational framework for the design of ultra-high strength, high toughness steels. Comput Methods Appl Mech Eng 193(17–20):1865–1908
3. Liu WK, McVeigh C (2008) Predictive multiscale theory for design of heterogeneous materials. Comput Mech 42(2):147–170
4. Gitman IM, Askes H, Sluys LJ (2007) Representative volume: existence and size determination. Eng Fract Mech 74(16):2518–2534
5. Martin O-S (2006) Material spatial randomness: from statistical to representative volume element. Probab Eng Mech 21(2):112–132
6. Feyel F, Chaboche JL (2000) FE2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials. Comput Methods Appl Mech Eng 183(3–4):309–330
7. Kochmann J, Wulfinghoff S, Reese S, Mianroodi JR, Svendsen B (2016) Two-scale FE-FFT- and phase-field-based computational modeling of bulk microstructural evolution and macroscopic material behavior. Comput Methods Appl Mech Eng 305:89–110
8. Cosserat E, Cosserat F (1909) Théorie des corps déformables
9. Mindlin RD (1964) Micro-structure in linear elasticity. Arch Ration Mech Anal 16(1):51–78
10. Cemal Eringen A, Suhubi ES (1964) Nonlinear theory of simple micro-elastic solids-i. Int J Eng Sci 2(2):189–203
11. Germain P (1973) The method of virtual power in continuum mechanics. part 2: microstructure. SIAM J Appl Math 25(3):556–575
12. McVeigh C, Vernerey F, Liu WK, Cate Brinson L (2006) Multiresolution analysis for material design. Comput Methods Appl Mech Eng 195(37–40):5053–5076

13. McVeigh C, Liu WK (2008) Linking microstructure and properties through a predictive multiresolution continuum. Comput Methods Appl Mech Eng 197(41–42):3268–3290
14. Vernerey F, Liu WK, Moran B (2007) Multi-scale micromorphic theory for hierarchical materials. J Mech Phys Solids 55(12):2603–2651
15. Elkhodary KI, Steven Greene M, Tang S, Belytschko T, Liu WK (2013) Archetype-blending continuum theory. Comput Methods Appl Mech Eng 254:309–333
16. Greene SM, Li Y, Chen W, Liu WK (2014) The archetype-genome exemplar in molecular dynamics and continuum mechanics. Comput Mech 53(4):687–737
17. Kouznetsova V, Geers MGD, Marcel Brekelmans WA (2002) Multi-scale constitutive modelling of heterogeneous materials with a gradient-enhanced computational homogenization scheme. Int J Numer Meth Eng 54(8):1235–1260
18. Kouznetsova VG, Geers MGD, Brekelmans WAM (2004) Multi-scale second-order computational homogenization of multi-phase materials: a nested finite element solution strategy. Comput Methods Appl Mech Eng 193(48–51):5525–5550
19. Dvorak GJ (1992) Transformation field analysis of inelastic composite materials. In: Proceedings of the royal society of London A: mathematical, physical and engineering sciences, volume 437, pp. 311–327. The Royal Society
20. Michel JC, Suquet P (2003) Nonuniform transformation field analysis. Int J Solids Struct 40(25):6937–6955
21. Caglar O, Jacob F (2007) Eigendeformation-based reduced order homogenization for failure analysis of heterogeneous materials. Comput Methods Appl Mech Eng 196(7):1216–1243
22. Julien Y, He Q-C (2007) The reduced model multiscale method (R3M) for the non-linear homogenization of hyperelastic media at finite strains. J Comput Phys 223(1):341–368
23. Matouš K, Geers MGD, Kouznetsova VG, Gillman A (2017) A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials. J Comput Phys 330:192–220
24. Liu Z, Bessa MA, Liu WK (2016) Zeliang Liu, MA Bessa, and Wing Kam Liu. Self-consistent clustering analysis. Comput Methods Appl Mech Eng 306:319–341
25. 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
26. Liu Z, Kafka OL, Yu C, Liu WK (2018) Data-driven self-consistent clustering analysis of heterogeneous materials with crystal plasticity. In: Advances in Computational Plasticity, pp. 221–242. Springer
27. Kafka OL, Yu C, Shakoor M, Liu Z, Wagner GJ, Liu WK (2018) Data-driven mechanistic modeling of influence of microstructure on high-cycle fatigue life of nickel titanium. JOM, pp. 1–5
28. Yu C, Kafka OL, Liu WK (2019) Self-consistent clustering analysis for multiscale modeling at finite strains. Comput Methods Appl Mech Eng 349:339–359
29. Han X, Gao J, Fleming M, Chenghai X, Xie W, Meng S, Liu WK (2020) Efficient multiscale modeling for woven composites based on self-consistent clustering analysis. Comput Methods Appl Mech Eng 364:112929
30. He C, Gao J, Li H, Ge J, Chen Y, Liu J, Fang D (2020) A data-driven self-consistent clustering analysis for the progressive damage behavior of 3D braided composites. Compos Struct 249:112471
31. Gao J, Shakoor M, Domel G, Merzkirch M, Zhou G, Zeng D, Xuming S, Liu WK (2020) Predictive multiscale modeling for unidirectional carbon fiber reinforced polymers. Compos Sci Technol 186:107922
32. Levent O (2006) Impulse functions over curves and surfaces and their applications to diffraction. J Math Anal Appl 322(1):18–27
33. Kroner E (1972) Statistical continuum mechanics, vol 92. Springer, Berlin
34. Michel JC, Moulinec H, Suquet P (2001) A computational scheme for linear and non-linear composites with arbitrary phase contrast. Int J Numer Meth Eng 52(1–2):139–160
35. 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
36. Li H, Kafka OL, Gao J, Cheng Yu, Nie Y, Zhang L, Tajdari M, Shan Tang X, Guo GL et al (2019) Clustering discretization methods for generation of material performance databases in machine learning and design optimization. Comput Mech 64(2):281–305
37. Moulinec H, Suquet P (1998) A numerical method for computing the overall response of nonlinear composites with complex microstructure. Comput Methods Appl Mech Eng 157(1–2):69–94
38. Matthias K, Thomas B, Matti S (2014) Efficient fixed point and Newton-Krylov solvers for FFT-based homogenization of elasticity at large deformations. Comput Mech 54(6):1497–1514
39. Mura T (1987) Micromechanics of defects in solids. Martinus Nijhoff Publishers
40. Zhang L, Tang S, Yu C, Zhu X, Liu WK (2019) Fast calculation of interaction tensors in clustering-based homogenization. Comput Mech 64(2):351–364
41. François W (2015) Fourier-based schemes for computing the mechanical response of composites with accurate local fields. Comptes Rendus Mécanique 343(3):232–245
42. Liu WK, Belytschko T, Chang H (1986) An arbitrary lagrangian-eulerian finite element method for path-dependent materials. Comput Methods Appl Mech Eng 58(2):227–245
43. Liu J, Shu X, Kanazawa H, Imaoka K, Mikkola A, Ren G (2018) A model order reduction method for the simulation of gear contacts based on Arbitrary Lagrangian Eulerian formulation. Comput Methods Appl Mech Eng 338:68–96
44. Claes J, Peter H (1992) Adaptive finite element methods in computational mechanics. Comput Methods Appl Mech Eng 101(1–3):143–181